

КОМПЛЕКТ ЛЕКЦІЙ

дисципліна	Інформатика (шифр і назва навчальної дисципліни)
спеціальність	121 Інженерія програмного забезпечення (шифр і назва спеціальності)
освітня програма	Розробка програмного забезпечення (назва освітньої програми)
відділення	Комп'ютерної та програмної інженерії (назва відділення)

Розробив: Буряк Г.І.

Розглянуто та схвалено
на засіданні циклової комісії

математики та інформатики

Протокол № _____ від _____ 2023 р

Голова комісії _____ Малик О.М.

Лекція № 1

з навчальної дисципліни **Інформатика**

Тема Інформаційні технології в суспільстві

Мета заняття Охарактеризувати сучасні інформаційні технології та системи, узагальнити поняття: інформація, повідомлення, дані, інформаційні процеси

Матеріально-технічне забезпечення та дидактичні засоби, ТЗН

1 Інтерактивний вид діяльності, створений засобами Moodle

2 Мультимедійний презентаційний матеріал

Час – 2 години

План проведення лекції

Структура лекції	Відведений час	Методичні вказівки
1 Організаційна частина	5 хв.	Включає в себе: <i>привітання</i> , яке має на меті привернути увагу, забезпечити контакт лектора з аудиторією; <i>визначення присутності студентів на занятті</i>
2 Актуалізація опорних знань, перевірка вивченого матеріалу та мотивація навчальної діяльності студентів	10 хв.	Включає в себе: <i>вступне зауваження, мотивацію актуальності теми, перевірку попереднього матеріалу, формулювання мети лекції, огляд головних питань теми (вказати питання та завдання для перевірки знань студентів)</i>
3 Основна частина (викладення навчальних питань лекції)	60 хв.	Головне місце приділяється <i>викладенню основного змісту матеріалу теми, його аналізу, узагальненню висунутих положень</i> . Для успіху лекції важливе значення має поділ матеріалу на розділи, основні питання
4 Заклучна частина Домашнє завдання: (відповідно до робочої програми)	5 хв.	Включає в себе: <i>узагальнення, теоретичні і фактичні висновки, закріплення вивченого на лекції матеріалу</i>

Література

1 Інформатика (рівень стандарту) : підруч. для 10-го (11-го) кл. закл. заг. серед. освіти / Й.Я. Ривкід Т.І. Лисенко, Л.А. Чернікова, В.В. Шакотько. — Київ : Генеза, 2018. — 144 с. : іл.

2 Інформатика (рівень стандарту) : підруч. для 10-го (11-го) кл. закл. заг. серед. освіти / О.О. Бондаренко, В.В. Ластовецький, О.П. Пилипчук, Є.А. Шестопапов. — Харків : Вид-во «Ранок», 2018. — 175 с. : іл.

Навчальні матеріали лекції

ВСТУП

Інформатика є дуже молодого наукою, порівняно з іншими науками, такими, наприклад, як математика або фізика. Їй близько сімдесяти років. Так, історія математики, приміром, налічує кілька тисячоліть. Як і в кожній науці, в інформатиці є основні поняття. До них належать такі: «інформація», «повідомлення», «дані», «інформаційні процеси», «інформаційні технології», «інформаційні системи», «алгоритм», «код», «програма» тощо.

ІНФОРМАЦІЯ, ПОВІДОМЛЕННЯ, ДАНІ

В інформатиці, що закономірно для молодого науки, тривають дискусії вчених щодо підходів до пояснення основних понять цієї науки. Їх погляди можна узагальнити в кілька теорій залежно від того, як вони пояснюють поняття «інформація».

Прибічники «традиційного підходу» опираються на наукові праці засновників теорії зв'язку Р. Хартлі (1888–1970) та К. Шеннона (1916–2001). У цій теорії термін «інформація» є синонімом терміна «повідомлення», зміст повідомлення ігнорується, а саме повідомлення розглядається як набір сигналів різного виду залежно від способів передавання та фіксації повідомлень.

«Атрибутивний підхід» базується на твердженні, що інформація є невід'ємною частиною матеріальних об'єктів, є їх описом, атрибутом. За такого підходу інформація є об'єктивною характеристикою матеріального об'єкта та існує незалежно від свідомості людини.

Представники «філософського підходу» розглядають інформацію як результат відображення у свідомості людини реалій навколишнього світу. При цьому враховується зміст повідомлень, що передаються, опрацьовуються, зберігаються. У цьому підручнику пропонуємо нижченаведені пояснення основних понять інформатики.

Інформація — одне з основних понять інформатики, строге й універсальне означення якого неможливе. Інформація є відображенням реалій навколишнього світу у свідомості людини.

Не можна казати про кількісні характеристики відображення людиною реалій зовнішнього світу, оскільки неможливо однозначно передбачити результат такого відображення для різних людей і навіть для однієї і тієї самої людини за різних умов. Тому неможливо говорити про вимірювання інформації та її кількісні характеристики.

Так, враження від переглянутого кінофільму в різних людей будуть різними. Різне враження може бути навіть в однієї і тієї самої людини за різного емоційного стану, настрою, стану здоров'я тощо. По-різному ми оцінюємо смак страв, стилі одягу тощо. Тобто ми по-різному сприймаємо навколишній світ і відповідно по-різному відображаємо його у своїй свідомості.

Повідомлення — це деякі сигнали чи послідовності сигналів, які сприймаються людиною через органи чуття (зір, слух, дотик тощо). У технічних пристроях для операцій з повідомленнями використовують різноманітні датчики, що можуть

сприймати навіть ті сигнали, які не сприймаються органами чуття людини, наприклад рентгенівське випромінювання.

Повідомлення можуть подаватися різними способами: електричними сигналами, малюнками, звуками, схемами, текстом тощо. Слід зауважити, що за одного й того самого подання повідомлення може нести зовсім різний зміст, залежно від обставин, у яких воно передається і приймається, підготовленості до його тлумачення людини, яка приймає й аналізує повідомлення та робить (синтезує) відповідні висновки.

Дані — це повідомлення, які подано у вигляді, зручному для зберігання, передавання та опрацювання.

Зв'язки між інформацією та повідомленням можна проілюструвати на прикладі, поданому на рисунку 1.1.

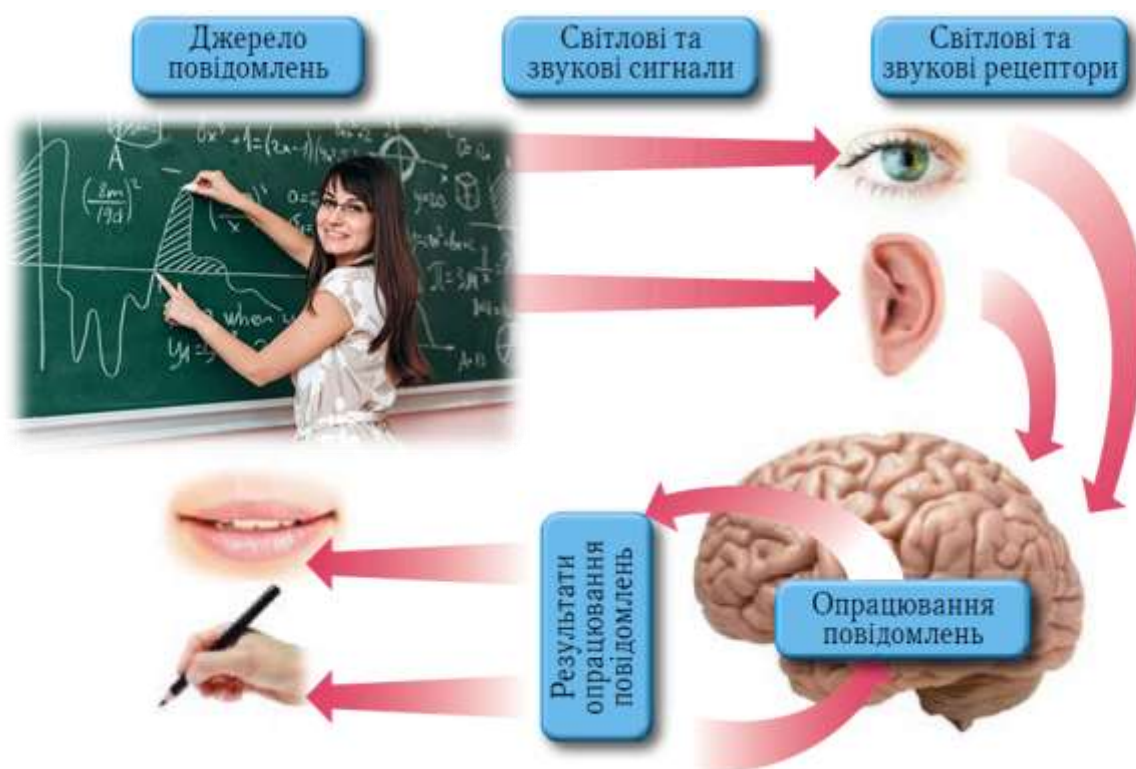


Рисунок 1.1 – Схема операцій з повідомленнями у процесі опрацювання студентом лекційного матеріалу

Пояснення викладачем правил виконання математичних обчислень студент сприймає з використанням рецепторів звуку та зору. Від рецепторів сигнали потрапляють у головний мозок, де опрацьовуються. Для усвідомлення змісту повідомлень лектора студент використовує зі своєї пам'яті відомості про позначення чисел, арифметичних дій тощо, які вже відомі.

Залежно від уважності на занятті, правильності опрацювання сигналів, наявних у пам'яті математичних знань, студент усвідомлює зміст повідомлень викладача. В такий спосіб можна вважати, що здобувач освіти отримав інформацію про математичні обчислення, які пояснював викладач. Ця інформація буде збережена у пам'яті студента. Якщо потрібно буде використати набуту інформацію для усної відповіді, студент, після відповідного опрацювання, здійснить передачу повідомлень з використанням, органів мовлення.

ІНФОРМАЦІЙНІ ПРОЦЕСИ

Інформаційними процесами називають процеси, що здійснюються над повідомленнями. Основними інформаційними процесами є процеси передавання, опрацювання та зберігання повідомлень. Розрізняють також і більш складні інформаційні процеси, такі як пошук, використання, групування, розповсюдження, сортування, кодування, захист, накопичення повідомлень тощо.

На рисунку 1.2 наведено схему інформаційного процесу використання повідомлень.



Рисунок 1.2 – Схема інформаційного процесу

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА СИСТЕМИ

технології здійснення операцій над текстами, графічними зображеннями, презентаціями, числовими, мультимедійними та іншими даними з використанням комп'ютерів називають інформаційними технологіями. Іншими словами, інформаційні технології описують особливості здійснення інформаційних процесів з використанням комп'ютерної техніки.

Інформаційні технології класифікують:

- залежно від типів даних, які опрацьовуються під час їх реалізації, наприклад, текстові, графічні, числові, мультимедійні інформаційні технології;
- за провідним інформаційним процесом, який реалізує ця технологія, наприклад, інформаційні технології передавання, опрацювання, зберігання даних;
- за основною метою здійснення інформаційних процесів, наприклад, пошуку, стиснення, передавання, кодування, захисту даних.

Мета застосування інформаційних технологій — створення та опрацювання інформаційних ресурсів, до яких належать програми, документи, графічні зображення, аудіо та відеодані, веб-сторінки тощо.

Для реалізації окремих інформаційних технологій, а особливо їх комплексу, створюються інформаційні системи. Сукупність взаємопов'язаних елементів, яка призначена для реалізації інформаційних процесів, називається інформаційною системою. Інформаційні системи мають дві основні складові — апаратну та програмну.

Апаратна складова — це комплекс технічних засобів, який включає пристрої опрацювання і зберігання даних, пристрої введення та виведення даних, засоби міжкомп'ютерного зв'язку.

Програмна складова — це комплекс програм, які забезпечують реалізацію інформаційних процесів пристроями інформаційної системи. Доволі часто в інформаційних системах виділяють інформаційну складову, під якою розуміють програмну складову та дані, які отримують, опрацьовують, зберігають і передають користувачам різні складові інформаційної системи.

На рисунку 1.3 наведено Типовий склад технічної інформаційної системи.



Рисунок 1.3 – Типовий склад технічної інформаційної системи

ЛЮДИНА В ІНФОРМАЦІЙНОМУ СУСПІЛЬСТВІ

Сучасне суспільство дедалі більше орієнтується на види діяльності, пов'язані з умінням своєчасно, швидко та професійно опрацьовувати величезні потоки відомостей, ефективно використовувати технології пошуку, добору, систематизації, оцінювання достовірності та актуальності різноманітних повідомлень, приймати на основі аналізу цих повідомлень рішення щодо подальшої діяльності. Зазначені знання та вміння повинні формуватися в людини ще під час навчання у школі, передусім це:

- знання основ теорії інформації;

- наявність навичок ефективного збирання, зберігання, опрацювання, передавання та захисту повідомлень;
- уміння аналізувати, класифікувати, оцінювати нові повідомлення, синтезувати нові ідеї та рішення;
- готовність не тільки оволодівати новими знаннями, а й пропагувати нові ефективні ідеї та технології;
- готовність сприймати різноманітні повідомлення, навіть такі, що ламають усталені стереотипи;
- стійкі навички міжособистісного спілкування, у тому числі з використанням комп'ютерних технологій і різними мовами;
- уміння аргументовано вести дискусії, готовність визнати себе переможеному цій дискусії;
- знання норм і правил, що регламентують використання інтелектуальної власності, та готовність незаперечно дотримуватися їх тощо.

Висновки

Отже, в процесі опрацювання матеріалів лекції можна прийти до висновків, що цінність має не стільки обсяг знань, які накопичила людина, скільки вміння ефективно використовувати ці знання у своїй повсякденній і професійній діяльності. Усе знати і все вміти неможливо, але вміти швидко адаптуватися до змін у суспільстві, до змін у техніці та технологіях — необхідна умова успішної реалізації задумів і мрій людини. Така адаптація неможлива без оволодіння навичками самостійно здобувати знання, у чому особливу роль відіграють сучасні інформаційно-комунікаційні технології.

Питання та завдання до контролю знань студентів

- 1 Які основні поняття інформатики ви знаєте?
- 2 Чому, на ваш погляд, учені не можуть дійти єдиної думки стосовно пояснення поняття «інформація»?
- 3 Поясніть, що таке інформація Чому не можна виміряти інформацію?
- 4 Що таке повідомлення?
- 5 Як можна подати повідомлення? Наведіть приклади
- 6 Що таке дані? Наведіть приклади даних
- 7 Які інформаційні процеси ви знаєте?
- 8 Що таке інформаційна технологія та інформаційна система?
- 9 Що входить до складу інформаційної системи?
- 10 Чому, сучасній людині потрібно добре володіти інформаційними технологіями?

Лекція № 2

з навчальної дисципліни **Інформатика**

Тема	Причини загострення проблеми забезпечення безпеки інформаційних технологій та види заходів протидії загрозам безпеки
Мета заняття	Виявити джерела, а також шляхи реалізації загроз, оцінити характери небезпеки, установити канали несанкціонованого доступу до даних

Матеріально-технічне забезпечення та дидактичні засоби, ТЗН

1 Інтерактивний вид діяльності, створений засобами Moodle

2 Мутимедійний презентаційний матеріал

Час – 2 години

План проведення лекції

Структура лекції	Відведений час	Методичні вказівки
1 Організаційна частина	5 хв.	Включає в себе: <i>привітання</i> , яке має на меті привернути увагу, забезпечити контакт лектора з аудиторією; <i>визначення присутності студентів на занятті</i>
2 Актуалізація опорних знань, перевірка вивченого матеріалу та мотивація навчальної діяльності студентів	10 хв.	Включає в себе: <i>вступне зауваження, мотивацію актуальності теми, перевірку попереднього матеріалу, формулювання мети лекції, огляд головних питань теми (вказати питання та завдання для перевірки знань студентів)</i>
3 Основна частина (викладення навчальних питань лекції)	60 хв.	Головне місце приділяється <i>викладенню основного змісту матеріалу теми, його аналізу, узагальненню висунутих положень</i> . Для успіху лекції важливе значення має поділ матеріалу на розділи, основні питання
4 Заклучна частина Домашнє завдання: (відповідно до робочої програми)	5 хв.	Включає в себе: <i>узагальнення, теоретичні і фактичні висновки, закріплення вивченого на лекції матеріалу</i>

Література

1 Інформатика (рівень стандарту) : підруч. для 10-го (11-го) кл. закл. заг. серед. освіти / Й.Я. Ривкід Т.І. Лисенко, Л.А. Чернікова, В.В. Шакотько. — Київ : Генеза, 2018. — 144 с. : іл.

2 Інформатика (рівень стандарту) : підруч. для 10-го (11-го) кл. закл. заг. серед. освіти / О.О. Бондаренко, В.В. Ластовецький, О.П. Пилипчук, Є.А. Шестопалов. — Харків : Вид-во «Ранок», 2018. — 175 с. : іл.

Навчальні матеріали лекції

ВСТУП

У зв'язку зі зростаючою роллю інформаційно-комунікаційних технологій у сучасному суспільстві проблема захисту даних від втрати, викрадення, спотворення або пошкодження потребує посиленої уваги. Вирішення цієї проблеми сприяє забезпеченню інформаційної безпеки як окремої особистості, організації, так і всієї держави.

ПОНЯТТЯ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

Інформаційна безпека — це стан захищеності систем опрацювання й зберігання даних, при якому забезпечено конфіденційність, доступність і цілісність інформації, або комплекс заходів, спрямованих на забезпечення захищеності інформації від несанкціонованого доступу, використання, оприлюднення, руйнування, внесення змін, ознайомлення, перевірки, запису чи знищення.

Принципи, на яких базується інформаційна безпека наведено на рисунку 2.1.



Рисунок 2.1 – Принципи інформаційної безпеки

Під конфіденційністю розуміють забезпечення доступу до даних на основі розподілу прав доступу, захист від несанкціонованого ознайомлення. До деяких даних право доступу мають усі користувачі, до інших — певні групи людей, а є особисті дані, доступ до яких може мати тільки одна людина.

Доступність означає забезпечення доступу до загальнодоступних даних усім користувачам і захист цих даних від блокування злоумисниками.

Цілісність передбачає захист даних від їх злоумисного або випадкового знищення чи спотворення.

Також під інформаційною безпекою розуміють комплекс заходів, спрямованих на забезпечення захищеності даних від несанкціонованого доступу, використання, оприлюднення, внесення змін чи знищення.

Останнім часом до питань інформаційної безпеки включено питання інформаційного впливу на особистість і суспільство. У лютому 2017 року указом Президента України було затверджено Доктрину інформаційної безпеки України, яка визначає національні інтереси України в інформаційній сфері, загрози їх реалізації, напрями і пріоритети державної політики в інформаційній сфері. Життєво важливими інтересами суспільства та держави визнано такі:

- захист українського суспільства від агресивного впливу деструктивної пропаганди;
- захист українського суспільства від агресивного інформаційного впливу, спрямованого на пропаганду війни, розпалювання національної та релігійної ворожнечі, зміну конституційного ладу насильницьким шляхом або порушення суверенітету і територіальної цілісності України;
- усебічне задоволення потреб громадян, підприємств, установ і організацій усіх форм власності в доступі до достовірних та об'єктивних відомостей тощо.

Кримінальним кодексом України передбачено кримінальну відповідальність за порушення інформаційної безпеки.

ЗАГРОЗИ ПІД ЧАС РОБОТИ В ІНТЕРНЕТІ

З технічної точки зору, залежно від результату шкідливих дій, можна виділити такі види загроз інформаційній безпеці:

- отримання несанкціонованого доступу до секретних або конфіденційних даних;
- порушення або повне припинення роботи комп'ютерної інформаційної системи;
- отримання несанкціонованого доступу до керування роботою комп'ютерної інформаційної системи;
- знищення та спотворення даних.

Значна частина загроз інформаційній безпеці виникає внаслідок користування ресурсами Інтернету. Серед них основними загрозами є такі:

- потрапляння в інформаційну систему шкідливого програмного забезпечення: вірусів, троянських програм, мережевих хробаків, клавіатурних шпигунів, рекламних систем тощо;
- інтернет-шахрайство, наприклад фішинг — вид шахрайства, метою якого є виманювання персональних даних у клієнтів онлайн-аукціонів, сервісів з переказу або обміну валюти, інтернет-магазинів тощо;
- несанкціонований доступ до інформаційних ресурсів та інформаційно-телекомунікаційних систем, наприклад у результаті цілеспрямованої хакерської атаки, що спрямовані на порушення штатного режиму функціонування системи, порушення доступності її сервісів, отримання несанкціонованого доступу до конфіденційних відомостей, порушення цілісності даних тощо;
- потрапляння комп'ютера до ботнет-мережі (англ. botnet від robot і network — робот і мережа) через приховане встановлення програмного забезпечення, яке

використовується зловмисником для виконання певних, найчастіше протиправних, дій з використанням ресурсів інфікованих комп'ютерів. Такими діями можуть бути розсилання спаму, добір паролів перебором усіх можливих варіантів, отримання персональних даних про користувачів, крадіжка номерів кредитних карток, паролів доступу, атаки з метою відмови в обслуговуванні, так звані DDoS-атаки (англ. Distributed Denial of Service — розподілена відмова в обслуговуванні), щоб порушити доступ до деякого інтернет-сервісу шляхом перевантаження його обчислювальних ресурсів тощо;

– «крадіжка особистості» — несанкціоноване заволодіння персональними даними особи, що дає можливість зловмиснику здійснювати діяльність (підписувати документи, отримувати доступ до ресурсів, користуватися послугами, знімати кошти з банківських рахунків тощо) від її імені.

Активування більшості шкідливих програм потребує безпосередньої участі користувача в захисті інформації. Загальновідомо, що коли на незнайомому сайті пропонують щось завантажити, то слід негайно відмовитися. Деякі зловмисники полюють на приватну інформацію користувача (банківські реквізити, електронне листування тощо). Деякі намагаються отримати віддалений доступ до персональних пристроїв — комп'ютерів, планшетів або мобільних телефонів. Розглянемо засоби браузерів, які призначені для захисту інформації користувача, на прикладі найбільш популярного браузера Google Chrome.

На рисунку 2.2 наведено пособи захисту інформації в Інтернеті.



Рисунок 2.2 – Способи захисту інформації в Інтернеті.

У Google Chrome передбачено захист акаунта Gmail від потенційно небезпечних листів, що можуть містити шкідливі програми, та попередження в разі спроби

завантажити файл, у якому під виглядом рінгтона або PDF-документа поширюється небезпечний код.

У Google Chrome використовуються додаткові модулі (плагіни), які дозволяють користувачу переглядати мультимедійні файли або відео. Плагіни також можуть використовуватися зловмисниками для зламу комп'ютера. Якщо виявиться застарілий плагін з уразливою системою безпеки, Google Chrome запропонує користувачу виконати оновлення і блокує плагін, доки він цього не зробить.

Якщо користувач спробує відкрити небезпечний сайт, Google Chrome попередить про це.

Спам — рекламні повідомлення (наприклад, про нові послуги, варіанти відпочинку, дешеві товари тощо), що надходять на поштову скриньку осіб, які не давали дозвіл на їх отримання. Окремі спам-повідомлення можуть нести комп'ютерні віруси, тому ніколи не слід відповідати на такі листи або переходити за вміщеними посиланнями.

Фішинг — масове розсилання електронних листів від імені популярних брендів із метою отримання доступу до логіна та пароля користувача, банківських рахунків та інших персональних даних.

У листах може міститися посилання на сайт, який зовні майже не відрізняється від справжнього. На проблемній веб-сторінці шахраї в різний спосіб намагатимуться примусити увести логін і пароль.

Із метою підвищення надійності захисту від фішингу виробники основних браузерів домовилися про застосування однакових способів інформування користувачів про перехід на підозрілий сайт.

Ознайомимося з типами підозрілих сайтів.

Підроблені сайти — метою таких сайтів є фішинг та інші види соціальної інженерії; вони обманом намагаються отримати доступ до особистих даних і паролів користувачів.

Небезпечні сайти — мета таких сайтів полягає в установленні на пристрій користувача шкідливого або небажаного програмного забезпечення, яке може порушити роботу системи й викликати неполадки в мережі.

Посилання на підозрілі сайти можуть поширюватися в різний спосіб, наприклад, електронною поштою. Для безпечного перегляду браузері складають список сайтів, підозрюваних у фішингу та поширенні шкідливих програм, аналізують веб-сторінки і надсилають попередження, якщо сайт становить загрозу.

Досвід користування деякими ресурсами свідчить, що іноді зображення замка, який є на сторінках деяких ресурсів, можна вважати простою картинкою. Це означає, що в сайті відсутній сертифікат справжності, не підтримується і SSL-з'єднання.

SSL (англ. Secure Sockets Layer — рівень захищених сокетів) — криптографічний протокол, який гарантує встановлення безпечного з'єднання між клієнтом і сервером. Технологія SSL забезпечує кращий захист і збереження конфіденційності інформації на відміну від незашифрованого веб-з'єднання. Протокол забезпечує конфіденційність обміну даними між клієнтом і сервером, які використовують TCP/IP.

Якщо з'єднання, яке потребує введення конфіденційної інформації, є незахищеним, то слід скористатися іншим ресурсом.

SSL-сертифікат — це сертифікат безпеки; цифровий спосіб ідентифікації, який підтверджено Центром сертифікації (ЦС). У сертифікаті містяться ідентифікаційна інформація, період дії, публічний ключ, серійний номер та цифрові підписи.

Сучасні браузери мають вбудовану систему безпеки, підтверджену сертифікатами, отриманими в спеціальних центрах. Система браузера вважатиме сертифікат справжнім, якщо ЦС, який його видав, є в переліку дійсних закладів. На рисунку 2.3 наведено принцип отримання інформації про сертифікат.

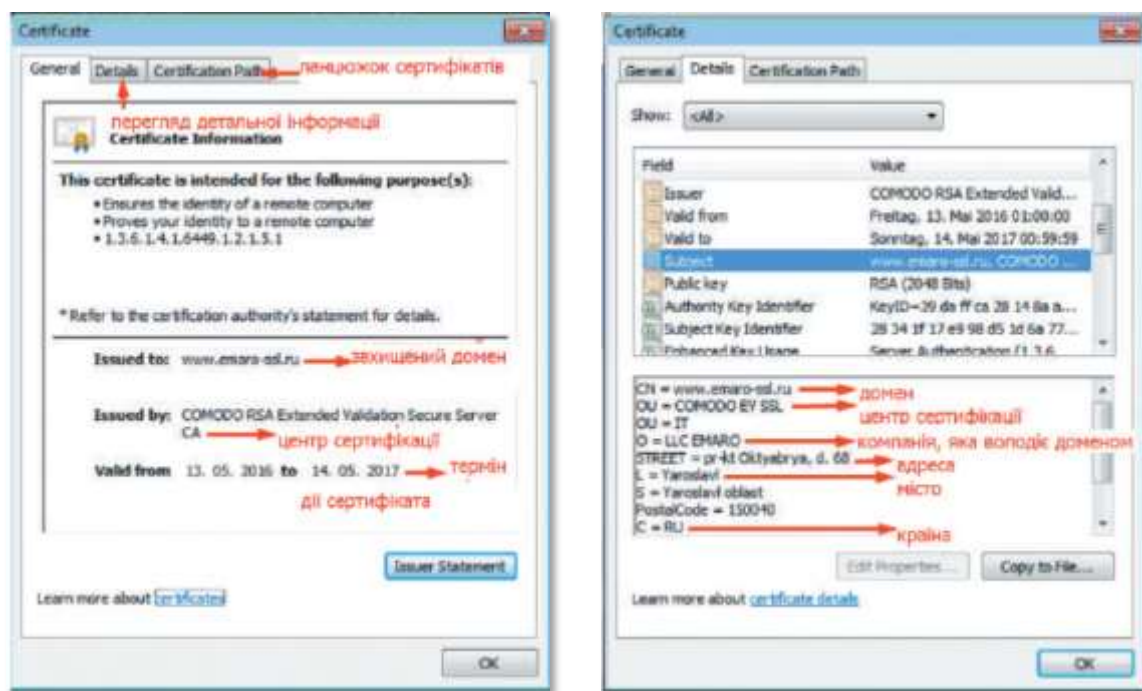


Рисунок 2.3 – Інформація про сертифікати

ЗАГРОЗИ ДЛЯ МОБІЛЬНИХ ПРИСТРОЇВ

Сучасні мобільні телефони (смартфони), доповнені функціями персонального комп'ютера, із власною операційною системою та різноманітним програмним забезпеченням. Тому для смартфонів характерні ті самі загрози, що і для стаціонарних комп'ютерів: віруси, троянські програми, мережеві хробаки, рекламні модулі тощо, орієнтовані на різні типи мобільних пристроїв. Як і стаціонарні комп'ютери, смартфони можуть потрапити до ботнет-мережі.

Найчастіше смартфон постійно увімкнено, має підключення до мережі Інтернет, завжди розташований поруч із власником, містить різноманітні пристрої введення/виведення: мікрофон, відеокамеру, GPS-навігатор тощо. Сьогодні смартфон забезпечує доступ до грошових рахунків в оператора мобільного зв'язку, у системі онлайн-банкінгу або інших. Усе це підсилює небезпеку.

Існують шпигунські програми, які зловмисники використовують для шпигування за користувачем смартфона. Використовуючи їх, можна перехоплювати повідомлення про всі здійснені дзвінки, дізнаватися вміст СМС-листування та дані про відвідані сайти, знімати камерою пристрою оточення користувача, визначати його місце розташування, включати мікрофон і записувати всі розмови.

Ще один аспект загроз для користувачів мобільних телефонів полягає в роботі з платними послугами. Підписка з використанням СМС на сервіс, який вимагає регулярну оплату, може призводити до списування з рахунку значних коштів. Іноді такі СМС можуть надсилатися троянськими програмами.

Однак не всі користувачі дбають про безпеку та встановлюють антивірусне програмне забезпечення на свої смартфони.

СОЦІАЛЬНА ІНЖЕНЕРІЯ

Соціальна інженерія — це наука, що вивчає людську поведінку та фактори, які на неї впливають.

У наш час результати досліджень із соціальної інженерії часто використовують зловмисники для маніпуляції, щоб спонукати людину виконати певні дії чи розголосити конфіденційну інформацію.

За даними української антивірусної лабораторії Zillya! Антивірус (zillya.ua), наразі більшість заражень шкідливими програмами комп'ютерів і мереж відбувається шляхом обману користувачів з використанням методів соціальної інженерії.

Найбільш поширені прийоми, які використовують зловмисники:

- надсилання електронних листів, зміст яких спонукає користувача відкрити прикріплений до листа файл. Як наслідок, може бути активована троянська програма. Зловмисники розраховують на емоційну реакцію користувача на повідомлення в листі або на звичайну цікавість;

- створення сайтів, які дуже схожі на справжні, для отримання логінів і паролів користувачів. Це один з прийомів фітінгу. Шахрайство базується на некоректно введених у браузері адресах сайтів, на підміні пошукових запитів;

- комбінація двох попередніх методів — надсилання електронного листа з пропозицією перейти на фішинговий сайт.

Людські слабкості — жадібність, нечесність, честолубство та інші — також часто використовують для досягнення зловмисної мети. Троянські програми найчастіше потрапляють на комп'ютер під час спроби користувача використати неліцензійне, «зламане», програмне забезпечення, у якому міститься прихований троянський модуль. Також троянські програми містяться в генераторах кодів і програмах для «зламування» платних програмних засобів.

ПРАВИЛА БЕЗПЕЧНОЇ РОБОТИ В ІНТЕРНЕТІ

Для того щоб максимально уникнути загроз під час роботи в Інтернеті, варто дотримуватися певних правил. Наведемо поради, що надані CERT-UA (англ. Computer Emergency Response Team of Ukraine — команда України з реагування на комп'ютерні надзвичайні події) — спеціалізованим структурним підрозділом Державного центру кіберзахисту та протидії кіберзагрозам Державної служби спеціального зв'язку та захисту інформації України (cert gov ua):

- 1 Використовуйте тільки ліцензійне програмне забезпечення. Установлюйте програми тільки з офіційних джерел. Перед установленням читайте відгуки інших користувачів, якщо вони доступні.

2 Установлюйте та оновлюйте антивірусне програмне забезпечення як на стаціонарні, так і на мобільні комп'ютери. Бажано, щоб оновлення антивірусних баз здійснювалося регулярно та автоматично.

3 Використовуйте надійні паролі. Не використовуйте на різних інтернет-ресурсах один і той самий пароль, змінюйте його регулярно.

4 Приєднуйтеся тільки до перевірених Wi-Fi-мереж. Не відправляйте важливі дані (дані кредитних карток, онлайн-банкінгу тощо) через публічні та незахищені Wi-Fi-мережі.

5 Установлюйте фільтр спливаючих вікон у браузері.

6 Перевіряйте сертифікат безпеки сайтів у вигляді замка в адресному рядку браузера та URL-адреси веб-сайтів, щоб визначити, чи не підроблений сайт ви відвідуєте.

7 Не відкривайте повідомлення електронної пошти від невідомих вам осіб і прикріплені до них файли, яких ви не очікуєте.

8 Подумайте про можливі ризики для вас перед тим, як викласти щось у мережу Інтернет. Дуже легко розмістити дані в мережі Інтернет, але дуже складно їх видалити з неї.

9 Створюйте резервні копії важливих для вас даних, зберігайте їх на носіях даних, відключених від мережі Інтернет.

Корисним є створення в операційній системі Windows облікового запису користувача, який не має прав адміністратора. Якщо під час роботи в Інтернеті з таким обліковим записом на комп'ютер потрапить троянська програма, вона не буде запущена на виконання.

Для користувачів електронної пошти та соціальних мереж рекомендується використовувати додаткові заходи безпеки. Один з них — двоетапна перевірка.

Двоетапна перевірка (або двофакторна авторизація) — це спосіб входу до облікового запису, при якому потрібно, крім введення логіна та пароля, виконати певну додаткову дію, наприклад увести код, отриманий в СМС, на електронну пошту або в голосовому повідомленні. Також для дуже важливих акаунтів використовуються унікальні зовнішні накопичувачі та зчитувачі біометричних даних.

Для користувачів смартфонів є окремі рекомендації:

- не телефонуйте на незнайомі номери;
- уважно контролюйте послуги, на які ви підписуєтеся;
- установлюйте мобільні додатки лише з офіційних магазинів: PlayMarket, (Android), AppStore (iOS), Marketplace (WindowsPhone);
- уважно стежте за тим, які дозволи вимагає програма під час установлення та оновлення програмного забезпечення на мобільних пристроях.

Висновки

Отже, заради уникнення загроз підчас взаємодії із Інтернет, слід дотримуватися зазначених правил. Бажано пам'ятати, що частіше за все, зловмисники використовують для досягнення зловмисної мети саме людські слабкості — жадібність, цікавість, нечесність, честолюбство тощо.

Питання та завдання до контролю знань студентів

- 1 Що таке інформаційна безпека?
- 2 На яких принципах базується інформаційна безпека?
- 3 Що таке спам, фішинг?
- 4 Назвіть найпоширеніші протоколи захищеного зв'язку.
- 5 Яка інформація міститься в сертифікаті безпеки сайта?
- 6 Відкрийте у браузері Google Chrome п'ять сайтів, які ви відвідуєте найчастіше.
- 7 За алгоритмом, наведеним наприкінці параграфа, перевірте
- 8 ці сайти на валідацію (англ. valid — справжність (дійсність)).
- 9 Зробіть висновки про результати роботи та опублікуйте на своєму Google-диску.
- 10 Надішліть звітні матеріали в системі дистанційного навчання Moodle.

Лекція № 3

з навчальної дисципліни **Інформатика**

Тема Навчання в Інтернеті

Мета заняття Ознайомитися із попитом ринку професії майбутнього, дослідити роль інформаційних технологій у роботі сучасного працівника

Матеріально-технічне забезпечення та дидактичні засоби, ТЗН

1. Інтерактивний вид діяльності, створений засобами Moodle

2. Мультимедійний презентаційний матеріал

Час – 2 години

План проведення лекції

Структура лекції	Відведений час	Методичні вказівки
1 Організаційна частина	5 хв.	Включає в себе: <i>привітання</i> , яке має на меті привернути увагу, забезпечити контакт лектора з аудиторією; <i>визначення присутності студентів на занятті</i>
2 Актуалізація опорних знань, перевірка вивченого матеріалу та мотивація навчальної діяльності студентів	10 хв.	Включає в себе: <i>вступне зауваження, мотивацію актуальності теми, перевірку попереднього матеріалу, формулювання мети лекції, огляд головних питань теми (вказати питання та завдання для перевірки знань студентів)</i>
3 Основна частина (викладення навчальних питань лекції)	60 хв.	Головне місце приділяється <i>викладенню основного змісту матеріалу теми, його аналізу, узагальненню висунутих положень</i> . Для успіху лекції важливе значення має поділ матеріалу на розділи, основні питання
4 Заклучна частина Домашнє завдання: (відповідно до робочої програми)	5 хв.	Включає в себе: <i>узагальнення, теоретичні і фактичні висновки, закріплення вивченого на лекції матеріалу</i>

Література

1 Інформатика (рівень стандарту) : підруч. для 10-го (11-го) кл. закл. заг. серед. освіти / Й.Я. Ривкід Т.І. Лисенко, Л.А. Чернікова, В.В. Шакотько. — Київ : Генеза, 2018. — 144 с. : іл.

2 Інформатика (рівень стандарту) : підруч. для 10-го (11-го) кл. закл. заг. серед. освіти / О.О. Бондаренко, В.В. Ластовецький, О.П. Пилипчук, Є.А. Шестопапов. — Харків : Вид-во «Ранок», 2018. — 175 с. : іл.

Навчальні матеріали лекції

ВСТУП

Потреби на ринку праці залежать від багатьох чинників. Вирішальне значення мають безпосередньо процеси, що відбуваються в соціально-економічному житті суспільства, у тому числі глобальні на світовому рівні. Розвиток ІТ змінює наші уявлення про способи отримання знань і змушує серйозно переосмислити звичний підхід до навчального процесу. Це означає, що в майбутньому ІТ-фахівці в галузі освіти будуть затребуваними.

Для сучасної людини важливими є навички самостійного отримання необхідних відомостей з використанням різноманітних засобів. Такими засобами у сьогоденні є наявні освітні ресурси мережі Інтернет.

ВИМОГИ, ЯКІ ВИСУВАТИМУТЬСЯ ДО ФАХІВЦІВ

Майбутні професії будуть вельми різноманітними. Тому і вимоги, які висуватимуться до фахівців є такими:

- крос-функціональність — найбільше цінуються фахівці, які працюють «на стику» професій. Багато компаній прагнуть сформувати команди професіоналів, що володіють знаннями відразу в кількох галузях;
- мобільність — експерти стверджують, що нині часта зміна місця роботи і навіть сфери діяльності є нормою. Американські дослідники припускають, що незабаром людина до 38 років зможе освоїти до 10–14 професій;
- універсальність — зростає попит на фахівців, які володіють кількома іноземними мовами, знають основи менеджменту, маркетингу та фінансів і вміють працювати з big data;
- уміння працювати в команді — важливо узгоджувати свої дії з членами команди, уміти довіряти іншим, делегувати роботу, відстоювати свою точку зору та прислухатися до чужої позиції.

НАВЧАННЯ У ІНТЕРНЕТІ

Однією з основних навичок людини стає здатність до самоосвіти, спроможність швидко засвоювати нові професії. Зростає роль освітніх платформ, які пропонують масові відкриті безкоштовні онлайн-курси від провідних університетів світу, наприклад, Гарвардського, стенфордського, університету Берклі тощо. Вони вже зараз до зволіють мільйонам людей здобувати знання в будь-якому куточку світу. Як було зазначено на Всесвітньому економічному форумі в Давосі в 2013 році, онлайн-освіта змінює світ.

Існують сотні відкритих курсів на десятках відкритих освітніх платформ, де зареєстровано мільйони користувачів. Популярності набули відкриті курси провідних університетів світу, що розміщено на платформах Coursera, Udacity, edX та інших, а також відкриті онлайн-курси на україномовних платформах Prometheus, EdEra тощо.

На рисунку 3.1 наведено піктограми найпопулярніших освітніх платформ світу та України.



Рисунок 3.1 – Піктограми найпопулярніших освітніх платформ світу та України

Coursera — масовий відкритий онлайн-курс, який заснували Ендрю Нг і Дафна Коллер, професор і професорка інформатики Стенфордського університету, у квітні 2012 року. Має 100 університетів-партнерів, у тому числі 33 університети США, та близько 5 млрд користувачів. Доступ до курсів обмежений за часом, кожне завдання або тест має бути виконане до певної дати. Після закінчення курсу, за умови виконання проміжних завдань і складання заключного іспиту, слухач отримує сертифікат.

Coursera пропонує близько 500 курсів за освітніми категоріями, що охоплюють гуманітарні науки, біологію, медицину, соціальні науки, математику, економіку та фінанси, комп'ютерні науки тощо. Coursera пропонує не окремі лекції, а повноцінні курси, які включають відеолекції з субтитрами, конспекти лекцій, домашні завдання, тести й підсумкові іспити.

Udacity — комерційна освітня організація, яку засновано професором комп'ютерних наук Стенфордського університету Себастьяном Траном, пропонує масові відкриті онлайн-курси, орієнтуючись насамперед на «професійні курси для професіоналів». Курси Udacity присвячено переважно комп'ютерним наукам, фізиці та математиці, і головна їх ідея — сприяння подальшому працевлаштуванню.

Курси на Udacity та Coursera містять цілий комплекс взаємопов'язаних складових. Нині вони почали надавати університетські кредити на проходження курсів, які провідні університети вже зараховують як складову навчальної програми.

edX — спільний проект Гарвардського університету і Массачусетського технологічного інституту, який було започатковано 2012 року. edX безкоштовно пропонує більш ніж 250 онлайн-курсів найвідоміших університетів та інститутів, переважно з комп'ютерних наук та електроніки. Є можливість отримати сертифікат edX від цих університетів.

Prometheus — перший український громадський проект масових відкритих онлайн-курсів. Головна мета — безкоштовне надання онлайн-доступу до курсів університетського рівня всім бажаючим. На платформі розміщено цикл курсів для всіх, хто хоче підготуватися до ЗНО та ДПА з української мови та літератури, історії України та математики.

EdEra — другий за популярністю україномовний онлайн-ресурс. Наразі створено понад 1000 освітніх роликів, запущено близько 20 відкритих онлайн-курсів та інтерактивних підручників. EdEra ставить на меті зробити освіту якісною та доступною, орієнтуючись насамперед на базову середню освіту. Автори створюють онлайн-курси повного циклу — від лекцій до книжок.

EdEra Books — це соціальний проект інтерактивної освітньої літератури зі вбудованими відео, автоматизованими тестами і можливістю коментувати кожну сторінку. Проект незамінний для тих, хто хоче підготуватися до складання ЗНО.

Упевнені навички при використанні можливостей офісних пакетів (Microsoft Office, Libre Office, Open Office), уміння застосовувати у своїй діяльності сучасні ІТ стають одними з основних компонентів професійної підготовки будь-якого фахівця.

КОМП'ЮТЕРНО-ОРІЄНТОВАНІ ЗАСОБИ ПЛАНУВАННЯ, ВИКОНАННЯ, ДОСЛІДНОЇ, НАВЧАЛЬНОЇ І ПРАКТИЧНОЇ ДІЯЛЬНОСТІ

Кожна справа, за яку ми беремося, вимагає чіткого планування: з яких етапів складається, скільки часу потребуватиме виконання? Якщо це справа колективна, дуже важливо узгодити свої дії з іншими членами команди. Можна, звісно, узяти аркуш паперу і розписати весь ланцюжок дій, а потім ще попросити кожного занотувати. А можна використати спеціальні сервіси — інструменти для індивідуального (колективного) планування.

Існують кілька видів онлайн-інструментів планування, які наведено на рисунку 3.2.



Рисунок 3.2 – Онлайн-інструменти планування

Органайзер — інструмент організації інформації про особисті контакти та події, засіб управління часом.

Оайм-трекінг — інструмент управління часом, відстеження часу на різні завдання (<http://www.yast.com/>, <http://www.rememberthemilk.com/>).

Планувальник завдань — ToDo-лист, інструмент, що дає змогу організувати список завдань (<http://todo.ly/>).

Окрім зазначених, існує ціла низка подібних органайзерів — онлайн-та офлайн-версії, мобільні та веб-додатки, платні й безкоштовні, орієнтовані на індивідуальну та колективну роботу, навчальну та бізнесову діяльність. Переважна більшість із них має інтуїтивно зрозумілий інтерфейс.

Зазвичай керування проектами (Project management) передбачає колективну роботу. Для організації мережових проектів із великою кількістю завдань та численними співавторами й учасниками призначені спеціальні веб-інструменти. Такі сервіси дозволяють зберігати інформацію про проекти в Інтернеті, керувати проектами спільно з колегами — розподіляти рівні доступу й відповідальність, планувати завдання та призначати виконавців із подальшим відстеженням їх виконання.

Для планування етапів дослідних завдань, які мають складну організацію, часто використовуються нескінченні віртуальні дошки. На них можна працювати з будь-яким візуальним контентом (прикріплювати картинки, малювати схеми та графіки, створювати колажі й ін.) індивідуально або з командою. На рисунку 3.3 наведено популярні сервіси.



Рисунок 3.3 – Онлайн-сервіси для віртуальних дошок

Twiddla (<http://www.twiddla.com>) — позиціонується як онлайн-дошка для сучасного класу. Це інструмент онлайн-співпраці в реальному часі. Простий та доступний: немає плагінів або завантажень, розширених розкладів, не потрібно працювати в складних брендмауерах. Єдина вимога — комп'ютер із під'єднанням до Інтернету та веб-браузером.

SpiderScribe (<http://www.spiderscribe.net>) — онлайн-інструмент для створення інтелект-карт та організації мозкового штурму. Він дає змогу структурувати ідеї, підключивши нотатки, файли, події календаря тощо в картках вільної форми. Є можливість співпрацювати та обмінюватися інформацією онлайн.

Lino (<http://linoit.com>) і **RealtimeBoard** — віртуальні дошки, створені для спільної роботи.

Особливу роль відіграють у навчальній діяльності Інтернет-сервіси. Наприклад, як інструменти для управління й організації навчального проекту можуть бути використані універсальні веб-сервіси з колективним доступом, функцією створення

wiki-сторінок, а також можливістю створення списків завдань і вбудовування віджетів від зовнішніх сервісів, таких як Google Календар.

ВІРТУАЛЬНІ ЦИФРОВІ ПОМІЧНИКИ

Не менш важливим є уміння організувати свій час, так званий тайм-менеджмент. Роботу значно полегшує залучення віртуальних цифрових помічників. На рисунку 3.4 наведено піктограми найпоширеніших віртуальних цифрових помічників.



Рисунок 3.4 – Піктограми найпоширеніших віртуальних цифрових помічників

Віртуальний цифровий помічник (від англ. Virtual Digital Assistant, скорочено VDA) — програмний додаток, який фактично виконує роль особистого секретаря користувача: вирішує завдання планування графіка, організації і виконання повсякденних справ, контекстного пошуку інформації для потреб конкретної людини тощо.

Такі програмні додатки є незамінними для осіб із обмеженими фізичними можливостями та вадами зору.

Віртуальний цифровий помічник може створювати нагадування, полегшити пошук і онлайн-бронювання квитків та столиків, замовлення таксі. У процесі виконання завдань він здатний до самонавчання, аналізу поведінки й інтересів користувача. Наразі найвідомішими помічниками є Siri, Google Assistant, Microsoft Cortana.

Siri — персональний помічник, адаптований під iOS. Цей програмний додаток спілкується природною мовою, щоб відповідати на питання та давати рекомендації. Siri пристосовується до кожного користувача індивідуально, вивчаючи його особливості протягом тривалого часу.

Google Assistant — розумний персональний асистент, розроблений компанією Google і представлений на конференції Google I/O 18 травня 2016 року. Є додаткова функція — збір персональних даних користувачів у корпоративній системі машинного навчання.

Microsoft Cortana — персональна помічниця, покликана передбачати потреби користувача. За бажанням їй можна дати доступ до особистих даних — електронної пошти, історії пошуків у мережі. Всі ці дані вона використовуватиме для передбачення потреб власника.

Висновки

Отже, навчання сьогоднішнього дня повинно допомогти здобувачам освіти обрати гідну професію та підготуватися до ефективного виконання професійних обов'язків у майбутньому. Обираючи професію, важливо орієнтуватися в тенденціях ринку праці, щоб бути впевненим в можливості отримати роботу після завершення навчання. Для ознайомлення з вакансіями на ринку праці можна скористатися ресурсами Інтернету, наприклад, Труд (www.trud.gov.ua), Work.ua (www.work.ua/ua), Rabota.ua (rabota.ua/ua).

Питання та завдання до контролю знань студентів

- 1 Наведіть приклади нових професій, пов'язаних з використанням ІТ-технологій, у сфері машинобудування, банківської справи, медицини.
- 2 Які основні вимоги висувуються в сучасному світі до майбутніх фахівців, незалежно від обраної професії?
- 3 Наведіть приклади відомих вам відкритих освітніх платформ. Чи доводилося вам займатися самоосвітою за їх допомогою?
- 4 Чи існують україномовні відкриті освітні онлайн-курси? Якщо так, назвіть їх.
- 5 Що означає VDA? Назвіть найвідоміші приклади. Які види інструментів для полегшення планування, виконання та прогнозування результатів діяльності за допомогою комп'ютера ви можете назвати?
- 6 Що таке Project management та навіщо він потрібен? Який подібний інструмент від українських розробників ви можете назвати?
- 7 Навіщо потрібні віртуальні дошки? Наведіть приклади сервісів.
- 8 Для чого в навчальному процесі може бути використано популярний онлайн-сервіс Google Sites?
- 9 Наведіть приклад навчального онлайн-сервісу, за допомогою якого викладач може в режимі реального часу стежити за роботою студентів над віртуальним проектом.
- 10 Відкрийте сторінку з переліком онлайн-курсів на одній із україномовних платформ: Prometheus (<https://prometheus.org.ua/courses/>) або EdEra (<https://www.ed-era.com/>).
- 11 Зареєструйтеся на вибраному сайті та виберіть курс, який вас найбільше зацікавив.
- 12 Пройдіть курс у зручний для вас час. Розмістіть копію сертифіката в системі дистанційного навчання Moodle.

Лекція № 4

з навчальної дисципліни Інформатика

Тема Мультимедійні та гіпертекстові документи

Мета заняття Ознайомити із технологіями опрацювання мультимедійних даних, дослідити роль електронних медійних засобів в житті сучасного суспільства

Матеріально-технічне забезпечення та дидактичні засоби, ТЗН

1 Інтерактивний вид діяльності, створений засобами Moodle

2 Мультимедійний презентаційний матеріал

Час – 2 години

План проведення лекції

Структура лекції	Відведений час	Методичні вказівки
1 Організаційна частина	5 хв.	Включає в себе: <i>привітання</i> , яке має на меті привернути увагу, забезпечити контакт лектора з аудиторією; <i>визначення присутності студентів на занятті</i>
2 Актуалізація опорних знань, перевірка вивченого матеріалу та мотивація навчальної діяльності студентів	10 хв.	Включає в себе: <i>вступне зауваження, мотивацію актуальності теми, перевірку попереднього матеріалу, формулювання мети лекції, огляд головних питань теми (вказати питання та завдання для перевірки знань студентів)</i>
3 Основна частина (викладення навчальних питань лекції)	60 хв.	Головне місце приділяється <i>викладенню основного змісту матеріалу теми, його аналізу, узагальненню висунутих положень</i> . Для успіху лекції важливе значення має поділ матеріалу на розділи, основні питання
4 Заклучна частина Домашнє завдання: (відповідно до робочої програми)	5 хв.	Включає в себе: <i>узагальнення, теоретичні і фактичні висновки, закріплення вивченого на лекції матеріалу</i>

Література

1 Інформатика (рівень стандарту) : підруч. для 10-го (11-го) кл. закл. заг. серед. освіти / Й.Я. Ривкід Т.І. Лисенко, Л.А. Чернікова, В.В. Шакотько. — Київ : Генеза, 2018. — 144 с. : іл.

2 Інформатика (рівень стандарту) : підруч. для 10-го (11-го) кл. закл. заг. серед. освіти / О.О. Бондаренко, В.В. Ластовецький, О.П. Пилипчук, Є.А. Шестопапов. — Харків : Вид-во «Ранок», 2018. — 175 с. : іл.

Навчальні матеріали лекції

ВСТУП

Технології опрацювання мультимедіа зараз є одним із найперспективніших і найпопулярніших напрямків сучасної інформатики. Мета — створення продукту, який передає інформацію шляхом упровадження та використання нових технологій, набору зображень, текстів і даних, що супроводжуються звуком, відео, анімацією й іншими візуальними ефектами.

Розвиток мультимедійних технологій був логічним продовженням загального впровадження інформаційних систем в усі аспекти життя.

РОЛЬ ЕЛЕКТРОННИХ МЕДІЙНИХ ЗАСОБІВ У ЖИТТІ ЛЮДИНИ

Життя сучасної людини важко уявити без електронних медійних засобів. З появою різноманітних мобільних пристроїв, оснащених пристроями та програмами роботи з фото, аудіо та відео, значно змінилася, наприклад, організація навчання. Пояснення викладачем нового матеріалу може бути записати з використанням комп'ютерних засобів. Дослід Е. Резерфорда можна детально розглянути у відеофільмі на сайті YouTube. Розклад навчальних занять з нагадуванням про особливості підготовки до певних з них можна занести в електронний календар, синхронізувавши його з Google Календарем.

Кількість електронних газет і журналів зараз значно перевищує кількість друкованих, а їх оперативність реагування на події не варто й порівнювати. Книжки в електронному форматі поступово витісняють їх друковані аналоги.

Ще до початку цього тисячоліття створенням відеофільмів займалися лише окремі користувачі, що мали відеокамери. А редагувати їх вміли лише окремі з них. Така сама ситуація була й з аудіо даними. Розвиток інформаційних технологій опрацювання мультимедійних даних зробив створення власних відео та аудіозаписів, їх редагування не складнішою операцією, ніж написання текстового твору. Технології опрацювання мультимедійних даних стали загальнодоступними і використовуються в усіх галузях діяльності сучасної спільноти.

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ВЗАЄМОДІЇ З МУЛЬТИМЕДІЙНИМИ ДАНИМИ

Мультимедіа — комп'ютеризована технологія, яка об'єднує роботу з усіма джерелами даних, засіб подання різних видів інформації у цифровому вигляді.

Ведення особистих мережних щоденників, блогів, де кожен користувач може персоналізувати свою певну зону сайту — додати особисті аудіо та відеофайли, зображення, публікувати свої статті або ділитися новинами, створювати співтовариства, у яких будь-хто може залишити своє повідомлення, поділитися проблемою, почути безліч різноманітних думок із того чи іншого приводу та висловити свою, — стало можливим завдяки реалізації мультимедійних технологій у Всесвітній павутині.

На рисунку 4.1 наведено об'єкти мультимедіа.



Рисунок 4.1 – Об'єкти мультимедіа

ОГЛЯД ТЕХНОЛОГІЙ РЕАЛІЗАЦІЇ МУЛЬТИМЕДІА

Розглянемо деякі приклади таких технологій.

AJAX (англ. Asynchronous Java Script And XML — асинхронний JavaScript (мова програмування) та XML (мова розмітки веб-сторінок)) — дає змогу створювати набагато зручніші веб-інтерфейси користувача на тих сторінках сайтів, де необхідна активна взаємодія. До того ж користувач може переглядати далі вміст сайту, поки сервер все ще обробляє запит. Браузер лише довантажує потрібні користувачеві дані. Рівень використання AJAX значно підвищився після того, як компанія Google почала активно використовувати його у створенні своїх сайтів, таких як Gmail, GoogleMaps і GoogleSuggest.

Кешування — процес збереження часто запитуваних документів на проксісерверах (проміжних серверах, які виконують роль посередників між цільовими серверами та клієнтами) або машині користувача з метою запобігання їх постійному завантаженню з сервера-джерела і зменшення трафіку (об'єм інформації, який передається комп'ютерною мережею за певний час). Таким чином, інформація переміщується ближче до користувача.

WebRTC (англ. Web Real Time Communication — веб-комунікація в режимі реального часу) — відкритий стандарт мультимедійного зв'язку в реальному часі, який працює безпосередньо у веб-браузері. Створений, щоб усунути ризики для конфіденційності в умовах доступності локальних сервісів і мультимедійних потоків, підвищуючи якість онлайн-нарад, відеоконференцій та подібних сервісів. До кінця 2018 року кількість користувачів WebRTC сягне 1 млрд, а смартфонів, планшетів і комп'ютерів із підтримкою WebRTC — зросте до 4,7 млрд.

LongPolling (англ. «довге опитування») — технологія, яка створює можливість отримувати дані про нові події за допомогою «відкладеного запиту». Сервер отримує запит, але відправляє відповідь на нього не відразу, а лише тоді, коли відбудеться якась подія (наприклад, прийде нове повідомлення) або закінчиться заданий термін очікування.

Веб-сокети (англ. WebSockets — веб-гніздо, роз'єм) — технологія, яка дає змогу створювати інтерактивне з'єднання між клієнтом (браузером) та сервером для обміну повідомленнями в режимі реального часу.

Веб-сокети допомагають працювати з двонаправленим потоком даних. Під час їх використання немає потреби постійно повторювати запити. Достатньо виконати один запит і чекати відгуку сервера, який буде відправляти повідомлення в міру готовності.

Веб-сокети незамінні в процесі розробки чат-додатків, програмних додатків Інтернету речей та багатокористувацьких ігор.

Усі перелічені технології націлені саме на прискорення процесу передачі мультимедійної інформації та полегшення організації процесу інтерактивного спілкування користувачів в Інтернеті.

ПОНЯТТЯ ПРО МОВУ РОЗМІТКИ ГІПЕРТЕКСТОВОГО ДОКУМЕНТА

Ознакою сучасного Інтернету є спільна діяльність та активне спілкування користувачів. А на початку ХХІ століття панували класичні веб-ресурси, для яких були характерні статичні веб-сторінки, створені спеціалістами (розробниками команд мови розмітки гіпертекстового документа — HTML) та наповнені авторами, а користувач виступав лише в ролі читача.

HTML (англ. Hyper Text Markup Language — мова розмітки гіпертексту) — стандартна мова розмітки документів у всесвітній павутині, яка обробляється спеціальними програмами (браузерами) і відображається у вигляді документа у зручній для людини формі.

Гіпертекст — електронний документ, який містить зв'язки з іншими електронними документами. Такі зв'язки називаються гіперпосиланнями.

Слід зазначити, що HTML не є мовою програмування, вона призначена лише для розмітки сторінки, надання певного вигляду її складовим.

Контент (англ. content — вміст) — це інформаційне наповнення сайту.

Будь-який документ мовою HTML є набором елементів, водночас початок і кінець кожного елемента позначається спеціальними позначками — тегами.

Теги — команди мови HTML. HTML-теги — це ключові слова або символи, які записуються в кутові дужки.

Теги бувають двох видів: парні й непарні (їх ще називають поодинокими). Парні теги складаються з відкриваючого і закриваючого тегів. Теги нечутливі до регістра (тобто регістронезалежні), тому можуть бути написані як великими, так і малими літерами. Теги визначають, де починається й де закінчується HTML-елемент.

Текстові документи, що містять розмітку мовою HTML (такі документи зазвичай мають розширення .html або .htm), опрацьовуються спеціальними додатками, які відображають документ у його відформатованому вигляді.

ОГЛЯД HTML-ТЕГІВ ДЛЯ РОЗМІТКИ ДОКУМЕНТА

Теги заголовків `<h1>`, `<h2>`, `<h3>` та інші — використовуються для визначення заголовків різного рівня важливості на сторінці.

Теги `<p>` — використовуються для відображення тексту на сторінці.

Теги `<img>` — використовуються для відображення зображень на сторінці.

Теги `<a>` — використовуються для визначення посилань на інші сторінки або ресурси.

Теги `<ul>` та `<ol>` — використовуються для відображення списків з позначками або номерами відповідно.

Теги `<div>` та `<span>` — використовуються для групування та форматування контенту на сторінці.

Теги `<table>`, `<tr>`, `<th>` та `<td>` — використовуються для створення таблиць на сторінці.

Теги `<form>`, `<input>` та `<button>` — використовуються для створення форм та елементів управління на сторінці.

Теги `<script>` та `<style>` — використовуються для вставки JavaScript коду та CSS стилів на сторінку відповідно.

На рисунку 4.2 наведено структуру документа, що містить розмітку мовою HTML.

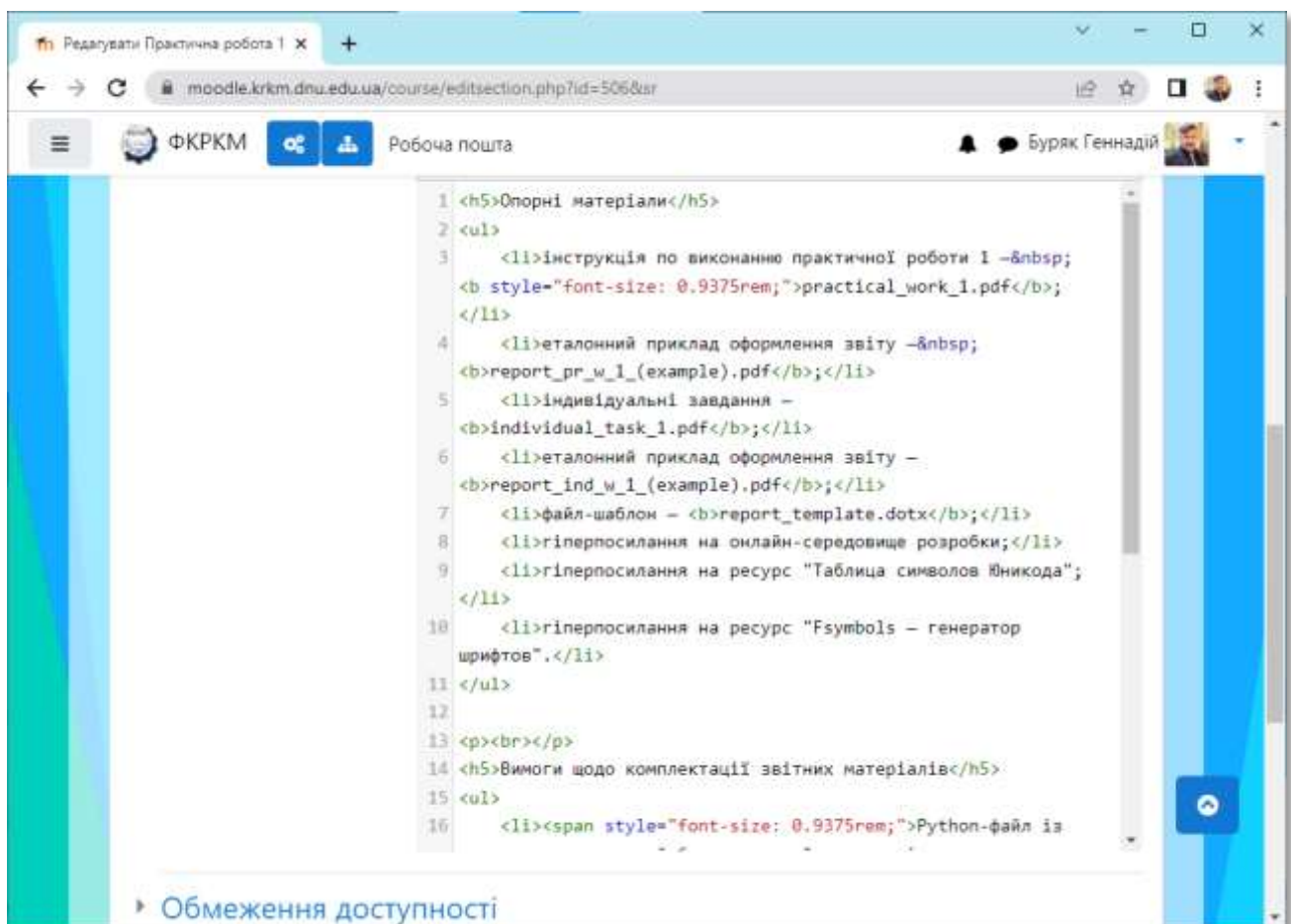


Рисунок 4.2 – Структура документа, що розмічений мовою HTML

Програмні додатки, які здатні обробляти документи, розмічені мовою HTML називаються браузером. Браузером (веб-переглядачем) називають програмне забезпечення для комп'ютера або іншого електронного пристрою, під'єднаного до Інтернету, який дає змогу користувачеві взаємодіяти з текстом, малюнками або іншою інформацією на гіпертекстовій веб-сторінці.

За допомогою HTML-тегів браузер розпізнає структуру документа. Отримавши цю інформацію, браузер використовує вбудовані в нього за замовчуванням правила про те, як відображати контент сторінки. Без використання HTML-тегів браузер виведе невідформатований текст, без відступів, заголовків, абзаців тощо.

Нині найпопулярнішими браузерами є MozillaFirefox, AppleSafari, Google-Chrome, Opera, Internet Explorer (Microsoft Edge).

На рисунку 4.3 наведено результат обробки браузером Google-Chrome структури документа, що розмічений мовою HTML.

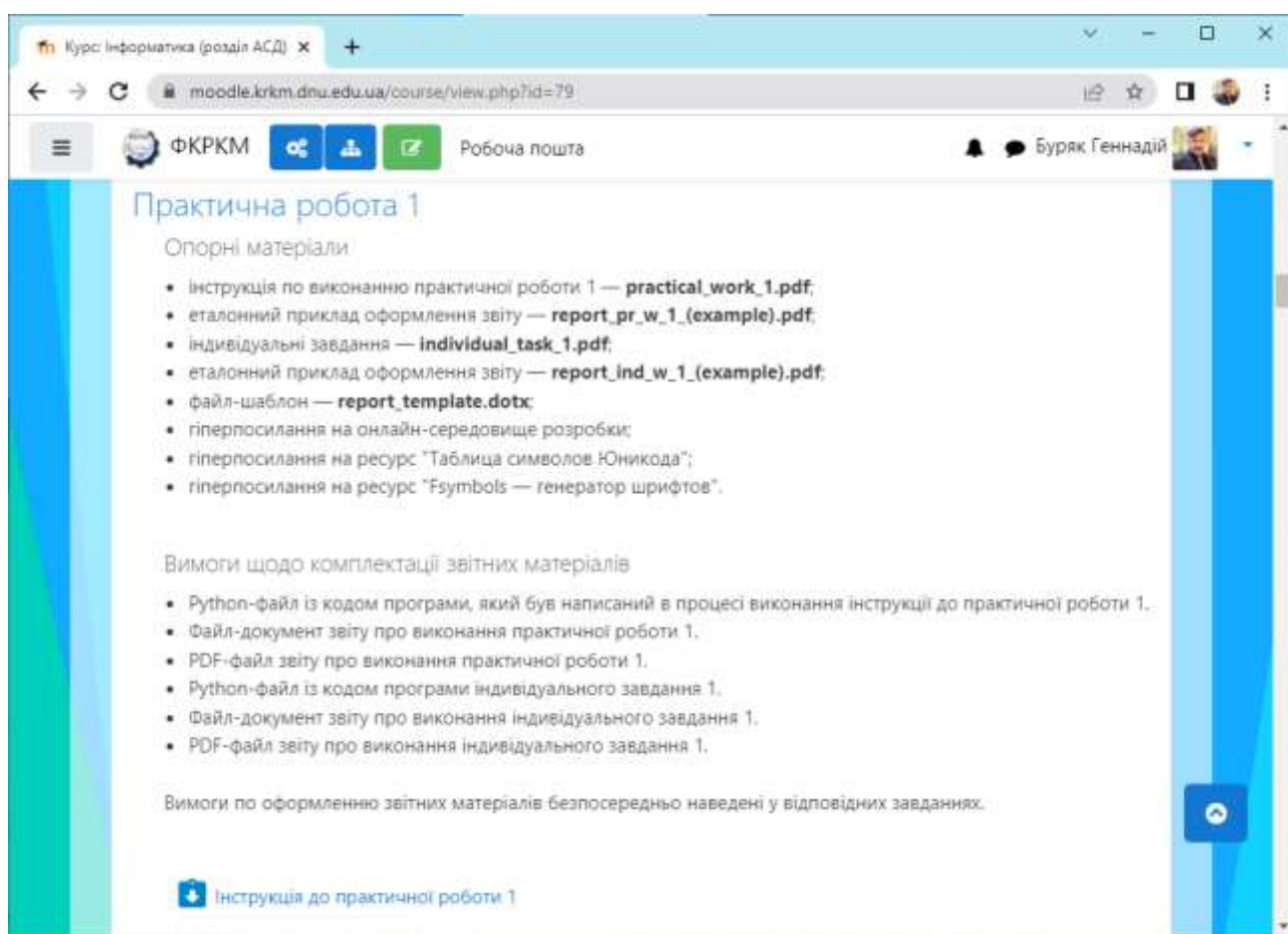


Рисунок 4.3 – Результат обробки документа браузером Google-Chrome

СТРУКТУРА СТАНДАРТНОЇ HTML-СТОРІНКИ

Сторінка завжди починається з відкриваючого тегу `<html>` та закінчується закриваючим тегом `</html>` і складається з двох обов'язкових блоків — голови (head) та тіла (body), які записуються послідовно.

У блоці `<head></head>` зберігається службова інформація, призначена допомогти браузеру в роботі з даними. Тут розташовані мета-теги, які використовуються для

зберігання інформації, призначеної для пошукових систем, а саме: опис сайта, ключові слова тощо.

Інформація є зазвичай невидимою для пересічного користувача, крім тегу `<title>`, у якому відображається назва сторінки сайта. Крім того, тут зберігаються посилання на фавікон, таблицю стилів (CSS) і скрипти (команди, записані мовою JavaScript), які використовуються на веб-сторінці.

Favicon (англ. favoritesicon) — іконка сайта, що відображається біля заголовка сторінки у вкладці або в списку закладок браузера.

Каскадні таблиці стилів (англ. Cascading Style Sheets, або скорочено CSS) — спеціальна мова, що використовується для опису зовнішнього вигляду сторінок, написаних мовами розмітки даних.

JavaScript (JS) — динамічна мова програмування, яка використовується для організації взаємодії з користувачем, обміну даними з сервером, зміни структури та зовнішнього вигляду веб-сторінки.

HTML5 (HyperText Markup Language version 5) — це остання версія мови розмітки гіпертексту, що використовується для створення веб-сторінок. HTML5 був прийнятий як стандарт в 2014 році і містить багато нових функцій та можливостей порівняно зі своїми попередниками.

Деякі з функцій HTML5 включають:

- підтримку багатьох нових тегів, таких як `<article>`, `<header>`, `<footer>`, `<nav>`, `<section>`, `<video>` тощо;
- покращену підтримку графіки та мультимедіа, зокрема вбудований плеєр для відео та аудіо;
- покращену підтримку форм та елементів вводу, таких як календарі, місцезнаходження та перетягування файлів;
- підтримку геолокації, що дозволяє веб-додаткам отримувати інформацію про місцезнаходження користувача;
- підтримку роботи в автономному режимі, який дозволяє веб-сторінкам зберігати дані та працювати без підключення до Інтернету;
- Підтримку нових API, таких як WebSocket, Web Workers та Web Storage, що дозволяють створювати більш складні веб-додатки;
- HTML5 дозволяє розробникам створювати більш динамічні та інтерактивні веб-сторінки, що підвищує їх функціональність та користувацький досвід.

СИСТЕМИ КЕРУВАННЯ ВМІСТОМ ДЛЯ ВЕБ-РЕСУРСІВ

Щоб створити власний сайт, ми маємо розробити макет, створити його за допомогою тегів HTML, селекторів CSS і сценаріїв JS. розмістити на певному сервері. Наповнити контентом і постійно оновлювати. Бажано зв'язати його із соціальними мережами, надати можливість зворотного зв'язку. Це означає, що ми повинні знати й уміти використовувати різноманітні API, програмувати мовою PHP і використовувати безліч інших технологій. Тобто ми одночасно маємо бути front-end та back-end розробниками.

Front-end — інтерфейс взаємодії між користувачем та базовою апаратно-адміністративною частиною (back-end). Інакше кажучи, frontend — це користувацький інтерфейс, а **back-end** — адміністративний, пов'язаний із керуванням сайтом.

API (англ. Application Programming Interface — інтерфейс програмування додатків) створює можливість використовувати ресурси інших програм.

PHP (англ. Hypertext Preprocessor — гіпертекстовий процесор) — мова програмування, створена для генерації HTML-сторінок на веб-сторінці сайта.

Система керування вмістом (англ. Content Management System, або CMS) — програмне забезпечення для організації спільного процесу створення, редагування й управління контентом веб-сайтів. На рисунку 4.5 наведено основні функції системи керування вмістом.



Рисунок 4.5 – Основні функції CMS

Зазвичай дизайн сайта змінюється рідко, тоді як зміни контенту можуть відбуватися не лише щодня, а й щогодини. Тому у своїй роботі CMS використовують так звані шаблони — спеціальні «порожні» заготовки сторінок, у яких дизайн сайта вже прописаний, і залишилося лише наповнити їх інформацією.

Принцип роботи всіх CMS заснований на поділі контенту (змісту) і дизайну (оформлення) сайта.

Кожна CMS має в своєму складі WYSIWYG-редактор (акронім від англ. What You See Is What You Get — що бачиш, те й отримуєш). За зовнішнім виглядом він дуже схожий на звичні текстові редактори офісних додатків, тому користувачеві не надто складно освоїти його. А наявність у системі великої кількості готових шаблонів дає змогу обрати потрібний дизайн буквально за лічені хвилини.

У процесі роботи над сайтом за допомогою візуального редактора сторінка відображається одразу в тому вигляді, начебто вона переглядається браузером, на відміну від звичайних редакторів, де записується послідовність команд-тегів. Для відображення такої сторінки необхідно викликати браузер, при цьому теги стають невидимими, і сторінка буде мати вигляд відповідно встановленим тегами розмітці. Наразі найбільш популярними є наступні безкоштовні CMS.

WordPress дуже популярний, постійно вдосконалюється, є прекрасним інструментом для новачків. WordPress — найкращий вибір для початківців завдяки простоті у використанні. Він особливо добре працює для невеликих та середніх веб-сайтів, блогів і невеликих магазинів електронної торгівлі.

Joomla теж проста у використанні, але для новачка вона буде складнішою за WordPress, оскільки для неї потрібні навички програмування для будь-якої зміни дизайну. Joomla є найліпшою для електронної комерції або веб-сайтів соціальних мереж.

Drupal орієнтована насамперед на програмістів (аж ніяк не призначена для новачка), має вельми потужний інструментарій із широкими можливостями. Тому вона буде ідеальним вибором для великих і серйозних сайтів, для яких необхідні знання програмування. Drupal потребує глибоких знань HTML, CSS і PHP.

Приклад стартових сторінок CMS наведено на рисунку 4.6.



Рисунок 4.6 – Стартові сторінки популярних CMS

ЗАГАЛЬНІ ПРИНЦИПИ АДМІНІСТРУВАННЯ САЙТОМ

Адміністрування сайту — це комплекс заходів щодо підтримки чіткого функціонування сайту, його працездатності, швидкої роботи, зручності.

Адміністрування сайту умовно можна розподілити на інформаційне та технічне.

Інформаційне адміністрування сайтом полягає у регулярному додаванні контенту:

- робота з заповнення сайту матеріалами, редагування та виправлення помилок розміщених матеріалів;
- оптимізація та підвищення зручності використання розміщеного контенту;
- підтримка актуальності та практичної цінності матеріалів сайту;
- своєчасне видалення й архівація застарілої інформації.

Технічне адміністрування сайтом полягає у забезпеченні будь-якому користувачеві постійного стабільного доступу до сайту, вибір та налаштування хостингу, цілодобовий контроль стану сервера, підтримка сайту в цілодобовому робочому стані.

ЕРГОНОМІКА РОЗМІЩЕННЯ ВІДОМОСТЕЙ НА ВЕБ-СТОРІНЦІ

Ергономіка (від грец. *ergon* — робота, *nomos* — закон) — наука, яка вивчає робочі процеси з метою створення оптимальних умов праці, що сприяє підвищенню її продуктивності.

Ергономічний сайт має бути добре пристосований для зручної та безпечної роботи користувача. Для цього існують певні правила, які наведено на рисунку 4.7.



Рисунок 4.7 – Правила ергономіки сайта

Юзабіліті (англ. *Usability* — зручність і простота використання) — це підхід, покликаний зробити сайти простими у використанні для користувача, який не потребує додаткового навчання, тобто має бути орієнтованим на нього інтерфейс. Потрібно створити умови, щоб користувач мав змогу інтуїтивно пов'язувати дії, які йому необхідно виконати на веб-сторінці.

Висновки

Отже, для ефективної роботи сайта необхідно забезпечити цілодобовий і безперебійний доступ до нього. Таку можливість створюють хостинг-провайдери, які надають в оренду певний дисковий простір на своїх серверах. Створення сайта з

урахуванням ергономіки може бути визначено як здатність ефективно реагувати на потреби користувачів і забезпечувати їм комфорт під час перегляду сторінки.

За допомогою інфографіки статистика на сайті перетворюється на візуально зрозумілу картинку з цікавим змістом. Широкоформатні фонові фото і відео для оформлення сайта надають динаміку веб-ресурсу і допомагають занурити користувача в певну історію. В такий спосіб підвищується популярність так званих синематографів — статичних картинок з елементами анімації другорядних деталей.

Питання та завдання до контролю знань студентів

- 1 Наведіть приклади об'єктів мультимедіа. Що таке кешування? Назвіть переваги його використання.
- 2 Де використовується інтерактивне спілкування користувачів?
- 3 Проведіть дослідження й наведіть приклади мультимедійних технологій, використання яких полегшує роботу користувача в Інтернеті.
- 4 Які інструменти необхідно використовувати, щоб створити власний сайт?
- 5 Що таке контент, фавікон, браузер? Який браузер ви використовуєте? Чому?
- 6 Знайдіть в Інтернеті відомості про розробника мови HTML. Назвіть причини появи CSS.
- 7 Проведіть порівняльний аналіз HTML5 та HTML4. У чому різниця?
- 8 Чим відрізняється front-end від back-end? Що таке CMS?
- 9 Назвіть функції систем керування вмістом. Що таке шаблон сторінки?
- 10 Що таке WYSIWYG? Наведіть приклади безкоштовних візуальних редакторів.
- 11 Дослідіть веб-системи керування вмістом сайтів. Зробіть висновки.
- 12 Що таке юзабіліті й ергономіка? Що, на вашу думку, їх відрізняє? Наведіть критерії ергономічного сайту. Які онлайн-сервіси підбору палітри ви знаєте?
- 13 Що таке синематографи? Наведіть приклади сайтів, де вони використовуються.
- 14 Виберіть сайт і проаналізуйте його з точки зору юзабіліті.
- 15 Проаналізуйте тренди поточного року та прослідкуйте, як вони змінювалися останнім часом.

Лекція № 5

з навчальної дисципліни Інформатика

Тема Документи та електронний документообіг, засоби обробки текстової інформації

Мета заняття Ознайомити із стандартами оформлення документації, розглянути уніфікованими засоби розробки документації, розвинути та оптимізувати уміння по обробці інформації офісного призначення

Матеріально-технічне забезпечення та дидактичні засоби, ТЗН

1. Інтерактивний вид діяльності, створений засобами Moodle

2. Мультимедійний презентаційний матеріал

Час – 2 години

План проведення лекції

Структура лекції	Відведений час	Методичні вказівки
1 Організаційна частина	5 хв.	Включає в себе: <i>привітання</i> , яке має на меті привернути увагу, забезпечити контакт лектора з аудиторією; <i>визначення присутності студентів на занятті</i>
2 Актуалізація опорних знань, перевірка вивченого матеріалу та мотивація навчальної діяльності студентів	10 хв.	Включає в себе: <i>вступне зауваження, мотивацію актуальності теми, перевірку попереднього матеріалу, формулювання мети лекції, огляд головних питань теми (вказати питання та завдання для перевірки знань студентів)</i>
3 Основна частина (викладення навчальних питань лекції)	60 хв.	Головне місце приділяється <i>викладенню основного змісту матеріалу теми, його аналізу, узагальненню висунутих положень</i> . Для успіху лекції важливе значення має поділ матеріалу на розділи, основні питання
4 Заклучна частина Домашнє завдання: (відповідно до робочої програми)	5 хв.	Включає в себе: <i>узагальнення, теоретичні і фактичні висновки, закріплення вивченого на лекції матеріалу</i>

Література

1 Інформатика (рівень стандарту) : підруч. для 10-го (11-го) кл. закл. заг. серед. освіти / Й.Я. Ривкід Т.І. Лисенко, Л.А. Чернікова, В.В. Шакотько. — Київ : Генеза, 2018. — 144 с. : іл.

Навчальні матеріали лекції

ВСТУП

В сучасному світі зростає кількість ділових транзакцій, які здійснюються в електронному форматі. Це стосується не лише фінансових транзакцій, а й обміну документами між підприємствами, установами та організаціями. Електронний документообіг — це процес створення, обміну, зберігання та управління документами в електронному форматі.

Основною метою електронного документообігу є покращення ефективності та швидкості обробки документів, зменшення витрат на їх обробку та зберігання, збільшення надійності та безпеки обміну даними. Електронний документообіг включає в себе не лише створення електронних документів, а й їх обробку, зберігання та передачу з використанням спеціалізованих програмних засобів.

За допомогою електронного документообігу можна здійснювати ефективний обмін даними між підприємствами, установами та організаціями, що забезпечує швидкість та точність обробки інформації. Крім того, використання електронного документообігу дозволяє зменшити витрати на паперову документацію, а також на її зберігання та обробку.

Зараз електронний документообіг стає все більш популярним серед компаній та організацій, що дозволяє підвищити їх ефективність та конкурентоспроможність. Тому вивчення та застосування електронного документообігу є актуальним завданням для багатьох фахівців у сфері бізнесу та управління.

ПОНЯТТЯ ЕЛЕКТРОННОГО ДОКУМЕНТООБІГУ

Електронний документообіг (ЕДО) — це система обміну документами в електронному форматі між різними учасниками, яка забезпечує швидкість та надійність передачі даних. ЕДО може використовуватися для обміну будь-якими документами, від рахунків-фактур до договорів та інших юридичних документів.

Основні переваги використання ЕДО полягають в тому, що він зменшує витрати на паперову документацію, скорочує час обробки документів та покращує комунікацію між учасниками процесу. Крім того, ЕДО дозволяє зберігати та керувати електронними даними, забезпечуючи більшу безпеку та контроль за їх зберіганням та передачею.

Серед інших переваг ЕДО можна виділити:

- зменшення ризиків втрати або пошкодження документів, оскільки вони зберігаються в електронному форматі та можуть бути забезпечені резервними копіями;
- швидкість обміну даними та можливість простої та швидкої перевірки стану документів в режимі реального часу;
- зменшення потреби в просторі для зберігання паперових документів та зменшення витрат на їх зберігання;
- покращення відносин з партнерами та покупцями завдяки більш прозорій та простій системі обміну даними.

– загалом, ЕДО є важливою частиною цифрової трансформації бізнесу та дозволяє підвищувати ефективність роботи, знижувати витрати та покращувати якість обслуговування клієнтів.

Офісні електронні документи — це електронні документи, які створюються в програмах для офісного користування, таких як текстові процесори, електронні таблиці, програми для презентацій та інші. Ці документи можуть включати різноманітну інформацію, таку як текст, зображення, відео, таблиці, графіки та інші типи даних.

Офісні електронні документи дозволяють вирішувати багато завдань в офісному середовищі, такі як створення документів, редагування тексту, складання бюджетів, створення презентацій, розрахунок фінансових показників та інші. До найбільш популярних програм для створення офісних документів належать Microsoft Office, Google Docs, LibreOffice та OpenOffice.

Офісні електронні документи можуть бути збережені у різних форматах, таких як DOCX, XLSX, PPTX, PDF та інші. Ці формати дозволяють зберігати документи в електронному вигляді та забезпечують їх збереження та передачу без змін на різних пристроях та програмах.

Офісні електронні документи стали невід'ємною частиною сучасного бізнесу та підприємництва. Вони дозволяють ефективно виконувати завдання, підвищувати продуктивність та ефективність роботи, а також зменшувати витрати на офісну документацію та її обробку.

ОГЛЯД ПРОГРАМНИХ ЗАСОБІВ ДЛЯ СТВОРЕННЯ ЕЛЕКТРОННИХ ДОКУМЕНТІВ

Існує багато програм для створення електронних документів, які можуть бути використані для різних потреб. Ось огляд деяких з найпопулярніших програм для створення електронних документів:

Microsoft Word — це програма, що входить до складу пакету Microsoft Office, яка дозволяє створювати, редагувати та форматовувати текстові документи. Microsoft Word є найпопулярнішою програмою для створення документів у форматі DOCX.

Google Docs — це безкоштовна хмарна програма для створення документів, що дозволяє працювати з документами в режимі онлайн. Google Docs може бути використана для створення та редагування текстових документів, таблиць та презентацій.

LibreOffice Writer — це безкоштовна альтернатива Microsoft Word, яка дозволяє створювати та редагувати текстові документи, а також зберігати їх у різних форматах, включаючи DOCX та PDF.

Adobe Acrobat — це програма для роботи з PDF-документами, яка дозволяє створювати, редагувати та конвертувати документи у форматі PDF. Adobe Acrobat також має можливості для захисту та підписування документів.

Evernote — це програма для зберігання та організації нотаток, що може бути використана для створення текстових документів, списків завдань та інших документів. Evernote також має можливості для синхронізації документів між різними пристроями.

Dropbox Paper — це безкоштовна хмарна програма для створення документів, яка дозволяє спільно працювати над документами з іншими користувачами. Dropbox Paper також має можливості для додавання зображень, відео та інших мультимедійних елементів до документів.

ОСОБЛИВОСТІ ВЗАЄМОДІЇ ІЗ MICROSOFT WORD

Microsoft Word є однією з найпопулярніших програм для створення документів та редагування тексту. Основним призначенням Microsoft Word є створення текстових документів, таких як листи, звіти, резюме, наукові статті, презентації та багато іншого.

Microsoft Word дозволяє використовувати шрифти різного розміру та стилю, колір тексту, підкреслення, курсив та жирність. Також є можливість вставки таблиць, списків та інших елементів для зручності читання та зрозумілості тексту.

Microsoft Word має інструменти для редагування тексту, такі як корекція помилок, перенесення слів, зміна порядку слів та речень, а також можливість знаходження та заміни певних слів та фраз.

Microsoft Word може легко інтегруватися з іншими програмами, такими як Excel, PowerPoint, Outlook та OneNote, що дозволяє використовувати їх функціональність прямо в документі Word.

Microsoft Word дозволяє декільком користувачам одночасно працювати над одним документом, що дозволяє швидко та ефективно обговорювати та вносити зміни до документу.

Microsoft Word має можливості для створення таблиць та діаграм, що дозволяє легко та зручно представляти дані та інформацію.

Microsoft Word дозволяє зберігати документи в різних форматах, таких як DOCX, PDF, HTML, TXT, RTF та інші, що дозволяє легко розповсюджувати документи та використовувати їх у різних контекстах.

ОГЛЯД ІНСТРУМЕНТІВ ВЗАЄМОДІЇ З РЕДАКТОРОМ MICROSOFT WORD

Стрічка інструментів Microsoft Word — це набір піктограм, які містяться в верхній частині програми та які надають доступ до різноманітних інструментів для редагування, форматування та оформлення документа.

Група інструментів "Файл"

Група інструментів "Файл" редактора Microsoft Word містить основні операції з файлами, такі як відкриття, збереження, друк і експорт документа. Ці інструменти дозволяють користувачам працювати з документами в Microsoft Word, зберігати їх на комп'ютері, надсилати і друкувати, а також експортувати в інші формати.

До складу групи інструментів "Файл" входять такі елементи:

- "Відкрити" — цей інструмент дозволяє відкрити документ, який був збережений раніше або знайдений на комп'ютері;
- "Зберегти" — цей інструмент зберігає поточний документ в файл на комп'ютері.
- "Зберегти як" — цей інструмент дозволяє зберегти поточний документ з новою назвою або в іншому форматі;
- "Новий" — цей інструмент створює новий порожній документ;
- "Друк" — цей інструмент дозволяє надрукувати документ на принтері або в PDF-файл;
- "Експорт" — цей інструмент дозволяє експортувати документ у форматі PDF або іншому форматі;
- "Закрити" — цей інструмент закриває поточний документ.

Стрічка інструментів "Головна"

Група інструментів "Головна" редактора Microsoft Word містить основні інструменти форматування тексту, зміни розміру та типу шрифту, розташування та вирівнювання тексту, а також інші інструменти, які дозволяють змінювати зовнішній вигляд тексту в документі.

На рисунок 5.1 наведено стрічку інструментів "Головна".

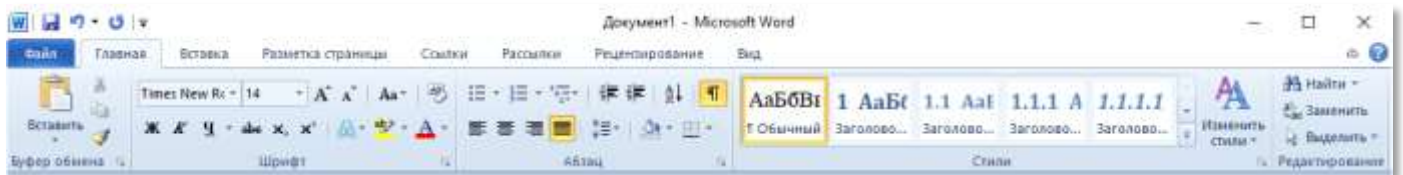


Рисунок 5.1 – Стрічка інструментів "Головна"

До складу групи інструментів "Головна" входять такі елементи:

- "Шрифт" — цей інструмент дозволяє змінити розмір та тип шрифту, а також встановити інші параметри шрифту, такі як жирність, курсивність, підкреслення і кольори;
- "Абзац" — цей інструмент дозволяє вирівняти текст за лівим, правим, центральним або обидвом краями сторінки, додати відступи, міжрядковий інтервал, інтервали між абзацами та інші параметри абзацу;
- "Таблиця" — цей інструмент дозволяє створювати таблиці, редагувати їх, додавати рядки та стовпці, змінювати розмір і форматування комірок;
- "Картинка" — цей інструмент дозволяє додавати зображення, фотографії та ілюстрації в документ;

- "Нумерований список" та "Маркований список" — ці інструменти дозволяють створювати списки з нумерованими або маркованими елементами;
- "Вирівняти" — цей інструмент дозволяє вирівняти текст за лівим, правим, центральним або обидвом краями сторінки.

Ці інструменти дозволяють користувачам формувати текст, створювати таблиці та списки, додавати зображення та змінювати їх розмір і форматування. Крім того, користувачі можуть вирівнювати текст за лівим, правим або центральним краєм сторінки.

Стрічка інструментів "Вставка"

Група інструментів "Вставка" редактора Microsoft Word містить інструменти для вставлення об'єктів в документ, таких як зображення, таблиці, графіки, гіперпосилання, символи, емодзі, відео та інші об'єкти. Ця група інструментів дозволяє додавати різні типи вмісту до документу та налаштовувати їх відображення і взаємодію з користувачем.

На рисунок 5.2 наведено стрічку інструментів "Вставка".

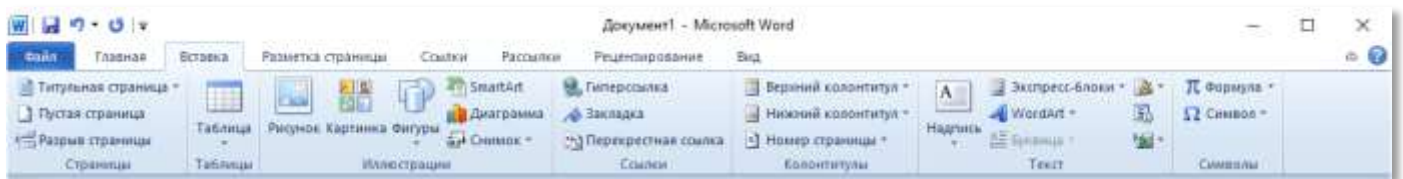


Рисунок 5.2 – Стрічка інструментів "Вставка"

До складу групи інструментів "Вставка" входять такі елементи:

- "Об'єкт" — цей інструмент дозволяє додавати в документ об'єкти, такі як Excel-таблиці, графіки, діаграми та інші;
- "Зображення" — цей інструмент дозволяє додавати зображення в документ, налаштовувати їх розмір та форматування;
- "Фігури" — цей інструмент дозволяє додавати різні фігури, такі як лінії, стрілки, прямокутники, еліпси та інші, а також налаштовувати їх розмір, кольори та інші параметри;
- "Таблиця" — цей інструмент дозволяє створювати таблиці, редагувати їх, додавати рядки та стовпці, змінювати розмір і форматування комірок;
- "Гіперпосилання" — цей інструмент дозволяє створювати гіперпосилання на інші сторінки, документи, веб-сторінки та інші об'єкти;
- "Символи" — цей інструмент дозволяє додавати спеціальні символи, такі як знаки пунктуації, математичні символи, символи валют та інші.

Ці інструменти дозволяють користувачам додавати різні типи об'єктів до документу, такі як зображення, таблиці, фігури, гіперпосилання та інші.

Стрічка інструментів "Розмітка сторінки"

Група інструментів "Розмітка сторінки" редактора Microsoft Word містить інструменти, які дозволяють налаштувати розмір, орієнтацію та поля сторінки документа. Ця група інструментів дозволяє налаштовувати сторінки документа залежно від потреб користувача та відповідно до вимог конкретної видавничої задачі.

На рисунок 5.3 наведено стрічку інструментів "Розмітка сторінки".



Рисунок 5.3 – Стрічка інструментів "Розмітка сторінки"

До складу групи інструментів "Розмітка сторінки" входять такі елементи:

- "Розмір" — цей інструмент дозволяє вибрати стандартний або налаштований розмір сторінки документа;
- "Орієнтація" — цей інструмент дозволяє вибрати портретну або кільцеву орієнтацію сторінки документа;
- "Поля" — цей інструмент дозволяє налаштувати розмір полів на сторінці документа;
- "Верхній колонтитул" — цей інструмент дозволяє налаштувати верхній колонтитул сторінки документа, додавати в нього текст, зображення, номер сторінки та інші елементи;
- "Нижній колонтитул" — цей інструмент дозволяє налаштувати нижній колонтитул сторінки документа, додавати в нього текст, зображення, номер сторінки та інші елементи.

Ці інструменти дозволяють налаштовувати розмір, орієнтацію та поля сторінки документа, а також додавати верхній та нижній колонтитули з різними елементами в залежності від потреб користувача. Крім того, інструменти групи "Розмітка сторінки" дозволяють користувачам налаштовувати документи для друку та електронного форматування, включаючи налаштування розміру та меж полів, орієнтації сторінки тощо.

Стрічка інструментів "Посилання"

Група інструментів "Посилання" редактора Microsoft Word містить інструменти, які дозволяють вставляти посилання на інші документи, веб-сторінки, мережеві ресурси, електронну пошту та інші джерела. Ці інструменти дозволяють створювати гіперпосилання, таблиці змісту, посилання на джерела, що використовуються в документі, та інші типи посилань.

На рисунок 5.4 наведено стрічку інструментів "Посилання".

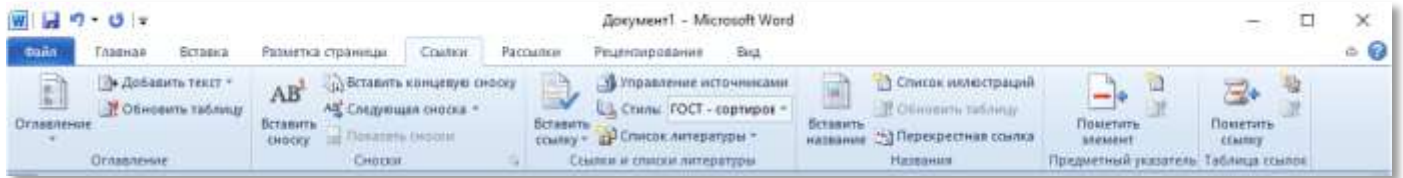


Рисунок 5.4 – Стрічка інструментів "Посилання"

До складу групи інструментів "Посилання" входять такі елементи:

- "Гіперпосилання" — цей інструмент дозволяє створювати посилання на інші документи, веб-сторінки, мережеві ресурси, електронну пошту та інші джерела. Крім того, цей інструмент дозволяє змінювати та видаляти існуючі гіперпосилання;
- "Таблиця змісту" — цей інструмент дозволяє створювати автоматично згенеровану таблицю змісту для документа, яка містить гіперпосилання на всі розділи, заголовки та підрозділи документа;
- "Посилання на джерела" — цей інструмент дозволяє створювати посилання на джерела, які використовуються в документі, такі як книги, журнали, статті та інші джерела;
- "Закладки" — цей інструмент дозволяє створювати закладки у документі, що можна використовувати для створення посилань на конкретні частини документа;
- "Перехід" — цей інструмент дозволяє створювати посилання на різні частини документа, такі як сторінки, заголовки та закладки;

Ці інструменти дозволяють користувачам створювати різні типи посилань у документі, такі як гіперпосилання, таблиці змісту, посилання на джерела та інші.

Стрічка інструментів "Розсилки"

Група інструментів "Розсилки" редактора Microsoft Word містить інструменти для створення та відправлення масових електронних листів, які містять персоналізовані дані, такі як ім'я, прізвище, адреса електронної пошти та інші дані.

Ці інструменти дозволяють користувачам створювати та відправляти масові електронні листи з персоналізованими даними, такі як ім'я та прізвище отримувача. Ці інструменти особливо корисні для бізнес-користувачів та маркетологів, які мають великі списки розсилки та потребують ефективного способу взаємодії зі своїми клієнтами та споживачами.

На рисунок 5.5 наведено стрічку інструментів "Розсилки".

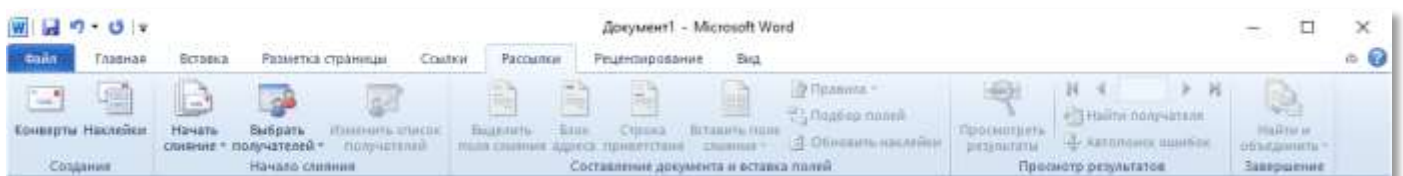


Рисунок 5.5 – Стрічка інструментів "Розсилки"

До складу групи інструментів "Розсилки" входять такі елементи:

- "Перелік розсилки" — цей інструмент дозволяє створювати список адрес електронної пошти, який можна використовувати для відправлення масових електронних листів;
- "Електронний лист" — цей інструмент дозволяє створювати шаблон електронного листа, який містить текст повідомлення та персоналізовані дані, такі як ім'я та прізвище отримувача;
- "Поля" — цей інструмент дозволяє вставляти в текст листа поля, які будуть автоматично заповнені даними зі списку розсилки, такі як ім'я та прізвище отримувача;
- "Відправити" — цей інструмент дозволяє відправити електронний лист на всі адреси електронної пошти зі списку розсилки.

Стрічка інструментів "Рецензування"

Група інструментів "Рецензування" редактора Microsoft Word містить інструменти для здійснення редагування та відстеження змін у документах. Вона дозволяє користувачам додавати коментарі, виправлення та інші зауваження до документів і зберігати їх відносно робочих процесів.

На рисунок 5.6 наведено стрічку інструментів "Рецензування".

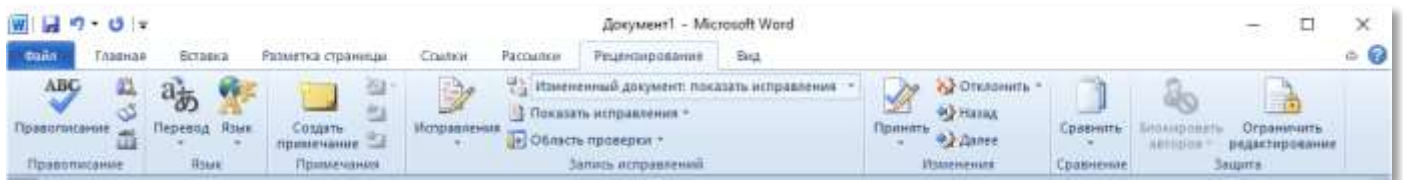


Рисунок 5.6 – Стрічка інструментів "Рецензування"

До складу групи інструментів "Рецензування" входять такі елементи:

- "Відстеження змін" — цей інструмент дозволяє відстежувати всі зміни, внесені до документа, включаючи видалені, переміщені та додані текстові блоки;
- "Коментарі" — цей інструмент дозволяє користувачам додавати коментарі до тексту документа, щоб надати замітки, поради та зауваження для інших членів команди;
- "Виправлення" — цей інструмент дозволяє користувачам вносити виправлення в текст документа без зміни його форматування;
- "Захист" — цей інструмент дозволяє користувачам захистити документ від неповноважного доступу та редагування, а також дозволяє встановлювати рівні дозволу для користувачів, які мають доступ до документа.
- "Перевірка" — цей інструмент дозволяє користувачам перевіряти правопис, граматику та інші помилки в документі перед його відправкою.

— 11 —

На рисунок 5.7 наведено стрічку інструментів "Вид".



"Масштаб" цей інструмент дозволяє збільшувати або

- Ці інструменти дозволяють користувачам налаштовувати вигляд документа для

Стрічка інструментів "Формат"

Вона дозволяє користувачам виконувати детальне форматування документа та налаштування його вигляду. Microsoft Word має ряд інструментів для редагування впроваджених рисунків, що дозволяє користувачам вносити зміни в зображення без необхідності використання зовнішніх програм. До таких інструментів належать:

- "Формат зображення" — цей інструмент дозволяє користувачам налаштовувати розмір, розташування та інші параметри зображення.
- "Кадрування" — цей інструмент дозволяє вибирати форму зображення, змінювати його розмір та вирівнювати відносно тексту.
- "Коригування кольорів" — цей інструмент дозволяє користувачам змінювати кольори зображення та налаштовувати яскравість, контрастність та інші параметри.
- "Повернення" — цей інструмент дозволяє користувачам повертати зображення на певний кут.
- "Розміщення тексту" — цей інструмент дозволяє користувачам вирівнювати текст відносно зображення, а також вставляти текст на зображення.
- "Компресування зображення" — цей інструмент дозволяє користувачам зменшувати розмір зображення, що зменшує об'єм файлу документа.

Ці інструменти дозволяють користувачам вносити зміни в зображення без необхідності використання зовнішніх програм, що зменшує час, необхідний для редагування документа та поліпшує продуктивність роботи.

На рисунок 5.8 наведено стрічку інструментів "Формат" із адаптованою палітрою інструментів редагування растрових зображень.

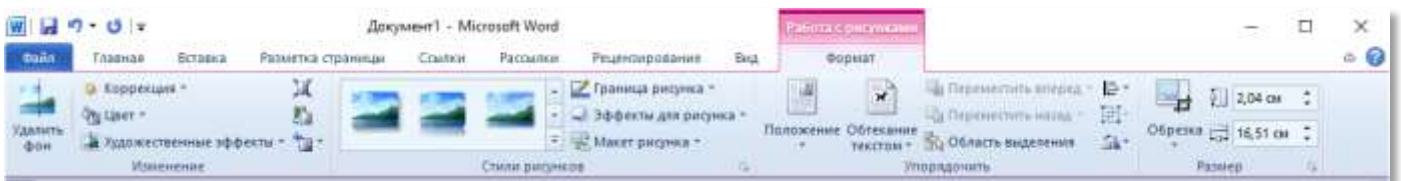


Рисунок 5.8 – Стрічка інструментів "Формат" для редагування растрових зображень

На рисунок 5.9 наведено стрічку інструментів "Формат" із адаптованою палітрою інструментів редагування векторних примітивів.



Рисунок 5.9 – Стрічка інструментів "Формат" для редагування векторних примітивів

Інструменти для оформлення таблиць засобами Microsoft Word

Оформлення та редагування таблиць засобами Microsoft Word здійснюється із залученням інструментів, які згруповані у палітрах "Конструктор" та "Макет". Зазначені палітри інструментів активуються уразі виділення таблиці.

Група інструментів "Конструктор" у редакторі Microsoft Word дозволяє користувачам створювати та налаштовувати таблиці в документі. Ці інструменти дозволяють легко формувати та організовувати дані у вигляді таблиці. На рисунок 5.10 наведено стрічку інструментів "Конструктор".



Рисунок 5.10 – Стрічка інструментів "Конструктор"

До інструментів групи "Конструктор" належать:

- "Вставити таблицю" — цей інструмент дозволяє користувачам вставляти таблиці в документ;
- "Розмір таблиці" — цей інструмент дозволяє користувачам визначити кількість рядків та стовпців у таблиці;
- "Формат таблиці" — цей інструмент дозволяє користувачам налаштовувати ширину стовпців та висоту рядків, а також визначати колір тла та рамок таблиці.
- "Об'єднати комірки" — цей інструмент дозволяє користувачам об'єднувати комірки в таблиці для створення більш складних макетів;
- "Розтягнути комірки" — цей інструмент дозволяє користувачам розтягувати комірки в таблиці, щоб вміст виглядав більш чітко та зручно;
- "Форматування таблиці" — цей інструмент дозволяє користувачам налаштовувати форматування тексту в комірках таблиці, таке як шрифт, розмір шрифту, колір тексту та інші параметри;
- "Сортування та фільтрація" — цей інструмент дозволяє користувачам сортувати та фільтрувати дані в таблиці за різними критеріями.

Група інструментів "Макет" дозволяє користувачам редагувати вигляд таблиці та налаштовувати її макет. Ці інструменти дозволяють користувачам змінювати розташування таблиці в документі та додавати елементи дизайну, що покращують її вигляд. На рисунок 5.11 наведено стрічку інструментів "Макет".



Рисунок 5.11 – Стрічка інструментів "Макет"

До інструментів групи "Макет" належать:

- "Розміщення" — цей інструмент дозволяє користувачам вибирати різні способи розміщення таблиці в документі, такі як горизонтальне або вертикальне розміщення;
- "Обтікання текстом" — цей інструмент дозволяє користувачам встановлювати параметри обтікання текстом для таблиці, такі як розміщення тексту поруч з таблицею або відстань між текстом та таблицею;
- "Дизайн" — цей інструмент дозволяє користувачам налаштовувати дизайн таблиці, такий як колір фону, стиль рамки, шрифт та колір тексту;
- "Формат таблиці" — цей інструмент дозволяє користувачам налаштовувати формат таблиці, такий як ширина стовпців та висота рядків, а також встановлювати автоматичне розширення стовпців;
- "Вставити" — цей інструмент дозволяє користувачам вставляти елементи дизайну в таблицю, такі як заголовки, підзаголовки, номери сторінок та інші елементи.

Ці інструменти дозволяють користувачам створювати та налаштовувати таблиці в документі, що полегшує організацію та візуалізацію даних, а також налаштовувати макет таблиці та додавати елементи дизайну, що покращують її вигляд. Вони допомагають зробити таблицю більш привабливою та зрозумілою для читачів.

Інструменти для оформлення формул засобами Microsoft Word

Microsoft Word має ряд інструментів для оформлення формул, які розташовані в групі "Інструменти формул" на вкладці "Рівняння". У Microsoft Word доступні два редактори формул.

Редактор формул Microsoft.

Це стандартний редактор формул, який доступний у всіх версіях Microsoft Word. Він має велику кількість функцій та можливостей для створення складних математичних формул.

LaTeX

Це додатковий редактор формул, який можна встановити окремо в Microsoft Word або використовувати в інших програмах для наукових розрахунків. Він базується на системі верстки LaTeX, яка дозволяє створювати складні формули у вигляді текстового файлу з використанням спеціальних команд та символів.

Обидва редактори формул можуть бути корисними залежно від потреб користувача. Редактор формул Microsoft зручний для швидкої створення простих формул, тоді як LaTeX дозволяє створювати складні формули з більш точною настройкою і форматуванням.

На рисунк 5.12 наведено стрічку інструментів "Конструктор (Робота із формулами)".



Рисунок 5.12 – Стрічка інструментів "Конструктор (Робота із формулами)"

Деякі з цих інструментів включають:

Редактор формул

Це інструмент, який дозволяє користувачам створювати складні математичні формули за допомогою різних математичних символів та функцій. Редактор формул забезпечує підтримку виразів з різних галузей науки, таких як фізика, хімія та інженерія.

Галерея формул

Це інструмент, який містить шаблони популярних математичних формул, які можна вставити безпосередньо в документ.

Символи

Це інструмент, який містить символи, які можуть бути використані в формулах. У галереї символів можна знайти знаки операцій, математичні символи, грецькі літери та інші символи.

Розмір і позиція

Цей інструмент дозволяє користувачам змінювати розмір та позицію формули в документі.

Стилі формул

Це інструмент, який дозволяє користувачам змінювати стиль формули, такий як розмір шрифту, колір та інші параметри.

Ці інструменти дозволяють користувачам створювати професійно оформлені формули у Microsoft Word.

Висновки

За допомогою Microsoft Word можна створювати різноманітні текстові документи, включаючи листи, звіти, резюме, наукові статті та інші документи. Окрім того, Microsoft Word має можливості для спільної роботи над документами в режимі реального часу та можливість збереження документів у різних форматах, що дозволяє забезпечити сумісність з іншими програмними додатками. Редактор оснащено інтуїтивним інтерфейсом. Більшість функцій доступні у контекстних меню або у ергономічних панелях інструментів.

Отже, Microsoft Word є корисним інструментом для створення різноманітних текстових документів. Він має багато функцій та можливостей, що дозволяє забезпечити професійний вигляд документів та полегшити спільну роботу над ними.

Питання та завдання до контролю знань студентів

- 1 Які інструменти в Microsoft Word допомагають підвищити ефективність написання документів?
- 2 3. Які інструменти доступні у Microsoft Word для форматування тексту?
- 3 Як використовувати функцію автоматичного заповнення в Microsoft Word?
- 4 Як користуватися інструментами форматування тексту в Microsoft Word?
- 5 Як вставити рисунки та таблиці у документ Microsoft Word?
- 6 Як використовувати функцію перевірки правопису та граматики в Microsoft Word?
- 7 Як створити змінну в Microsoft Word та використовувати її у документі?
- 8 Які інструменти допомагають змінити розмір сторінки та орієнтацію в Microsoft Word?
- 9 Як зробити змінний номер сторінки в Microsoft Word?
- 10 Як створити власний шаблон документа в Microsoft Word?
- 11 Як вставити гіперпосилання у документ Microsoft Word?
- 12 Як вставити зображення або графіку в документ Microsoft Word?
- 13 Як зберегти документ у форматі PDF у Microsoft Word?
- 14 Як створити таблицю в Microsoft Word?
- 15 У який спосіб впровадити у документ таблицю Microsoft Excel?
- 16 Як до документа додати Microsoft Excel графіки та діаграми?
- 17 Як вставити посилання на інший документ або веб-сторінку у Microsoft Word?
- 18 Як відформатувати сторінку документа у Microsoft Word?
- 19 Як використовувати граматичний і правописний перевірка в Microsoft Word?
- 20 Як створити власний шаблон документа у Microsoft Word?

Лекція № 6

з навчальної дисципліни Інформатика

Тема Програмні засоби обробки табличної інформації

Мета заняття Дослідити програмні засоби, що призначені для обробки табличної інформації
повідомлення, дані, інформаційні процеси

Матеріально-технічне забезпечення та дидактичні засоби, ТЗН

1. Інтерактивний вид діяльності, створений засобами Moodle

2. Мультимедійний презентаційний матеріал

Час – 2 години

План проведення лекції

Структура лекції	Відведений час	Методичні вказівки
1 Організаційна частина	5 хв.	Включає в себе: <i>привітання</i> , яке має на меті привернути увагу, забезпечити контакт лектора з аудиторією; <i>визначення присутності студентів на занятті</i>
2 Актуалізація опорних знань, перевірка вивченого матеріалу та мотивація навчальної діяльності студентів	10 хв.	Включає в себе: <i>вступне зауваження</i> , <i>мотивацію актуальності теми</i> , <i>перевірку попереднього матеріалу</i> , <i>формулювання мети лекції</i> , <i>огляд головних питань теми</i> (вказати питання та завдання для перевірки знань студентів)
3 Основна частина (викладення навчальних питань лекції)	60 хв.	Головне місце приділяється <i>викладенню основного змісту матеріалу</i> теми, його <i>аналізу</i> , <i>узагальненню висунутих положень</i> . Для успіху лекції важливе значення має поділ матеріалу на розділи, основні питання
4 Заклучна частина Домашнє завдання: (відповідно до робочої програми)	5 хв.	Включає в себе: <i>узагальнення</i> , <i>теоретичні і фактичні висновки</i> , <i>закріплення вивченого на лекції матеріалу</i>

Література

1 Інформатика (рівень стандарту) : підруч. для 10-го (11-го) кл. закл. заг. серед. освіти / Й.Я. Ривкід Т.І. Лисенко, Л.А. Чернікова, В.В. Шакотько. — Київ : Генеза, 2018. — 144 с. : іл.

2 Інформатика (рівень стандарту) : підруч. для 10-го (11-го) кл. закл. заг. серед. освіти / О.О. Бондаренко, В.В. Ластовецький, О.П. Пилипчук, Є.А. Шестопапов. — Харків : Вид-во «Ранок», 2018. — 175 с. : іл.

Навчальні матеріали лекції

ВСТУП

У процесі дослідження навколишнього світу людина відображає результати спостережень, з одного боку, для подання їх у вигляді, зручному для аналізу, а з іншого, — для зберігання й передавання. Моделювання є обов'язковим етапом багатьох наукових досліджень, оскільки створює можливість вивчати об'єкти та процеси, які неможливо безпосередньо спостерігати або відтворити експериментально.

Залежно від області застосування розрізняють такі моделі:

- навчальні: тренажери, наочні засоби, навчальні програми;
- дослідні: моделі кораблів, літаків тощо;
- науково-технічні: наприклад, прилад для дослідження розряду блискавки;
- ігрові: ділові, економічні ігри;
- імітаційні: виявлення впливу нового лікарського засобу на тваринах.

КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ

Схематично процес моделювання можна подати, як показано на рисунку 6.1.



Рисунок 6.1 – Схема процесу моделювання

Основним завданням процесу моделювання є створення моделі, яка є найбільше наближеною до оригіналу та властивості якої відповідають меті дослідження.

Моделі можна класифікувати за різними ознаками: призначення, спосіб подання, стан. На рисунку 6.2 подано розподіл моделей за способом подання.

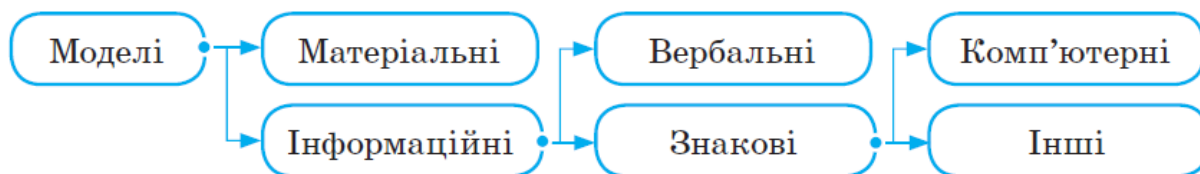


Рисунок 6.2 – Класифікація моделей за способом подання

Інформаційна модель — це модель, яка містить інформацію про суттєві властивості об'єкта, зв'язок між ними та призначена для дослідження стану об'єкта в разі зміни його властивостей.

Комп'ютерне моделювання — процес створення інформаційних моделей комп'ютерними засобами.

КОМП'ЮТЕРНИЙ ЕКСПЕРИМЕНТ, СТАТИСТИЧНИЙ АНАЛІЗ ДАНИХ

Комп'ютерний експеримент — інструмент дослідження комп'ютерних моделей з метою визначення, як змінюються їх параметри. У середо вищі програмного засобу за значеннями одних параметрів моделі обчислюються інші її параметри, і на цій основі робляться висновки про властивості об'єкта.

Статистика (від латин. status) — стан справ/речей. Дані статистики — кількісні характеристики явищ, об'єктів, процесів, мета статистики — виявлення особливих закономірностей.

Математична статистика — наука про математичні методи систематизації, опрацювання та використання статистичних даних для отримання наукових і практичних висновків.

Комп'ютерний експеримент не відображає, наприклад, природні явища, а є чисельною реалізацією створеної людиною математичної моделі цього явища. У будь-якому експерименті досліджують властивості певної кількості однотипних об'єктів, щоб дійти висновку про досліджувані властивості всіх таких об'єктів.

Вибірка, або вибіркова сукупність, — частина генеральної (загальної) сукупності об'єктів, яка охоплюється експериментом.

У статистичному аналізі опрацьовуються такі характеристики вибірки.

Обсяг вибірки — кількість елементів у вибірці.

Розмах — різниця між максимальним і мінімальним значеннями елементів вибірки.

Середнє арифметичне — це частка від ділення суми чисел на їх кількість (обсяг вибірки).

Мода — значення, яке трапляється найчастіше в ряді даних. Якщо дані у вибірці не повторюються, мода не обчислюється. Якщо в ряді даних є числа, які зустрічаються однаково кількість разів, мода буде мати кілька значень.

Медіана впорядкованого ряду даних із непарним числом членів — це серединне значення ряду. Медіана впорядкованого ряду із парним числом членів — середнє арифметичне двох значень, розташованих посередині. Отже, половина значень вибірки менші за медіану, а половина — більші.

Частота — число повторень значень вибірки в заданих інтервалах. Якщо інтервалом є вибірка, то частотою є повторення кожного значення у ній.

Відносна частота — це відношення частоти до загальної кількості даних у вибірці.

Стандартне відхилення (ще його називають середньоквадратичним відхиленням, СКВ) — показник розсіювання статистичних даних відносно середнього значення вибірки.

ОБЧИСЛЕННЯ ОСНОВНИХ СТАТИСТИЧНИХ ХАРАКТЕРИСТИК ВИБІРКИ ЗАСОБАМИ ТАБЛИЧНОГО ПРОЦЕСОРА

Зазвичай для обчислень статистичних характеристик користуються програмними засобами, наприклад MS Excel.

Наприклад, в Україні за результатами ЗНО з певного предмета (знайдіть дані про кількість випускників і випускниць) рахують кількість значень, які потрапляють у задані інтервали (1–3; 4–6; 7–9 і 10–12). Для цього зручно використовувати функцію FREQUENCY (ЧАСТОТА) табличного процесора.

Функції Excel згруповані за призначенням, кожна група на стрічці є розкривним списком, з якого й вибирають необхідну функцію. Відкривається вікно введення параметрів функції з підказками щодо значень кожного з параметрів. У поле введення параметрів прописують вираз або посилання на клітинку.

Зокрема MS Excel, у сьогоденні набирають популярності Google-додатки. На рисунку 6.3 показано функціонал для обчислення основних статистичних характеристик засобами онлайн-додатку Google Таблиці.

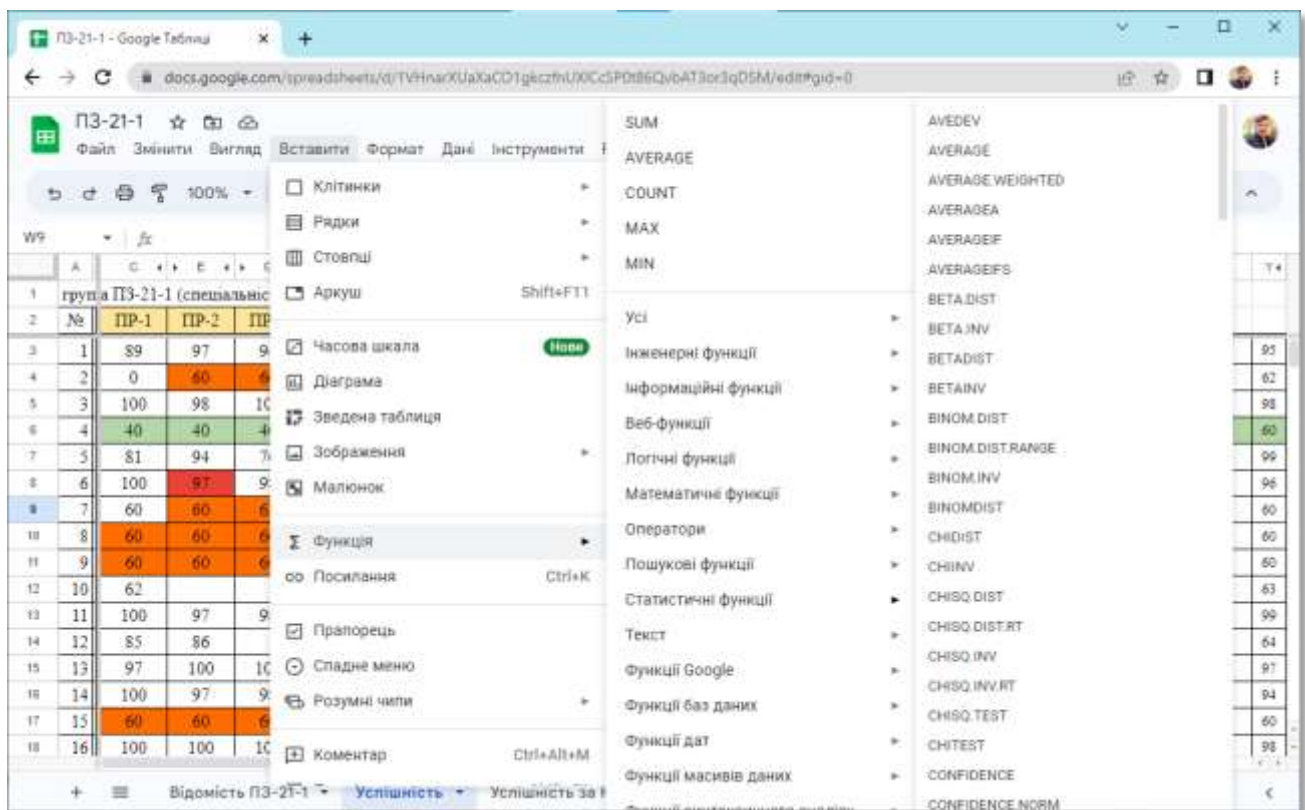


Рисунок 6.3 – Функціонал онлайн-додатку Google Таблиці

Постановка задачі

Поставимо перед собою задачу проаналізувати температурний режим повітря за 15 днів липня. Для обчислення статистичних характеристик можемо користуватися функціями популярних табличних процесорів — MS Excel або Google Таблиці.

Заповнення таблиці даних температурного режиму повітря бажано здійснити відповідно до рисунка 6.4.

	A	B	C	D	E
1	Липень				
2	День	Темп.	Мода	Частота	Відн. частота
3	1	24	=MODE.MULT(B3:B17)	=FREQUENCY(B3:B17;B3:B17)	=D3/\$D\$20
4	2	24	=MODE.MULT(B3:B17)	=FREQUENCY(B3:B17;B3:B17)	=D4/\$D\$20
5	3	23	=MODE.MULT(B3:B17)	=FREQUENCY(B3:B17;B3:B17)	=D5/\$D\$20
6	4	27	=MODE.MULT(B3:B17)	=FREQUENCY(B3:B17;B3:B17)	=D6/\$D\$20
7	5	30	=MODE.MULT(B3:B17)	=FREQUENCY(B3:B17;B3:B17)	=D7/\$D\$20
8	6	30		=FREQUENCY(B3:B17;B3:B17)	0
9	7	26		=FREQUENCY(B3:B17;B3:B17)	=D9/\$D\$20
10	8	30		=FREQUENCY(B3:B17;B3:B17)	0
11	9	26		=FREQUENCY(B3:B17;B3:B17)	0
12	10	18		=FREQUENCY(B3:B17;B3:B17)	=D12/\$D\$20
13	11	25		=FREQUENCY(B3:B17;B3:B17)	=D13/\$D\$20
14	12	27		=FREQUENCY(B3:B17;B3:B17)	0
15	13	25		=FREQUENCY(B3:B17;B3:B17)	0
16	14	21		=FREQUENCY(B3:B17;B3:B17)	=D16/\$D\$20
17	15	26		=FREQUENCY(B3:B17;B3:B17)	0
18					
19					
20			Обсяг	15	
21			СА	=AVERAGE(B3:B17)	
22			Мін	=MIN(B3:B17)	
23			Макс	=MAX(B3:B17)	
24			Розмах	=D23-D22	
25			Медіана	=MEDIAN(B3:B17)	

Рисунок 6.4 – Таблиця даних аналізу температурного режиму повітря

Результати розрахунку статистичних характеристики аналізу температурного режиму повітря наведено на рисунку 6.5.

	A	B	C	D	E
1	Липень				
2	День	Темп.	Мода	Частота	Відн. частота
3	1	24	30	2	0,13
4	2	24	26	0	0
5	3	23	#Н/Д	1	0,07
6	4	27	#Н/Д	2	0,13
7	5	30	#Н/Д	3	0,20
8	6	30		0	0,00
9	7	26		3	0,20
10	8	30		0	0
11	9	26		0	0
12	10	18		1	0,07
13	11	25		2	0,13
14	12	27		0	0
15	13	25		0	0
16	14	21		1	0,07
17	15	26		0	0
18				0	
19					
20			Обсяг	15	
21			СА	25,5	
22			Мін	18	
23			Макс	30	
24			Розмах	12	
25			Медіана	26	

Рисунок 6.5 – Результати розрахунку статистичних характеристик

Аргументи функцій — набір даних, для якого й обчислюється мода.

Функція **MODE.SNGL** (МОДА.ОДН(аргументи)) повертає одне значення. Її використовують для ряду значень, у яких групи з однаковими значеннями не повторюються.

Функція **MODE.MULT** (МОДА.НСК(аргументи)) гнучкіша.

Температура 26 і 30 повторюється частіше за інші — по 3 рази. Перша функція поверне результат 30, оскільки саме така температура трапиться вперше в запропонованому ряді значень. Якщо вказаний ряд значень упорядкувати, то отримаємо результат 26.

Функцію **MODE.MULT** (МОДА.НСК(аргументи)) використовують, поперше, для вертикального ряду значень, а по-друге, як функцію масиву. Тобто опрацьовується ряд значень — масив значень. І функція повертає не одне значення, а кілька — теж масив значень.

Отже, для вставлення формули з функцією одразу виділяють кілька клітинок (ми не знаємо, скільки, тому краще зарезервувати більше), після вказування аргументу функції — діапазону клітинок із значеннями — натискають сполучення клавіш **Ctrl + Shift + Enter**.

Якщо масив не містить однакових даних, функція **MODE.MULT** (МОДА.НСК(аргументи)) повертає значення помилки **#Н/Д**.

Якщо дані введено в рядок (**B3:P3**), тобто маємо горизонтальний ряд даних, то для отримання вертикального застосовують функцію транспонування. Загальний вигляд обчислення моди буде такий:

{=MODE.MULT(TRANSPOSE(B3:P3))} або **{=МОДА.НСК(ТРАН СП(B3:P3))}**

Фігурні дужки для формули вказують на опрацювання масиву даних — дужки проставляються автоматично після **Ctrl + Shift + Enter**.

Функція моди опрацьовує ряд значень вказаного діапазону, перетворений на вертикальний. Для функції виділено 5 клітинок, знайдено 2 значення, які найчастіше повторюються, тому в клітинках будуть ці значення — 30 і 26, а в решті — **#Н/Д**.

Для обчислення медіани викликається однойменна функція і як аргументи вставляється ряд даних (у прикладі це **B2:B16**).

Для підрахунку частот користуються однойменною функцією **FREQUENCY** (ЧАСТОТА), аргументами якої є ряд даних та інтервали, у яких і рахують повторення значень.

Постановка задачі

Розглянемо дані, які характеризують вік осіб. Необхідно порахувати, скільки серед них осіб молодше за 14 років, від 14 до 17, від 17 до 22, старше за 22. Початковими даними буде діапазон із віком, а в клітинках інтервалів будуть значення 14, 17, 22.

Отже, потрібно мати 4 клітинки для результату:

- 1 — для числа осіб віком молодше за 14 або 14 років;
- 2 — для числа осіб віком від 14 до 17 включно;
- 3 — для числа осіб віком від 17 до 22 включно;
- 4 — для числа осіб віком старших за 22.

Пригадайте, що для таких обчислень користувалися функцією COUNTIF (СЧЕТЕСЛИ) — функція опрацьовує як чисельні дані, так і текстові (для текстових даних користуються виключно COUNTIF (СЧЕТЕСЛИ)), а функція FREQUENCY (ЧАСТОТА) використовується для опрацювання цілих і дійсних чисел.

Під час порівняння результатів обчислень двох функцій функція FREQUENCY (ЧАСТОТА) повертає кількість повторень для значення, яке першим трапляється в ряді, для інших таких само значень повертається 0, а функція COUNTIF (СЧЕТЕСЛИ) повторює обчислюваний результат для всіх даних. У цьому випадку це не впливає на статистичні характеристики. Але в процесі обчислення відносної частоти у відсотках загальну кількість отримаємо 100 %.

Якщо рахувати кількість повторень через COUNTIF (СЧЕТЕСЛИ) і результат взяти за основу підрахунку відносної частоти, то під час перевірки отримаємо число більше за 100 %.

В Excel містяться кілька функцій для обчислення стандартного відхилення. Функції STDEVA (СТАН ДОТКЛОНА) і STDEVPA (СТАН ДОТКЛОНПА) відрізняються значенням середнього: у першій функції використовується середнє вибірки, у другій — середнє генеральної сукупності.

У наших прикладах не вказуються значення генеральної сукупності, вони однакові і для вибірки, і для генеральної сукупності. І обидві зазначені функції повернуть однаковий результат.

Функції як аргументи можуть набувати як чисельних значень, так і логічних. Інші функції Excel для обчислення СКВ працюють із чисельними значеннями.

Алгоритм обчислення статистичного відхилення наведено на рисунку 6.6.

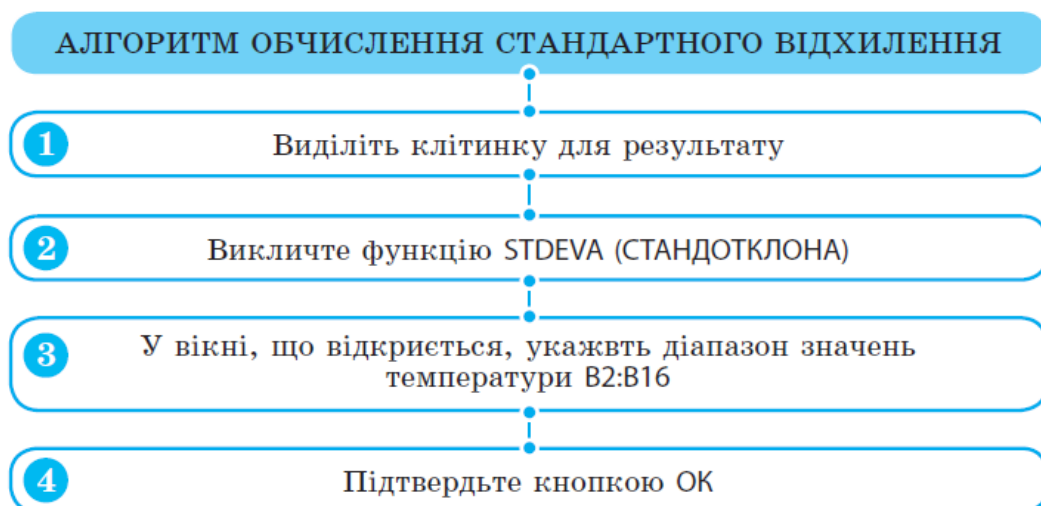


Рисунок 6.6 – Алгоритм обчислення статистичного відхилення

ВІЗУАЛІЗАЦІЯ РЯДІВ І ТРЕНДІВ ДАНИХ

Дослідження, експерименти, статистичний аналіз супроводжуються візуалізацією даних. Графіки й діаграми не тільки графічно відображають значення, а й мають змістове навантаження щодо теми дослідження. Крім того, акцентується увага на деяких елементах ряду даних, порівнюються фрагменти даних, демонструється тенденція зміни властивостей, що досліджуються, тощо.

Інфографіка (від латин. informatio — інформування, роз'яснення) — це візуальне відображення інформації, статистичних даних для простої і наочної демонстрації тенденцій, співвідношень, а також зацікавлення в предметі дослідження.

Гістограми використовують для наочного подання динаміки зміни даних у часі або розподілу даних у вигляді прямокутних областей, розмір яких відповідає значенням, отриманим у результаті статистичного дослідження.

Гістограми можуть бути як вертикальні — стовпчикові діаграми, так і лінійчасті. На рисунку 6.7 побудовано гістограму зросту за відносною частотою.

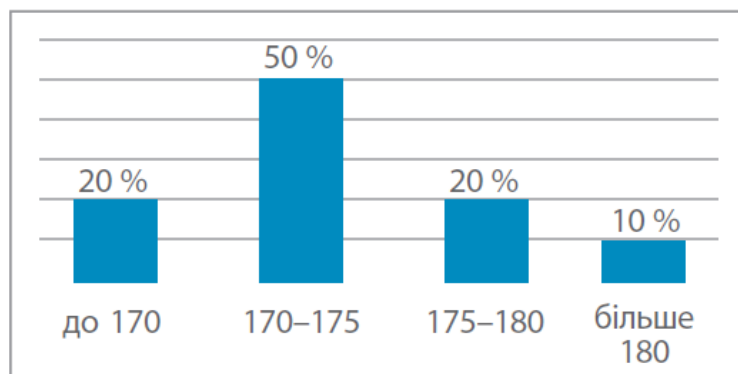


Рисунок 6.7 – Гістограма зросту за відносною частотою

За діаграмою легко дістатися висновку про відсоткове співвідношення кількості учнів за зростом у певних інтервалах.

Лінія тренду в MS Excel — це графічне подання загальної закономірності зміни ряду даних. Лінія тренду є графіком, який описується математичними формулами та до значень якого наближаються значення діаграми.

Лінія тренду є апроксимацією (апроксимація — наближення) значень діаграми до графіка, який описується математичними формулами. Її додають до гістограми, діаграми з областями, лінійчастої діаграми, графіків або точкових діаграм.

На гістограмі, що наведено на рисунку 6.8 лінія тренду (зображено червоним пунктиром) засвідчує підвищення рівня знань здобувача освіти.

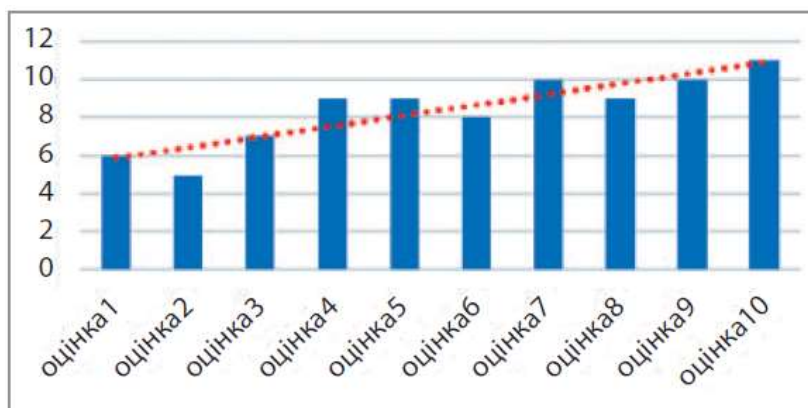


Рисунок 6.8 – Аналіз за рахунок відображення лінії тренду

У прикладі скористалися лінійною лінією тренду, бо саме цей тип найкраще описує простий лінійний набір даних і застосовується в тих випадках, коли точки даних розташовані близько до прямої.

Алгоритм додавання лінії тренду наведено на рисунку 6.9.

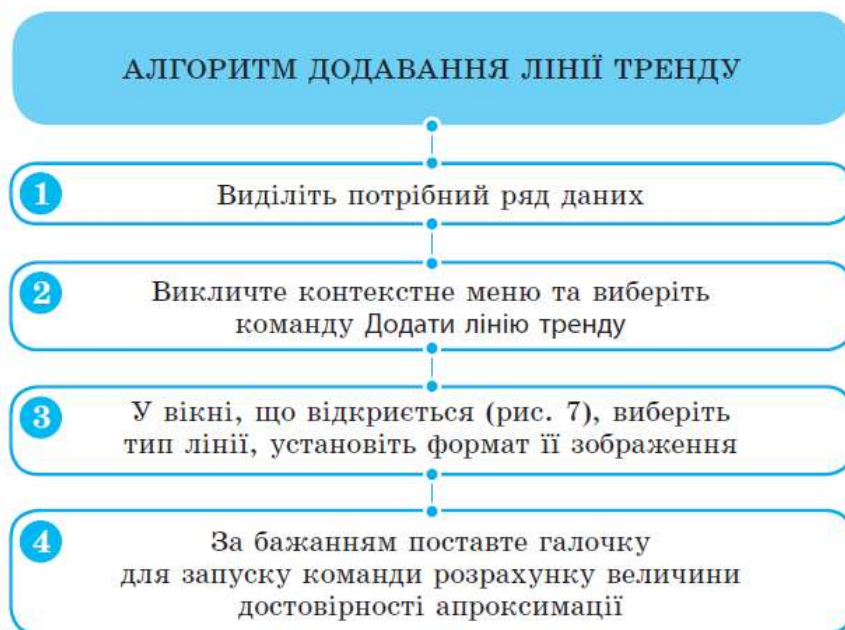


Рисунок 6.9 – Алгоритм додавання лінії тренду

РОЗВ’ЯЗУВАННЯ РІВНЯНЬ, СИСТЕМ РІВНЯНЬ, ОПТИМІЗАЦІЙНИХ ЗАДАЧ

Комп’ютерне моделювання ефективно використовується для розв’язування задач будь-якої галузі. Оскільки математичні моделі процесів, явищ реалізують у рівняннях або системах рівнянь, то в MS Excel додано засоби їх розв’язання.

Метод «Підбір параметра»

Приклад 6.1

Клієнт придбав 138 м чорного та червоного сукна за 540 грн. Скільки метрів кожного сукна придбав клієнт, якщо червоне коштувало 5 грн. за метр, а червоне — 3 грн.?

Математична модель розв’язування задачі. Нехай x — кількість метрів чорного сукна, яке придбав клієнт, тоді червоного сукна він придбав $138 - x$. Складемо рівняння: $3x + 5(138 - x) = 540$.

Складемо таблицю: у її клітинках подано відповідно формули і результати їх обчислення.

У клітинці C2 кількість чорного сукна, яку ми позначили як x , дорівнює 1. MS Excel не здійснює арифметичні операції з нечисловими даними, тому на початку замість x вставляють будь-яке значення (0 вставляти не варто, бо в рівняннях може виникнути ділення на 0). Із таким довільним значенням, звичайно, ми не отримаємо правильну вартість усього сукна з умови задачі (540), але головним є те, що в клітинці

D4 розміщено підсумкову формулу, результатом обчислення якої має бути значення з умови задачі.

Для цього потрібно скласти для задачі математичну модель у вигляді рівняння з одним невідомим. Вираз із невідомим — у лівій частині рівняння, а в правій — значення цього виразу. Метод добирає значення невідомого, поки не буде отримано необхідне значення результату обчислення виразу.

На рисунку 6.10 наведено результат виконаних дій.

	A	B	C	D
1		вартість	скільки	За сукно
2	Чорне сукно	3	1	=B2*C2
3	Червоне сукно	5		=B3*C3
4				=СУММ(D2:D3)

	A	B	C	D
1		вартість	скільки	За сукно
2	Чорне сукно	3	1	3
3	Червоне сукно	5	137	685
4				688

Рисунок 6.10 – Результати обчислення

Приклад 6.2

Кредит у 1000 у. о. видано на 5 років під 20 % річних за таких умов повернення: наприкінці другого року повернути 200 у. о., наприкінці кожного наступного — по 400 у. о. Визначити, яку суму необхідно внести наприкінці першого року, щоб кредит було погашено вчасно (до кінця п'ятого року).

Кожного року борг із кредитування обчислюється за формулою:

$$\text{Борг} = \text{Залишок_внеску} + \text{Залишок_внеску} * \text{Річні}/100.$$

Погашення кредиту означає, що борг відсутній: $\text{Борг} = 0$.

У моделі необхідно скласти рівняння для обчислення боргу наприкінці кожного року за наведеною формулою. Результат обчислення боргу за рік залежить від значення залишку боргу за попередній рік.

Отже, якщо x — повернення грошей у перший рік, то:

$$\text{борг1 (наприкінці першого року)} = 1000 - x + 0,2(1000 - x);$$

$$\text{борг2} = \text{борг1} - 200 + 0,2(\text{борг1} - 200);$$

$$\text{борг3} = \text{борг2} - 400 + 0,2(\text{борг2} - 400);$$

$$\text{борг4} = \text{борг3} - 400 + 0,2(\text{борг3} - 400);$$

$$\text{борг5} = \text{борг4} - 400 + 0,2(\text{борг4} - 400).$$

Оскільки кредит необхідно сплатити за 5 років, то $\text{борг5} = 0$, тобто:

$$\text{борг4} - 400 + 0,2(\text{борг4} - 400) = 0.$$

Таблицю зі значеннями після виклику методу «Підбір параметра». Зазначений зручно використовувати для складання математичної моделі з одним невідомим параметром.

Спочатку у клітинку B4 введемо 1 — сплачено в перший рік (х у моделі), у наступні клітинки стовпця B: 200 і по 400, як в умові.

Для зручності залишок внеску, як різницю боргу та сплаченого, обчислимо в стовпці C, і остаточний борг наприкінці року — у стовпці D. Вікно діалогу Підбору параметра зі введеними посиланнями та значеннями, які зображено на рисунку 6.11.

D8

Рисунок 6.11 – Результати обчислення

Метод «Пошук рішення»

Якщо методом «Підбір параметра» зручно користуватися для математичної моделі з одним невідомим, то для певного типу складніших завдань краще підійде надбудова «Розв’язувач». У математичній моделі розв’язування таких задач необхідно знайти значення параметрів, від яких залежить значення результату обчислення певної функції (у MS Excel ця функція подається формулою). Цей результат може бути максимальним (наприклад, прибуток організації), мінімальним (витрати на сировину) або дорівнювати певному значенню (потужність електричної мережі). У таких випадках говорять про оптимізацію результату. Цей тип задач належить до оптимізаційних задач, функція називається цільовою, а параметри — оптимізаційними.

Щоб розв’язок і значення параметрів були реальними, модель обов’язково містить обмеження або на значення параметрів, або на допоміжні розрахунки, які залежать від оптимізаційних параметрів. Обмеження завжди даються в умові.

Для розв’язування таких задач у MS Excel передбачено надбудову «Розв’язувач», який дає змогу: зазначати кілька клітинок для оптимізаційних параметрів; указувати обмеження на їх значення; знаходити розв’язок (цільова функція описується в клітинці формулою, у якій обов’язковим є посилання на клітинки з оптимізаційними параметрами); визначити відповідно до умови один із варіантів роботи з цільовою функцією (знайти її максимальне значення, мінімальне значення, конкретне значення); отримати кілька рішень.

Щоб модель реалізувати в MS Excel надбудовою Розв’язувач, необхідно дотримуватися певних правил:

- у таблиці обов’язково має бути діапазон клітинок для значень оптимізаційних параметрів;
- у таблиці має бути клітинка з формулою обчислення цільової функції;
- таблиця обов’язково містить клітинки із значеннями обмежень параметрів, щоб мати змогу на них посилається.

Приклад 6.3

Підприємство виготовляє вироби трьох видів — A , B і C . Денний дохід із кожного виду одиниці виробу — 13, 18 і 22 у. о. Необхідно визначити, за якої кількості виробів підприємство отримає максимальний дохід з урахуванням того, що:

1) загальний обсяг виробництва — усього 300 одиниць виробів на день;

2) підприємству необхідно здійснити виробництво 50 одиниць виробу A для виконання замовлення; 40 одиниць виробу B для виконання планового замовлення; а оскільки збут виробів C доволі невеликий, то слід виготовляти не більше 40 одиниць цього виробу.

Складемо математичну модель. Позначимо кількість виробів кожного типу через a , b і c . Тоді можна скласти рівняння:

$$\text{дохід} = 13a + 18b + 22c.$$

Цільова функція — дохід — залежить від значень параметрів a , b , c і має бути спрямована на отримання максимального значення функції.

Оптимізаційними параметрами є a , b , c — їх значення впливають на значення цільової функції.

Під час пошуку максимального значення доходу необхідно врахувати обмеження на значення параметрів a , b , c :

$$a + b + c = 300;$$

$$a \geq 50; b \geq 40; c \leq 40;$$

a , b , c — цілі додатні значення.

Розв'яжемо задачу в MS Excel за допомогою надбудови «Розв'язувач».

1. Скористаємося математичною моделлю для створення таблиці в середовищі Excel. У клітинки C2:C4 вводимо початкові довільні значення для виробів a , b , c (можна було ввести і по 1). Для використання надбудови «Розв'язувач» її необхідно встановити на стрічку «Дані». Пригадайте алгоритм установлення пакета «Аналіз даних».

2. Для встановлення пакета «Розв'язувач» переходимо в меню «ФАЙЛ» — «Параметри». Вибираємо «Надбудови».

3. У вікні, що відкриється, вибираємо пакет аналізу й активуємо кнопку «Перейти» — відкриється вікно «Надбудови».

4. Проставляємо галочку біля «Розв'язувач». Підтверджуємо — «ОК». На стрічці «Дані» з'явилася команда «Розв'язувач».

Результат правильно виконаних дій наведено на рисунку 6.12.

	A	B	C	D
1	Вироби	дохід з одиниці	кількість	дохід
2	а	13	100	1300
3	б	18	100	1800
4	с	22	100	2200
5	всього		300	5300



Рисунок 6.12 – Результати заповнення таблиці

5. Запускаємо команду «Розв’язувач» — відкриться вікно «Параметри» «Розв’язувача», у якому:

а) у поле Встановіть цільову клітинку вводимо клітинку D5 з підрахунком загального доходу і вказуємо, що функція прямує до максимуму;

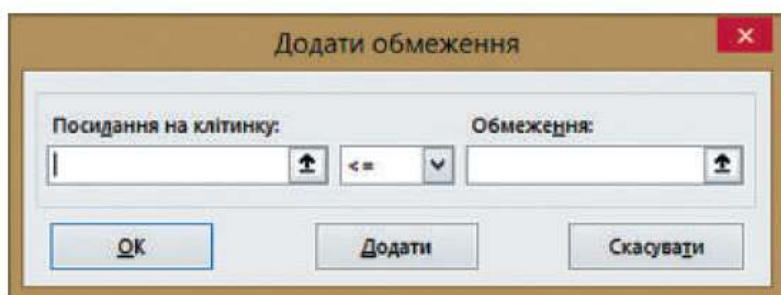
б) у поле Змінні клітинки вводимо клітинки C2:C4;

в) для введення в поле Обмеження скористаємося командою «Додати», обмеження вводять у вікні «Додати обмеження»;

г) підтверджуємо — кнопка «Знайти» розв’язок.

6. У вікні, що відкриться, залишаємо активною команду «Зберегти» знайдений розв’язок, клацаємо «ОК».

Дані таблиці будуть змінені відповідно до рисунка 6.13.



	A	B	C	D
1	Вироби	дохід з одиниці	кількість	дохід
2	а	13	100	1300
3	б	18	100	1800
4	с	22	100	2200
5	всього		300	5300

Рисунок 6.13 – Результати обчислень за методом «Пошук рішення»

Математичною моделлю розв’язування задач може бути система лінійних рівнянь із кількома невідомими. Із курсу математики ви знаєте кілька методів знаходження невідомих системи рівнянь. У MS Excel пропонуються такі методи, як метод оберненої матриці (аналогічний методу Крамера), метод Гауса й метод Зейделя.

РОЗВ'ЯЗУВАННЯ ЗАДАЧ ІЗ РІЗНИХ ПРЕДМЕТНИХ ГАЛУЗЕЙ

Для різних галузей науки й виробництва вибір програмних засобів для опрацювання даних у математичних моделях можуть бути використано велику кількість параметрів або рівняння будуть нелінійними.

На рисунку 6.14 наведено приклад розв'язування системи лінійних рівнянь: $3x + 2y = 5$; $2x - y = 8$.

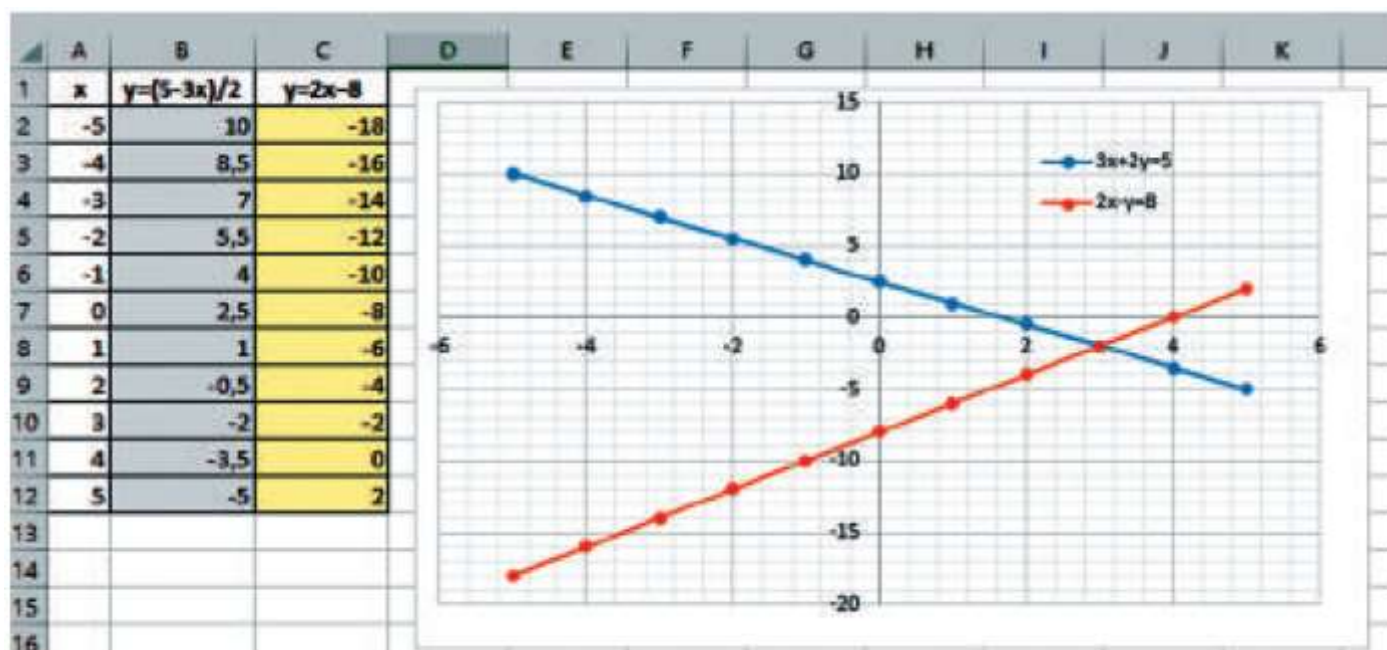


Рисунок 6.14 – Результат розв'язування системи лінійних рівнянь

Алгоритм отримання розв'язку показаний на рисунку 6.15.



Рисунок 6.15 – Алгоритм отримання розв'язку

Перетворимо рівняння у вигляд, потрібний для побудови графіків:
 $y = (5 - 3x) : 2$; $y = 2x - 8$.

Із діаграми видно, що графіки перетинаються в точці $(3; -2)$. Отже, розв'язком системи рівнянь є $x = 3$; $y = -2$.

Якщо виділити діаграму, з'являється можливість перейти до додаткової стрічки «ЗНАРЯДДЯ ДЛЯ ДІАГРАМ» — «Конструктор». Першою групою команд є «Додати елемент діаграми». Якщо на діаграмі не вистачає ліній сітки, слід скористатися саме цією групою. Також можна налагодити вигляд діаграм за запропонованими зразками.

Якщо внести зміни у друге рівняння в системі на $2x^2 - y = 8$ і знайти графічне розв'язання, то можна прийти до висновків, що отримано два розв'язки.

Результат правильно виконаних корегувань показаний на рисунку 6.16.

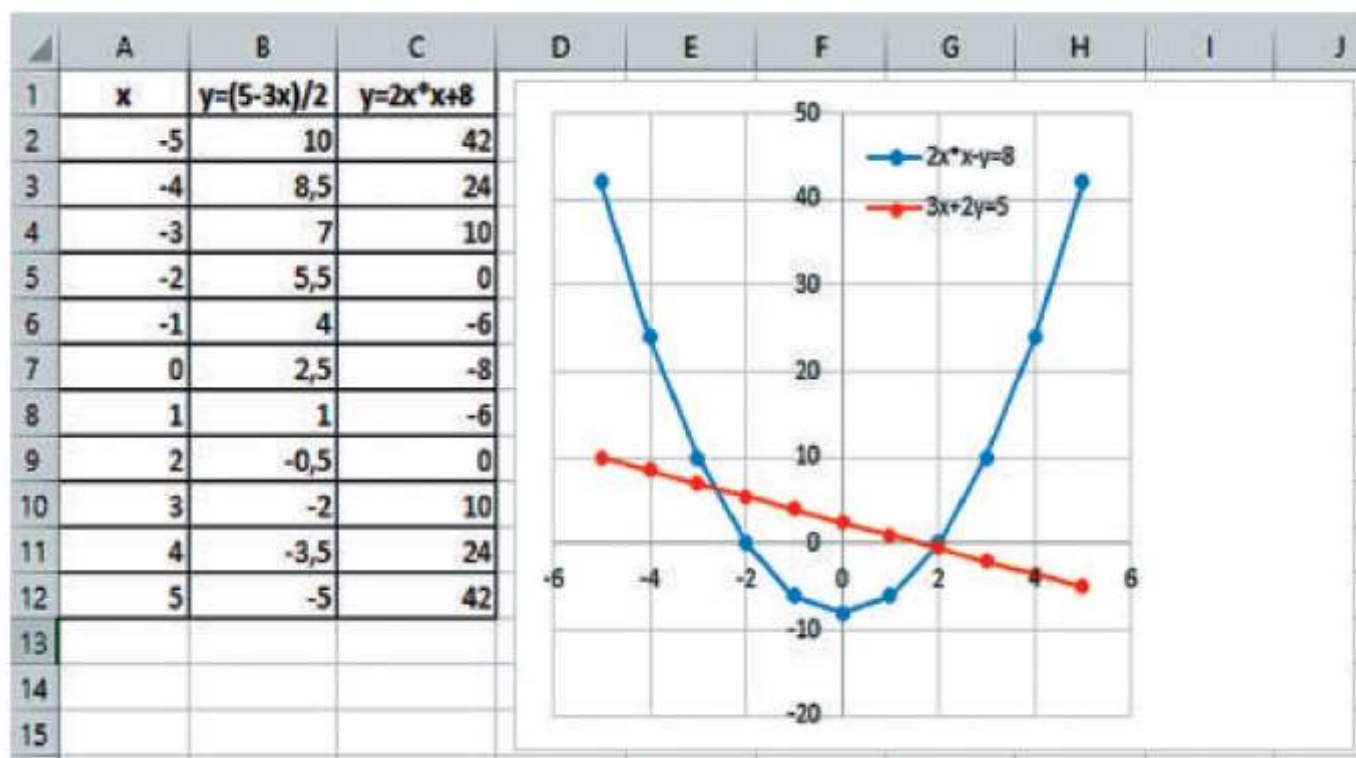


Рисунок 6.16 – Результат розв'язування скорегованої системи рівнянь

Висновки

Отже, Табличні процесори, такі як Microsoft Excel, Google Таблиці або LibreOffice Calc, мають безліч переваг, серед яких;

- ефективність: табличні процесори дозволяють швидко та ефективно обробляти великі обсяги даних, проводити аналіз та зробити розрахунки. таким чином, вони дозволяють збільшити продуктивність та ефективність роботи;

- спрощення обробки даних: табличні процесори дозволяють легко вводити, редагувати та форматовувати дані, зручно зберігати та організовувати інформацію, що значно полегшує роботу з даними;

- аналіз даних: табличні процесори мають багатий набір інструментів для аналізу даних, таких як вбудовані функції, графіки, діаграми та звіти. це дозволяє виконувати складні аналітичні завдання та робити швидкий аналіз даних;

- сумісність та широка підтримка: більшість табличних процесорів мають широку підтримку та популярність, тому їх легко використовувати та інтегрувати з іншими програмами та сервісами;
- можливості спільної роботи: табличні процесори дозволяють працювати з даними в режимі реального часу, що дозволяє кільком користувачам працювати з одними та ж даними одночасно, що забезпечує зручність та швидкість обробки даних;
- автоматизація: табличні процесори дозволяють автоматизувати багато рутинних операцій, таких як обчислення, форматування, сортування та фільтрування даних. це дозволяє значно збільшити продуктивність та ефективні.

Питання та завдання до контролю знань студентів

- 1 Поясніть сутність терміну «модель». Наведіть приклади моделей. Що називають моделюванням? Як ви розумієте поняття «комп'ютерний експеримент»?
- 2 Що вивчає статистика? Що таке вибірка даних? Дайте визначення статистичним характеристикам. Чому для обчислення моди вибірки резервують більше однієї клітинки?
- 3 Які групи функцій містять функції обчислення статистичних характеристик? Яким сполученням клавіш користуються для підтвердження опрацювання даних масиву?
- 4 Які функції містить MS Excel для обчислення моди? Чому для обчислення моди резервують більше однієї клітинки? Як ви розумієте «інтервал значень» як аргумент функції FREQUENCY (ЧаСТОТа)? Яке призначення статистичної характеристики відносної частоти? У якому форматі краще аналізувати?
- 5 Для чого використовується інфографіка? На прикладах поясніть, у яких випадках зручно для ряду даних створити гістограму, а в яких — кругову діаграму.
- 6 Як ви розумієте поняття лінії тренду? На що вказує лінія тренду? Як додати на діаграму лінію тренду? За яким параметром визначають відповідність лінії тренду значенням діаграми?
- 7 Для чого використовується метод «Підбір параметра» в MS Excel? Для чого використовується надбудова «Розв'язувач»? Якою має бути математична модель для використання методу «Підбір параметра»? Які правила побудови таблиці на основі математичної моделі для використання надбудови «Розв'язувач»?
- 8 Яким методом можна розв'язати систему лінійних рівнянь у середовищі MS Excel?
- 9 Які типи статистичних пакетів ви знаєте? Назвіть аргументи функцій для фінансових розрахунків. Яка різниця між простими та складними відсотками?
- 10 Наведіть приклад фінансової задачі, у якій використовують складний відсоток.

Лекція № 7

з навчальної дисципліни Інформатика

Тема Програмні засоби для розробки презентацій

Мета заняття Дослідити програмні засоби, що призначені для обробки інформації у форматі презентацій, розвинути навички по підготовці презентацій

Матеріально-технічне забезпечення та дидактичні засоби, ТЗН

1 Інтерактивний вид діяльності, створений засобами Moodle

2 Мультимедійний презентаційний матеріал

Час – 2 години

План проведення лекції

Структура лекції	Відведений час	Методичні вказівки
1 Організаційна частина	5 хв.	Включає в себе: <i>привітання</i> , яке має на меті привернути увагу, забезпечити контакт лектора з аудиторією; <i>визначення присутності студентів на занятті</i>
2 Актуалізація опорних знань, перевірка вивченого матеріалу та мотивація навчальної діяльності студентів	10 хв.	Включає в себе: <i>вступне зауваження</i> , <i>мотивацію актуальності теми</i> , <i>перевірку попереднього матеріалу</i> , <i>формулювання мети лекції</i> , <i>огляд головних питань теми</i> (вказати питання та завдання для перевірки знань студентів)
3 Основна частина (викладення навчальних питань лекції)	60 хв.	Головне місце приділяється <i>викладенню основного змісту матеріалу</i> теми, його <i>аналізу</i> , <i>узагальненню висунутих положень</i> . Для успіху лекції важливе значення має поділ матеріалу на розділи, основні питання
4 Заклучна частина Домашнє завдання: (відповідно до робочої програми)	5 хв.	Включає в себе: <i>узагальнення</i> , <i>теоретичні і фактичні висновки</i> , <i>закріплення вивченого на лекції матеріалу</i>

Література

1 Інформатика (рівень стандарту) : підруч. для 10-го (11-го) кл. закл. заг. серед. освіти / Й.Я. Ривкід Т.І. Лисенко, Л.А. Чернікова, В.В. Шакотько. — Київ : Генеза, 2018. — 144 с. : іл.

2 Інформатика (рівень стандарту) : підруч. для 10-го (11-го) кл. закл. заг. серед. освіти / О.О. Бондаренко, В.В. Ластовецький, О.П. Пилипчук, Є.А. Шестопапов. — Харків : Вид-во «Ранок», 2018. — 175 с. : іл.

Навчальні матеріали лекції

ВСТУП

Електронна презентація є інструментом для показування інформації широкій аудиторії за допомогою візуальних елементів, таких як текст, зображення, графіки, анімація та інші мультимедійні елементи. Основне призначення електронної презентації полягає в тому, щоб допомогти доповідачеві передати свої ідеї та повідомлення в більш доступній і зрозумілій формі, що дозволяє залучити увагу аудиторії та збільшити ефективність комунікації.

Електронні презентації можуть використовуватися в різних контекстах, таких як:

- ділові презентації — для презентації бізнес-ідей, результатів досліджень, звітування перед керівництвом, презентації продукту або послуги тощо;
- навчальні презентації — для навчання студентів, учнів або працівників у різних тематиках;
- культурні події — для презентації культурних продуктів, таких як мистецтво, література, музика, або для показування результатів досліджень у галузі культури та мистецтва;
- медіа — для створення відеопрезентацій для трансляції в мережі Інтернет або для демонстрації фільмів, документальних стрічок, відеоігор тощо.

В загальному, електронні презентації можуть використовуватися в будь-якому контексті, де потрібно показати інформацію за допомогою візуальних елементів.

ОГЛЯД ІСНУЮЧИХ ЗАСОБІВ РОЗРОБКИ ПРЕЗЕНТАЦІЙ

Існує багато програмних засобів для розробки презентацій. Ось кілька найпопулярніших:

Microsoft PowerPoint — це один з найбільш відомих програмних засобів для розробки презентацій. Він має багатий функціонал, включаючи можливість вставляти текст, зображення, відео, анімацію та інші елементи.

Google Slides — це безкоштовний інструмент для розробки презентацій від Google. Він має схожий функціонал з PowerPoint, але працює в хмарі і не потребує установки на комп'ютер.

Prezi — це програмний засіб, що дозволяє створювати динамічні та інтерактивні презентації з великою кількістю елементів. Prezi надає можливість створювати зображення з високою роздільною здатністю та робити розумні презентації.

Canva — це програмний засіб, який дозволяє створювати різні типи дизайну, включаючи презентації. Canva має велику кількість шаблонів та елементів дизайну, які можна використовувати для створення унікальної презентації.

Keynote — це програмний засіб для розробки презентацій від Apple. Він має розширений функціонал, включаючи можливість використовувати 3D-об'єкти, робити презентації у форматі кіно, додавати музику та інші елементи.

ВЗАЄМОДІЯ ІЗ ПРОГРАМИМ ЗАСОБОМ MICROSOFT POWERPOINT

PowerPoint — програмний додаток корпорації Microsoft, призначений програми допомогти користувачам створювати візуально привабливі та ефективні презентації, що дозволяють передати інформацію аудиторії в зрозумілій та легкій формі. Основні функції та можливості PowerPoint включають:

- створення слайдів — користувач може створювати слайди з візуальними елементами, такими як текст, зображення, графіки, таблиці, аудіо та відео;
- додавання анімації — користувач може додавати анімацію до візуальних елементів, що дозволяє їм з'являтися на екрані в певний момент часу та з певним ефектом;
- використання макетів — користувач може використовувати готові макети для створення презентацій, що дозволяє швидко створювати професійно виглядані слайди;
- редагування та форматування — користувач може редагувати та формувати візуальні елементи, такі як текст, зображення, графіки, для досягнення бажаного вигляду;
- додавання презентації — користувач може додавати звукові та відео файли до презентації для більш ефективної комунікації з аудиторією;
- поділ на розділи — користувач може створювати розділи в презентації, що дозволяє організувати інформацію та зробити її більш доступною для аудиторії;
- збереження та друк — користувач може зберігати презентації в різних форматах, таких як PPTX, PDF або друкувати їх для використання в офлайн-режимі.

Інструменти стрічки "Головна"

Стрічка "Головна" у Microsoft PowerPoint містить основні інструменти для створення та редагування електронних презентацій. На рисунку 7.1 наведено інструменти стрічки "Головна".

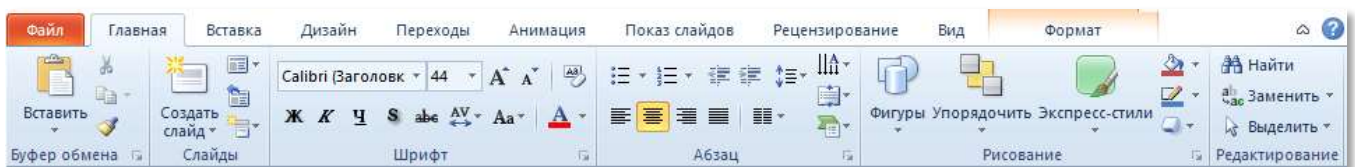


Рисунок 7.1 – Інструменти стрічки "Головна"

Основні інструменти на цій стрічці включають:

- "Новий слайд": інструмент для додавання нового слайда в презентацію;
- "Розміщення": інструмент для вибору макету сторінки для нового слайда, наприклад, зображення з тестом, графіком або таблицею;

- "Формат": інструмент для зміни формату вибраного тексту або об'єкту, такого як зміна шрифту, розміру або кольору;
- "Зв'язок": інструмент для створення гіперпосилань на інші слайди або документи;
- "Форми": інструмент для вставки та редагування форм, таких як прямокутники, кола, стрілки тощо;
- "Вставити": інструмент для вставки зображень, таблиць, графіків та інших мультимедійних елементів;
- "Зображення": інструмент для редагування зображень, такий як зміна розміру, обрізка, зміна кольору тощо;
- "Текст": інструмент для додавання та редагування тексту на слайдах;
- "Анімація": інструмент для додавання анімації до об'єктів на слайді, такого як зміна розміру, рух, з'явлення, зникнення тощо;
- "Запуск слайд-шоу": інструмент для запуску презентації в режимі слайд-шоу.

Ці інструменти дозволяють користувачам створювати візуально привабливі та ефективні електронні презентації.

Інструменти стрічки "Вставка"

Стрічка "Вставка" у Microsoft PowerPoint містить інструменти для вставки різноманітних об'єктів у презентацію. На рисунку 7.2 наведено інструменти стрічки "Вставка".

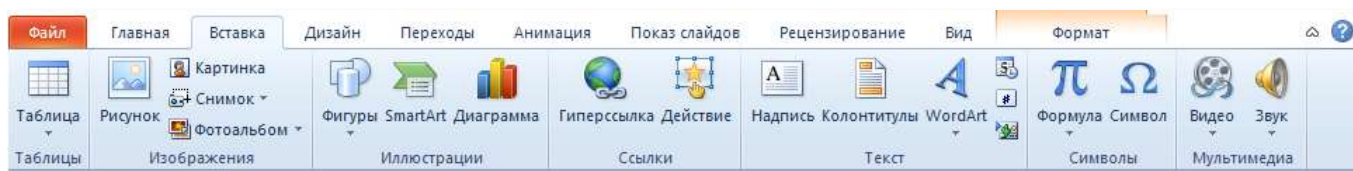


Рисунок 7.2 – Інструменти стрічки "Вставка"

Основні інструменти на цій стрічці включають:

- "Картинка": інструмент для вставки зображень з файлу, з веб-сторінки або з онлайн-галереї;
- "Фільм": інструмент для вставки відео з файлу або з веб-сторінки;
- "Аудіо": інструмент для вставки аудіофайлів, таких як музика або звукові ефекти;
- "Форми": інструмент для вставки та редагування форм, таких як прямокутники, кола, стрілки тощо;

- "SmartArt": інструмент для створення діаграм, процесів, ієрархій та інших складних об'єктів;
- "Графік": інструмент для вставки графіків та діаграм, таких як стовпчасті, кругові та лінійні графіки;
- "Таблиця": інструмент для вставки та редагування таблиць;
- "Ілюстрації": інструмент для вставки малюнків, векторних зображень та іконок;
- "3D-модель": інструмент для вставки та редагування тривимірних моделей;
- "Об'єкт": інструмент для вставки об'єктів з інших програм, таких як Word або Excel.

Ці інструменти дозволяють користувачам вставляти різноманітні об'єкти у презентацію, що дозволяє створювати більш цікаві та варіативні електронні презентації.

Інструменти стрічки "Дизайн"

Стрічка "Дизайн" у Microsoft PowerPoint містить інструменти для налаштування зовнішнього вигляду презентації. На рисунку 7.3 наведено інструменти стрічки "Дизайн".

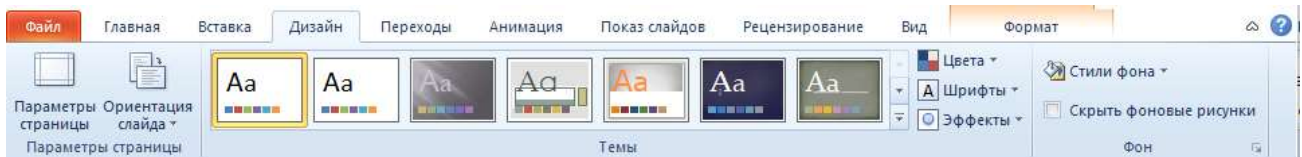


Рисунок 7.3 – Інструменти стрічки "Дизайн"

Основні інструменти на цій стрічці включають:

- "Теми": інструмент для вибору готової теми оформлення для презентації. Теми можуть включати фонові зображення, кольорові палітри та шрифти.
- "Змінити тему": інструмент для зміни теми оформлення презентації. Ви можете вибрати іншу тему або налаштувати кольорову палітру та шрифти самостійно.
- "Фон": інструмент для вибору фонового зображення або кольору для слайда або всієї презентації.
- "Водяний знак": інструмент для додавання водяного знака на слайди презентації.
- "Стилі": інструмент для налаштування стилів об'єктів, таких як заголовки, текст та списки.
- "Розмір": інструмент для налаштування розміру та орієнтації слайдів.

– "Переходи": інструмент для вибору ефекту переходу між слайдами презентації.

– "Анімація": інструмент для налаштування анімації об'єктів на слайді. Ви можете встановити, коли та які об'єкти з'являтимуться на слайді.

Ці інструменти дозволяють користувачам налаштовувати вигляд своєї презентації, щоб зробити її більш привабливою та професійною. Ви можете вибрати готову тему, налаштувати фон та стилі об'єктів, вибрати ефекти переходу між слайдами та налаштувати анімацію, щоб зробити свою презентацію більш варіативною та ефектною.

Інструменти стрічки "Переходи"

Стрічка "Переходи" у Microsoft PowerPoint містить інструменти для налаштування ефектів переходу між слайдами в презентації. На рисунку 7.4 наведено інструменти стрічки "Переходи".

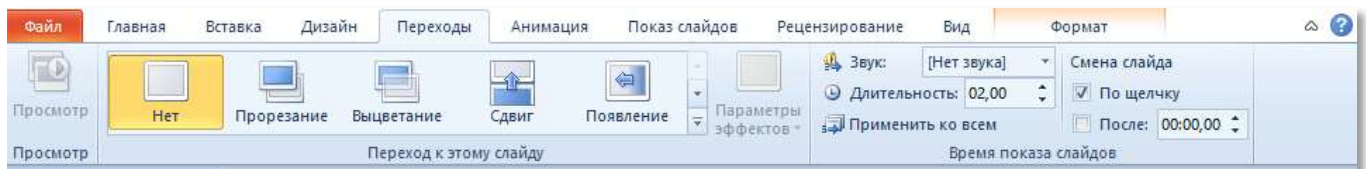


Рисунок 7.4 – Інструменти стрічки "Переходи"

Основні інструменти на цій стрічці включають:

– "Переходи": Інструмент для вибору ефекту переходу між слайдами презентації. Ви можете вибрати з більш ніж 50 різних ефектів переходу, таких як "Розкриття", "Затемнення" та "Вікно".

– "Тривалість": Інструмент для налаштування часу, протягом якого буде показано перехід між слайдами. Ви можете встановити тривалість від 0,1 до 59,9 секунд.

– "Затемнення під час переходу": Інструмент для вибору того, чи буде затемнюватися екран під час переходу між слайдами.

– "Інший звук": Інструмент для вибору звуку, який буде відтворюватися під час переходу між слайдами.

– "Швидкість": Інструмент для вибору швидкості переходу між слайдами. Ви можете вибрати швидкість "Швидко", "Середньо" або "Повільно".

– "Відмінити перехід": Інструмент для відміни налаштування переходу між слайдами.

Ці інструменти дозволяють користувачам налаштовувати ефекти переходу між слайдами в презентації, щоб зробити її більш цікавою та привабливою. Ви можете вибрати з більш ніж 50 різних ефектів переходу, встановити час, протягом якого буде

показано перехід, вибрати звук та швидкість переходу. Налаштування переходів між слайдами дозволяють підкреслити головні ідеї та визначити певну логіку розповіді в презентації.

Інструменти стрічки "Анімація"

Стрічка "Анімація" у Microsoft PowerPoint містить інструменти для додавання анімації до об'єктів на слайдах презентації. На рисунку 7.5 наведено інструменти стрічки "Анімація".

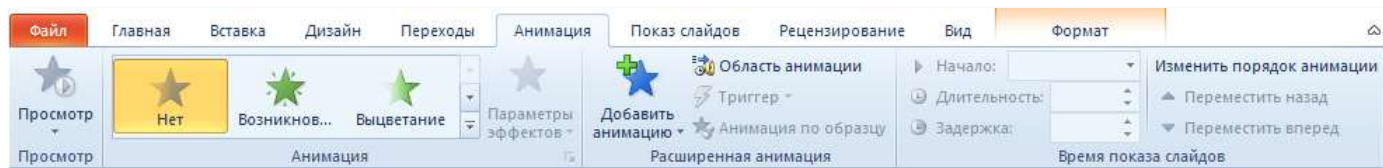


Рисунок 7.5 – Інструменти стрічки "Анімація"

Основні інструменти на цій стрічці включають:

- "Додати анімацію": Інструмент для додавання анімації до об'єктів на слайді. Ви можете вибрати з різних видів анімації, таких як "З'явлення", "Рух" та "Зникає".
- "Додаткові анімації": Інструмент для додавання додаткових анімацій до об'єктів на слайді.
- "Налаштування анімації": Інструмент для налаштування параметрів анімації, таких як час початку, тривалість та спосіб запуску.
- "Анімаційна панель": Інструмент для управління анімацією на слайді. Ви можете переміщати анімаційні об'єкти, змінювати їх порядок та налаштовувати параметри анімації.
- "Прев'ю анімації": Інструмент для перегляду анімації на слайді.
- "Відмінити анімацію": Інструмент для відміни анімації на слайді.

Ці інструменти дозволяють користувачам додавати анімацію до об'єктів на слайдах презентації, щоб зробити її більш динамічною та привабливою. Ви можете вибирати з різних видів анімації, налаштовувати їх параметри та управляти анімаційними об'єктами на слайді. Використання анімації може допомогти підкреслити ключові моменти та ідеї в презентації та зробити її більш запам'ятовуваною для аудиторії.

Інструменти стрічки "Показ слайдів"

Стрічка "Показ слайдів" у Microsoft PowerPoint містить інструменти для перегляду та управління презентацією в режимі показу слайдів. На рисунку 7.6 наведено інструменти стрічки "Показ слайдів".

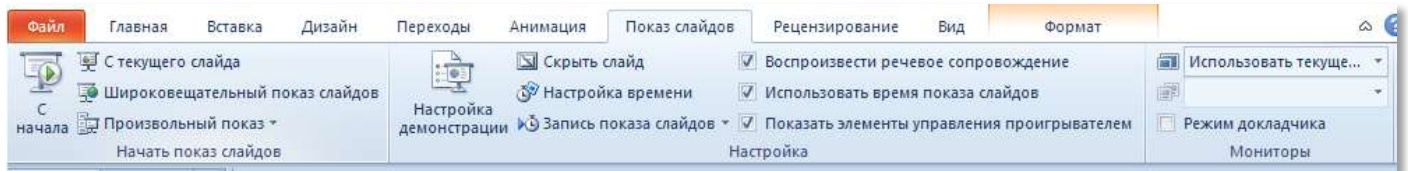


Рисунок 7.6 – Инструменты стрічки "Показ слайдів"

Основні інструменти на цій стрічці включають:

- "Запустити показ слайдів": Інструмент для запуску показу слайдів в повноекранному режимі.
- "Показ слайдів з початку": Інструмент для запуску показу слайдів з першого слайда.
- "Показ слайдів з поточного слайда": Інструмент для запуску показу слайдів з поточного слайда.
- "Налаштування показу слайдів": Інструмент для налаштування параметрів показу слайдів, таких як режим показу, час показу та наявність звуку.
- "Перегляд слайдів": Інструмент для перегляду всіх слайдів в презентації.
- "Зупинити показ слайдів": Інструмент для зупинки показу слайдів та повернення до режиму редагування.

Ці інструменти дозволяють користувачам легко запустити показ слайдів та управляти ним в режимі показу. Ви можете налаштувати параметри показу, щоб забезпечити оптимальний досвід для вашої аудиторії. Ви можете встановлювати час показу слайдів, вибирати режим показу, додавати звукові ефекти та багато іншого. Використання цих інструментів може допомогти підвищити ефективність вашої презентації та забезпечити її успішний вплив на вашу аудиторію.

Інструменти стрічки "Рецензування"

Стрічка "Рецензування" у Microsoft PowerPoint містить інструменти для співпраці та рецензування презентацій з іншими користувачами. На рисунку 7.7 наведено інструменти стрічки "Рецензування".

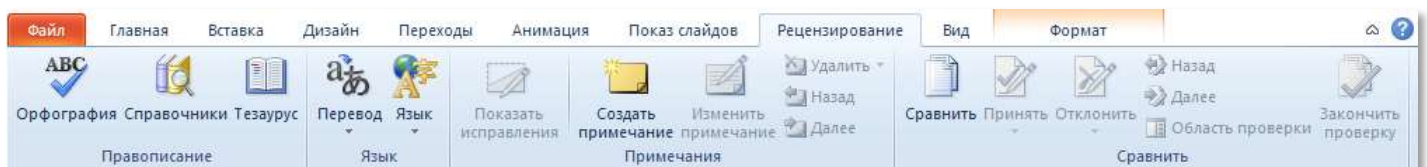


Рисунок 7.7 – Инструменты стрічки "Рецензування"

Основні інструменти на цій стрічці включають:

- "Перевірка правопису": Інструмент для перевірки правопису у тексті слайдів.

- "Переклад": Інструмент для перекладу тексту слайдів на іншу мову.
- "Коментарі": Інструмент для додавання коментарів до слайдів або відповіді на коментарі інших користувачів.
- "Підказки": Інструмент для додавання підказок до слайдів, які можна використовувати під час показу презентації.
- "Захист": Інструмент для захисту презентації від несанкціонованого доступу.

Ці інструменти дозволяють користувачам співпрацювати над презентацією та ділитися своїми коментарями і відгуками. Завдяки можливості додавати коментарі та підказки, користувачі можуть легко взаємодіяти та спілкуватися щодо вмісту слайдів. Крім того, інструмент "Перевірка правопису" дозволяє користувачам легко перевірити правильність написання тексту на слайдах. Інструмент "Захист" дозволяє захистити презентацію від несанкціонованого доступу та зберегти конфіденційність ваших даних. Загалом, ці інструменти роблять процес створення та редагування презентацій більш зручним та ефективним для користувачів.

Інструменти стрічки "Вид"

Стрічка "Вид" у Microsoft PowerPoint містить інструменти для налаштування вигляду презентації та її елементів. На рисунку 7.8 наведено інструменти стрічки "Вид".

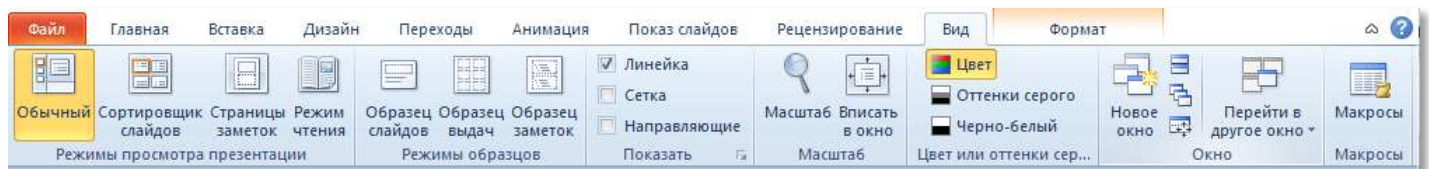


Рисунок 7.8 – Інструменти стрічки "Вид"

Основні інструменти на цій стрічці включають:

- "Масштаб": Інструмент для збільшення або зменшення масштабу слайдів.
- "Вигляд слайдів": Інструмент для вибору та налаштування макетів та дизайну слайдів.
- "Зона презентації": Інструмент для налаштування розміру та пропорцій сторін слайдів.
- "Майстер перегляду": Інструмент для налаштування способу показу презентації, зокрема налаштування перегляду слайдів, налаштування слайд-шоу, додавання звукових ефектів та інших елементів.
- "Огляд документа": Інструмент для швидкого перегляду всіх слайдів презентації та їх змісту у вигляді мініатюр.

Ці інструменти дозволяють користувачам налаштувати вигляд та формат презентації, вибрати необхідний макет та дизайн для слайдів, а також налаштувати

режим показу презентації та звукові ефекти. Інструмент "Огляд документа" дозволяє користувачам швидко знайти необхідний слайд та легко переглянути зміст всієї презентації. В цілому, інструменти на стрічці "Вид" допомагають користувачам налаштувати вигляд презентації так, щоб вона виглядала професійно та привабливо для аудиторії.

Інструменти стрічки "Формат"

Стрічка "Формат" у Microsoft PowerPoint містить інструменти для форматування об'єктів на слайдах презентації. На рисунку 7.9 наведено інструменти стрічки "Формат".

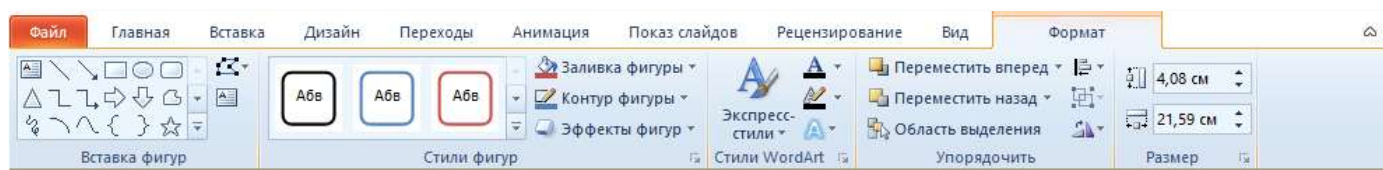


Рисунок 7.9 – Інструменти стрічки "Формат"

- Основні інструменти на цій стрічці включають:
- "Формат фігури": Інструмент для налаштування властивостей фігур на слайді, таких як розмір, контур, заливка, тінь та інші ефекти.
- "Формат зображення": Інструмент для налаштування властивостей зображень на слайді, таких як розмір, яскравість, контрастність, зміна кольору та інші ефекти.
- "Формат тексту": Інструмент для налаштування властивостей тексту на слайді, таких як шрифт, розмір, колір, вирівнювання та інші ефекти.
- "Формат таблиці": Інструмент для налаштування властивостей таблиць на слайді, таких як розмір, стиль, колір та інші ефекти.
- "Формат форми": Інструмент для налаштування властивостей форм на слайді, таких як розмір, стиль, заливка та інші ефекти.

Ці інструменти дозволяють користувачам налаштувати вигляд та формат об'єктів на слайдах презентації, зокрема фігур, зображень, тексту та таблиць. Вони дозволяють вибрати необхідні параметри для кожного об'єкту та налаштувати їх так, щоб вони виглядали привабливо та професійно. Наприклад, за допомогою інструменту "Формат тексту" можна вибрати підходящий шрифт, розмір та колір для тексту на слайді, щоб зробити його більш зрозумілим та привабливим для аудиторії. В цілому, інструменти на стрічці "Формат" дозволяють користувачам налаштувати вигляд кожного об'єкту на слайді так, щоб презентація виглядала професійно та привабливо.

Висновки

Microsoft PowerPoint є потужним інструментом для створення презентацій на різноманітні теми і з метою різних заходів. Він має широкий набір інструментів, що дозволяє додавати до слайдів текст, зображення, графіки, аудіо та відео матеріали, а також застосовувати анімацію та переходи між слайдами. PowerPoint надає можливість користувачеві редагувати слайди та оформлення, додавати коментарі, виконувати рецензування та виправлення помилок. Крім того, програма має можливості для показу презентації в режимі повного екрану з можливістю керування за допомогою клавіш на клавіатурі або спеціальних контролерів.

Завдяки своїй простоті та легкості використання, PowerPoint є часто використовуваним інструментом для створення презентацій на зустрічах, конференціях, лекціях та інших подібних заходах. Програма також дозволяє ефективно передавати інформацію, залучаючи слухачів та допомагаючи зберегти увагу протягом тривалого часу. Отже, у загальному, PowerPoint є потужним інструментом для створення презентацій, який може бути використаний в різних областях та на різноманітних заходах для зручного та ефективного представлення інформації.

Питання та завдання до контролю знань студентів

- 1 Як основні функції PowerPoint?
- 2 Як інструменти використовуються в PowerPoint для створення слайдів?
- 3 Як додати зображення до презентації в PowerPoint?
- 4 Як додати відео до презентації в PowerPoint?
- 5 Як додати аудіо до презентації в PowerPoint?
- 6 Як додати текст до презентації в PowerPoint?
- 7 Як вставити графік до презентації в PowerPoint?
- 8 Як використовувати фонові зображення та текстури в PowerPoint?
- 9 Як встановити анімацію для елементів на слайдах в PowerPoint?
- 10 Як використовувати переходи між слайдами в PowerPoint?
- 11 Як відобразити презентацію в режимі повного екрану в PowerPoint?
- 12 Як відобразити зміст презентації на двох моніторах в PowerPoint?
- 13 Як додати коментарі до слайдів в PowerPoint?
- 14 Як скористатися інструментом рецензування в PowerPoint?
- 15 Як зберегти презентацію в PowerPoint для подальшої роботи з нею?

Лекція № 8

з навчальної дисципліни Інформатика

Тема Графічний дизайн, як засіб візуальної комунікації

Мета заняття Охарактеризувати поняття дизайну, стилістичної єдності та композиції, здійснити класифікацію комп'ютерної графіки та засобів її обробки

Матеріально-технічне забезпечення та дидактичні засоби, ТЗН

1 Інтерактивний вид діяльності, створений засобами Moodle

2 Мультимедійний презентаційний матеріал

Час – 2 години

План проведення лекції

Структура лекції	Відведений час	Методичні вказівки
1 Організаційна частина	5 хв.	Включає в себе: <i>привітання</i> , яке має на меті привернути увагу, забезпечити контакт лектора з аудиторією; <i>визначення присутності студентів на занятті</i>
2 Актуалізація опорних знань, перевірка вивченого матеріалу та мотивація навчальної діяльності студентів	10 хв.	Включає в себе: <i>вступне зауваження, мотивацію актуальності теми, перевірку попереднього матеріалу, формулювання мети лекції, огляд головних питань теми (вказати питання та завдання для перевірки знань студентів)</i>
3 Основна частина (викладення навчальних питань лекції)	60 хв.	Головне місце приділяється <i>викладенню основного змісту матеріалу теми, його аналізу, узагальненню висунутих положень</i> . Для успіху лекції важливе значення має поділ матеріалу на розділи, основні питання
4 Заклучна частина Домашнє завдання: (відповідно до робочої програми)	5 хв.	Включає в себе: <i>узагальнення, теоретичні і фактичні висновки, закріплення вивченого на лекції матеріалу</i>

Література

1 Інформатика (рівень стандарту) : підруч. для 10-го (11-го) кл. закл. заг. серед. освіти / Й.Я. Ривкід Т.І. Лисенко, Л.А. Чернікова, В.В. Шакотько. — Київ : Генеза, 2018. — 144 с. : іл.

2 Комп'ютерна графіка: навч. посіб. для студентів напряму підгот. / А.С. Василюк, Н.І. Мельникова. «Системний аналіз». — Львів: Вид-во Львів. політехніки, 2016. — 308 с. : іл.

Навчальні матеріали лекції

ВСТУП

Графічний дизайн — це мистецтво творення візуальної комунікації, що використовує зображення, колір, форму, текст та інші елементи для передачі повідомлень. У сучасному світі, дизайн використовується скрізь — від логотипів та пакувань до книжок та веб-сторінок. Графічний дизайнер повинен розуміти принципи дизайну, вміти працювати з програмними засобами та мати здатність передавати ефективно повідомлення через візуальні засоби. У цьому вступі ми розглянемо, як графічний дизайн може допомогти візуальній комунікації та як він використовується в сучасному світі.

У сучасному сьогоднішньому світі візуалізувати ідеї та концепції допомагають комп'ютерні технології. Комп'ютерна графіка є невід'ємною частиною сучасного світу, оскільки дозволяє створювати віртуальні об'єкти, які можуть бути використані у різних сферах, від реклами до медицини. Вона включає в себе різноманітні технології, такі як 2D та 3D графіка, рендеринг, комп'ютерну анімацію, віртуальну реальність та багато іншого.

НАПРЯМКИ КОМП'ЮТЕРНОГО ДИЗАЙНУ

Комп'ютерний дизайн є широкою галуззю, яка охоплює різноманітні напрямки та спеціалізації. Зазначимо деякі з найпоширеніших напрямків комп'ютерного дизайну.

Веб-дизайн

Розробка та створення дизайну для веб-сайтів та інтернет-проектів. Це включає в себе розміщення елементів, кольорову палітру, типографіку та інші фактори, що сприяють створенню естетично привабливого та функціонального веб-дизайну.

Графічний дизайн

Розробка графічних елементів для друкованих матеріалів, які можуть включати логотипи, постери, брошури, рекламні матеріали та інше. Графічний дизайнер працює з типографією, кольорами, композицією та іншими важливими аспектами, щоб створити візуально привабливі продукти.

UX/UI-дизайн

Розробка інтерфейсів користувача (UI) та взаємодії користувача (UX) для програмного забезпечення, мобільних додатків та інших цифрових продуктів. UX/UI-дизайнер створює інтуїтивно зрозумілі та ефективні інтерфейси, які забезпечують зручну взаємодію користувачів з продуктом.

Анімація та візуальні ефекти

Розробка рухомих зображень, відео та спеціальних візуальних ефектів для фільмів, реклами, ігор та інших мультимедійних проектів. Це включає створення 2D або 3D анімації, моделювання об'єктів та створення реалістичних сцен.

Промисловий дизайн

Промислові дизайнери використовують комп'ютерні програми та інструменти для проектування та моделювання продуктів з урахуванням їх функціональності, форми, матеріалів, виробничих процесів та масштабування виробництва.

СФЕРИ ЗАСТОСУВАННЯ КОМП'ЮТЕРНОЇ ГРАФІКИ

Комп'ютерна графіка — це розділ інформатики, у якому вивчаються методи створення й опрацювання зображень за допомогою комп'ютера.

На даний момент комп'ютерну графіку розглядають, як окрему дисципліну, в якій, з одного боку, накопичено значний багаж знань, з іншого боку, здійснюється постійний розвиток методів, алгоритмів та практичних застосувань.

Основним об'єктом комп'ютерної графіки є графічне зображення: малюнок або фотографія, діаграма або графік, схема або технічне креслення, архітектурний ескіз або рекламний плакат, кадр з мультфільму або відеоролика тощо.

У комп'ютерній графіці розглядаються такі завдання:

- представлення зображення в комп'ютерній графіці;
- підготовка зображення до візуалізації;
- створення зображення;
- здійснення дій із зображенням.

Основні сфери застосування технологій комп'ютерної графіки такі:

- графічний інтерфейс користувача;
- спецефекти, кінематографія й телебачення;
- цифрове телебачення, Інтернет, відеоконференції;
- обробка цифрових фотографій;
- комп'ютерні ігри, системи віртуальної реальності.

Сучасні апаратні та програмні засоби значно розширюють спектр застосування комп'ютерної графіки. Це зручні інструменти для роботи з графічними зображеннями, не лише для професіоналів, а й для пересічних користувачів.

На рисунку 8.1 наведено структурну схему засобів комп'ютерної графіки.



Рисунок 8.1 – Структурна схема засобів комп'ютерної графіки

КЛАСИФІКАЦІЯ КОМП'ЮТЕРНОЇ ГРАФІКИ

Комп'ютерна графіка може бути класифікована за різними критеріями. Критерії класифікації комп'ютерної графіки включають напрямки та аспекти, наведено на рисунку 8.2.



Рисунок 8.2 – Структурна схема класифікації комп'ютерної графіки

За рівнем деталізації моделей

Векторна графіка використовує геометричні об'єкти, такі як точки, лінії та криві, для представлення графічних об'єктів. Цей тип графіки часто використовується для 2D-ілюстрацій, дизайну логотипів та іншого векторного малювання.

Растрова графіка використовується для представлення графічних об'єктів за допомогою пікселів або точок. Зображення складається з масиву пікселів, кожен з яких має визначене значення кольору. Растрова графіка використовується в ігровій графіці, комп'ютерних анімаціях та фотографіях.

За метою використання

Графіка для відображення

Тип графіки, що включає розробку графічних інтерфейсів користувача (GUI) для програм, веб-сторінок та мобільних додатків. Цей тип графіки використовується для створення кнопок, меню, іконок та інших елементів, з якими користувач може взаємодіяти.

Графіка для моделювання і візуалізації:

Тип графіки, що використовується для створення комп'ютерних моделей об'єктів, їх анімації та візуалізації результатів. Він широко застосовується в галузі комп'ютерної графіки, комп'ютерного моделювання, архітектури, інженерії та інших галузях.

Тип алгоритмів і технологій

Рендеринг — алгоритми і технології для створення реалістичних зображень, включаючи алгоритми трасування променів, растрове і векторне рендерінги.

Анімація — техніки створення рухливих зображень та анімаційних ефектів.

Віртуальна реальність — технології для створення іммерсивних віртуальних середовищ, включаючи віртуальний світ, віртуальний тренажер і інші VR-додатки.

Спосіб взаємодії

За способом взаємодії з користувачем комп'ютерну графіку можна розділити на класи:

Статична графіка — зображення, які не реагують на дії користувача.

Інтерактивна графіка — зображення, які можуть реагувати на дії користувача, наприклад, рухатися або реагувати на кліки миші.

За рівнем реалізації

Комп'ютерна графіка може бути класифікована за рівнем реалізації, що означає, яким чином графіка обробляється та відображається.

Реальний час (Real-time):

Реально-часова комп'ютерна графіка означає, що графічні об'єкти генеруються та відображаються миттєво без помітної затримки. Це важливо для ігрової графіки, віртуальної реальності, комп'ютерної симуляції та інших застосувань, де важлива взаємодія в реальному часі з користувачем.

Пререндерінг (Pre-rendered):

Пререндерена комп'ютерна графіка використовується для створення графічних об'єктів, які не вимагають миттєвого відображення. Це включає промальовкування і анімацію, які можуть бути вироблені заздалегідь і потім відтворені безпосередньо. Пререндерена графіка використовується в кіноіндустрії, рекламі, архітектурному візуалізації та інших сферах, де важлива висока якість зображення та деталізація.

Інтерактивна графіка (Interactive Graphics):

Інтерактивна комп'ютерна графіка дозволяє користувачам взаємодіяти з графічними об'єктами у реальному часі. Це включає взаємодію з 2D- та 3D-об'єктами, переміщення, обертання, масштабування та інші дії, що впливають на зображення. Вона може охоплювати створення та маніпулювання двомірними об'єктами, такими як текст, форми, малюнки, а також створення, редагування та взаємодію з тривимірними об'єктами, такими як моделі, анімації та сцени.

Висновки

Отже, комп'ютерний дизайн є невід'ємною частиною сучасного світу, він використовується в багатьох сферах, починаючи від реклами та маркетингу до розробки інтерфейсів та продуктів. Комп'ютерний дизайн допомагає створювати візуально привабливі продукти, що привертають увагу та створюють позитивне враження у користувачів. Це особливо важливо в рекламі та маркетингу, де ефективність залежить від здатності привернути увагу аудиторії.

Комп'ютерний дизайн не обмежується лише зовнішнім виглядом, він також спрямований на забезпечення зручності та функціональності використовуваних продуктів. Це включає в себе розробку ефективних інтерфейсів, логіку взаємодії тощо. забезпечення зручності використання.

Комп'ютерний дизайн знаходить застосування в різних сферах, таких як веб-дизайн, графічний дизайн, промисловий дизайн, архітектурний дизайн. Він дозволяє створювати інноваційні рішення та поліпшувати користувацький досвід у багатьох галузях. Комп'ютерний дизайн допомагає ефективно комунікувати і передавати інформацію за допомогою візуальних засобів.

Питання та завдання до контролю знань студентів

- 1 Які основні принципи та елементи дизайну використовуються в комп'ютерному дизайні?
- 2 Які програми та інструменти найчастіше використовуються для створення графічних проектів?
- 3 Як впливає комп'ютерний дизайн на рекламу та маркетингові стратегії?
- 4 Які основні принципи ергономіки враховуються при розробці інтерфейсів та веб-дизайну?
- 5 Які критерії використовуються для оцінки якості комп'ютерного дизайну?
- 6 Як використовується комп'ютерний дизайн в індустрії виробництва ігор?
- 7 Які нові тренди та інновації спостерігаються в комп'ютерному дизайні?
- 8 Як комп'ютерний дизайн впливає на архітектурне проектування та візуалізацію будівель?
- 9 Які виклики та проблеми пов'язані зі створенням мультимедійних проектів в комп'ютерному дизайні?
- 10 Як комп'ютерний дизайн сприяє створенню інтерактивних дослідницьких інструментів та симуляцій?

Лекція № 9

з навчальної дисципліни Інформатика

Тема Растрова та векторна графіка

Мета заняття Ознайомити із засобами обробки растрової, векторної графіки, досягти оволодіння принципами розробки та редагування графічних об'єктів

Матеріально-технічне забезпечення та дидактичні засоби, ТЗН

1 Інтерактивний вид діяльності, створений засобами Moodle

2 Мутимедійний презентаційний матеріал

Час – 2 години

План проведення лекції

Структура лекції	Відведений час	Методичні вказівки
1 Організаційна частина	5 хв.	Включає в себе: <i>привітання</i> , яке має на меті привернути увагу, забезпечити контакт лектора з аудиторією; <i>визначення присутності студентів на занятті</i>
2 Актуалізація опорних знань, перевірка вивченого матеріалу та мотивація навчальної діяльності студентів	10 хв.	Включає в себе: <i>вступне зауваження, мотивацію актуальності теми, перевірку попереднього матеріалу, формулювання мети лекції, огляд головних питань теми</i> (вказати питання та завдання для перевірки знань студентів)
3 Основна частина (викладення навчальних питань лекції)	60 хв.	Головне місце приділяється <i>викладенню основного змісту матеріалу теми, його аналізу, узагальненню висунутих положень</i> . Для успіху лекції важливе значення має поділ матеріалу на розділи, основні питання
4 Заклучна частина Домашнє завдання: (відповідно до робочої програми)	5 хв.	Включає в себе: <i>узагальнення, теоретичні і фактичні висновки, закріплення вивченого на лекції матеріалу</i>

Література

1 Інформатика (рівень стандарту) : підруч. для 10-го (11-го) кл. закл. заг. серед. освіти / О.О. Бондаренко, В.В. Ластовецький, О.П. Пилипчук, Є.А. Шестопапов. — Харків : Вид-во «Ранок», 2018. — 175 с. : іл.

2 Responsive Web Design with Adobe Photoshop 1st Edition, Adobe Press / Dan Rose. — 1st Edition (January 12, 2015) – 250 p.

3 Adobe Illustrator: A Complete Course and Compendium of Features Paperback – Rocky Nook; Illustrated Edition / Jason Hoppe. – 388p.

Навчальні матеріали лекції

ВСТУП

За останні десятиліття комп'ютерна графіка пройшла значний шлях розвитку, завдяки постійному зростанню потужності комп'ютерів і появі нових програмних засобів. Вона знайшла широке застосування в різних галузях, таких як реклама, мультимедіа, інтерактивні додатки, дизайн, архітектура, наукові дослідження та виробництво ігор.

Одним із основних завдань комп'ютерної графіки є створення візуально збагачених інтерфейсів, які допомагають нам взаємодіяти з комп'ютерами та іншими пристроями. Вона дозволяє нам відтворювати реальні об'єкти та сцени, створювати вражаючі візуальні ефекти та переносити нас у віртуальні світи.

Крім того, комп'ютерна графіка використовується в наукових дослідженнях та симуляціях для візуалізації складних процесів, моделювання реальних явищ та аналізу даних. Це сфера, що поєднує технології, творчість та візуальне мистецтво для створення, редагування та відображення графічних зображень на комп'ютерах. За допомогою комп'ютерної графіки можна створювати реалістичні сцени, дизайни, анімацію, спеціальні ефекти та багато іншого.

РАСТРОВА КОМП'ЮТЕРНА ГРАФІКА

Растрова комп'ютерна графіка — це тип комп'ютерної графіки, в якому графічні об'єкти представлені у вигляді масиву пікселів або точок, розташованих у вигляді растра (сітки). Кожен піксель має визначений колір або значення, яке відображається на візуальному виході, такому як монітор, екран телевізора або друкована поверхня.

Основні особливості растрової комп'ютерної графіки

Піксель — найменша адресована одиниця зображення, яка має власну позицію та колір. Піксель являє собою базовий елемент растрової графіки. Щільність пікселів визначається роздільною здатністю (роздільною здатністю) пристрою відображення.

Роздільна здатність визначає кількість пікселів, які можуть бути відображені на одиницю довжини або площі. Вища роздільна здатність дозволяє отримати більш деталізоване та гладке зображення.

Формати файлів являє стандарт кодування у якому растрова графіка зберігається у форматі файлу, такому як JPEG, PNG, BMP або GIF. Кожен формат має свої особливості, які впливають на якість зображення, розмір файлу та підтримувані функції, такі як прозорість або анімація.

Редакція: — особливості обробки через застосування графічних редакторів, таких як Adobe Photoshop, GIMP або Paint.NET. Вони дозволяють створювати, редагувати та маніпулювати растровими зображеннями.

Переваги та недоліки растрової графіки

Растрова комп'ютерна графіка має свої власні переваги:

Реалістичність: растрова графіка може набагато точніше відтворити складні деталі та текстури, що дозволяє створювати більш реалістичні зображення. За допомогою пікселів різного кольору та яскравості можна досягти більшої візуальної глибини та різноманітності.

Можливість деталізації: растрові зображення можуть мати велику кількість пікселів на одиницю площі, що дозволяє отримати високу роздільну здатність та деталізацію. Це особливо важливо для зображень з великою кількістю дрібних деталей, які необхідно відтворити точно.

Фотореалізм: растрова графіка забезпечує можливість створювати фотореалістичні зображення, які дуже близькі до фотографій. Вона дозволяє передати різні відтінки, тіні, градієнти, освітлення та інші характеристики, що забезпечують відчуття реалізму.

Обробка фотографій: растрова графіка є основним форматом для фотографій, і вона дозволяє робити різні операції над зображеннями, такі як ретушування, кольорова корекція, фільтри, зміна розміру, роздільна здатність та інші. Растровий формат є універсальним для роботи з фотографіями та їх подальшої обробки.

Анімація: растрова графіка дозволяє створювати анімацію з високою якістю. Анімація растрової графіки включає створення рухомих зображень за допомогою послідовності растрових зображень, які показують різні стани об'єктів протягом часу.

Серед недоліків растрової графіки можна зазначити такі:

Залежність від роздільної здатності: растрова графіка залежить від роздільної здатності, тобто кількості пікселів на одиницю площі. При збільшенні розміру зображення без використання високої роздільної здатності, можуть виникати проблеми з розтягуванням, розмиттям та втратою деталей.

Скалінг: при зміні масштабу растрового зображення, пікселі просто розтягуються або стискаються, що може призводити до втрати якості та розмиття зображення. Особливо це помітно при збільшенні розміру зображення, де пікселі стають видимими.

Об'єм файлу: растрові зображення можуть бути великими в розмірі, особливо якщо мають високу роздільну здатність або багато деталей. Це може призвести до проблем зі зберіганням, передачею та завантаженням файлів, особливо в мережових середовищах з обмеженою пропускну здатністю.

Обробка та редагування: растрові зображення вимагають спеціального програмного забезпечення для їх редагування та обробки. Вносити зміни в растрове зображення може бути складніше, особливо якщо потрібно зберегти деталі та якість.

Сталість деталей: растрові зображення мають фіксовану кількість пікселів, і вони не можуть бути збільшені без втрати деталей. Якщо растрове зображення має низьку роздільну здатність або низьку якість, важко покращити його

ВЕКТОРНА КОМП'ЮТЕРНА ГРАФІКА

Векторна комп'ютерна графіка — це метод представлення та обробки графічних об'єктів, в якому використовуються геометричні об'єкти, такі як точки, лінії, криві та полігони, для опису форм та зображень. У векторній графіці графічні об'єкти представлені у вигляді математичних рівнянь або векторних даних, які визначають їх розташування, форму, розмір та кольори.

Особливості векторної комп'ютерної графіки

Масштабованість є однією з ключових переваг векторної графіки, можливість без втрати якості змінювати розмір об'єктів. Оскільки графіка представлена у вигляді математичних рівнянь, вона може бути легко масштабована без розмивання або розтягування.

Редагованість векторні зображення можуть бути легко редаговані та модифіковані. Вони складаються з окремих елементів, таких як точки та криві, які можна переміщувати, змінювати форму, з'єднувати та видаляти. Це дає дизайнерам більшу гнучкість та контроль над графічними об'єктами.

Компактність векторні зображення зазвичай займають менше місця в пам'яті, оскільки вони описуються у вигляді математичних рівнянь та векторних даних. Це особливо важливо при збереженні та передачі графічних файлів через мережу.

Роздільна здатність векторні зображення можуть бути відтворені з високою роздільною здатністю на будь-яких пристроях відображення. Оскільки векторна графіка представлена математичними рівняннями та векторами, роздільна здатність відповідає точності та деталізації цих об'єктів. Це означає, що векторна графіка може мати дуже високу роздільну здатність та деталізацію без втрати якості зображення.

Переваги та недоліки векторної графіки

Векторна графіка базується на математичних рівняннях та векторах, що дозволяє без втрати якості змінювати розмір зображень.

Векторні зображення зазвичай мають набагато менший розмір файлу порівняно з растровими зображеннями.

Векторні зображення легко редагуються та модифікуються. Об'єкти та елементи можуть бути пересунуті, масштабовані, змінені форми та кольори. Можна вносити зміни в окремі елементи без впливу на решту зображення. Це робить їх більш гнучкими для редагування та налаштування.

Векторна графіка зберігає високу якість друку незалежно від роздільної здатності використовуваного пристрою. Вона не має обмежень, пов'язаних з пікселями, тому друк з векторних зображень дає чіткі та різкі результати навіть при великому масштабуванні.

Векторна графіка є ідеальним форматом для створення логотипів, ілюстрацій та графічних елементів, оскільки вона дозволяє точне та детальне відтворення форм, ліній та кольорів.

Незважаючи на багато переваг, векторна графіка також має деякі недоліки. Ось декілька з них:

Обмежена для складних фотореалістичних зображень

Векторна графіка не є ідеальним варіантом для створення складних фотореалістичних зображень, таких як фотографії або зображення з багато деталей. Растрова графіка зазвичай краще підходить для таких завдань.

Обробка затінків та текстур

Векторна графіка зазвичай не має вбудованих можливостей для обробки затінків, текстур та інших візуальних ефектів. Для досягнення більш складених візуальних ефектів, може бути необхідне використання растрової графіки або спеціалізованого програмного забезпечення.

Вимоги до обробки

Робота з векторною графікою вимагає спеціалізованого програмного забезпечення, яке може бути відносно складним для навчання та використання. Для складних проєктів може знадобитися додатковий час та навички для створення векторних зображень.

Великі розміри файлів

Векторні зображення, що містять багато складних об'єктів або деталей, можуть мати великий розмір файлу. Це може становити проблему при зберіганні, передачі або завантаженні файлів, особливо в мережевих середовищах з обмеженою пропускнуою здатністю.

Недостатня підтримка деяких програм та пристроїв

Хоча векторні зображення підтримуються багатьма програмами та пристроями, є деякі програми, які не підтримують цей формат. Тому, для певних випадків, може знадобитися конвертація векторного зображення у растровий формат.

ЗАСОБИ ОБРОБКИ РАСТРОВОЇ ГРАФІКИ

Adobe Photoshop

Це один з найпопулярніших програмних засобів для редагування та обробки растрових зображень. Він має багатий набір інструментів для корекції кольору, налаштування контрастності, ретушування, створення ефектів та іншого.

GIMP

Це безкоштовний і відкритий редактор зображень, який надає можливості для редагування та маніпулювання растровими зображеннями. Він має подібні функції до Adobe Photoshop, включаючи шари, фільтри, інструменти малювання тощо.

Corel PaintShop Pro

Це інша популярна програма для редагування та обробки растрової графіки. Вона має велику кількість інструментів для налаштування зображень, ретушування, створення колажів, векторизації та багато іншого.

Pixlr

Це онлайн-редактор зображень, який надає базові функції редагування растрових зображень. Він може бути зручним для швидкого редагування та обробки зображень без необхідності установки програмного забезпечення.

Інтерфейси популярних редакторів растрової графіки

Хоча популярні редактори растрової графіки можуть мати різні компоновки інтерфейсу головного вікна, деякі елементи, такі як головне меню, панель інструментів та палітри, можуть бути присутні в багатьох редакторах, їх розташування та організація можуть не дуже відрізнятися. Компоновка інтерфейсу залежить від дизайну та концепції кожного редактора. Компанії-розробники намагаються забезпечити зручний та ефективний робочий потік для користувачів, але це може призводити до різних розташувань та організацій елементів.

На рисунку 9.1 наведено компоновку інтерфейсу вікна Adobe Photoshop.

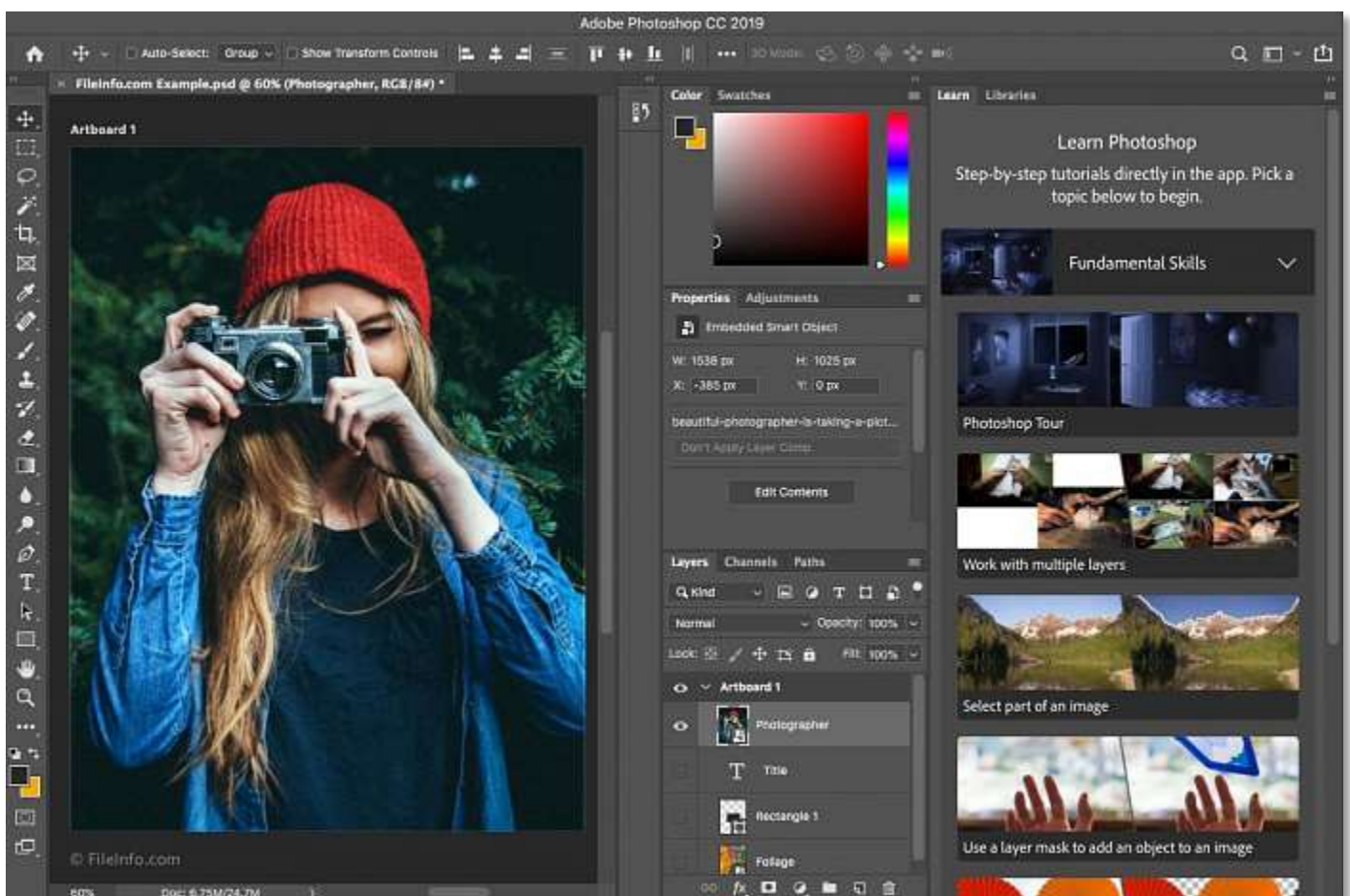


Рисунок 9.1 – Компоновка інтерфейсу голоного вікна редактора Adobe Photoshop

Редактор графіки Adobe Photoshop має типову компоновку з головним меню у верхній частині, панеллю інструментів зліва та палітрами праворуч. Стандартно головне вікно редактора має таку компоновку:

- головне меню розташоване у верхній частині вікна, з доступом до всіх основних команд та налаштувань;
- панель інструментів знаходиться зліва від області робочого простору і містить різні інструменти для вибору, малювання, ретуші та інших операцій;
- палітри, такі як Layers (Шари), Channels (Канали), History (Історія) розташовані зазвичай праворуч, надаючи доступ до різних налаштувань та контролю над зображенням;
- область робочого простору займає основну частину вікна, де відображається редаговане зображення.

Редактор GIMP також має схожий розподіл елементів, але може мати відмінності в організації палітр та меню. Стандартно головне вікно редактора має таку компоновку:

- головне меню розташоване у верхній частині вікна і містить різні команди та налаштування;
- панель інструментів знаходиться зліва від області робочого простору і містить інструменти для малювання, ретуші, вибору тощо;
- палітри, такі як Layers (Шари), Channels (Канали), Paths (Траєкторії), розташовані зазвичай праворуч, надаючи доступ до налаштувань та контролю над зображенням;
- область робочого простору відображає редаговане зображення.

Corel PaintShop Pro також має свою власну компоновку, яка може відрізнятися від інших редакторів. Стандартно головне вікно редактора має таку компоновку:

- головне меню знаходиться у верхній частині вікна, з доступом до різних команд та налаштувань;
- панель інструментів знаходиться зліва або праворуч від області робочого простору і містить різні інструменти для редагування та малювання;
- палітри, такі як Layers (Шари), Adjustments (Налаштування), розташовані зазвичай праворуч, надаючи доступ до різних налаштувань та контролю над зображенням.

Таким чином, компоновка інтерфейсу головного вікна популярних редакторів растрової графіки може варіюватися, але загальні елементи та функціональність залишаються подібними.

Сфера використання засобів растрової графіки

Як правило, один із розглянутих редакторів растрової графіки може виконати широкий спектр задач, пов'язаних з редагуванням та обробкою растрових зображень. Розширені функціональні можливості для створення, редагування та оптимізації растрових зображень дозволяють застосовувати будь-який із розглянутих редакторів растрової графіки для розв'язання таких задач:

- видалення дефектів, розмиття, виправлення кольорових недоліків, видалення небажаних об'єктів або елементів зображення;
- створення колажів, об'єднання різних елементів зображень, створення композицій, ретуш та обробка портретних фотографій;
- додавання фільтрів, стилів, текстур, ефектів освітлення, створення графічних елементів для веб-дизайну, розробки логотипів та інших графічних матеріалів;
- налаштування кольорової палітри, корекція кольору, зміна тону, насиченості та яскравості, створення кольорових градієнтів та палітр
- зміна розміру та роздільної здатності зображення, оптимізація для вебу, створення форматів файлів, підготовка друкованих матеріалів.

ЗАСОБИ ОБРОБКИ ВЕКТОРНОЇ ГРАФІКИ

Існує кілька популярних комп'ютерних засобів для обробки векторної графіки, які надають широкий набір інструментів та функцій для створення, редагування та маніпулювання векторними зображеннями. Ось огляд кількох з них:

Adobe Illustrator

Adobe Illustrator є одним з найпопулярніших програмних засобів для векторної графіки. Він має багатий набір інструментів для створення векторних малюнків, ілюстрацій, логотипів та графічних дизайнів. Програма надає можливість працювати з об'єктами, шляхами, кривими Безьє та багатьма іншими елементами векторного зображення. Вона також підтримує імпорт та експорт файлів у різних форматах.

CorelDRAW

CorelDRAW є ще одним популярним векторним графічним редактором. Він використовується для створення векторних малюнків, дизайну логотипів, рекламних матеріалів та інших графічних проектів. CorelDRAW має багато інструментів для маніпулювання об'єктами, створення кривих, редагування тексту та застосування ефектів до зображень.

Inkscape

Inkscape є безкоштовним векторним графічним редактором з відкритим вихідним кодом. Він пропонує різні інструменти для створення та редагування векторних зображень. Inkscape підтримує роботу зі шляхами, формами, текстом та шаблонами. Він також підтримує імпорт та експорт різних форматів файлів.

Affinity Designer

Affinity Designer є високопотужним редактором векторної графіки, розробленим компанією Serif. Це популярне програмне забезпечення, яке використовується професіоналами в галузі графічного дизайну, ілюстрацій та веб-дизайну. Функціонал додатку оснащений потужним редактором точок, який дозволяє точно керувати формою і шляхом об'єктів.

Інтерфейси популярних редакторів растрової графіки

Інтерфейс популярних векторних редакторів може варіюватися в залежності від конкретного програмного забезпечення. Однак, вони часто мають деякі загальні риси та подібні елементи інтерфейсу, оскільки базуються на загальних принципах векторної графіки та редагування. Основні компоненти інтерфейсу векторних редакторів включають:

- більшість векторних редакторів мають меню з різними опціями та панелі інструментів, які містять набір інструментів для створення, редагування та маніпулювання об'єктами;
- редактори векторної графіки зазвичай мають вікна та панелі налаштувань, які дозволяють контролювати параметри об'єктів, кольору, форми, штрихів та інших властивостей;
- полотно або робоча область — це основний простір, воно представлене у вигляді прямокутного полотна, на якому можна розміщувати й організовувати об'єкти;
- шари використовуються для організації та керування елементами вашого проекту;
- інструменти векторного редагування: дозволяють маніпулювати точками, вузлами, кривими та шляхами об'єктів, що складаються з векторних форм.

Сфери використання редакторів векторної графіки

Редактори векторної графіки широко використовуються в різних сферах для створення, редагування та маніпулювання векторними зображеннями.

Графічний дизайн

Редактори векторної графіки є незамінним інструментом для створення логотипів, постерів, брендінгу, рекламних матеріалів, пакування, друкованих дизайнів та іншої графічної продукції.

Ілюстрація

Векторні редактори використовуються художниками та ілюстраторами для створення цифрових малюнків, коміксів, карикатур, ілюстрацій до книг, артових робіт та багатьох інших творів мистецтва.

Веб-дизайн

Редактори векторної графіки використовуються для створення веб-елементів, іконок, кнопок, банерів, векторної графіки для інтерфейсу користувача та іншого графічного контенту для веб-сайтів та програмного забезпечення.

Архітектурне та промислове проектування

Редактори векторної графіки використовуються для створення планів будівель, схем, технічних креслень, дизайну інтер'єрів та інших графічних матеріалів, пов'язаних з архітектурою та промисловим проектуванням.

Мобільний дизайн та розробка ігор

Редактори векторної графіки використовуються для створення графіки для мобільних додатків, ігор та інтерактивних медіа. Вони дозволяють створювати векторні спрайти, персонажі, об'єкти, інтерфейси та інші графічні ефекти.

Висновки

Комп'ютерна графіка є незамінним інструментом у багатьох галузях, включаючи графічний дизайн, ілюстрацію, веб-дизайн, архітектурне та промислове проектування, мобільний дизайн та розробку ігор. Комп'ютерна графіка дозволяє створювати професійний та привабливий графічний контент, що відображається в рекламі, медіа, друкованих матеріалах, веб-сайтах та програмах.

Завдяки комп'ютерній графіці можливе збереження часу та зусиль у процесі створення графічних матеріалів, а також здійснення швидкої редакції та внесення змін до вже існуючих зображень. Розвиток технологій комп'ютерної графіки сприяє появі нових інновацій, таких як віртуальна реальність, розширена реальність та 3D-моделювання, що відкривають нові можливості для творчості та взаємодії з графічним контентом.

Питання та завдання до контролю знань студентів

- 1 Що таке растрова графіка і які основні принципи лежать в її основі? Що таке векторна графіка і як вона відрізняється від растрової графіки?
- 2 Які фактори впливають на якість растрових зображень, таких як роздільна здатність та глибина кольору?
- 3 Які методи компресії використовуються для зменшення розміру файлів растрових зображень?
- 4 Які переваги та недоліки використання растрової графіки порівняно з векторною графікою?
- 5 Як растрова графіка застосовується у виробництві ігор та анімації?
- 6 Які основні критерії використовуються для оцінки якості растрових зображень?
- 7 Які основні принципи та принципи роботи векторної графіки?
- 8 Які програми та інструменти найчастіше використовуються для створення та редагування векторної графіки?
- 9 Які переваги має використання векторної графіки порівняно з растровою графікою?
- 10 Які типи об'єктів можна створити за допомогою векторної графіки?
- 11 Які методи масштабування використовуються для зміни розміру векторних зображень?
- 12 Як використовується векторна графіка в дизайні логотипів та ідентичності бренду?
- 13 Як векторна графіка використовується в архітектурному проектуванні та ілюстраціях?
- 14 Які особливості малювання кривих та обробки шляхів векторної графіки?
- 15 Які критерії використовуються для оцінки якості векторних зображень?

Лекція № 10

з навчальної дисципліни **Інформатика**

Тема Системи автоматизованої обробки інформації

Мета заняття Дослідити системи автоматизованого проектування, ознайомитися із основними сферами використання, отримати загальні навички по застосуванню

Матеріально-технічне забезпечення та дидактичні засоби, ТЗН

1 Інтерактивний вид діяльності, створений засобами Moodle

2 Мультимедійний презентаційний матеріал

Час – 2 години

План проведення лекції

Структура лекції	Відведений час	Методичні вказівки
1 Організаційна частина	5 хв.	Включає в себе: <i>привітання</i> , яке має на меті привернути увагу, забезпечити контакт лектора з аудиторією; <i>визначення присутності студентів на занятті</i>
2 Актуалізація опорних знань, перевірка вивченого матеріалу та мотивація навчальної діяльності студентів	10 хв.	Включає в себе: <i>вступне зауваження, мотивацію актуальності теми, перевірку попереднього матеріалу, формулювання мети лекції, огляд головних питань теми (вказати питання та завдання для перевірки знань студентів)</i>
3 Основна частина (викладення навчальних питань лекції)	60 хв.	Головне місце приділяється <i>викладенню основного змісту матеріалу теми, його аналізу, узагальненню висунутих положень</i> . Для успіху лекції важливе значення має поділ матеріалу на розділи, основні питання
4 Заклучна частина Домашнє завдання: (відповідно до робочої програми)	5 хв.	Включає в себе: <i>узагальнення, теоретичні і фактичні висновки, закріплення вивченого на лекції матеріалу</i>

Література

1 Інформатика (рівень стандарту) : підруч. для 10-го (11-го) кл. закл. заг. серед. освіти / Й.Я. Ривкід Т.І. Лисенко, Л.А. Чернікова, В.В. Шакотько. — Київ : Генеза, 2018. — 144 с. : іл.

2 Комп'ютерна графіка: навч. посіб. для студентів напряму підгот. / А.С. Василюк, Н.І. Мельникова. «Системний аналіз». — Львів: Вид-во Львів. політехніки, 2016. — 308 с. : іл.

Навчальні матеріали лекції

ВСТУП

У сучасному світі, де швидкість, точність та ефективність є критичними факторами у розробці та проектуванні, системи автоматизованого проектування (САПР) виявляються незамінними інструментами для професіоналів різних галузей. Від інженерії та промислового дизайну до архітектури та механічного проектування, САПР забезпечують надійну та швидку розробку складних проектів.

САПР перетворили традиційний процес проектування, у якому використовуються паперові малюнки та ручні креслення, на віртуальне середовище, де інженери та дизайнери можуть працювати з об'єктами у цифровому форматі. Це дає їм можливість ефективно моделювати, аналізувати та оптимізувати свої проекти до їх фізичної реалізації. Завдяки автоматизації багатьох процесів, САПР допомагають зберегти час та ресурси, уникнути помилок та забезпечити високу якість проектів.

Системи автоматизованого проектування (САПР) є невід'ємною складовою сучасного інженерного та архітектурного середовища. Вони надають інженерам, дизайнерам та архітекторам потужні інструменти та технології для створення складних проектів у віртуальному середовищі. Завдяки автоматизації багатьох процесів проектування, САПР спрощують роботу та прискорюють розробку інноваційних продуктів і конструкцій.

СФЕРИ ВИКОРИСТАННЯ СИСТЕМ АВТОМАТИЧНОГО ПРОЕКТУВАННЯ

Системи автоматичного проектування (Computer-Aided Design, CAD) знаходять широке використання в різних сферах і промисловостях.

Застосування САПР у машинобудуванні

CAD-системи використовуються для проектування та моделювання деталей, складальних одиниць та машин. Вони дозволяють інженерам створювати 2D- та 3D-моделі, виконувати аналіз міцності та динаміки, підготовляти технічні креслення та виробничі документи. Системи автоматизованого проектування (САПР) в машинобудуванні знаходять широке використання та є невід'ємною частиною процесу проектування та розробки машин та механізмів. Вони можуть бути залучені до таких видів діяльності:

- тривимірне моделювання, що дозволяє візуалізувати та перевіряти конструкції перед фізичним виготовленням, а також виявляти можливі проблеми та конфлікти;
- моделювання руху та кінематика, що дозволяє вивчати взаємодію деталей, аналізувати та здійснити оптимізацію кінематичних параметрів, такі як швидкість, прискорення, сили та моменти;
- аналіз міцності та навантаження, що дозволяє проводити аналіз міцності конструкцій та виявляти можливі проблеми з навантаженням;

- генерація технічної документації, дозволяє автоматично генерувати технічні креслення, специфікації, список деталей та іншу документацію, необхідну для виробництва машин;
- симуляція та віртуальне тестування, дозволяє інженерам моделювати та симулювати роботу механізмів та машин.

Архітектура та будівництво

CAD-системи допомагають архітекторам та інженерам проектувати будівлі, створювати детальні моделі будівель та ландшафтів, виконувати аналіз стійкості та енергоефективності, а також генерувати креслення та специфікації для будівництва. Системи автоматизованого проектування (САПР) в архітектурі та будівництві знаходять широке застосування і відіграють важливу роль у процесі проектування, розробки та будівництва. Ось деякі основні застосування САПР в цих галузях:

- проектування будівель, дозволяють архітекторам і інженерам створювати детальні 2D- та 3D-моделі будівель;
- аналіз стійкості та енергоефективності, дозволяють проводити аналіз стійкості будівельних конструкцій та оцінювати їх енергоефективність;
- комунікація та співпраця, дозволяють архітекторам, інженерам та іншим учасникам проекту спілкуватися та співпрацювати;
- документація та креслення, дозволяють автоматично генерувати технічну документацію, креслення, специфікації та інші документи, необхідні для будівництва.

Виробництво та промисловість

CAD-системи застосовуються в промисловості для проектування та моделювання виробничих ліній, обладнання та інструментів. Вони допомагають виробникам покращити ефективність процесів, зменшити час розробки та виробництва, а також виконати аналіз стійкості, монтажу та ергономіки. Системи автоматизованого проектування (САПР) мають широке застосування в виробництві та промисловості і відіграють важливу роль у вдосконаленні процесів проектування, виробництва та управління. Ось деякі з основних застосувань САПР у цих галузях:

- проектування та розробка продукту: САПР дозволяють проектувати та моделювати продукти з високою точністю і деталізацією;
- виробництво та виготовлення — використовуються для планування та керування процесами виробництва;
- управління життєвим циклом продукту (PLM) — часто інтегруються з системами управління життєвим циклом продукту.

Автомобілебудування

CAD-системи широко використовуються у розробці автомобілів для проектування кузовів, двигунів, систем паливного споживання та інших компонентів. Вони дозволяють інженерам виконувати аналіз ефективності палива, безпеки та ергономіки, а також оптимізувати дизайн та виробничі процеси.

Системи автоматизованого проектування (САПР) в автомобілебудуванні відіграють важливу роль у розробці та виробництві автомобілів. Вони дозволяють інженерам та дизайнерам створювати та моделювати автомобільні компоненти, системи та весь автомобільний прототип. Ось деякі з основних застосувань САПР у цій галузі:

- дизайн та концептуалізація, дозволяють автомобільним дизайнерам створювати віртуальні 3D-моделі автомобілів з різними деталями, формами та стилями;
- механічне проектування, використовуються для проектування різних механічних компонентів автомобіля, таких як двигуни, шасі, підвіски, трансмісії та інші;
- електричне та електронне проектування, використовуються для проектування електричних та електронних систем в автомобілях, таких як системи керування двигуном, системи безпеки, системи розваг тощо;
- віртуальне тестування та аналіз, дозволяють проводити віртуальне тестування автомобілів на різних етапах розробки.

Електроніка та мікроелектроніки

Системи автоматизованого проектування (САПР) в електроніці та мікроелектроніці відіграють важливу роль у розробці та виробництві електронних пристроїв, мікросхем та інших компонентів. Вони дозволяють інженерам створювати та моделювати складні електронні системи, виконувати аналізи, вирішувати проблеми та вдосконалювати продукти. Ось деякі з основних застосувань САПР у цих галузях:

Схематичне проектування: САПР використовуються для створення електричних схем електронних пристроїв;

Моделювання та симуляція, дозволяють інженерам моделювати та симулювати роботу електронних пристроїв та мікросхем;

Розміщення компонентів, допомагають інженерам розміщувати компоненти на платі або мікросхемі.

СИСТЕМИ АВТОМАТИЗОВАНОГО ПРОЕКТУВАННЯ AUTODESK

Компанія Autodesk займається розробкою та постачанням програмного забезпечення для різних галузей, таких як архітектура, інженерія, будівництво, виробництво, медіа та розваги. Вони пропонують широкий спектр інструментів для комп'ютерного проектування (CAD), комп'ютерного виробництва (CAM), комп'ютерного інженерінгу (CAE) та інших областей дизайну та візуалізації.

Autodesk відома своїми інноваціями та активним впровадженням нових технологій, таких як штучний інтелект, віртуальна реальність та аддитивне виробництво, у своїх продуктах.

Компанія Autodesk є однією з провідних у світі у своїй галузі, і її програмне забезпечення використовують професіонали та компанії у багатьох галузях. Вона також надає послуги хмарного зберігання та співпраці, що дозволяє користувачам працювати над проектами разом та зберігати дані в безпечному місці.

Основні програмні продукти САПР Autodesk

На рисунку 10.1 наведено офіційний сайт компанії Autodesk із переліком програмних продуктів.

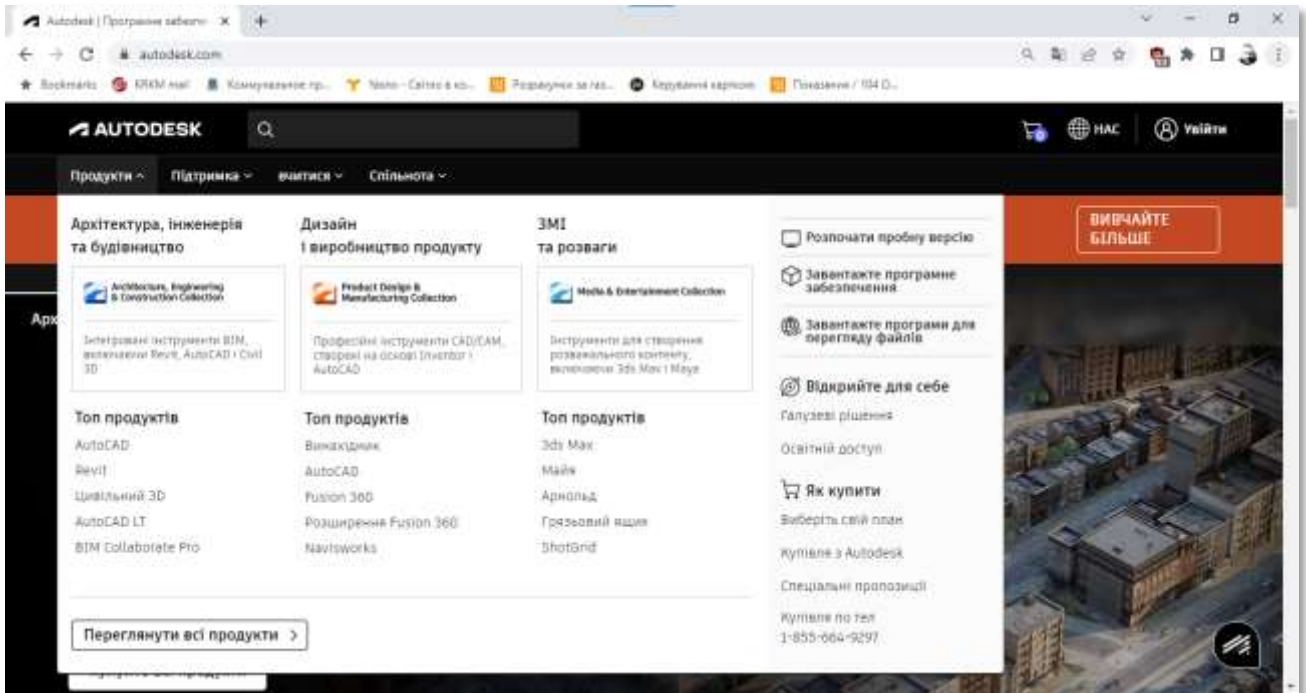


Рисунок 10.1 – Перелік програмних продуктів Autodesk

Засіб придбання програмного продукту САПР AutoCAD наведено на рисунку 10.2.

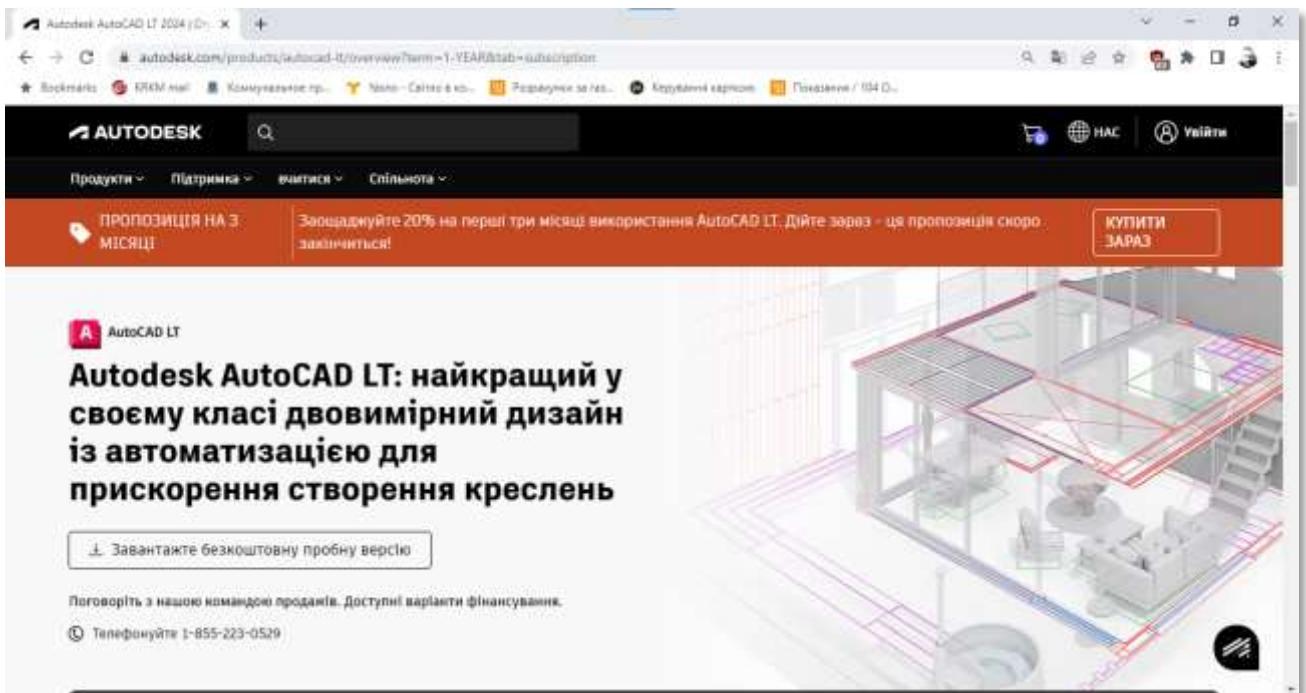


Рисунок 10.2 – Засіб придбання програмного продукту AutoCAD

Популярний програмний продукт для 2D та 3D-комп'ютерного проектування, який використовується в багатьох галузях, зокрема в архітектурі, машинобудуванні та інженерії.

Засіб придбання програмного продукту САПР Inventor наведено на рисунку 10.3.

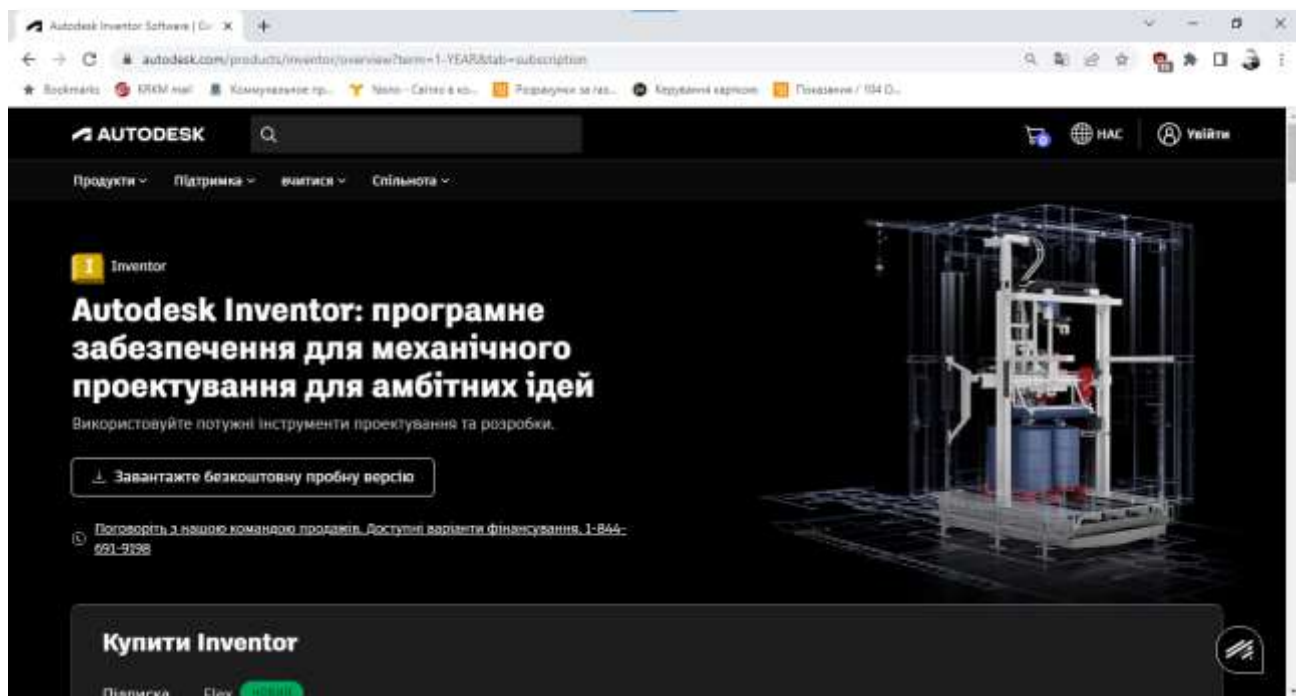


Рисунок 10.3 – Засіб придбання програмного продукту Inventor

Потужна система автоматизованого проектування, спеціально розроблена для машинобудування, промислового дизайну та інженерних рішень.

Засіб придбання САПР Fusion 360 наведено на рисунку 10.4.

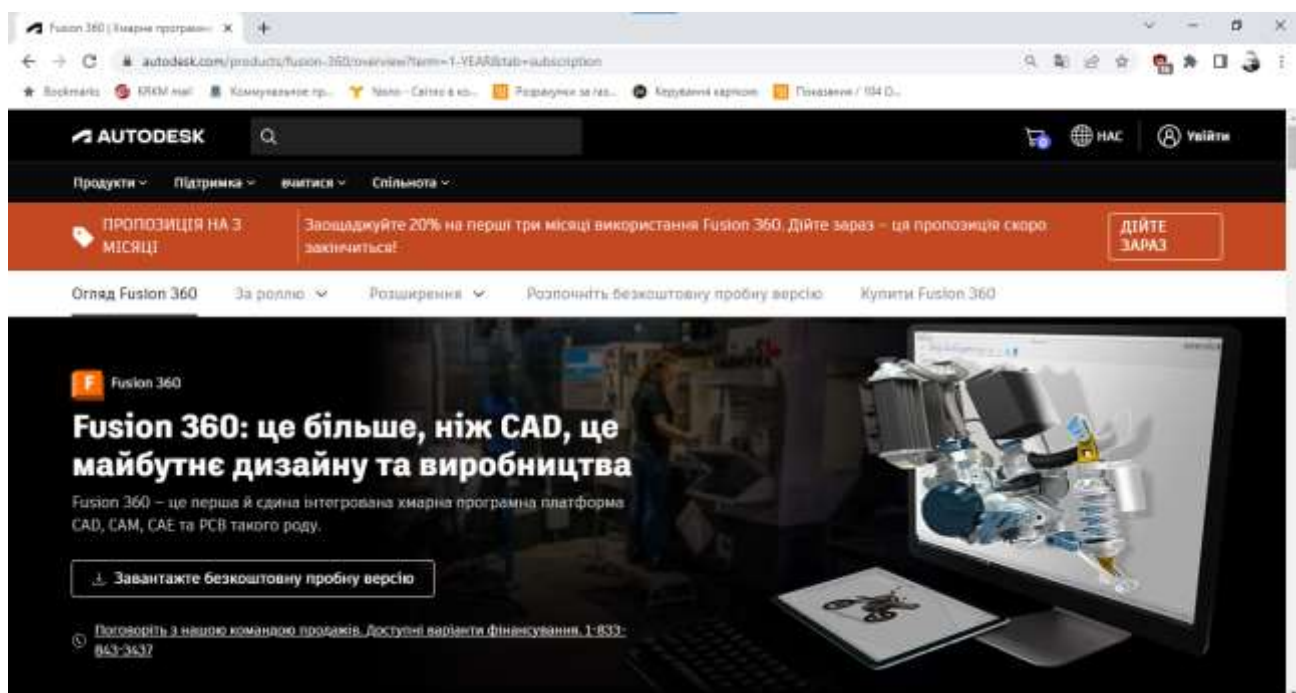


Рисунок 10.4 – Засіб придбання програмного продукту Fusion 360

Засіб придбання САПР Civil 3D наведено на рисунку 10.5.

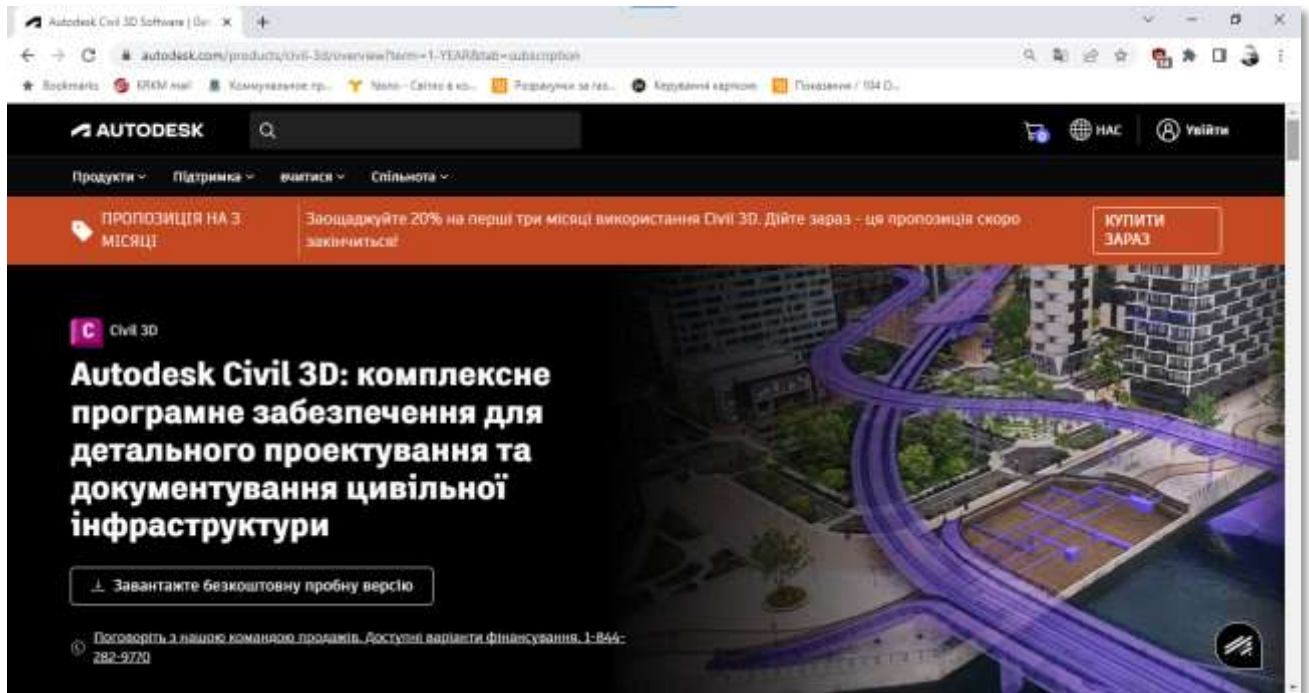


Рисунок 10.5 – Засіб придбання програмного продукту Civil 3D

Autodesk Civil 3D є програмним рішенням, спеціально розробленим для геодезії, проектування і документування інфраструктурних проектів, таких як дороги, міста, каналізація, водопостачання, ландшафтна архітектура та інші.

Засіб придбання САПР Revit наведено на рисунку 10.6.

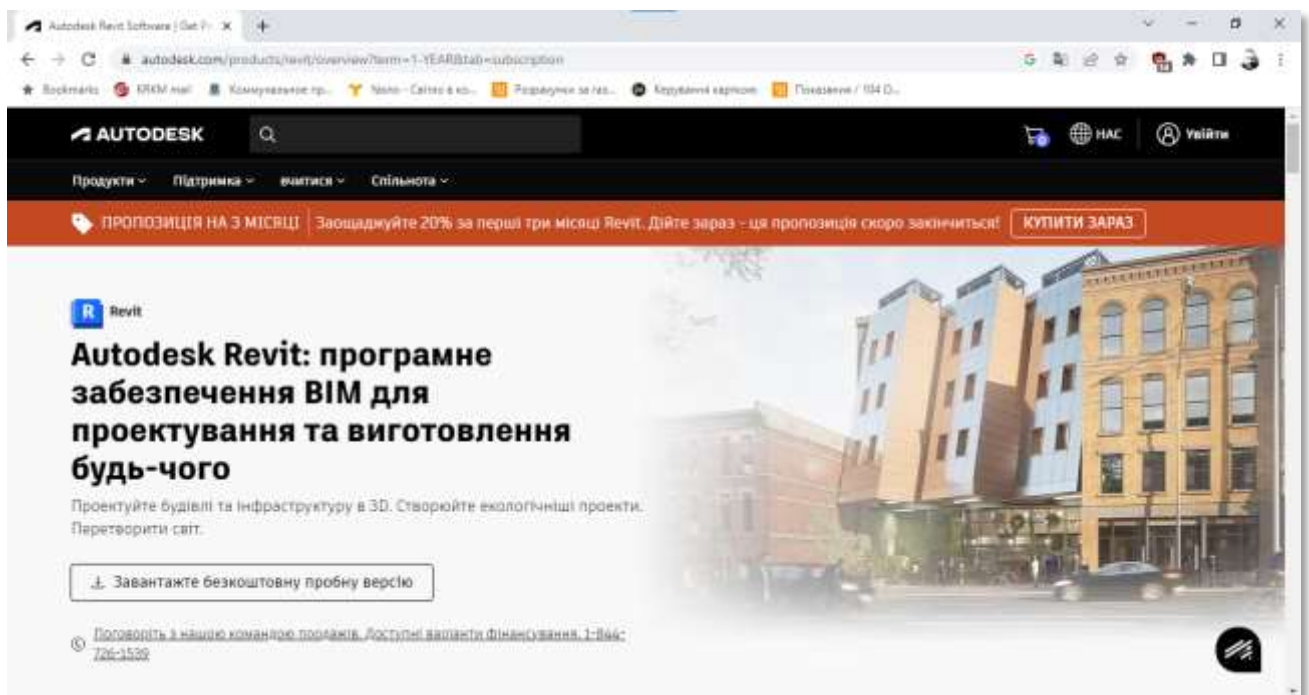


Рисунок 10.6 – Засіб придбання програмного продукту Revit

Програмне забезпечення для інтегрованого проектування будівель та споруд. Воно спеціалізується на інформаційному моделюванні будівлі (BIM) та дозволяє проектувати, моделювати та координувати роботу всіх учасників будівельного процесу.

Організація навчання Autodesk

Компанія Autodesk пропонує різноманітні можливості для навчання та підвищення кваліфікації на їхніх програмних продуктах. Навчання у компанії Autodesk організоване через декілька різних методів і ресурсів.

Autodesk University

Autodesk University (AU) — це щорічна конференція та навчальна подія, де професіонали з різних галузей збираються, щоб дізнатися про програмне забезпечення Autodesk, тенденції у галузі та передові практики. AU пропонує комбінацію занять у форматі лекцій, воркшопів та онлайн-уроків, які проводяться експертами галузі (рисунок 10.7).

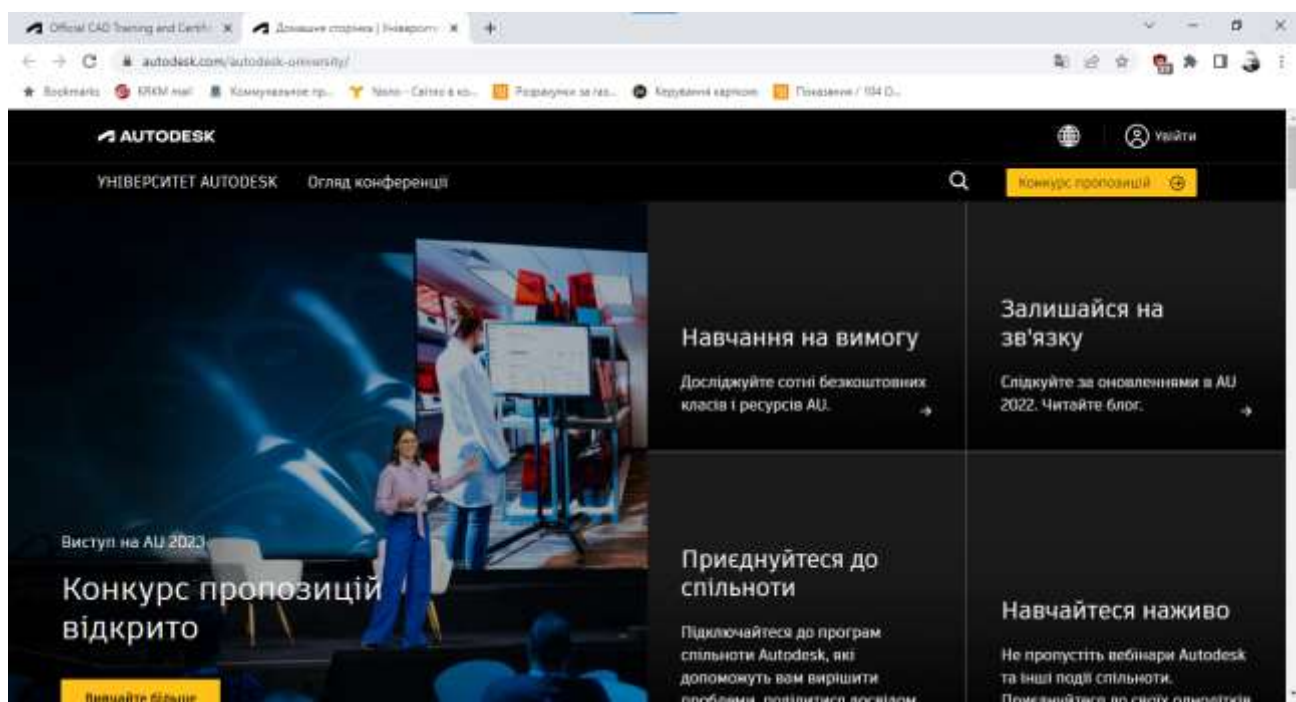


Рисунок 10.7 – Головна сторінка Autodesk University

Autodesk має мережу авторизованих центрів навчання під назвою "Партнери з навчання Autodesk" (ALP). ALP пропонують курси навчання з допомогою інструктора та програми сертифікації з програмних продуктів Autodesk. Ці курси проводять сертифіковані тренери та забезпечують можливість отримання практичних навичок.

Autodesk надає широкий спектр онлайн-курсів, документації та ресурсів навчання на їхньому офіційному веб-сайті. Ці ресурси включають відеоуроки, крок за кроком інструкції, форуми користувачів та статті бази знань. Користувачі можуть мати доступ до цих матеріалів для самостійного навчання та усунення проблем.

Autodesk співпрацює з видавцями, щоб створити офіційні навчальні книги та посібники з використання програмних продуктів. Ці книги охоплюють різні аспекти використання програм Autodesk (рисунок 10.8).

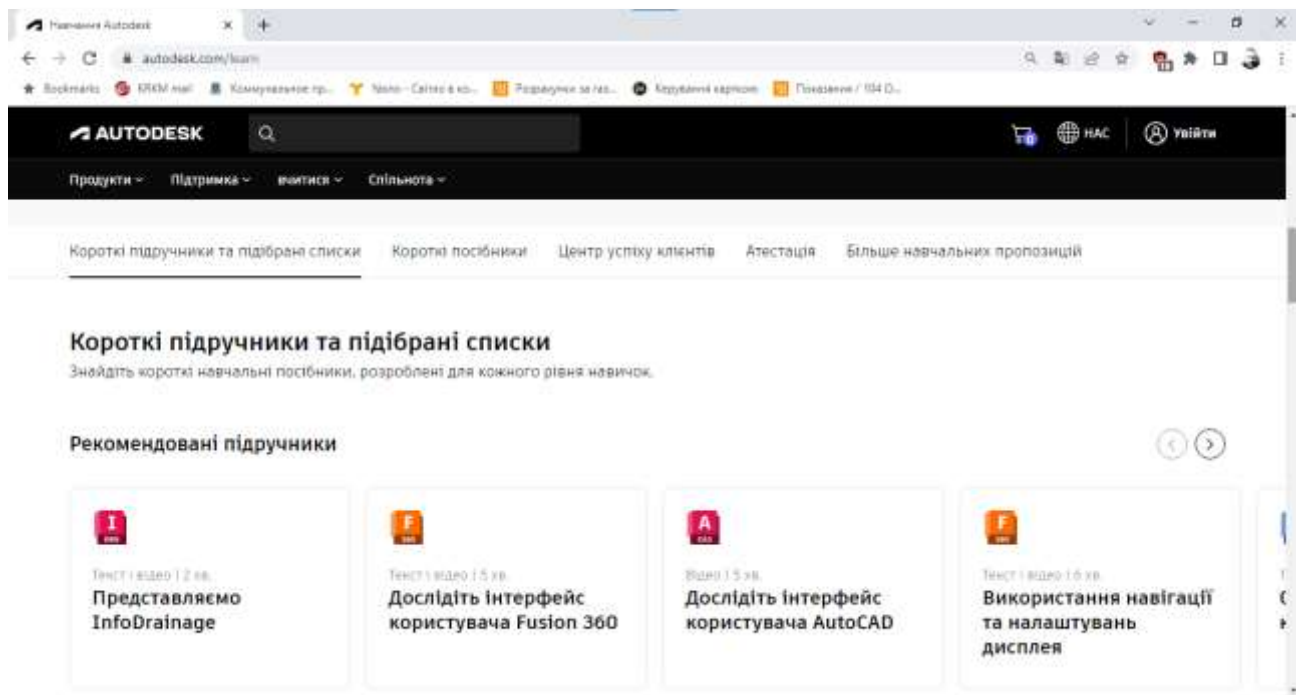


Рисунок 10.8 – Підручники САПР Autodesk

Autodesk надає широкий спектр онлайн-посібників, документації та навчальних ресурсів на своєму офіційному веб-сайті. Ці ресурси включають відеоуроки, крок за кроком інструкції, форуми користувачів та статті бази знань. Користувачі можуть мати доступ до цих матеріалів для самостійного навчання та усунення проблем.

Організація підтримки студентів компанією Autodesk

Компанія Autodesk надає підтримку студентам у навчанні та використанні їхніх програмних продуктів через наступні ініціативи:

Autodesk Education Community

Autodesk Education Community — це онлайн-ресурс, призначений спеціально для студентів, викладачів та навчальних закладів. Студенти можуть зареєструватися на цій платформі безкоштовно, щоб отримати доступ до навчальних матеріалів, підручників, відеоуроків та інших ресурсів Autodesk.

Безкоштовні ліцензії для студентів

Autodesk надає студентам можливість безкоштовно отримати ліцензії на використання їхніх програмних продуктів для навчальних цілей. Це дозволяє студентам отримати доступ до професійних інструментів та практикуватися у проектуванні та моделюванні.

Autodesk Design Academy

Autodesk Design Academy — це онлайн-платформа з навчальними ресурсами, яка допомагає студентам розвивати навички проектування і використання програм Autodesk. Вона містить курси, проекти, завдання та практичні вправи, які допомагають студентам вдосконалювати свої навички та знання.

Конкурси та стипендії

Autodesk організовує різноманітні конкурси та програми стипендій для студентів, які використовують їхні програмні продукти. Ці ініціативи дозволяють студентам продемонструвати свої навички та креативність, отримати визнання та нагороди за свої досягнення.

На рисунку 10.9 наведено відповіді на запитання стосовно організації безкоштовного використання продуктів компанії Autodesk.

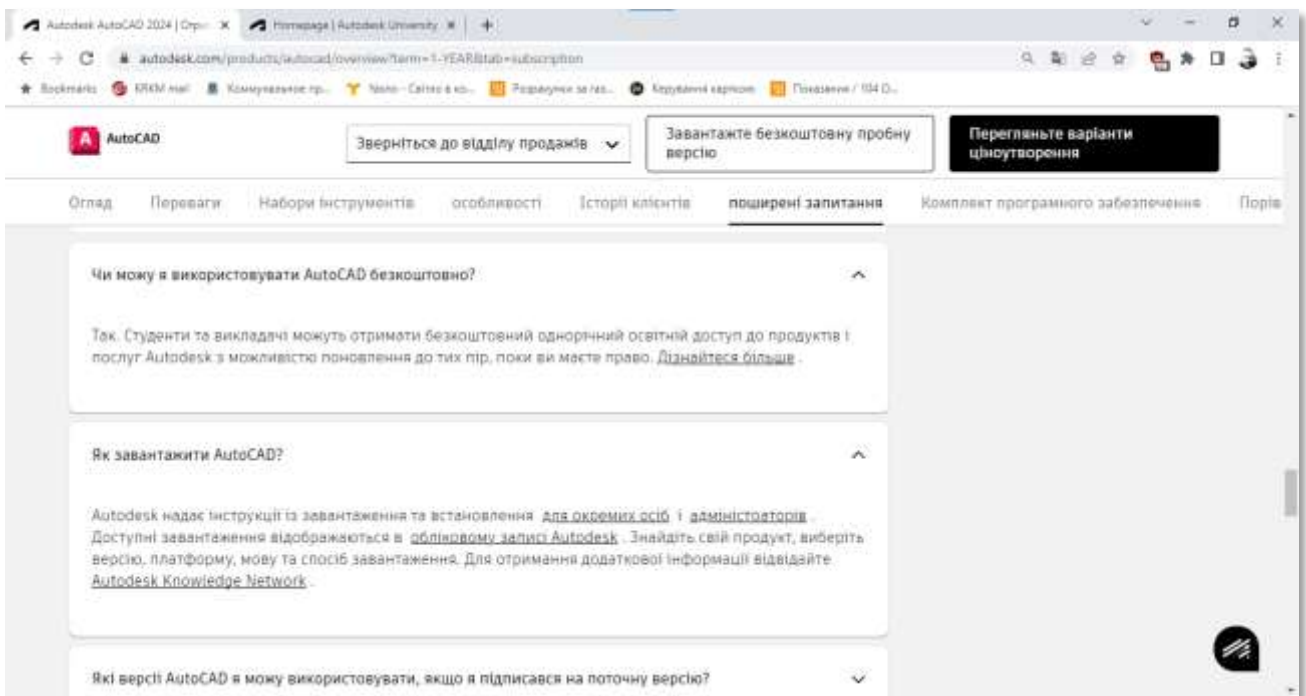


Рисунок 10.9 – Умови безкоштовного використання продуктів Autodesk

Крім того, Autodesk Education Community пропонує безкоштовний доступ до онлайн-курсів та підручників для студентів, які бажають вивчити основи використання програмного забезпечення Autodesk. Також, студенти можуть звернутися до підтримки користувачів Autodesk (Autodesk User Support), яка надає безкоштовну підтримку та консультації з використання програмного забезпечення компанії.

Ці ініціативи підтримки студентів допомагають забезпечити доступність та розвиток навичок використання програм Autodesk для молодих професіоналів у галузі дизайну, інженерії та архітектури.

Висновки

Отже, системи автоматичного проектування мають значний потенціал. Вони дозволяють здійснювати швидкий і точний проектувальний процес, сприяють підвищенню продуктивності і ефективності роботи. Застосування САПР у різних галузях, таких як машинобудування, архітектура, будівництво, виробництво та інші, дозволяє значно скоротити час проектування і витрати на розробку, а також забезпечує високу якість і точність проектів.

САПР забезпечують можливість використовувати 3D-моделювання, віртуальну реальність та симуляцію, що дозволяє проводити віртуальні тести та аналізувати різні сценарії перед фізичною реалізацією проектів. Це сприяє виявленню помилок, зниженню ризиків і оптимізації проектних рішень. Застосування САПР сприяє поліпшенню комунікації і співпраці між різними спеціалістами та відділами. Всі учасники проекту можуть працювати з однією централізованою базою даних, обмінюватися інформацією та вносити зміни в реальному часі, що підвищує ефективність колективної роботи. Використання САПР сприяє створенню інноваційних рішень та розвитку нових технологій. Вона дозволяє випереджати конкуренцію, швидше впроваджувати нові ідеї та забезпечувати постійне вдосконалення продуктів і процесів.

Питання та завдання до контролю знань студентів

- 1 Які основні переваги використання систем автоматизованого проектування в порівнянні з традиційними методами проектування?
- 2 Які галузі та сфери застосування САПР знаходяться в сучасному світі?
- 3 Які основні функціональні можливості мають сучасні системи автоматизованого проектування?
- 4 Які переваги надає використання 3D-моделювання в рамках САПР? Які можливості він відкриває для дизайнерів та інженерів?
- 5 Як САПР допомагає забезпечити співпрацю та комунікацію між різними учасниками проекту?
- 6 Які можливості надає віртуальна симуляція та віртуальне тестування в рамках САПР? Як це сприяє уникненню помилок та оптимізації проектних рішень?
- 7 Які основні методи навчання та підтримки надає компанія Autodesk для освоєння систем автоматизованого проектування?
- 8 Як САПР впливає на інноваційність та розвиток нових технологій у різних галузях?
- 9 Які виклики та обмеження можуть виникати при використанні систем автоматизованого проектування?
- 10 Які перспективи розвитку САПР в майбутньому та як вони можуть змінити спосіб, яким ми проектуємо та будуємо різні структури і вироби?

Лекція № 11

з навчальної дисципліни **Інформатика**

Тема Тривимірна графіка та фотореалістика

Мета заняття Оцінити концепцію створення багатополігональних тривимірних примітивів, оволодіти навичками по розробці тривимірних графічних об'єктів

Матеріально-технічне забезпечення та дидактичні засоби, ТЗН

1 Інтерактивний вид діяльності, створений засобами Moodle

2 Мультимедійний презентаційний матеріал

Час – 2 години

План проведення лекції

Структура лекції	Відведений час	Методичні вказівки
1 Організаційна частина	5 хв.	Включає в себе: <i>привітання</i> , яке має на меті привернути увагу, забезпечити контакт лектора з аудиторією; <i>визначення присутності студентів на занятті</i>
2 Актуалізація опорних знань, перевірка вивченого матеріалу та мотивація навчальної діяльності студентів	10 хв.	Включає в себе: <i>вступне зауваження, мотивацію актуальності теми, перевірку попереднього матеріалу, формулювання мети лекції, огляд головних питань теми (вказати питання та завдання для перевірки знань студентів)</i>
3 Основна частина (викладення навчальних питань лекції)	60 хв.	Головне місце приділяється <i>викладенню основного змісту матеріалу теми, його аналізу, узагальненню висунутих положень</i> . Для успіху лекції важливе значення має поділ матеріалу на розділи, основні питання
4 Заклучна частина Домашнє завдання: (відповідно до робочої програми)	5 хв.	Включає в себе: <i>узагальнення, теоретичні і фактичні висновки, закріплення вивченого на лекції матеріалу</i>

Література

1 Комп'ютерна графіка: навч. посіб. для студентів напряду підгот. / А.С. Василюк, Н.І. Мельникова. «Системний аналіз». — Львів: Вид-во Львів. політехніки, 2016. — 308 с. : іл.

2 Комп'ютерна графіка: навч. посібник / М. Ф. Пічугін, І. О. Канкін, В. В. Воротніков. Київ: Центр учбової літ., 2013. 346 с.

Навчальні матеріали лекції

ВСТУП

Сучасна технологія, яка відкриває перед нами захоплюючі можливості у візуалізації та сприйнятті інформації це тривимірна графіка. За допомогою тривимірної графіки ми можемо створювати віртуальні світи, моделі і сцени, які реалістично передають об'єкти і поведінку в реальному часі.

Вплив тривимірної графіки на сучасне життя неможливо переоцінити. Вона проникає в різні сфери нашого повсякденного життя, від розваг і реклами до освіти і промисловості. Ця технологія створює потужні засоби візуалізації та інтерактивності, які розширюють наші можливості та змінюють спосіб, яким ми сприймаємо світ навколо нас.

У сфері розваг тривимірна графіка дозволяє нам зануритися в захоплюючі віртуальні світи відеоігор, де реалістична графіка, динамічні ефекти та жива анімація створюють захоплюючий досвід для гравців. Такі технології, як віртуальна реальність та доповнена реальність, дозволяють нам взаємодіяти з цими віртуальними світами у більш іммерсивний спосіб.

У сфері дизайну та архітектури тривимірна графіка дає змогу створювати детальні 3D-моделі будівель, продуктів та інтер'єрів. Це допомагає архітекторам, дизайнерам та інженерам візуалізувати свої ідеї, а також проводити віртуальні тести та аналізи перед фізичною реалізацією проектів. Т

СФЕРИ ВИКОРИСТАННЯ ТРИВИМІРНОЇ ГРАФІКИ

Сфери використання тривимірної графіки є різноманітними і широкими. Поширення сфер застосування тривимірної графіки у різних галузях навіть не вимагає доведення доцільності по її використанню.

Кіноіндустрія

Тривимірна графіка використовується для створення спеціальних ефектів, візуальних ефектів, анімації персонажів та створення віртуальних світів у фільмах. Вона має велике значення в кіноіндустрії і впливає на всі аспекти створення фільмів — від створення вражаючих спеціальних ефектів до віртуального створення світів та персонажів.

Відеоігри

Тривимірна графіка є основою багатьох сучасних відеоігор. Вона дозволяє створювати реалістичні графічні образи, 3D-середовища та ефекти, що роблять геймплей більш іммерсивним.

Тривимірна графіка дозволяє створювати просторові середовища, які можуть бути досліджені гравцями. Це можуть бути великі відкриті світи, деталізовані міські ландшафти або фантастичні фантастичні місця. Гравці можуть взаємодіяти з цими середовищами, досліджуючи їх, виконуючи завдання та взаємодіючи з персонажами.

Архітектура та дизайн

Тривимірна графіка використовується для створення 3D-моделей будівель, інтер'єрів та ландшафтів. Це допомагає архітекторам та дизайнерам візуалізувати та зробити реалістичну передачу проєктів. Вона відіграє важливу роль у сучасній архітектурі і є незамінним інструментом для проєктування, візуалізації та аналізу будівельних проєктів. Дозволяє архітекторам та дизайнерам створювати детальні 3D-моделі будівель, інтер'єрів та ландшафтів, що допомагає їм краще розуміти просторові взаємозв'язки та впливати на всі аспекти проєкту.

Медицина

Тривимірна графіка використовується для створення точних моделей органів та систем тіла для медичних досліджень, діагностики та планування хірургічних втручань. Вона відіграє важливу роль у медицині і використовується для різних цілей, включаючи діагностику, планування лікування та навчання медичних процедур.

Реклама та маркетинг

Тривимірна графіка дозволяє створювати реалістичні рекламні матеріали, анімацію та візуалізацію продуктів, що допомагає залучити увагу споживачів та покращити ефективність рекламної кампанії.

Наука та дослідження

Тривимірна графіка (3D-графіка) має важливе застосування у науці та дослідженнях і використовується для візуалізації складних даних, моделювання та симуляцій. Вона допомагає дослідникам краще розуміти складні системи, виявляти закономірності і вирішувати складні наукові проблеми. Завдяки тривимірній графіці дослідники здатні отримувати нові інсайти та зрозуміти структуру та взаємозв'язки даних.

ПІДХОДИ ПО СТВОРЕННЮ ТРИВИМІРНОЇ ГРАФІКИ

Моделювання з використанням комп'ютерних програм

Моделювання з використанням комп'ютерних програм є одним з основних способів створення тривимірної графіки. Цей підхід використовує спеціалізовані програми, які надають художникам та дизайнерам інструменти для створення складних тривимірних об'єктів та сцен.

При створенні тривимірної графіки з використанням комп'ютерних програм використовуються різноманітні прийоми та техніки.

Моделювання

Цей прийом включає створення тривимірних об'єктів шляхом створення та маніпулювання їх геометричними формами. Фахівці можуть використовувати методи, такі як полігональне моделювання, субдівізіонна поверхня, моделювання з використанням кривих та інші для створення складних об'єктів.

Текстурування

Цей прийом дозволяє додати кольори, текстури та матеріали до тривимірних моделей для створення реалістичного вигляду. Розробники можуть застосовувати фотографічні текстури, процедурні генератори текстур, бамп-маппінг, нормалі картки та інші методи для досягнення бажаного візуального ефекту.

Анімація

Цей прийом включає створення руху та анімації тривимірних об'єктів. Спеціалісти можуть встановлювати ключові кадри, розробляти траєкторії руху, використовувати скелетну анімацію, симулювати фізичні ефекти та багато іншого.

Освітлення та рендеринг

Цей прийом дозволяє встановити освітлення на сцені та отримати готовий зображення тривимірної сцени. Користувачі можуть використовувати різні типи світлових джерел, установлювати їх параметри, використовувати тіні, відблиски, рефлексії та інші ефекти. Після цього сцена піддається процесу рендерингу, який перетворює тривимірну сцену.

Сканування реальних об'єктів

Один із способів створення тривимірної графіки полягає в скануванні реальних об'єктів. Цей процес дозволяє отримати точний цифровий відтір фізичного об'єкта з використанням спеціального обладнання. За допомогою спеціального обладнання, такого як 3D-сканери або фотограмметрія, можна отримати точний набір даних про форму та поверхневі деталі об'єкта, які потім можуть бути перетворені на тривимірну модель.

Сканування

Використовуючи сканер або 3D-сканер, об'єкт фізично сканується з усіх боків. Сканер вимірює форму, розміри та поверхневу структуру об'єкта, збираючи велику кількість точок даних.

Обробка даних

Після сканування отримані дані потрібно обробити, щоб створити повноцінну тривимірну модель. Цей процес включає видалення непотрібних даних, реконструкцію геометрії об'єкта та заміну недостатньо точних даних.

Ретопологія

Іноді отримані під час сканування дані можуть бути занадто деталізовані або незручні для подальшого використання. Тому може знадобитися процес ретопології, під час якого створюється нова геометрія з меншою кількістю полігонів, що полегшує редагування та анімацію моделі.

Текстурування та розфарбовування

Після отримання тривимірної моделі можна додати текстури та розфарбувати її. Це може включати використання фотографій, процедурного генерування текстур або ручного розфарбовування.

Використання анімаційних систем

Створення тривимірної графіки за допомогою анімаційних систем є популярним підходом, особливо в галузі кіно, відеоігор та анімації. Анімаційні системи надають художникам та аніматорам інструменти для створення руху, анімації персонажів та створення тривимірних сцен.

Багато анімаційних систем надають вбудовані інструменти для цих задач.

Анімаційні системи надають інтерфейс для керування об'єктами та персонажами за допомогою ключових кадрів, інтерполяції руху, фізичних симуляцій та інших методів. Аніматори можуть встановлювати позиції, рухи, вирази обличчя та інші атрибути для створення реалістичного руху та життєвості.

Анімаційні системи забезпечують контроль над рухом, освітленням, текстурами та іншими аспектами тривимірної сцени.

Художнє моделювання

Художнє моделювання є одним з підходів до створення тривимірної графіки, який поєднує мистецтво та технологію. В цьому підході художники використовують різні техніки та інструменти для створення детальних та естетично привабливих тривимірних моделей. Ось декілька способів, які використовуються у художньому моделюванні:

Ручне скульптурування

Цей метод базується на традиційному скульптурному процесі, але з використанням спеціалізованих програм та цифрових інструментів. Художники можуть створювати моделі шляхом формування цифрової глини або використовувати пенсиль, стилуси або графічні планшети для деталізації форми об'єкта.

Моделювання за допомогою полігонів

Цей підхід використовується для створення тривимірних моделей з використанням полігонів — малих геометричних фігур, що утворюють поверхню об'єкта. Художники створюють полігони, розташовуючи їх у просторі та формуючи бажану форму об'єкта.

Розтягування поверхні (Surface sculpting)

Цей метод дозволяє художникам створювати деталізовані тривимірні моделі, розтягуючи поверхню об'єкта із використанням спеціальних інструментів. Художник може витягувати, розтягувати, стискати або вигинати поверхню, надаючи об'єкту бажану форму і текстуру.

Розфарбовування та текстурювання

Розфарбовування та текстурювання є важливими етапами в художньому моделюванні тривимірної графіки. Ці процеси дозволяють надати моделі кольору, деталізації та реалістичності. Художник може використовувати програми для розфарбовування моделей, що надають широкі можливості для створення кольорових палітр, використання кистей, пензлів та інших інструментів для створення бажаного вигляду та ефектів.

ПРОГРАМНІ ЗАСОБИ ДЛЯ СТВОРЕННЯ ТРИВИМІРНОЇ ГРАФІКИ

Існує багато програмних засобів для створення тривимірної графіки. Деякі з них спеціалізовані на конкретних галузях, таких як кіно, відеоігри або архітектура, тоді як інші є універсальними інструментами. Ось кілька популярних програмних засобів для створення тривимірної графіки.

Autodesk Maya

Це одна з найпоширеніших програм для тривимірного моделювання, анімації та рендерингу. Вона має широкий набір інструментів для створення складних тривимірних сцен, персонажів та спеціальних ефектів.

Blender

Це безкоштовна та відкрита програма для тривимірного моделювання, анімації та рендерингу. Вона підтримує багато функцій, включаючи моделювання, текстурювання, скульптурування, анімацію, симуляцію та багато іншого.

3ds Max

Це програмне забезпечення, розроблене компанією Autodesk, яке спеціалізується на тривимірному моделюванні, анімації та рендерингу. Воно широко використовується в галузі кіно, відеоігор та візуалізації архітектурних проєктів.

Cinema 4D

Це програмне забезпечення з тривимірного моделювання, анімації та рендерингу, розроблене компанією Maxon. Воно використовується в різних галузях, включаючи рекламу, телебачення, кіно та відеоігри.

ZBrush

Це програмне забезпечення для цифрового скульптурування, яке дозволяє художникам створювати деталізовані тривимірні моделі. ZBrush надає потужні інструменти для цифрового скульптурування і створення деталізованих тривимірних моделей. Це корисно для художників, скульпторів та дизайнерів, які створюють художні моделі для мистецтва, створення фігурок, прототипів та багато іншого. Програмний додаток дозволяє художникам створювати деталізовані моделі з високою різкістю, що добре виглядають під час близьких планів.

Висновки

Отже, тривимірна графіка є надзвичайно важливим елементом сучасного візуального середовища. Вона знаходить широке застосування в різних галузях, включаючи кіноіндустрію, відеоігри, архітектуру, медицину, науку та багато інших.

Тривимірна графіка також знаходить широке використання в наукових дослідженнях, дозволяючи візуалізувати складні дані, моделювати та симулювати різні процеси та явища. Створення тривимірної графіки вимагає високої кваліфікації, технологічних знань та використання спеціалізованих програмних засобів.

За допомогою різних технік, таких як художнє моделювання, розфарбовування, текстуровування, анімація та рендеринг, створюються реалістичні та захоплюючі тривимірні сцени. Загалом, тривимірна графіка є потужним інструментом, що здатна забезпечити створення живих ігрових світів, вражаючих персонажів та реалістичної анімації.

Питання та завдання до контролю знань студентів

- 1 Які сфери використовують тривимірну графіку і як вона впливає на них?
- 2 Які програмні засоби використовуються для створення тривимірної графіки і що ви можете сказати про них?
- 3 Які основні техніки використовуються при створенні тривимірної графіки?
- 4 Як тривимірна графіка використовується в кіноіндустрії та відеоіграх? Які переваги вона надає у цих галузях?
- 5 Як тривимірна графіка впливає на архітектуру та як вона використовується для візуалізації архітектурних проектів?
- 6 Як тривимірна графіка допомагає у медицині, особливо в створенні тривимірних моделей органів та тканин?
- 7 Як тривимірна графіка використовується у науці та дослідженнях для візуалізації складних даних та моделювання процесів?
- 8 Які переваги і виклики пов'язані зі створенням тривимірної графіки через сканування реальних об'єктів?
- 9 Які можливості надає використання анімаційних систем для створення тривимірної графіки? В яких галузях вона застосовується найчастіше?
- 10 Які вимоги та навички необхідні для праці в галузі тривимірної графіки? Які можливості розвитку кар'єри існують у цій сфері?

Лекція № 12

з навчальної дисципліни Інформатика

Тема Аналіз, візуалізація та структури даних

Мета заняття Виробити навички по аналізу даних, дослідити можливості взаємодії різних типів даних, оволодіти уміннями по структуризації та візуалізації даних

Матеріально-технічне забезпечення та дидактичні засоби, ТЗН

1 Інтерактивний вид діяльності, створений засобами Moodle

2 Мультимедійний презентаційний матеріал

Час – 2 години

План проведення лекції

Структура лекції	Відведений час	Методичні вказівки
1 Організаційна частина	5 хв.	Включає в себе: <i>привітання</i> , яке має на меті привернути увагу, забезпечити контакт лектора з аудиторією; <i>визначення присутності студентів на занятті</i>
2 Актуалізація опорних знань, перевірка вивченого матеріалу та мотивація навчальної діяльності студентів	10 хв.	Включає в себе: <i>вступне зауваження, мотивацію актуальності теми, перевірку попереднього матеріалу, формулювання мети лекції, огляд головних питань теми (вказати питання та завдання для перевірки знань студентів)</i>
3 Основна частина (викладення навчальних питань лекції)	60 хв.	Головне місце приділяється <i>викладенню основного змісту матеріалу теми, його аналізу, узагальненню висунутих положень</i> . Для успіху лекції важливе значення має поділ матеріалу на розділи, основні питання
4 Заклучна частина Домашнє завдання: (відповідно до робочої програми)	5 хв.	Включає в себе: <i>узагальнення, теоретичні і фактичні висновки, закріплення вивченого на лекції матеріалу</i>

Література

1 Інформатика (профільний рівень) : підруч. для 10 кл. закл. заг. серед. освіти / В.Д. Руденко, Н.В. Речич, В.О. Потієнко. — Харків : Вид-во «Ранок», 2018.

2 Програмування числових методів мовою Python : підруч. / А. В. Анісімов, А. Ю. Дорошенко, С. Д. Погорілий, Я. Ю. Дорогий ; за ред. А. В. Анісімова. – К. : Видавничо-поліграфічний центр "Київський університет", 2014. – 640 с.

Навчальні матеріали лекції

ВСТУП

Історію комп'ютерних наук до певної міри можна подати як історію мов програмування, початок розвитку яких припадає на XIX ст., коли англійський учений Чарльз Беббідж розробив механічну обчислювальну машину. Програму для неї, як вам відомо, написала леді Ада Лавлейс. Мови програмування в сучасному розумінні фактично почали розвиватися з появою електронних обчислювальних машин.

Мови програмування є основою для створення комп'ютерних програм та веб-додатків. Завдяки мовам програмування розробники можуть створювати програми, які допомагають нам у повсякденному житті, від соціальних мереж та месенджерів до бізнес-додатків та комп'ютерних ігор. Кожна мова програмування має свої особливості, що дає розробникам можливість підібрати найкращу мову для конкретного проекту. Багато мов програмування постійно розвиваються та покращуються, що дозволяє розробникам створювати ще більш потужні та ефективні програми. У цьому контексті вивчення мов програмування є корисним навиком для тих, хто хоче розуміти, як працюють комп'ютерні програми, та для тих, хто бажає розробляти власні програмні продукти.

КЛАСИФІКАЦІЯ І СКЛАДОВІ МОВ ПРОГРАМУВАННЯ

Мова програмування (англ. Programming language) — формальний мовний засіб, який дозволяє розробникам писати інструкції. Це штучна мова, створена для розробки програм, які призначено для виконання на комп'ютері.

Комп'ютерна програма (англ. Computer Program) — це послідовність команд (інструкцій), що забезпечує реалізацію на комп'ютері конкретного алгоритму.

Команда (інструкція) — це вказівка, що визначає, яку дію (операцію) слід виконувати.

Сьогодні можна нарахувати понад 2 000 різних мов програмування та їх модифікацій, проте лише окремі набули широкого визнання.

Кожна з яких має свої властивості та використовується для різних цілей. Ось кілька найбільш популярних мов програмування:

- Java є однією з найпопулярніших мов програмування, яка використовується для створення додатків на платформі Android, розробки веб-сайтів та серверних додатків;
- Python є простою та ефективною мовою програмування, яка використовується для розробки веб-додатків, машинного навчання, аналізу даних та багатьох інших завдань;
- JavaScript є мовою програмування, яка використовується для створення динамічних веб-сайтів та веб-додатків;
- C# є мовою програмування, яка використовується для розробки додатків під управлінням операційної системи Windows, веб-додатків та ігор.

- C++ є мовою програмування, яка використовується для розробки операційних систем, драйверів, ігор та інших завдань, які вимагають високої продуктивності.
- PHP є мовою програмування, яка використовується для створення динамічних веб-сайтів та веб-додатків, яка працює на серверній стороні.
- Swift є мовою програмування, яка використовується для розробки додатків на платформі iOS та macOS.

Усі мови програмування можна умовно класифікувати за деякими основними ознаками, що наведено на рисунку 12.1.



Рисунок 12.1 – Класифікація мов програмування

Будь-яка мова програмування високого рівня, як і будь-яка інша мова, має основні складові: алфавіт, синтаксис, семантику. Мова програмування має словник — певну кількість слів, правила вживання яких визначено певною мовою і які мають строго визначене призначення. Такі слова називають зарезервованими (ключовими), наприклад, `for`, `input`, `if`, `print`. На рисунку 12.2 наведено складові мови програмування.

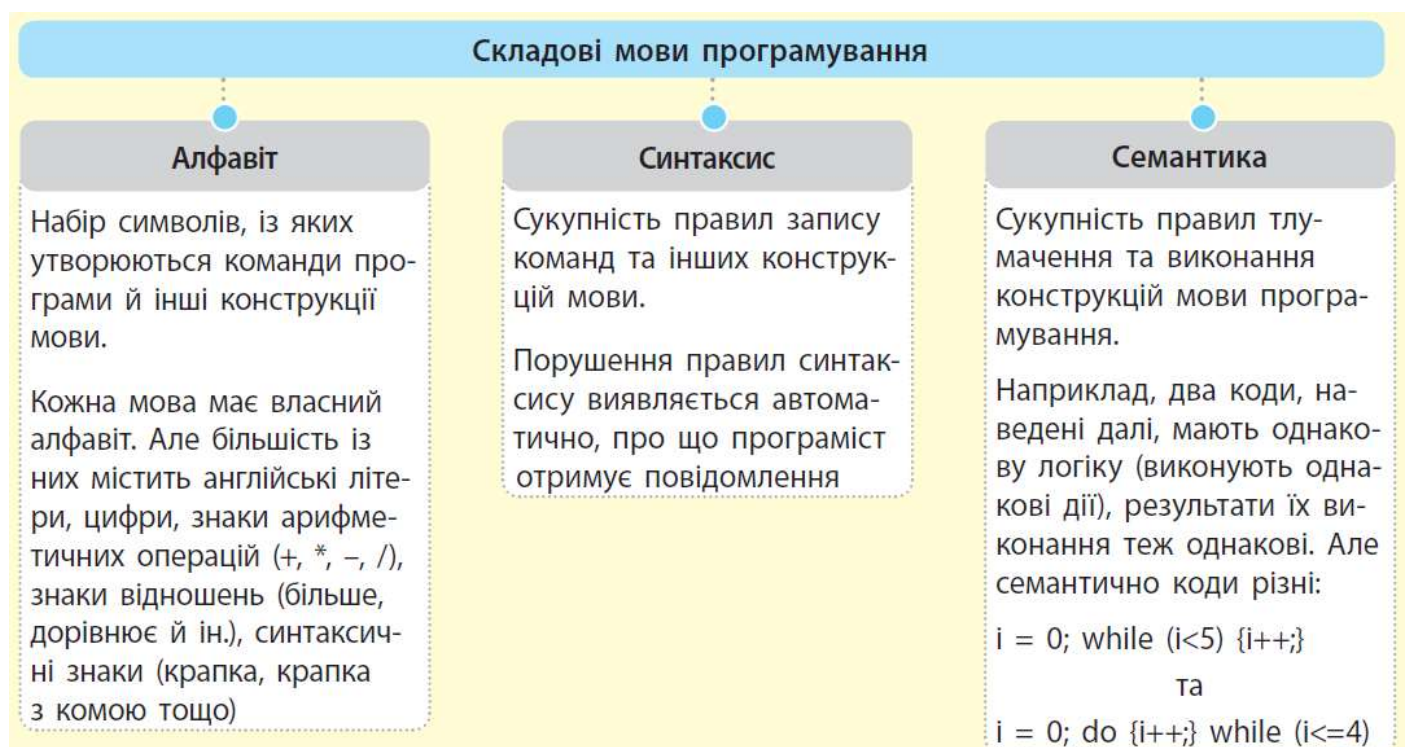


Рисунок 12.2 – Складові мови програмування

ПРИЗНАЧЕННЯ І СКЛАД СЕРЕДОВИЩА ПРОГРАМУВАННЯ

Для зручної розробки програм існують спеціальні засоби їх створення, — середовища (системи) програмування, які забезпечують весь цикл роботи з програмою — від її розроблення до виконання й отримання необхідних результатів.

Середовище програмування — це комплекс програмних засобів, які призначено для автоматизації процесу підготовки та виконання програм користувача.

На рисунку 12.3 наведені основні складові середовища програмування.



Рисунок 12.3 – Складові середовища програмування

Середовища програмування допомагають розробникам створювати програми більш ефективно та швидше. Вони надають різноманітні функції, такі як підказки коду, автоматичне заповнення, налагодження та інші, що полегшують роботу програмістів та збільшують їх продуктивність.

Середовища (системи) програмування часто іменуються за назвою мови, яка в них реалізується, наприклад середовище Pascal, середовище Delphi. Іноколи назва середовища містить префікс, який вказує на розробника середовища назва середовища Turbo-C означає, що її розробником є фірма Borland.

Нині все частіше використовуються інтегровані середовища програмування, які забезпечують роботу з кількома мовами.

Такими середовищами є, наприклад, IntelliJ IDEA, Eclipse. Варіант Ultimate Edition середовища IDEA забезпечує роботу з мовами програмування Java, PHP, Python.

Деякі з найбільш популярних середовищ програмування включають у себе:

- Visual Studio: це одне з найбільш популярних та повних середовищ програмування для розробки програм на різних мовах програмування, зокрема на C#, C++, Visual Basic та ін.

- Eclipse: це безкоштовне середовище програмування, яке підтримує різні мови програмування, зокрема Java, C++, Python та ін.

- IntelliJ IDEA: це інтегроване середовище розробки програмного забезпечення для мов програмування Java, Kotlin та ін.

- PyCharm: це середовище програмування для мови програмування Python.

- Sublime Text: це легкий текстовий редактор з підсвічуванням синтаксису та розширеними можливостями для розробки програм.

ОСНОВНІ ЕЛЕМЕНТИ МОВИ ПРОГРАМУВАННЯ PYTHON

Загальна характеристика та сфери використання мови Python

Комплекс програмних засобів, які призначені для автоматизації процесу підготовки та виконання алгоритмів розробника прийнято називати середовищем розробки.



Python — інтерпретована об'єктно-орієнтована мова програмування високого рівня зі строгою динамічною типізацією.

Інтерпретатор Python та стандартні модулі доступні як у скомпільованій, так і у вихідній формі на всіх основних платформах. В мові програмування Python підтримується кілька парадигм програмування, зокрема:

- об'єктно-орієнтована;
- функціональна;
- аспектно-орієнтована.

Python портована і працює майже на всіх відомих платформах — від КПК до мейнфреймів. Існують порти під Microsoft Windows, всі варіанти UNIX (включаючи FreeBSD та GNU/Linux), Plan 9, Mac OS та Mac OS X, iPhone OS 2.0 і вище, Palm OS, OS/2, Amiga, AS/400 та навіть OS/390, Symbian та Android.

Python — стабільна та поширена мова. Вона використовується в багатьох проектах та в різних якостях: як основна мова програмування або для створення розширень та інтеграції додатків.

На Python реалізована велика кількість проектів, вона може бути використана для розв'язання складних математичних задач, створення графічних зображень, розробки веб-сайтів, взаємодії із реляційними базами даних.

Програма мовою Python — це звичайний текстовий файл, інструкції (команди) якого виконуються інтерпретатором для кожного рядка. Програми можуть розроблятися в консольному режимі (такі програми мають розширення .py) і з графічним інтерфейсом (програми мають розширення .pyw). Під час першого запуску програми створюється байт-код, який зберігається у окремий файл. Якщо після цього програма не змінювалася, то в процесі наступних її запусків буде виконуватися байт-код.

Основні елементи мови програмування Python

Первино символи кодуються в системі UTF-8.



Мінімальна одиниця мови програмування, що має певне самостійне значення та яку розуміє транслятор називається лексема.

Розрізняють такі види лексем:

- зарезервовані — ключові слова (keywords);
- ідентифікатори (identifiers);
- літерали (константи);
- операції;
- знаки пунктуації.

У Python використовується кілька десятків зарезервованих слів (наприклад, `int`, `list`, `input`, `print`, `float` тощо).



Ідентифікатори використовуються для позначення різноманітних об'єктів (змінних, функцій, тощо).

Ідентифікатор змінної може складатися з латинських літер, цифр і знака підкреслення. Першим символом імені не може бути цифра або знак підкреслення.



Мова програмування Python — є регістрозалежною до використаних ідентифікаторів.

Однакові імена з літерами на різних регістрах сприймаються як різні імена. Наприклад, ідентифікатори `student` та `Student` — сприймаються, як різні імена змінних.

В свою чергу, імена змінних використовуються для доступу до даних. Дані в мові Python подано у формі об'єктів.



Об'єкт — це ділянка пам'яті з певним значенням та можливими операціями його опрацювання.

Кожний об'єкт мови Python має свій тип. Розрізняють базові об'єктні типи, що є вбудованими в мову Python та ті, що розробляються власноруч засобами самої мови або сторонніми засобами.



Змінні зберігають не сам об'єкт, а посилання на об'єкт — адресу об'єкта в пам'яті.

Усі типи даних мови Python є класами. Найчастіше вживані базові типи об'єктів, а також приклади їх літералів наведено в таблиці А.1 додатку А.

Тип змінних в мові Python визначається автоматично в процесі присвоювання.

Тип змінної визначається значенням, розташованим праворуч від оператора присвоювання, а сама змінна в процесі виконання коду може посилатися на об'єкти з різними типами даних. В такий спосіб можна прийти до висновку, що змінна містить виключно посилання на об'єкт.



Python підтримує динамічну типізацію змінних, де тип даних — це характеристика об'єкта, а не змінної.

Для визначення останнього типу даних, на який посилається змінна, використовується функція `type` (ім'я змінної). Набір операцій, які можуть виконуватися над типами даних строго визначені для кожного вбудованого базового типу мови Python.

Усі класи містять методи.



Метод — це частина класу (об'єкта), що викликається з використанням точкової нотації, яка не може бути використана окремо від нього.

За своїм призначенням методи і функції Python — схожі, проте, на відміну від методів функції являють собою самостійні сутності.



Функція — це набір інструкцій, об'єднаних для виконання одного завдання.

ОГЛЯД СТАНДАРТНИХ МОДУЛІВ PYTHON

Основні функції перетворення одного типу даних на інший

Функція `bool(об'єкт)` використовується для перетворення об'єкта на логічний тип.

Функція `int(об'єкт, система числення)` використовується для перетворення об'єкта у ціле число. Система числення, у якій подається об'єкт, може бути десятковою, вісімковою, шістнадцятковою. За замовчуванням — десяткова система.

Функція `float(ціле число або рядок)` використовується для перетворення цілого числа або рядка на число дійсного типу.

Функція `str(об'єкт)` використовується для перетворення об'єкта на рядок.

Функція `list(послідовність)` використовується для перетворення елементів послідовності на список.

Базові функції роботи із об'єктами

Числа можуть подаватися в десятковій, двійковій, вісімковій і шістнадцятковій системах числення. У процесі виконання арифметичних операцій над числами в різних системах числення вони автоматично перетворюються на десяткову систему числення.

В таблиці 12.1 наведено перелік вбудованих у мову програмування базових типів об'єктів.

Таблиця 12.1 – Вбудовані базові типи об’єктів

Тип об’єкта	Клас	Приклад літерала
Цілі числа	int	3, 27, 145 -15, -4, 0
Дійсні числа	float	-0.75, -0.1, -0.01 .5, .65, 9.4, 318.043
Рядки	str	'Rocket-and-Space Engineering College' "ФКРКМ ДНУ ім. О. Гончара"
Логічні дані	bool	True False
Списки	list	[-2, 1.03, "Дніпро", 7, True, 'college'] [[1, 2, 3, 4, 5], ['зараховано', 'не зараховано'], [0, 1]]
Словники	dict	{121: 'Інженерія програмного забезпечення', 123: 'Комп’ютерна інженерія', 071: 'Облік і оподаткування', 133: 'Галузеве машинобудування'}
Кортежі	tuple	(-2.5, 'FKRKM', 27, "Україна") ([True, False], {1: "січень", 2: "лютий"}, (0, 120, 255))
Множини	set	set('abc') {'a', 'b', 'c'}

Функції модуля math

Модуль math містить стандартні константи та функції, що використовуються для роботи з числовими типами даних. Для залучання модуля math необхідно імпортувати його за допомогою інструкції `import math`. Модуль math містить константи чисел π та e .

В таблиці 12.2 наведено перелік основних функцій стандартного модуля math.

Таблиця А.3 – Найуживаніші функції модуля math

Функція	Призначення
<code>sqrt(аргумент)</code>	повертає значення квадратного кореня за вказаним аргументом
<code>sin(аргумент)</code> <code>cos(аргумент)</code> <code>tan(аргумент)</code>	за вказаним аргументом повертає значення стандартних тригонометричних функцій
<code>asin(аргумент)</code> <code>acos(аргумент)</code> <code>atan(аргумент)</code>	за вказаним аргументом повертає значення обернених тригонометричних функцій
<code>degrees(аргумент)</code>	за вказаним аргументом перетворює значення радіанів на градуси
<code>radians(аргумент)</code>	за вказаним аргументом перетворює значення градусів на радіани

Продовження таблиці 12.2

Функція	Призначення
<code>exp(аргумент)</code>	повертає значення експонента за заданим аргументом
<code>log(аргумент, основа логарифма)</code>	повертає значення логарифма за вказаними значеннями аргументу та основи
<code>log(аргумент)</code>	повертає значення натурального логарифма за вказаним аргументом
<code>log10(аргумент)</code>	повертає значення десяткового логарифма за вказаним аргументом
<code>factorial(аргумент)</code>	повертає значення факторіалу за вказаним аргументом
<code>ceil(аргумент)</code>	повертає округлене значення аргументу до найближчого цілого значення типу <code>int</code> , яке дорівнює або перевищує вказаний аргумент
<code>floor(аргумент)</code>	повертає округлене значення аргументу, як найближче ціле значення типу <code>int</code> , менше або рівне вказаного
<code>trunc(аргумент)</code>	повертає цілу частину числа вказаного аргументу у вигляді об'єкта типу <code>int</code>
<code>hypot(аргумент_1, аргумент_2)</code>	за послідовно вказаними аргументами повертає результат обчислення гіпотенузи чи довжини вектора

Функції модуля `random`

Модуль `random` містить функції, які використовуються для генерації випадкових чисел. Для застосування функцій модуля необхідно попередньо впровадити його за допомогою інструкції `import random`.

В таблиці 12.3 наведено перелік найуживаніших функцій модуля `random`.

Таблиця 12.3 – Найуживаніші функції модуля `random`

Функція	Призначення
<code>random()</code>	генерує випадкове число від 0.0 до 1.0 [0.0, 1.0) 1.0 не включено до діапазону
<code>uniform(початок, кінець)</code>	генерує дійсне випадкове число в діапазоні від «початок» до «кінець». 1.0 включено до діапазону
<code>randint(початок,кінець)</code>	генерує ціле випадкове число в діапазоні від «початок» до «кінець». 1.0 не включено до діапазону
<code>choice(послідовність)</code>	вибирає з послідовності (рядка, списку або кортежу) випадковий елемент

Висновки

Отже, аналіз, візуалізація та структури даних відіграють ключову роль у сучасному інформаційному світі. Ці процеси дозволяють розуміти, інтерпретувати і використовувати великі обсяги даних для прийняття розумних рішень і забезпечення ефективної комунікації. Аналіз даних допомагає виділити корисну інформацію з великих наборів даних, розкриває залежності, тренди тощо. Він дозволяє зрозуміти сутність даних, виявити причинно-наслідкові зв'язки і зробити достовірні прогнози. Аналіз даних забезпечує підтримку прийняття рішень на основі об'єктивних фактів і допомагає виявити нові можливості та вдосконалення.

Візуалізація даних сприяє зручному сприйняттю інформації, спілкуванню та передачі знань. Це допомагає зрозуміти дані швидше і ефективніше, виявляти тенденції, порівнювати дані і виявляти відхилення.

Використання відповідних структур даних дозволяє оптимізувати доступ до даних, забезпечувати цілісність і ефективність.

Мова програмування Python відіграє важливу роль у аналізі та структурі даних завдяки своїм особливостям та багатому екосистемі. Простий синтаксис, робить її доступною для початківців та дозволяє швидко приступити до аналізу і обробки даних. Широкий вибір бібліотек для аналізу та обробки даних, таких як NumPy, Pandas, Matplotlib, SciPy і багато інших надають потужні функції для роботи з даними, візуалізації, моделювання та машинного навчання. Python легко інтегрується з іншими мовами програмування та інструментами, що дозволяє поєднувати його з різними інструментаріями для аналізу даних, баз даних тощо.

Питання та завдання до контролю знань студентів

- 1 Як основні бібліотеки Python використовуються для аналізу даних?
- 2 Як Python допомагає у візуалізації даних і побудові графіків?
- 3 Як основні методи та функції Python використовуються для обробки та маніпуляції даними?
- 4 Як Python допомагає у роботі з великими обсягами даних (Big Data)?
- 5 Як інструменти Python можна використовувати для машинного навчання та статистичного аналізу даних?
- 6 Як Python інтегрується з базами даних і які інструменти доступні для взаємодії з реляційними та нереляційними системами?
- 7 Як Python допомагає у виконанні дослідницьких аналізів та створенні висновків на основі даних?
- 8 Як Python використовується для автоматизації рутинних задач у процесі аналізу даних?
- 9 Як інструменти Python можна використовувати для роботи з географічними даними та геопросторовим аналізом?
- 10 Як Python допомагає у побудові прогностичних моделей і передбаченні майбутніх тенденцій на основі наявних даних?

Лекція № 13

з навчальної дисципліни Інформатика

Тема Реалізація базових алгоритмічних конструкцій

Мета заняття Виробити вміння стосовно залучання базових алгоритмічних конструкцій, розвинути уміння розробки розгалужених алгоритмів, оволодіти навичками по створенню циклічних алгоритмів

Матеріально-технічне забезпечення та дидактичні засоби, ТЗН

1 Інтерактивний вид діяльності, створений засобами Moodle

2 Мультимедійний презентаційний матеріал

Час – 2 години

План проведення лекції

Структура лекції	Відведений час	Методичні вказівки
1 Організаційна частина	5 хв.	Включає в себе: <i>привітання</i> , яке має на меті привернути увагу, забезпечити контакт лектора з аудиторією; <i>визначення присутності студентів на занятті</i>
2 Актуалізація опорних знань, перевірка вивченого матеріалу та мотивація навчальної діяльності студентів	10 хв.	Включає в себе: <i>вступне зауваження, мотивацію актуальності теми, перевірку попереднього матеріалу, формулювання мети лекції, огляд головних питань теми (вказати питання та завдання для перевірки знань студентів)</i>
3 Основна частина (викладення навчальних питань лекції)	60 хв.	Головне місце приділяється <i>викладенню основного змісту матеріалу теми, його аналізу, узагальненню висунутих положень</i> . Для успіху лекції важливе значення має поділ матеріалу на розділи, основні питання
4 Заключна частина Домашнє завдання: (відповідно до робочої програми)	5 хв.	Включає в себе: <i>узагальнення, теоретичні і фактичні висновки, закріплення вивченого на лекції матеріалу</i>

Література

1 Інформатика (профільний рівень) : підруч. для 10 кл. закл. заг. серед. освіти / В.Д. Руденко, Н.В. Речич, В.О. Потієнко. — Харків : Вид-во «Ранок», 2018.

2 Програмування числових методів мовою Python : підруч. / А. В. Анісімов, А. Ю. Дорошенко, С. Д. Погорілий, Я. Ю. Дорогий ; за ред. А. В. Анісімова. – К. : Видавничо-поліграфічний центр "Київський університет", 2014. – 640 с.

Навчальні матеріали лекції

ВСТУП

Алгоритм програмування — це точний набір інструкцій, які необхідно виконати, щоб вирішити певну задачу або досягти певної мети. Алгоритми програмування можуть бути написані на будь-якій мові програмування і зазвичай передбачаються для виконання комп'ютером.

Алгоритми програмування є ключовим елементом розробки програмного забезпечення, оскільки вони дозволяють розробникам зосередитися на розв'язанні конкретної задачі, не втрачаючи час на розгляд всіх можливих варіантів виконання. Вони допомагають зменшити складність розв'язання проблеми та покращити продуктивність.

Алгоритми програмування можуть бути розроблені для вирішення широкого спектру завдань, включаючи сортування даних, пошук елементів, розрахунок математичних функцій, обробку даних та інше. Ці алгоритми можуть бути реалізовані у вигляді функцій, підпрограм, модулів, класів та інших елементів програми.

Важливо мати на увазі, що алгоритми програмування не завжди є оптимальним варіантом для вирішення певних задач. Розробники повинні ретельно розглядати всі можливі варіанти розв'язання завдання та обирати найкращий з них, з урахуванням того, що він повинен бути не тільки ефективним, а й легко зрозумілим та зручним у використанні.

ОСНОВНІ ТИПИ АЛГОРИТМІВ ПРОГРАМУВАННЯ

Алгоритми програмування можуть бути різних типів залежно від їхнього призначення та характеристик. Основні типи алгоритмів програмування включають:

Лінійні алгоритми — алгоритми, в яких кожна інструкція виконується послідовно, одна за одною, без перерви. Такий алгоритм можна представити у вигляді послідовності кроків, де кожен наступний крок виконується після завершення попереднього.

Розгалужені алгоритми — алгоритми, що містять різні шляхи виконання, в залежності від заданих умов. Такі алгоритми зазвичай використовуються для розв'язання задач, в яких може бути кілька варіантів розв'язку.

Циклічні алгоритми — алгоритми, що виконують один або декілька кроків циклічно, доки не будуть виконані певні умови. Такі алгоритми зазвичай використовуються для розв'язання задач, пов'язаних з обробкою багатьох елементів даних.

Рекурсивні алгоритми — алгоритми, які викликають самі себе для розв'язання певної задачі. Вони зазвичай використовуються для розв'язання задач, які можуть бути розбиті на менші підзадачі.

Паралельні алгоритми — алгоритми, що виконуються на кількох процесорах або комп'ютерах одночасно з метою прискорення роботи та збільшення продуктивності.

Евристичні алгоритми — алгоритми, що базуються на використанні евристичних методів для розв'язання задач, які не можуть бути розв'язані за допомогою стандартних алгоритмів.

ПРИКЛАДИ РЕАЛІЗАЦІЇ ЛІНІЙНИХ АЛГОРИТМІВ МОВОЮ PYTHON

У мові програмування Python лінійні алгоритми можуть бути реалізовані за допомогою послідовності інструкцій. У лістингу 13.1 наведено приклад лінійного алгоритму для обчислення середнього арифметичного двох чисел.

Лістинг 13.1 – Обчислення середнього арифметичного двох чисел

```
# Введення двох чисел
num1 = float(input("Введіть перше число: "))
num2 = float(input("Введіть друге число: "))

# Обчислення середнього арифметичного
avg = (num1 + num2) / 2

# Виведення результату
print("Середнє арифметичне чисел", num1, "та", num2, "дорівнює", avg)
```

У цьому прикладі послідовність інструкцій починається з введення двох чисел, після чого обчислюється їхнє середнє арифметичне та виводиться результат.

Для реалізації лінійного алгоритму, який полягає у заміні значень між двома змінними мовою програмування Python можна використати допоміжну змінну, або використати множинне присвоєння. У лістингу 13.2 наведено приклад лінійного алгоритму заміни значень між двома змінними через впровадження допоміжної змінної.

Лістинг 13.2 – Впровадження допоміжної змінної для заміни значень між двома змінними

```
# Послідовне присвоєння значень змінним
a = 5
b = 10

temp = a
a = b
b = temp

# Виведення результату
print("a =", a)
print("b =", b)
```

Реалізація алгоритму заміни значень між двома змінними за допомогою множинного присвоєння наведено у лістингу 13.3.

Лістинг 13.3 – Заміна значень між двома змінними за допомогою множинного присвоєння

```
# Послідовне присвоєння значень змінним
a = 5
b = 10

# Множинне присвоєння значень змінним
a, b = b, a

# Виведення результату
print("a =", a)
print("b =", b)
```

Реалізація лінійного алгоритму для переведення температури з градусів Фаренгейта у градуси Цельсія наведено у лістингу 13.4.

Лістинг 13.4 – Переведення температури з градусів Фаренгейта у градуси Цельсія

```
# Зчитуємо температуру в градусах Фаренгейта з клавіатури
fahrenheit = float(input("Введіть температуру в градусах Фаренгейта: "))

# Переводимо температуру в градуси Цельсія
celsius = (fahrenheit - 32) * 5 / 9

# Виводимо результат на екран
print(fahrenheit, "градусів Фаренгейта = ", celsius, "градусів Цельсія")
```

Спочатку програма зчитує введені користувачем значення температури у градусах Фаренгейта за допомогою функції `input()`. Потім виконується формула для переведення градусів Фаренгейта у градуси Цельсія, результат зберігається у змінній `celsius`. Нарешті, результат виводиться на екран за допомогою функції `print()`.

У лістингу 13.5 наведено реалізацію лінійного алгоритму обчислення складних відсотків.

Лістинг 13.5 – Обчислення складних відсотків

```
# Вхідні дані
p = float(input("Введіть річну процентну ставку: "))
n = int(input("Введіть кількість років: "))
k = float(input("Введіть початковий вклад: "))

# Розрахунки
m = 12 # кількість місяців у році
r = p / 100 # річна процентна ставка у десяткових дробах
f = k * (1 + r/m) ** (m*n) # кінцева сума на рахунку

# Вивід результату
print("Через", n, "років на рахунку буде:", round(f, 2), "грн.")
```

Спочатку програма зчитує введені користувачем дані: річну процентну ставку p , кількість років n та початковий вклад k . Далі виконуються розрахунки за формулою для складних відсотків. Нарешті, результат виводиться на екран за допомогою функції `print()`. У наведеному прикладі здійснюється округлення результату до 2 знаків після коми за допомогою функції `round()`.

РЕАЛІЗАЦІЯ АЛГОРИТМІВ ІЗ РОЗГАЛУЖЕННЯМ У PYTHON

У більшості програм на певному етапі їх виконання виникає потреба вибору того чи іншого варіанту подальшого розвитку подій. Умовою вибору того чи іншого варіанту є виконання деякої умови. Сама умова повинна бути булевого типу, тобто приймати лише два значення `true` або `false`.

Одноальтернативне розгалуження

Одноальтернативне розгалуження виконується так. Перевіряється умова, і якщо вона істинна (`True`), то виконується оператор розгалуження. Інакше буде виконуватися оператор, який розташовано безпосередньо за оператором розгалуження. На рисунку 13.1 наведено графічну схему одноальтернативного розгалуження.

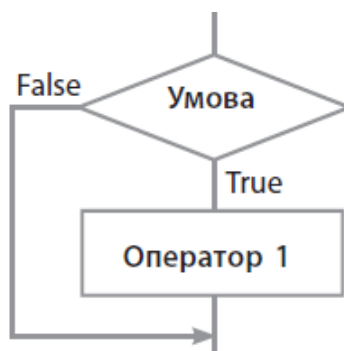


Рисунок 13.1 – Графічна схема одноальтернативного розгалуження

У мові Python цей тип розгалуження реалізується оператором умовного переходу в неповній формі:

```
код основної гілки
if умова:
    інструкція 1
    інструкція 2
    ...
код основної гілки
```

Приклад реалізації такої конструкції мовою програмування Python наведена у лістингу 13.6.

Лістинг 13.6 – Реалізації конструкції «if»

```
a = 10
if a > 0:
    print("a – додатна")
```

Двоальтернативне розгалуження

Двоальтернативне розгалуження виконується так: перевіряється умова і, якщо вона має значення True, то виконується Оператор 1. Інакше виконується Оператор 2. Графічну схему двоальтернативного розгалуження зображено на рисунку 13.2.

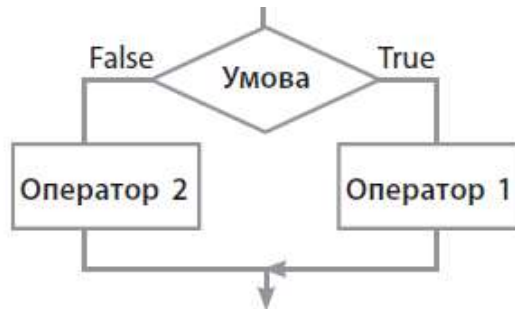


Рисунок 13.2 – Графічна схема двоальтернативного розгалуження

У мові Python цей тип розгалуження реалізується оператором такої структури:

код основної гілки

if умова:

інструкція 1

інструкція 2

...

код основної гілки

Якщо необхідно передбачити альтернативний варіант виконання програми, тобто при умові — True, потрібно виконати один набір інструкцій, а при False — інший, тоді використовується конструкція «if-else»

Приклад реалізації конструкції «if-else» наведено у лістингу 13.7.

Лістинг 13.7 – Реалізації конструкції «if-else»

```
print("Ви бажаєте вивчати мову Python?\n 1. так\n2. ні")

answer = int(input("Оберіть варіант відповіді: "))

if answer == 1:
    print("Ви не пошкодуєте!")

else:
    print("Шкода")

print("Отже, Ви зробили свій вибір...")
```

Багатоальтернативне розгалуження

У цьому типі розгалуження реалізується розгалуження з багатьма варіантами вибору. Кожне значення виразу має бути унікальним константним виразом. Значення виразу порівнюється з кожним значенням в умовних операторах у порядку їх розташування. Якщо певне значення виразу збігається зі значенням оператора, то виконується той блок операторів, що розташований безпосередньо за цим оператором. Якщо значення вираз не збігається з жодним значенням в операторах, виконується останній блок умови.

Графічну схему розгалуження з трьома можливими варіантами зображено на рисунку 13.3.

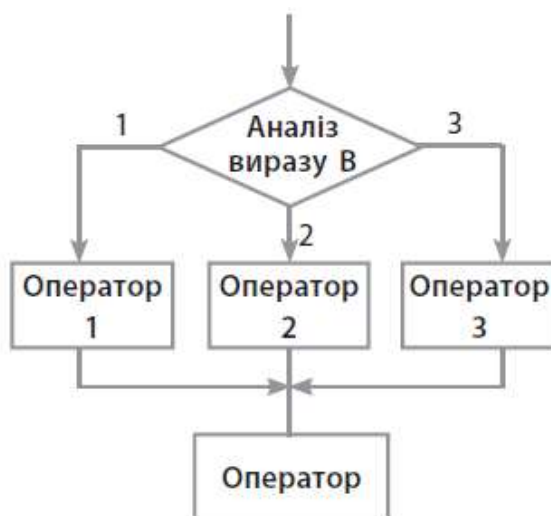


Рисунок 13.3 – Графічна схема багатоальтернативного розгалуження

Якщо виникає потреба у реалізації вибору з декількох альтернатив використовується конструкція «if-elif-else».

Приклад реалізації конструкції «if-elif-else» наведений у лістингу 13.8.

Лістинг 13.8 – Реалізації конструкції «if-elif-else»

```
print("Яку мову ви бажаєте вивчати?")
print("1. Python")
print("2. JavaScript")
print("3. C#")
print("4. PHP")
print("5. Go")
answer = int(input("Оберіть варіант відповіді: "))
```

```
if answer == 1:
    print("Python – мова програмування...")
elif answer == 2:
    print("JavaScript – мова програмування...")
elif answer == 3:
    print("C# – мова програмування...")
elif answer == 4:
    print("PHP – мова програмування...")
elif answer == 5:
    print("Go – мова програмування...")
else:
    print("Шкода")
print("Отже, Ви зробили свій вибір...")
```

код основної гілки

if умова 1:

інструкція 1

...

elif умова 2:

інструкція 2

...

elif умова 3:

інструкція 3

...

elif умова n:

інструкція n

...

else:

інструкція n+1

...

код основної гілки

РЕАЛІЗАЦІЯ ЦИКЛІЧНИХ АЛГОРИТМІВ У PYTHON

Програмування — процес досить трудомісткий, проте, він може бути компенсований здатністю формулювати інструкції виконання циклічних алгоритмів. Для реалізації алгоритмів із повторенням мовою Python використовують два види операторів циклу:

- з параметром (зі змінною циклу);
- з умовою.



Цикл — алгоритмічна структура, за допомогою якої одна й та сама послідовність дій виконується багаторазово.

Серію інструкцій, які виконуються багаторазово під час виконання циклу, називають тілом циклу.

Цикли з параметрами

У циклах із параметром (циклах зі змінною циклу) кількість повторень інструкцій тіла циклу заздалегідь відома. Цикли з параметром у мові Python реалізуються оператором циклу `for` (для).

На початку виконання циклу змінна циклу набуває значення першого елемента об'єкта. Оператор `in` генерує логічне значення `True`, і виконується блок інструкцій тіла циклу. У наступному етапі змінна циклу набуде значення другого елемента об'єкта й також буде генеруватися значення `True`, у результаті чого буде виконано блок інструкцій тіла циклу. Після того як усі елементи об'єкта будуть перебрані, оператор `in` згенерує значення `False`, блок інструкцій тіла циклу не виконається, а управління буде передано інструкції, що розташована безпосередньо за блоком інструкцій тіла циклу. На рисунку 13.4 наведено фрагмент графічної схеми циклу з параметрами.

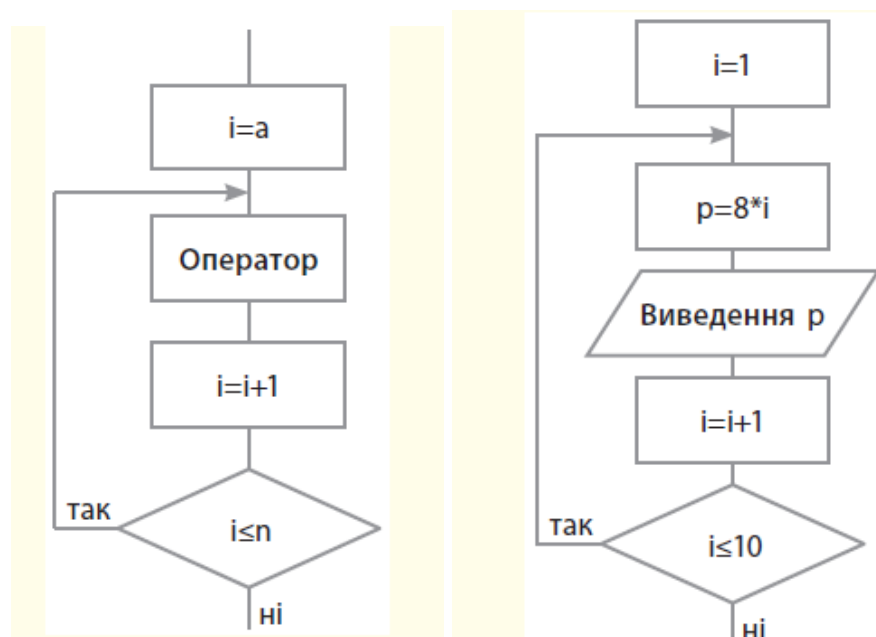


Рисунок 13.3 – Фрагмент графічної схеми з параметрами

Приклад використання циклу з параметрами для виведення на екран послідовності натуральних чисел до 10 включно наведено у лістингу 13.9.

Лістинг 13.9 – Виведення на екран натуральних чисел до 10 включно

```
for i in range(1, 11):  
    print(i)
```

У цьому коді `range(1, 11)` генерує послідовність чисел від 1 до 10 включно, а `print(i)` виводить кожне число на екран.

У таких циклах в якості параметра можуть бути залучені ітеровані об'єкти: списки, словники, кортежі, множини.

Приклад використання циклу з параметрами для виведення на екран елементів списку наведено у лістингу 13.10.

Лістинг 13.10 – Виведення на екран елементів списку

```
my_list = [1, 2, 3, 4, 5]  
for item in my_list:  
    print(item)
```

У цій програмі цикл `for` проходить по всіх елементах списку `my_list`, присвоюючи кожен елемент змінній `item` та виводячи її на екран.

Приклад використання циклу з параметрами для виведення на екран окремо кожного символу заданого рядка наведено у лістингу 13.11.

Лістинг 13.11 – Виведення на екран окремо кожного символу рядка

```
college = "ФКРКМ ДНУ"  
for letter in college:  
    print(letter)
```

Для поліпшення розуміння наведено приклад суми кінцевого ряду.

$$\sum_{i=1}^{n=5} i = 1 + 2 + 3 + 4 + 5 = 15$$

Реалізація поставленої задачі здійснена через розробку алгоритму циклу з параметрами, якій наведено у лістингу 13.12.

Лістинг 13.12 – Реалізація обчислення суми простого кінцевого ряду

```
a = 5 # Перший член арифметичної прогресії  
d = 1 # Різниця арифметичної прогресії  
Sum = 0.0 # сума ряду, результат  
  
for I in range(1, a + 1):  
    Sum += I  
print(Sum)
```

Зверніть увагу : в `range()` ми передали `n`, збільшене на одиницю ($n + 1$). Вся справа в тому, що цикл `for` зупиняється тоді, коли спрацьовує умова $i < n$ (в нашому ж випадку, коли $4 < 5$). Збільшуючи `n` на одиницю, ми отримуємо можливість дійти до $i = 5$ і порахувати суму вірно.

Цикли з умовою

У циклах з умовою кількість виконання інструкцій тіла циклу заздалегідь невідома: вона завершується після досягнення певної умови.

У мові Python цикли з передумовою реалізуються оператором `while`. `While` є універсальним циклом. Розрізняють типи циклів:

- з передумовою, де умова виконання інструкцій тіла циклу перевіряється перед початком їх виконання (рисунк 13.4 а);
- з післяумовою, де спочатку виконуються оператори тіла циклу, а потім перевіряється умова (рисунк 13.4 б).

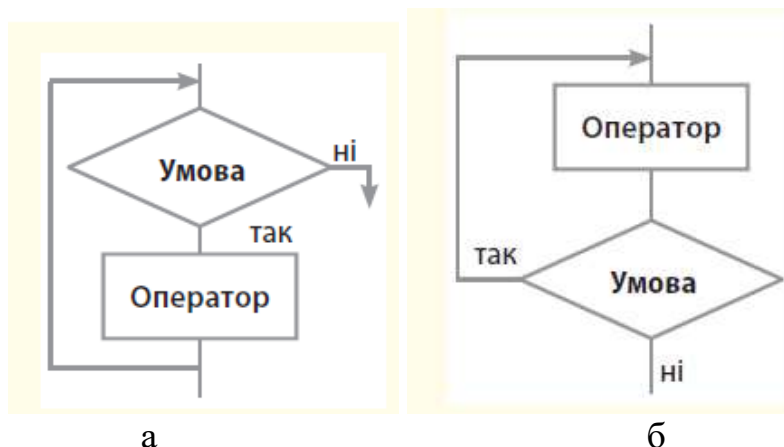


Рисунок 13.4 – Фрагмент графічної схеми з параметрами

У лістингу 13.13 наведено приклад реалізації циклічного алгоритму із передумовою.

Лістинг 13.13 – Реалізація циклічного алгоритму із передумовою

```
# Просимо користувача ввести число
number = int(input("Введіть ціле число: "))

# Перевіряємо, чи введене число є додатнім
while number < 0:
    # Якщо число від'ємне, то просимо користувача ввести інше число
    print("Введене число від'ємне. Введіть додатнє число.")
    number = int(input("Введіть ціле число: "))

# Виводимо повідомлення з введеним числом
print("Введене число:", number)
```

У наведеному прикладі користувачу пропонується ввести ціле число. Якщо введене число є від'ємним, програма просить користувача ввести інше число. Цей

процес повторюється, доки користувач не введе додатне число. Потім програма виводить повідомлення з введеним додатнім числом.

Приклад циклічного алгоритму із післяумовою наведено у лістингу 13.14.

Лістинг 13.13 – Реалізація циклічного алгоритму із передумовою

```
# Значення паролю
password = "secret"

# Просимо користувача ввести пароль
is_password_correct = input("Введіть пароль: ")

# Запускаємо цикл, поки пароль не буде правильним
while password != is_password_correct:
    print("Ви ввели неправильний пароль!")
    is_password_correct = input("Повторно введіть пароль: ")

# Код, що виконується після введення правильного паролю
print("Підтвердження правильно введеного паролю.")
```

У цій програмі користувач вводить пароль. Якщо введений пароль є правильним, програма виводить повідомлення про це та завершується. Якщо введений пароль неправильний, програма просить користувача ввести пароль ще раз. Цей процес повторюється, доки користувач не введе правильний пароль. Після введення правильного паролю програма виводить повідомлення про завершення роботи.

Висновки

Отже, Python — це високорівнева мова програмування, що дозволяє легко створювати лінійні алгоритми. Лінійний алгоритм складається з послідовно виконуваних команд, які розв'язують певну задачу. Для складання лінійних алгоритмів на Python можна використовувати різні типи даних (наприклад, числа, рядки, списки, словники тощо). При складанні таких алгоритмів важливо бажано передбачати введення та виведення даних через використання функції `input()` та `print()`. Перед виконанням складного лінійного алгоритму на Python важливо спростувати його до менш складених частин та перевіряти правильність розв'язку на кожному етапі. Для підтримки зрозумілості та читабельності лінійних алгоритмів на Python важливо використовувати зрозумілі та точні назви змінних, дотримуватися стандартів оформлення коду.

Вкладені умовні оператори дозволяють спростити складні розгалужені алгоритми на Python. Коректне використання розгалужених алгоритмів може значно поліпшити ефективність та зручність програми. Однак, слід уникати надмірного використання розгалужених алгоритмів, оскільки це може призвести до складності та заплутаності коду.

Циклічні алгоритми можуть бути дуже ефективними при роботі з великими об'ємами даних або при повторенні певних дій, що дозволяє скоротити код та зменшити його складність. Коректне використання циклічних алгоритмів може значно поліпшити ефективність та зручність програми.

Питання та завдання до контролю знань студентів

- 1 За якими ознаками класифікують мови програмування? Як класифікуються мови програмування за орієнтацією на клас задач? Як класифікуються мови програмування за принципами програмування?
- 2 Для чого призначено середовища програмування? Які функції виконує редактор тексту?
- 3 Поясніть сутність інтерпретації програм. Які переваги та недоліки мають компілятори й інтерпретатори програм?
- 4 Для чого призначено редактор зв'язку?
- 5 Які основні дії виконуються в процесі налагодження програм?
- 6 Назвіть сучасні інтегровані середовища програмування. Які переваги мають інтегровані середовища програмування?
- 7 Поясніть структуру проекту мовою Python та яку структуру може мати модуль мовою Python?
- 8 Яким ідентифікатором позначаються цілі числа? Яким ідентифікатором позначаються дійсні числа?
- 9 Які базові типи даних використовуються найчастіше?
- 10 Сформулюйте правило запису ідентифікаторів змінних. Що зберігається в змінних?
- 11 Поясніть сутність динамічної типізації змінних.
- 12 Поясніть сутність поняття «посилання на об'єкт».
- 13 Як можна одним оператором присвоїти різні значення кільком об'єктам?
- 14 Що називають лексемою? Які існують види лексем?
- 15 Яка функція призначена для визначення типу даних? У яких системах числення можуть подаватися числа?
- 16 Які операції можуть виконуватися над даними типу `int`?
- 17 На скільки типів даних може посилатися змінна в програмі?
- 18 Для чого призначено модуль `math`?
- 19 Поясніть сутність вкладеного оператора умовного переходу. Поясніть правило запису блоку операторів у вкладеному операторі умовного переходу.
- 20 Які існують види циклічних алгоритмів? Для чого призначено оператор `break`; оператор `continue`?

Лекція № 14

з навчальної дисципліни Інформатика

Тема Елементи функціонального програмування

Мета заняття Виробити вміння стосовно залучання базових алгоритмічних конструкцій, ознайомити із поняттям рекурсії, розвинути уміння по створенню і реалізації користувацьких функцій, а також обробці винятків

Матеріально-технічне забезпечення та дидактичні засоби, ТЗН

1 Інтерактивний вид діяльності, створений засобами Moodle

2 Мультимедійний презентаційний матеріал

Час – 2 години

План проведення лекції

Структура лекції	Відведений час	Методичні вказівки
1 Організаційна частина	5 хв.	Включає в себе: <i>привітання</i> , яке має на меті привернути увагу, забезпечити контакт лектора з аудиторією; <i>визначення присутності студентів на занятті</i>
2 Актуалізація опорних знань, перевірка вивченого матеріалу та мотивація навчальної діяльності студентів	10 хв.	Включає в себе: <i>вступне зауваження, мотивацію актуальності теми, перевірку попереднього матеріалу, формулювання мети лекції, огляд головних питань теми</i> (вказати питання та завдання для перевірки знань студентів)
3 Основна частина (викладення навчальних питань лекції)	60 хв.	Головне місце приділяється <i>викладенню основного змісту матеріалу</i> теми, його <i>аналізу, узагальненню висунутих положень</i> . Для успіху лекції важливе значення має поділ матеріалу на розділи, основні питання
4 Заклучна частина Домашнє завдання: (відповідно до робочої програми)	5 хв.	Включає в себе: <i>узагальнення, теоретичні і фактичні висновки, закріплення вивченого на лекції матеріалу</i>

Література

1 Інформатика (профільний рівень) : підруч. для 10 кл. закл. заг. серед. освіти / В.Д. Руденко, Н.В. Речич, В.О. Потієнко. — Харків : Вид-во «Ранок», 2018.

2 Head First. Pythonб / Е. Фрімен, Е. Робсон, Б. Бейтс, П. Беррі, К. Сієрра (переклад укр. Г. Якубовська) — Харків : Вид-во «Фабула», 2021. – 624 с.

Навчальні матеріали лекції

ВСТУП

Функціональне програмування — це парадигма програмування, в якій програми будуються за допомогою функцій. Самі функції, в свою чергу, виконують обчислення та маніпулюють даними без використання змінних стану.

Функціональне програмування базується на математичній теорії функцій та інших формальних методах, що дозволяє створювати програми, які мають більшу надійність, зручність у використанні та легкість розуміння.

ОСНОВНІ ПРИНЦИПИ ФУНКЦІОНАЛЬНОГО ПРОГРАМУВАННЯ:

Функції являються основним елементом програми, тобто вони є самодостатніми та повинні мати чітко визначену область визначення та область значень.

Функції — це універсальний засіб структурування програм, вони дають змогу розбити складну програму на окремі частини, можуть викликатися у виразах і застосовуватися в умовних інструкціях та інструкціях циклу.

У процесі кожного звернення до функції вона виконує одні й ті самі дії над різними значеннями даних і повертає отримане значення до основної програми. Наприклад, функція обчислення суми членів арифметичної прогресії під час кожного звернення до неї може обчислювати суму її членів із різними значеннями різниці та кількості членів прогресії. Сутність програми з двома функціями пояснюється схемою, яку наведено на рисунку 14.1.

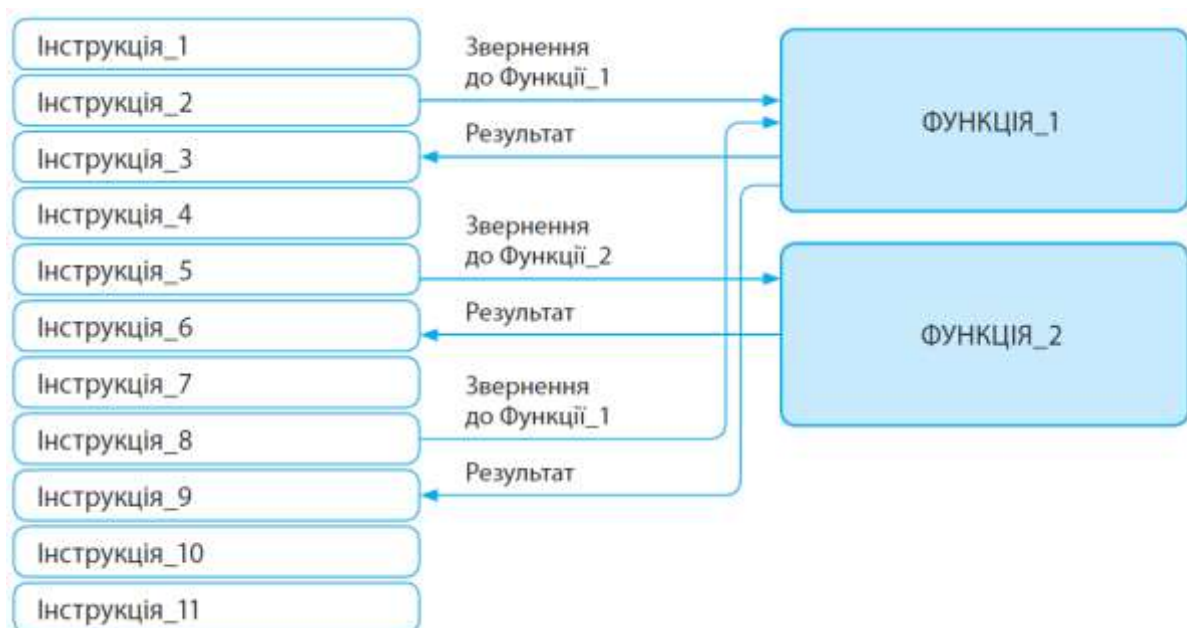


Рисунок 14.1

Всі обчислення виконуються за допомогою функцій, тобто програми не мають стану та не містять глобальних змінних. Функції можуть бути передані як аргументи іншим функціям, що дозволяє створювати більш складні функції з малих та простих. Програми повинні бути написані таким чином, щоб забезпечувати відсутність побічних ефектів, тобто зміни стану системи, які не пов'язані зі зміною результату функції.

Парадигма функціонального програмування пропонує використання рекурсії як основного механізму повторення та ітерації. Процес рекурсії полягає в формуванні алгоритму у якому функція здійснює виклик самої себе.

Функціональне програмування спрямоване на створення коду, який є більш зрозумілим та простим для розуміння, що сприяє зменшенню кількості помилок та полегшує розробку.

ФУНКЦІЇ КОРИСТУВАЧА ТА МОДУЛІ МОВИ PYTHON

Користувацька функція — це невеликий, логічно завершений програмний код, до якого можна звертатися багаторазово з різних місць основної програми.

Користувацькі функції у мові програмування Python визначаються за допомогою інструкції `def`, структуру якої наведено у лістингу 14.1:

Лістинг 14.1 – Структура визначення функції

```
def <ім'я функції> ([параметри]):  
    <тіло функції>  
    [return<результат>]
```

Як бачимо, існують функції з параметрами і без параметрів.

Ім'я функції — це звичайний унікальний ідентифікатор латинськими буквами. Ім'я функції складається з малих символів.

Тіло функції — це сукупність інструкцій, які реалізують певне завдання, наприклад, інструкції обчислення середнього значення списку чисел.

Параметри — це імена об'єктів (змінних, списків тощо), які отримують конкретні значення (атрибути) під час звернення до функцій. Параметри відокремлюються один від одного комою. Якщо функція не має параметрів, зазначаються лише порожні круглі дужки.

Якщо тіло функції не містить інструкцій, слід розмістити оператор `pass`, який жодних дій не виконує. Приклад користувацької функції, яка не містить інструкцій наведено у лістингу 14.2.

Лістинг 14.2 – Користувацька функція, що не містить інструкцій

```
def funkt_1:  
    pass
```

Визначення функцій визначається зазвичай на початку програми, одразу після імпортування необхідних модулів.

Одна й та сама функція може виконувати різні дії над об'єктами. Яка конкретно дія буде виконуватися, залежить від типу об'єктів, над якими виконуються відповідні операції.

ЗАСТОСУВАННЯ ЗМІННИХ У КОРИСТУВАЦЬКИХ ФУНКЦІЯХ

Змінні поділяються на локальні та глобальні.



Локальні змінні — це змінні, які оголошені всередині користувацької функції.

Локальні змінні доступні лише програмному коду тіла функції. Вони існують виключно під час безпосереднього виконання функції.



Глобальні змінні — це змінні, які оголошені в основній програмі, за межами функції.

Вони доступні в будь-якій частині програми, у тому числі всередині функції. Локальні та глобальні змінні можуть мати однакові імена. Але операції над локальними змінними всередині функцій не впливають на значення однойменних глобальних змінних. Для запобігання плутанині рекомендується не використовувати однаковими іменами локальних та глобальних змінних.

У мові Python саме об'єкти визначають синтаксичну сутність операцій, які будуть виконуватися над ними.



Поліморфізм — явище, коли синтаксична сутність операцій залежить від типу об'єктів, які опрацьовуються.

Поліморфними є фактично всі операції у мові Python, а також відповідні функції. Інтерпретатор мови сам перевіряє сумісність типів об'єктів. Якщо виявиться, що вони несумісні, видає повідомлення про синтаксичну помилку. Поліморфізм є одним із фундаментальних понять об'єктно-орієнтованого програмування.

Параметрам функцій можна передавати не лише значення літералів, змінних, а й значення кортежів, списків, словників. Якщо параметрам функції передаються значення словника, то в інструкції звернення до функції перед її іменем ставлять два символи зірочка (**). А якщо списку або кортежу, то одну зірочку. Одночасно можна передавати параметрам функції значення різних типів.

Інструкція `def` у процесі оголошення функції створює об'єкт, який має тип `function`, і зберігає посилання на цей об'єкт в ідентифікаторі, яким є ім'я функції. Це посилання на функцію можна зберегти в іншій змінній.

ПРИКЛАДИ ВИЗНАЧЕННЯ КОРИСТУВАЦЬКИХ ФУНКЦІЙ

Приклад 1

У лістингу 14.3 наведено приклад визначення користувацької функції з назвою `multiply`, яка приймає два аргументи, a та b .

Лістинг 14.3 – Визначення функції `multiply`

```
def multiply(a, b):                # визначення користувацької функції
    result = a * b
    return result

product = multiply(5, 7)          # виклик користувацької функції
print(product)                   # виведення результату
```

У тілі функції реалізовано обчислення добутку аргументів, і повертаємо його за допомогою ключового слова `return`. Далі, за необхідністю, здійснюється виклик користувацької функції у основному алгоритмі програми.

Приклад 2

У лістингу 14.4 наведено приклад визначення користувацької функції для обчислення факторіалу числа.

Лістинг 14.4 – Визначення користувацької функції `factorial`

```
def factorial(n):                  # визначення користувацької функції
    result = 1
    for i in range(1, n + 1):
        result *= i
    return result

fct = factorial(5)                # виклик користувацької функції
print(fct)                       # виведення результату
```

Визначена функція, приймає аргумент n . У тілі функції створюємо змінну `result` зі значенням 1, а потім за допомогою циклу `for` обчислюємо добуток чисел від 1 до n . Кожен раз, коли ми проходимо через цикл, ми множимо `result` на поточне значення i . Нарешті, ми повертаємо значення `result` за допомогою ключового слова `return`.

Приклад 3

У лістингу 14.5 наведено приклад визначення функції для виведення на екран результату множення цілого числа на послідовність від 1 до 10 включно.

Лістинг 14.4 – Визначення користувацької функції `factorial`

```
def multiply_list(x):              # визначення користувацької функції
    for i in range(1, 11):
        result = x * i
        print(f"{x} · {i} = {result} ")

multiply_list(3)
```

Висновки

Функціональне програмування спрямоване на створення коду, який є більш зрозумілим та простим для розуміння, що сприяє зменшенню кількості помилок та полегшує розробку. Функціональний код зазвичай має менше побічних ефектів, що робить його більш передбачуваним та легшим для тестування та відлагодження. Функціональні програми зазвичай складаються з меншої кількості коду, що дозволяє зменшити час на розробку та підтримку.

Питання та завдання до контролю знань студентів

- 1 Які функції називають користувацькими? Що називають тілом функції?
- 2 Яка різниця між аргументами й параметрами? Поясніть різницю між локальними та глобальними змінними. Як передаються значення аргументів необов'язковим параметрам?
- 3 Поясніть сутність звернення до функції із основної програми та які переваги надає використання функції у програмуванні?
- 4 Яку роль виконує інструкція `return` у тілі функції? Для чого застосовується оператор `pass`?
- 5 Поясніть сутність поліморфізму. Які функції називають анонімними? Як оголошують анонімні функції?
- 6 Як реалізується передавання параметрам у довільному порядку значення аргументів?
- 7 Як можна реалізувати передавання параметрам довільної кількості значень аргументів? Поясніть порядок використання у функції комбінації різних параметрів.
- 8 Що називають рекурсією в програмуванні? Із якою метою в тілі рекурсії передбачається умовний оператор? На якій структурі даних засновано реалізацію рекурсивних функцій?
- 9 Які дефекти має рекурсивне обчислення? Поясніть сутність «заглиблення» та виходу зі стеку в процесі реалізації рекурсії.
- 10 Складіть програму з використанням функції без параметрів для обчислення об'єму конуса.
- 11 Складіть програму з використанням функцій із параметрами для обчислення двох різних за розміром прямокутних трикутників із відомими значеннями їх катетів і обчисліть площі кола з відомим радіусом.
- 12 Кубик із цифрами від 1 до 6 підкидають п'ять разів. Складіть програму обчислення середнього значення цих чисел із використанням функції з параметрами.
- 13 Складіть програму обчислення значення виразу $a*b-c/d$ із використанням функції з параметрами для значень списку `[3, 5, 2, 7]` і кортежу `(4, 1, 5, 6)`.
- 14 Генеруються три цілі випадкові числа в діапазоні від 3 до 7. Складіть програму з використанням функції з параметрами для обчислення суми квадратів цих чисел.
- 15 З'ясуйте в Інтернеті, через які області України протікає Дніпро, та знайдіть їх площі. Складіть програму визначення областей із максимальною та мінімальною площами. Обчисліть середню площу. Використайте функції з параметрами.

Лекція № 15

з навчальної дисципліни Інформатика

Тема Застосування засобів розробки графічного інтерфейсу

Мета заняття Оволодіти навичками по розробці структури GUI програмних додатків засобами стандартного модуля Tkinter, досягти оволодіння розробки GUI-додатків засобами модуля ttk

Матеріально-технічне забезпечення та дидактичні засоби, ТЗН

1 Інтерактивний вид діяльності, створений засобами Moodle

2 Мультимедійний презентаційний матеріал

Час – 2 години

План проведення лекції

Структура лекції	Відведений час	Методичні вказівки
1 Організаційна частина	5 хв.	Включає в себе: <i>привітання</i> , яке має на меті привернути увагу, забезпечити контакт лектора з аудиторією; <i>визначення присутності студентів на занятті</i>
2 Актуалізація опорних знань, перевірка вивченого матеріалу та мотивація навчальної діяльності студентів	10 хв.	Включає в себе: <i>вступне зауваження, мотивацію актуальності теми, перевірку попереднього матеріалу, формулювання мети лекції, огляд головних питань теми</i> (вказати питання та завдання для перевірки знань студентів)
3 Основна частина (викладення навчальних питань лекції)	60 хв.	Головне місце приділяється <i>викладенню основного змісту матеріалу теми, його аналізу, узагальненню висунутих положень</i> . Для успіху лекції важливе значення має поділ матеріалу на розділи, основні питання
4 Заключна частина Домашнє завдання: (відповідно до робочої програми)	5 хв.	Включає в себе: <i>узагальнення, теоретичні і фактичні висновки, закріплення вивченого на лекції матеріалу</i>

Література

1 Інформатика (профільний рівень) : підруч. для 10 кл. закл. заг. серед. освіти / В.Д. Руденко, Н.В. Речич, В.О. Потієнко. — Харків : Вид-во «Ранок», 2018.

2 Head First. Pythonб / Е. Фрімен, Е. Робсон, Б. Бейтс, П. Беррі, К. Сієрра (переклад укр. Г. Якубовська) — Харків : Вид-во «Фабула», 2021. – 624 с.

Навчальні матеріали лекції

ВСТУП

Графічний інтерфейс програмного забезпечення (GUI) — це спосіб взаємодії користувача з комп'ютерною програмою, який використовує графічні елементи, такі як кнопки, поля введення, вікна тощо. Мова Python забезпечує взаємодію з різними сторонніми графічними модулями. Завдяки таким модулям можна займатися розробкою програмних додатків з розвинутим графічним інтерфейсом користувача.

Проте, стандартний модуль tkinter має достатньо потужні графічні можливості.

ЗАГАЛЬНИЙ ПОРЯДОК СТВОРЕННЯ ГРАФІЧНОГО ІНТЕРФЕЙСУ

Основним призначенням модуля tkinter є створення графічних інтерфейсів програмних додатків, а також реалізації проектів, що можуть бути пов'язані із створенням статичної та динамічної графіки. За допомогою модуля tkinter можна створювати також програмні додатки для візуалізації даних.



Віджет — компонент графічного інтерфейсу користувача, тобто об'єкт відповідного класу.

Щоб скористатися можливостями графічного інтерфейсу, розробнику слід передбачити розміщення у вікні програмного додатку необхідних графічних об'єктів (віджетів), а також ті дії, які мають виконуватися в процесі взаємодії користувача з цими об'єктами. Такий підхід розробки програмних продуктів, неодмінно збільшує обсяг програмного коду та ускладнює його структуру, але суттєво спрощує взаємодію з програмним продуктом.

РОЗРОБКА ВІКНА ПРОГРАМНОГО ДОДАТКУ

Ініціалізація основного вікна програмного додатку

Розробка конфігурації вікна програмного додатку розпочинається після впровадження модуля tkinter. У лістингу 15.1 наведено рекомендований метод впровадження модуля та інструкції генерації основного вікна додатку.

Лістинг 15.1 – Рекомендований імпорт модуля tkinter та інструкції по генерації основного вікна

```
from tkinter import *  
  
root = Tk()  
root.mainloop()
```

Вікно програмного додатку реалізується, як об'єкт класу Tk, ідентифікатор якого традиційно іменують root. Інструкція наступного рядку ініціює нескінченний цикл, який реалізовано через залучання методу mainloop(). Результатом виконання програми є створення вікна програмного додатку розмірами 250 px × 250 px у довільній дільниці екрану (рисунок 15.1).

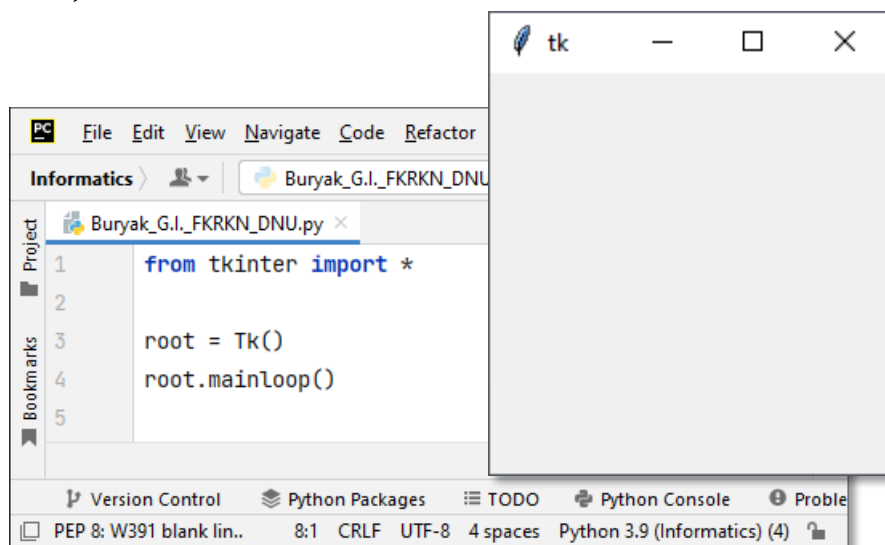


Рисунок 15.1 – Створення вікна програмного додатку

Об'єкт root успадковує базові властивості від класу Tk. Основне вікно додатку оснащено стандартними інструментами взаємодії, а також має механіку типову для елементів графічного інтерфейсу користувача. Тому завершення виконання програмного додатку, без залучання додаткових методів, здійснюється через знайомі інструменти взаємодії із графічним інтерфейсом користувача.

Кастомізація вікна програмного додатку

Методи кастомізації вікон однаково можуть бути застосовані для основного вікна програмного додатку (root), вікон верхнього рівня (toplevel), а також вікон діалогу. У лістингу 15.2 наведено найуживаніші з методів кастомізації основного вікна програмного додатку.

Лістинг 15.2 – Найуживаніші методи кастомізації основного вікна

```
root.title("ФКРКМ ДНУ ім. О. Гончара")  
root.iconphoto(True, PhotoImage(file="img/rocket.png"))  
root.geometry("350x250+500+200")  
root.config(background="deep sky blue")  
root.resizable(False, False)
```

У перших двох рядках інструкції задіяні методи кастомізації рядка заголовку основного вікна додатку. Метод title генерує текстовий вміст рядка заголовку, приймаючи єдиний аргумент str-типу. Метод iconphoto генерує відображення іконки у рядку заголовку, приймаючи два аргументи. Перший аргумент bool-типу із значенням True призначає розповсюдження іконки, як на основі так і на дочірні вікна

програмного додатку. У якості другого аргументу вказано функцію PhotoImage із зазначеним місцем розташування файлу.

Метод `geometry` відповідає за розміри та фіксоване місце розташування вікна у момент його генерації. Метод приймає аргумент `str`-типу у форматі "ширина x висота + зсув по вісі x + зсув по вісі y".

Метод `config` у якості аргумента приймає властивість `background` із параметром назви кольору фону вікна. В мові Python можуть бути використані назви кольорів у вигляді констант `str`-типу, а також доступні можливості відображення кольорів у форматі `rgb` та в інших форматах.

Метод `resizable` може бути задіяний, якщо розробником програмного додатку не передбачена можливість зміни розмірів вікна. Метод приймає два аргументи `bool`-типу, де перше значення `False` призначає обмеження зміни розмірів вікна по ширині, а друге значення `False` — по висоті.

Залучання методів по кастомізації вікон програмних додатків здійснюється в залежності від прикладного призначення програмного продукту.

МЕТОДИ ВПРОВАДЖЕННЯ КОМПОНЕНТІВ ГРАФІЧНОГО ІНТЕРФЕЙСУ

Будь-який компонент графічного інтерфейсу (віджет) може бути впроваджений через вибір одного з менеджерів розташування. Модуль `tkinter` оснащено трьома типами менеджерів розташування: `pack`, `place`, `grid`. Вони залучаються як звичайні методи до вказаних об'єктів — віджетів.

Для вказаного об'єкта (віджета) можна використовувати лише один обраний тип менеджера розташування. При спробі змішування різних типів менеджерів розташування середовище розробки, швидше за все, попередить про некоректне формулювання інструкцій.

Менеджер розташування `pack`



`pack` — менеджер розташування, який використовують для впровадження віджетів один за одним (зліва направо або зверху донизу).

Основною перевагою зазначеного методу є простота у використанні, оскільки він може бути задіяний навіть без зазначення властивостей. При застосуванні менеджера `pack` без додавання властивостей відбувається розташування віджетів один за одним, із вирівнюванням зверху, посередині.

Як правило, менеджер розташування `pack` використовується для розробки простої конфігурації графічних інтерфейсів. Безумовно, розробка складної конфігурації компонентів графічного інтерфейсу із застосуванням менеджера `pack` можлива, але потребує врахування порядку впровадження компонентів інтерфейсу, а також залучання додаткових віджетів, кожен з яких вимагає дворядкового опису інструкції по їх впровадженню. А це, в свою чергу, неодмінно впливає на загальний обсяг коду та погіршує його структуру.

Менеджер розташування grid



`grid` — менеджер розташування, який використовують для впровадження віджетів в уявні комірки таблиці.

Популярність використання зазначеного менеджера розташування обумовлене зручними умовами по конструюванню конфігурації графічного інтерфейсу. Уявити компоновання елементів інтерфейсу у вигляді комірок таблиці достатньо легко, а за рахунок властивостей, які забезпечують об'єднання рядків чи стовбців можлива розробка досить складних реалізацій графічного інтерфейсу користувача.

Менеджер розташування place



`place` — менеджер розташування, який використовують для впровадження віджетів у фіксованому місці з фіксованим розміром.

Зазначений менеджер розташування є не дуже зручним, проте, надає повну свободу в розміщенні віджетів. Власне тому менеджер `place` може бути задіяний для розробки складної конфігурації інтерфейсу користувача оскільки дозволяє вказувати координати розташування у відносних одиницях для реалізації «гумового» розміщення. При використанні менеджера розташування `place` необхідно вказувати координати кожного впровадженого віджета.

Методом `place` віджету вказується його положення або в пікселях, або в частках вікна. Також абсолютно і відносно можна задавати розмір самого віджета.

ОГЛЯД КОМПОНЕНТІВ ГРАФІЧНОГО ІНТЕРФЕЙСУ

Компонент класу Label



`Label` — клас віджету, що являє собою напис без можливості його редагування.

У лістингу 15.3 наведено інструкції впровадження віджета класу `Label` в парі із менеджером розташування `pack`.

Лістинг 15.3 – Впровадження віджета класу Label

```
name = Label(master=root, text="Вкажіть ім'я")
name.pack()
```

Як показано в прикладі впровадження віджета реалізується дворядковою інструкцією. У першому рядку об'єкту `name` призначається клас `Label`, вказується місце розташування — об'єкт `root` та задається єдина властивість `text`, параметром якої

є текст в лапках — вміст віджета. У другому рядку об'єкту `name` призначено менеджер розташування — метод `pack` без зазначення властивостей.



Зауваження

Текст інструкцій конфігурації основного вікна програми, а також інструкцій впроваджених віджетів має бути записаний до призначення методу `mainloop` об'єкту `root`, тобто `root.mainloop()` — останній рядок програми

Компонент класу Entry



`Entry` — клас віджету, що являє собою елемент керування для введення однорядкового тексту.

У лістингу 15.4 наведено інструкції впровадження віджета класу `Entry` в парі із менеджером розташування `pack`.

Лістинг 15.4 – Впровадження віджета класу Entry

```
enter_name = Entry(master=root, width=30)
enter_name.pack()
```

В наведеному прикладі показана реалізація віджета класу `Entry` із майже мінімальним набором властивостей щодо впровадження. У першому рядку об'єкту `enter_name` призначається клас `Entry`, вказується місце розташування — об'єкт `root`. Віджет буде відображений і без вказаної властивості `width`, однак, її залучання є майже обов'язковим. У другому рядку об'єкту `enter_name` призначено менеджер розташування — метод `pack` без зазначення властивостей.

Компонент класу Button



`Button` — клас віджету, що являє собою елемент керування у вигляді кнопки.

У лістингу 15.6 наведено інструкції впровадження віджета класу `Button` в парі із менеджером розташування `pack`.

Лістинг 15.6 – Впровадження віджета класу Button

```
button_1 = Button(master=root, text="Обробити дані")
button_1.pack()
```

У першому рядку об'єкту `button_1` призначається клас `Button`, вказується властивість місця розташування (`master`) — об'єкт `root`, а також задається властивість `text`, параметром якої є текст в лапках — вміст віджета. У другому рядку об'єкту

button_1 призначено менеджер розташування — метод pack без зазначення властивостей.

Результат впровадження розглянутих віджетів наведено рисунку 15.2.

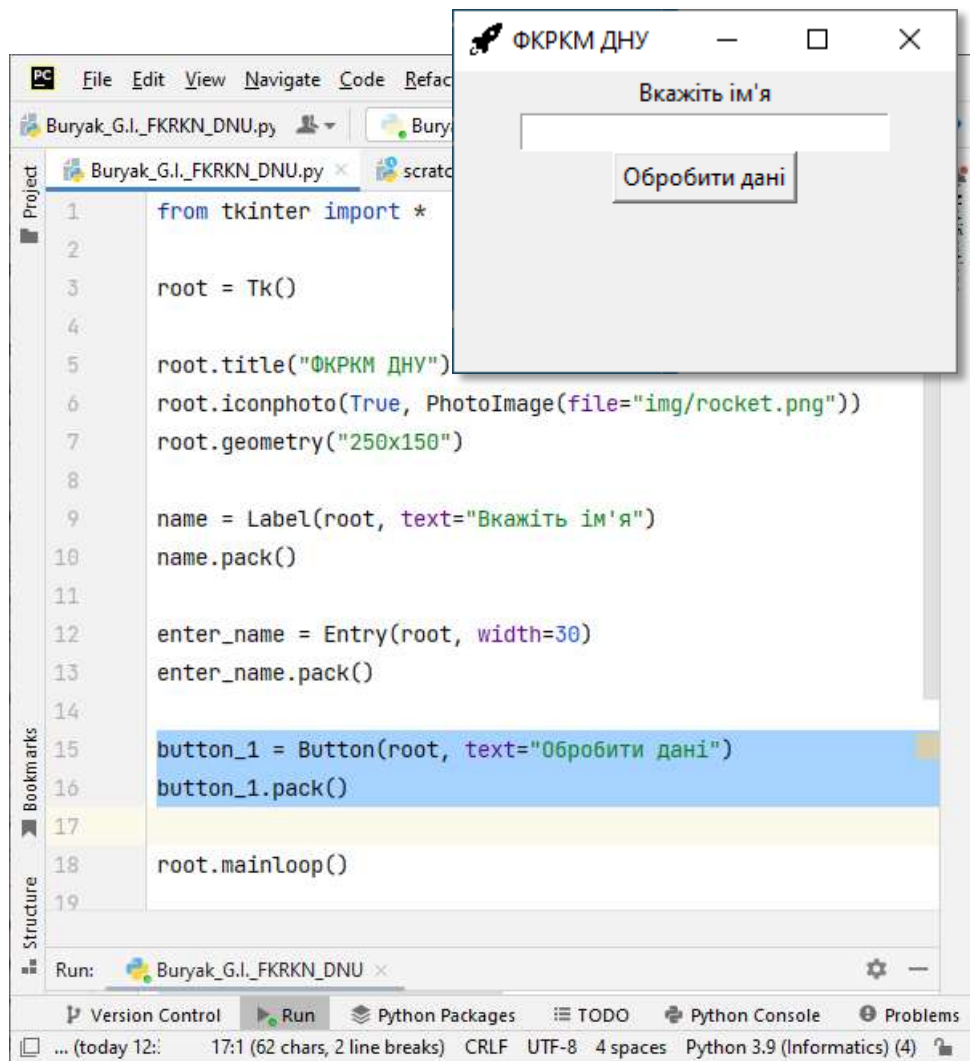


Рисунок 15.2 – Впровадження віджета класу Button

Компонент класу Spinbox



Spinbox — клас віджету, який являє собою поле введення даних, та керувальні елементи переміщення за пунктами списку.

На рисунку 15.3 наведено приклад впровадження об'єктів класу Spinbox. За своїм призначенням віджет класу Spinbox схожий на віджет класу Entry.

Об'єкту класу Spinbox присвоюють ідентифікатор, який обертають довільним. В таблиці Г.3 наведено властивості віджета класу Spinbox.

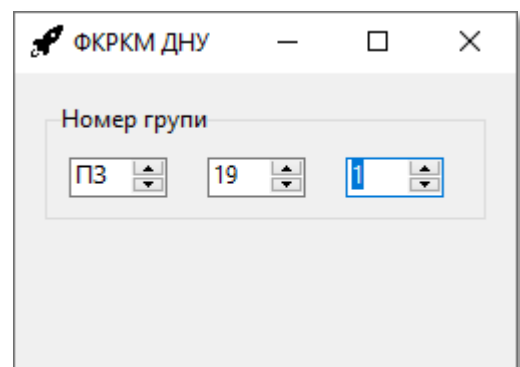


Рисунок 15.3

У лістингу 15.7 наведено інструкції впровадження віджета класу Spinbox в парі із менеджером розташування pack.

Лістинг 15.7 – Впровадження віджета класу Spinbox

```
list_number = [i for i in range(1, 11)]

choice_num = Spinbox(master=root, width=5, values= list_number)
choice_num.pack()
```

В наведеному прикладі показана генерація списку list_number у реалізації першого рядку інструкції. Далі об'єкту choice_num призначається клас Spinbox, вказується місце його розташування — об'єкт root, а також призначаються властивості: width та values. Властивість width відповідає за ширину віджета, властивість values відображає значення із згенерованого списку list_number. Розташування віджета здійснюється із використанням методу pack із без зазначення властивостей.

Компоненти класів Frame та LabelFrame

Якщо інтерфейс програмного додатку перенасичений елементами керування та потребує його розподілу на функціональні дільниці, то для такої реалізації використовують віджети класів Frame або LabelFrame.



Frame та LabelFrame — класи віджетів, які призначені для компонування групи впроваджених віджетів.

Об'єкти класів Frame та LabelFrame мають ідентифікатори які обертають довільними. У якості першого аргументу вказують місце розташування віджетів, а далі, через кому, перелічують властивості із зазначеними параметрами.

У лістингу 15.8 наведено інструкції впровадження віджетів класів Frame та LabelFrame в парі із менеджером розташування pack.

Лістинг 15.8 – Впровадження віджетів класів Frame та LabelFrame

```
fr = Frame(master=root)
fr.pack()

lb_fr = LabelFrame(master=root, text="Вихідні дані")
lb_fr.pack()
```

Хоча об'єкти зазначених класів мають спільні властивості, впроваджений в інтерфейс об'єкт класу Frame не спостерігається користувачем на відміну від об'єкта класу LabelFrame, якій містить наочну межу та назву.

Розглянемо реалізацію групування віджетів через залучання компонентів класів Frame та LabelFrame із застосуванням менеджера розкошування pack.

Компонент класу Radiobutton



Radiobutton — клас віджету, який являє собою елемент списку з місцем для мітки за умови можливості виставити лише одну мітку.

На рисунку 15.4 наведено приклад впровадження віджетів класу Radiobutton в основне вікно додатку.

Об'єкту класу Radiobutton присвоюють ідентифікатор, який обертають довільним. У якості першого аргументу вказують місце розташування віджета, а далі, через кому, перелічують властивості із зазначеними параметрами.

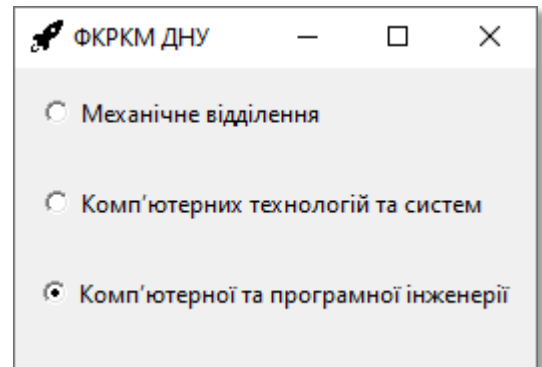


Рисунок 15.4

У лістингу 15.8 наведено інструкції впровадження віджетів класу Radiobutton в парі із менеджером розташування pack.

Лістинг 15.8 – Впровадження віджетів класу Radiobutton

```
v = IntVar()
rbutton_1 = Radiobutton(root,
                        text='Механічне відділення',
                        variable=v,
                        value=1)
rbutton_2 = Radiobutton(root,
                        text='Комп'ютерних технологій та систем',
                        variable=v,
                        value=2)
rbutton_3 = Radiobutton(root,
                        text='Комп'ютерної та програмної інженерії',
                        variable=v,
                        value=3)
rbutton_1.pack(padx=10, pady=10, anchor=W)
rbutton_2.pack(padx=10, pady=10, anchor=W)
rbutton_3.pack(padx=10, pady=10, anchor=W)
```

Об'єкти класу Radiobutton завжди використовують у групі, причому увімкненим може бути лише один перемикач.

Компонент класу Checkbutton



Checkbutton — клас віджету, який являє собою елемент списку з місцем для мітки із можливістю виставити декілька міток.

Об'єкту класу Checkbutton присвоюють ідентифікатор, який обертають довільним. У якості першого аргументу вказують місце розташування віджета, а далі, через кому, перелічують властивості із зазначеними параметрами.

На рисунку 15.5 наведено приклад впровадження віджетів класу `Checkbutton` в основне вікно додатку.

У лістингу 15.9 наведено інструкції впровадження віджетів класу `Checkbutton` в парі із менеджером розташування `pack`.

Лістинг 15.9 – Впровадження віджетів класу `Checkbutton`

```
groups = ["спец. 121 ПЗ", "спец. 123 КС"]

var_1 = IntVar()
var_2 = IntVar()

check_1 = Checkbutton(root,
                       text=groups[0], variable=var_1,
                       onvalue=1, offvalue=0)
check_2 = Checkbutton(root,
                       text=groups[1], variable=var_2,
                       onvalue=1, offvalue=0)

check_1.pack(padx=25, pady=5, anchor=W)
check_2.pack(padx=25, pady=5, anchor=W)
```

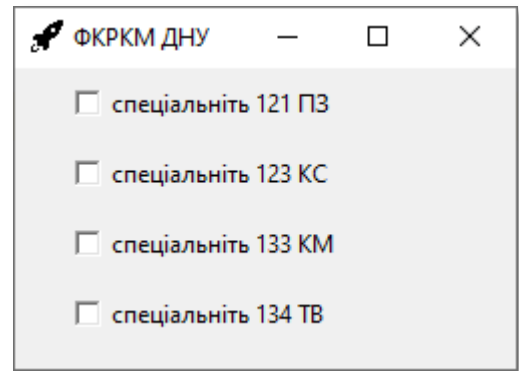


Рисунок 15.5

При залучанні кількох елементів списку потрібно для кожного використати власну змінну.

Компонент класу `Text`



`Text` — клас віджету, який являє собою елемент керування для введення довільного багаторядкового тексту.

На рисунку 15.6 наведено приклад впровадження віджету класу `Text` в основне вікно додатку.

Об'єкту класу `Text` присвоюють ідентифікатор, який обертають довільним. У якості першого аргументу вказують місце розташування віджета, а далі, через кому, перелічують властивості із зазначеними параметрами.

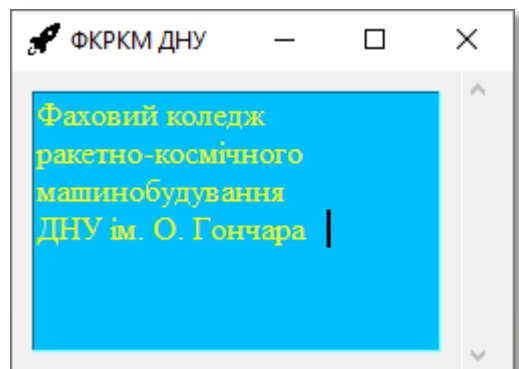


Рисунок 15.6

Незалежно від габаритів основного вікна додатку, чи розмірів самого віджета класу `Text`, його реалізація, доволі часто, передбачає наявність таких елементів керування, як горизонтальна чи вертикальна смуги прокрутки.

Компонент класу Scrollbar



Scrollbar — клас віджету, який являє собою елемент керування: смугу прокручування (скроллер).

Об'єкту класу Scrollbar присвоюють ідентифікатор, який обертають довільним. У якості першого аргументу вказують місце розташування віджета, а далі, через кому, перелічують властивості із зазначеними параметрами.

У лістингу 15.10 наведено інструкції впровадження віджетів класу Text та класу Scrollbar із застосуванням менеджера розташування pack.

Лістинг 15.10 – Впровадження віджетів класу Text та класу Scrollbar

```
text = Text(root,
             width=25,
             height=7,
             bg="deep sky blue",
             fg="yellow",
             font=("Times New Roman", 12),
             wrap=WORD)
text.pack(padx=10, pady=10, side=LEFT)

scroll = Scrollbar(command=text.yview)
scroll.pack(side=LEFT, fill=BOTH)

text.config(yscrollcommand=scroll.set)
```

Важливою властивістю віджета класу Text — є wrap, що відповідає за перенесення тексту. Щоб переносити словами, потрібно надати їй значення параметру WORD.

Компонент класу Listbox



Listbox — клас віджету, який являє собою список, з елементів якого користувач може вибрати один або кілька пунктів.

На рисунку 15.7 наведено приклад впровадження об'єктів класу Listbox.

Об'єкту класу Listbox присвоюють ідентифікатор, який обертають довільним. У якості першого аргументу вказують місце розташування віджета та перелічують властивості із зазначеними параметрами.

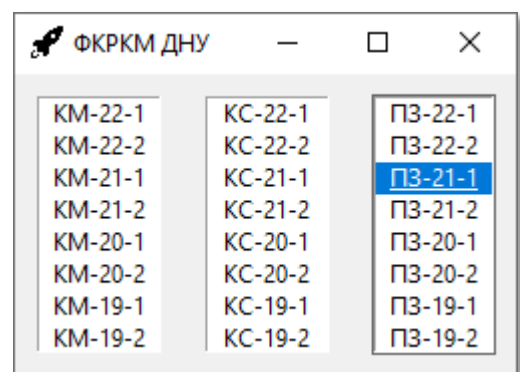


Рисунок 15.7

У лістингу 15.11 наведено інструкції впровадження віджетів класу Listbox із застосуванням менеджера розташування pack.

Лістинг 15.11 – Впровадження віджетів класу Listbox

```
group_list = Listbox(root, height=8, width=10,  
                      justify=CENTER, selectmode=EXTENDED)  
group_list.pack(padx=10, pady=10, side=LEFT)  
list_ps = ["ПЗ-22-1", "ПЗ-22-2", "ПЗ-21-1", "ПЗ-21-2",  
           "ПЗ-20-1", "ПЗ-20-2", "ПЗ-19-1", "ПЗ-19-2"]  
for i in list_ps:  
    group_list.insert(END, i)
```

В наведеному приладі відображення вмісту списку реалізовано через цикл for. В таблиці Г.9 наведено властивості віджета класу Listbox.

Компонент класу Canvas



Canvas — клас віджету, який являє собою полотно для відображення графічних зображень.

На рисунку 15.8 наведено приклад впровадження об'єкта класу Canvas, який здатний не лише відображати статичні зображення, а також змінювати і переміщати їх у процесі виконання програми.

Об'єкту класу Canvas присвоюють ідентифікатор, який обертають довільним. У якості першого аргументу вказують місце розташування віджета, а далі, через кому, перелічують властивості із зазначеними параметрами.

У лістингу 15.12 наведено інструкції впровадження віджета класу Canvas із застосуванням менеджера розташування pack.



Рисунок 15.8

Лістинг 15.12 – Впровадження віджета класу Canvas

```
canvas_logo = Canvas(master=root,  
                     width=300,  
                     height=300)  
canvas_logo.pack()
```

Як показано у прикладі ініціалізація об'єкта класу Canvas здійснюється типово, через призначення доступних властивостей. На полотні Canvas можуть бути розташувати такі об'єкти: arc — дуга; bitmap — двоколірне зображення; image — багатоколірне зображення; line — відрізок прямої; oval — коло або еліпс; polygon — багатокутник; rectangle — прямокутник; text — текст; window — вікно.

Для впровадження графічного зображення на полотні об'єкту класу Canvas призначають метод `create_` тип об'єкта. У якості аргументів методу передають спочатку обов'язкові — значення властивостей-координат, а далі перелічують решту властивостей.

Метод `focus_set` використовують для фокусування уваги на полотні, щоб мати можливість прив'язати події до виконання методів полотна.

Для видалення графічних зображень з полотна використовують метод `delete`. Якщо виникла необхідність звільнити від графічних зображень всієї ділянки полотна, то замість ідентифікаторів залучають сталу `ALL`.

Компонент класу Scale



Scale — клас віджету, що являє собою елемент керування у вигляді повзункової шкали, який дозволяє вибрати значення із заданого діапазону.

На рисунку 15.9 наведено приклад впровадження об'єкта класу `Scale`.

Об'єкту класу `Scale` присвоюють ідентифікатор, який обертають довільним. У якості першого аргументу вказують місце розташування віджета, а далі, через кому, перелічують властивості із зазначеними параметрами.

Оскільки віджет класу `Scale` є елементом керування, то для його використання слід передбачити наявність користувацької функції.

У лістингу 15.13 наведено інструкції впровадження віджета класу `Scale` в парі із менеджером розташування `pack`.

Лістинг 15.13 – Впровадження віджету класу `Scale`

```
def size(val):
    label.config(font=("Times New Roman", val))

scale = Scale(root,
               orient=VERTICAL, length=200,
               from_=10, to=30,
               tickinterval=10, resolution=10,
               command=size)
scale.pack(side=LEFT)

label = Label(root, text='ФКРКМ', font=("Times New Roman", 14))
label.pack(pady=20)
```

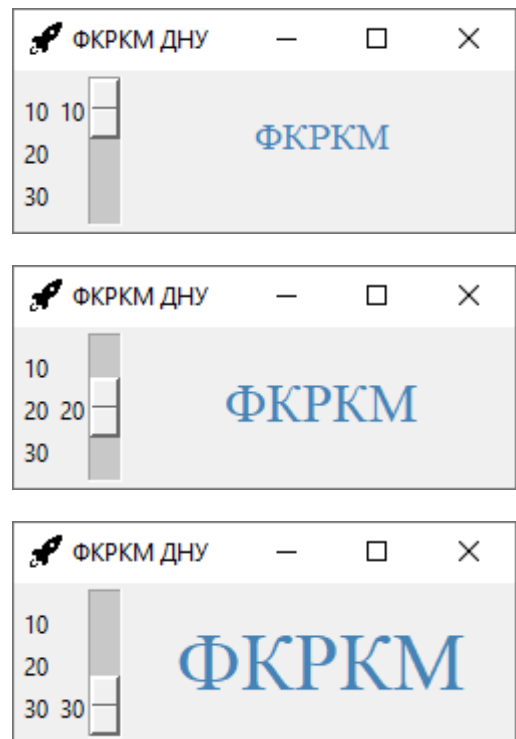


Рисунок 15.9

Компонент класу Menu



Menu — клас віджету, який застосовують для структуризації елементів керування.

На рисунку 15.10 наведено приклад впровадження об'єкта класу Menu.

Об'єкту класу Menu присвоюють ідентифікатор, який може бути обраний довільним, але традиційно присвоюють mainmenu. У якості аргументу вказують місце розташування віджета: основне вікно додатку — об'єкт root.

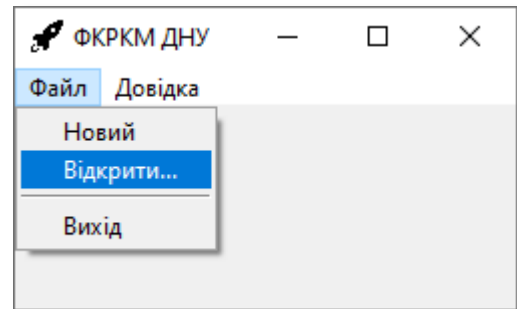


Рисунок 15.10

Принцип розробки конфігурації меню наведено у лістингу 15.14.

Лістинг 15.14 – Конфігурація та впровадження віджета класу Menu

```
mainmenu = Menu(root)
root.config(menu=mainmenu)

filemenu = Menu(mainmenu, tearoff=0)
filemenu.add_command(label="Новий")
filemenu.add_command(label="Відкрити...")
filemenu.add_separator()
filemenu.add_command(label="Вихід")

helpmenu = Menu(mainmenu, tearoff=0)
mainmenu.add_cascade(label="Файл", menu=filemenu)
mainmenu.add_cascade(label="Довідка", menu=helpmenu)
```

Конфігурація головного меню виконана в такий спосіб: об'єкт mainmenu класу Menu — головне меню додатку, об'єкти filemenu та helpmenu класу Menu являють собою підменю. Для додавання команд у відповідне меню використовується метод add_command. Метод add_separator залучають при намірі відокремити групи впроваджених команд. Метод add_cascade може бути задіяний для випадаючого меню.

Висновки

Отже, GUI є найпоширенішим способом взаємодії з програмами, який використовується в більшості сучасних операційних систем. Незважаючи на досить широкий вибір методів кастомізації, рекомендується в процесі розробки графічного інтерфейсу користувача дотримуватися естетичної єдності, намагатися реалізовувати ергономічну та інтуїтивно зрозумілу компоновку елементів керування програмних додатків. GUI зазвичай дозволяє користувачам взаємодіяти з програмою за допомогою інтуїтивно зрозумілих графічних елементів, що знижує кількість часу та зусиль, необхідних для навчання користувачів використовувати програму.

Візуальне відображення інформації та стан програмного додатку допомагає користувачам легше розуміти, що відбувається в процесі взаємодії та дозволяє виконувати багато задач швидше та ефективніше, ніж за допомогою командного рядка.

Питання та завдання до контролю знань студентів

- 1 Що таке бібліотека Tkinter Python та які є її основні функції?
- 2 Які є різниці між віджетами Label та Button у бібліотеці Tkinter Python?
- 3 Як можна створити вікно за допомогою бібліотеки Tkinter Python? Які є основні параметри вікна?
- 4 Які є основні віджети у бібліотеці Tkinter Python та як їх використовувати?
- 5 Як можна змінювати розміри віджетів у бібліотеці Tkinter Python? Які є способи вирівнювання віджетів?
- 6 Як можна додавати зображення та інші елементи до віджетів у бібліотеці Tkinter Python?
- 7 Як можна змінювати кольори та інші властивості віджетів у бібліотеці Tkinter Python?
- 8 Як можна створювати події та пов'язувати їх з віджетами у бібліотеці Tkinter Python?
- 9 Як можна використовувати розташування Grid та Pack у бібліотеці Tkinter Python для організації віджетів на вікні?
- 10 Як можна створювати меню та підменю у бібліотеці Tkinter Python? Які є основні параметри меню та підменю?

Лекція № 16

з навчальної дисципліни Інформатика

Тема Візуалізація даних засобами Matplotlib та NumPy

Мета заняття Оволодіти уміннями по використанню функцій та методів інтегрованих бібліотек Matplotlib та NumPy, систематизувати знання стосовно побудови графіків

Матеріально-технічне забезпечення та дидактичні засоби, ТЗН

1 Інтерактивний вид діяльності, створений засобами Moodle

2 Мутимедійний презентаційний матеріал

Час – 2 години

План проведення лекції

Структура лекції	Відведений час	Методичні вказівки
1 Організаційна частина	5 хв.	Включає в себе: <i>привітання</i> , яке має на меті привернути увагу, забезпечити контакт лектора з аудиторією; <i>визначення присутності студентів на занятті</i>
2 Актуалізація опорних знань, перевірка вивченого матеріалу та мотивація навчальної діяльності студентів	10 хв.	Включає в себе: <i>вступне зауваження, мотивацію актуальності теми, перевірку попереднього матеріалу, формулювання мети лекції, огляд головних питань теми (вказати питання та завдання для перевірки знань студентів)</i>
3 Основна частина (викладення навчальних питань лекції)	60 хв.	Головне місце приділяється <i>викладенню основного змісту матеріалу теми, його аналізу, узагальненню висунутих положень</i> . Для успіху лекції важливе значення має поділ матеріалу на розділи, основні питання
4 Заклучна частина Домашнє завдання: (відповідно до робочої програми)	5 хв.	Включає в себе: <i>узагальнення, теоретичні і фактичні висновки, закріплення вивченого на лекції матеріалу</i>

Література

1 Інформатика (профільний рівень) : підруч. для 10 кл. закл. заг. серед. освіти / В.Д. Руденко, Н.В. Речич, В.О. Потієнко. — Харків : Вид-во «Ранок», 2018.

2 Head First. Pythonб / Е. Фрімен, Е. Робсон, Б. Бейтс, П. Беррі, К. Сієрра (переклад укр. Г. Якубовська) — Харків : Вид-во «Фабула», 2021. – 624 с.

Навчальні матеріали лекції

ВСТУП

Візуалізація даних — це процес відображення даних у вигляді графічних елементів, таких як графіки, діаграми, гістограми, кругові діаграми та інші типи візуальних представлень. Візуалізація даних може бути корисною для розуміння інформації, що міститься у наборах даних, та для виявлення зв'язків та тенденцій, які можуть бути приховані у даних. Вона може бути корисна для комунікації результатів досліджень або аналізу, дозволяє відображати дані в більш доступний та зрозумілий спосіб, зокрема для тих, хто не має спеціалізованої освіти в цій галузі.

У загальному розумінні, візуалізація даних може бути розглянута як інструмент для розуміння даних, виявлення тенденцій та зв'язків між даними, та сприяти розвитку та підвищенню ефективності діяльності в різних сферах, включаючи бізнес, медицину, соціальні дослідження та багато інших.

ОГЛЯД ЗАСОБІВ ВІЗУАЛІЗАЦІЇ ДАНИХ ЗАСОБАМИ PYTHON

Візуалізація даних мовою програмування Python може бути реалізована через залучання бібліотеки Matplotlib. Зазначена бібліотека надає можливість створювати графіки, діаграми, гістограми, графіки розсіювання та інші типи графічних візуалізацій даних.

Matplotlib може бути використаний для реалізації таких завдань:

- відображення даних у вигляді графіків, діаграм та інших типів візуалізацій;
- побудова графіків різних типів, таких як лінійні, кругові, стовпчикові графіки, також графіки розсіювання тощо;
- налаштування візуального вигляду графіків, включаючи колір, шрифт, стиль ліній та маркерів, межі та наповнення дільниць;
- додавання підписів, легенд та анотацій до графіків;
- розширення можливостей Matplotlib за допомогою додаткових пакетів, таких як Seaborn, що надають більш складні типи візуалізацій даних.

Бібліотека Matplotlib працює у тандемі з бібліотекою NumPy (Numeric Python), яка, у свою чергу, є однією з основних бібліотек для наукового обчислення у мові програмування Python. Її основна мета — пропонувати ефективні маніпуляції з числовими даними у вигляді масивів та матриць.

Деякі з основних функцій бібліотеки NumPy:

- підтримка многовимірних масивів: NumPy дозволяє створювати та працювати з масивами з будь-якою кількістю вимірів, що можуть містити числа, рядки або булеві значення;
- універсальні функції: NumPy містить багато вбудованих функцій, що можуть бути застосовані до масивів, таких як математичні функції, операції з логікою, статистичні операції тощо;
- робота з матрицями: NumPy має розширені можливості для обробки матриць, включаючи матричне множення, знаходження визначника, оберненої матриці та інші;

- широкі можливості для обробки та аналізу даних: Numpy містить багато функцій для обробки та аналізу даних, таких як сортування, фільтрація, обчислення статистичних показників, генерація псевдовипадкових чисел тощо;
- інтеграція з іншими бібліотеками: Numpy підтримує роботу з іншими бібліотеками наукового обчислення для Python, такими як Scipy, Pandas, Matplotlib та інші;
- таким чином, бібліотека Numpy є важливим інструментом для наукового обчислення, статистики, машинного навчання та інших областей, що вимагають ефективної обробки числових даних.

ВЗАЄМОДІЯ ІЗ БІБЛІОТЕКОЮ MATPLOTLIB

Імпорт модуля (бібліотеки) в тіло програми може бути здійснений у традиційний спосіб, проте, якщо бібліотека не впроваджена у проект віртуального оточення, то звертання до неї буде супроводжуватися повідомлення про помилку відповідно до рисунка 16.1.

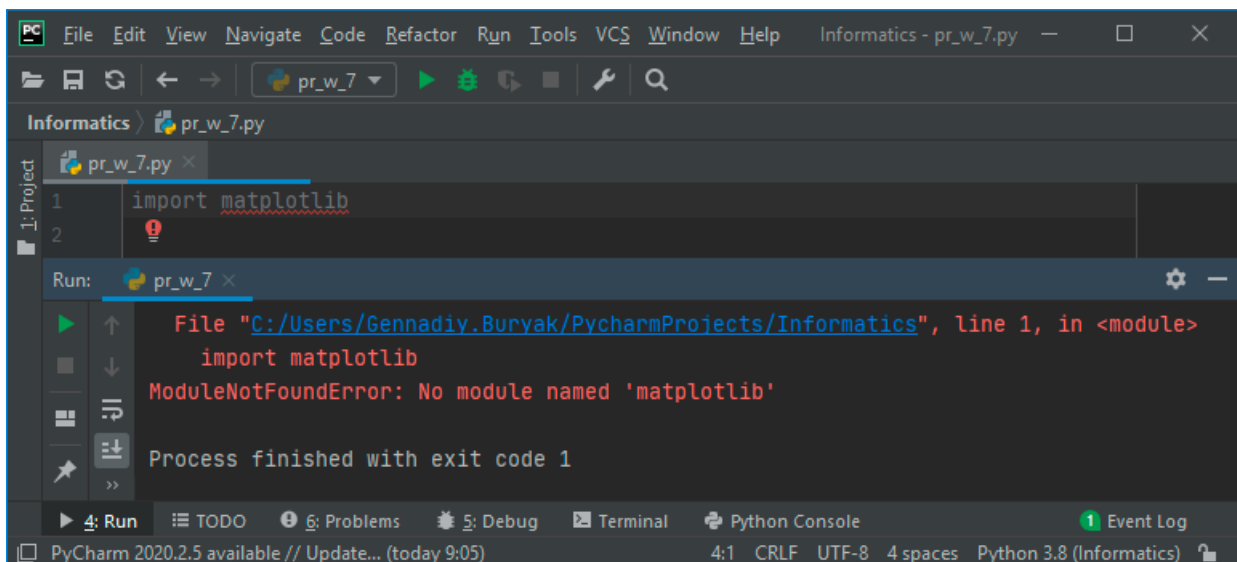


Рисунок 16.1 – Повідомлення про помилку «модуль не знайдено»

Інсталяція модуля (бібліотеки) Matplotlib може бути виконана у терміналі середовища розробки PyCharm у стандартний спосіб. В процесі інсталяції модуля Matplotlib автоматично буде встановлено модуль (бібліотека) Numpy (рисунок 16.2).

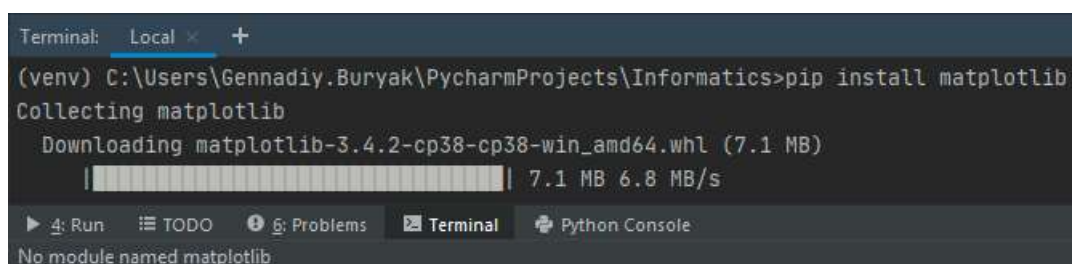


Рисунок 16.2 – Інсталяція модуля Matplotlib

Приклади застосування Matplotlib для візуалізації даних

Процес візуалізації даних засобами Matplotlib здійснюється через генерацію вікна «Figure...», із залучанням стандартного модуля Tkinter. У лістингу 16.1 наведено алгоритм генерації вікна «Figure 1» (рисунок 16.3).

Лістинг 16.1 – Генерація вікна для візуалізації даних

```
import matplotlib.pyplot as plt

fig = plt.figure()
axes = plt.subplot()

plt.show()
```

Вікно візуалізації даних містить інструменти взаємодії, що забезпечують здійснювати навігацію, аналіз, кастамізацію побудованих графіків, а також функціонал, який здатний здійснити даних, які було візуалізовано.

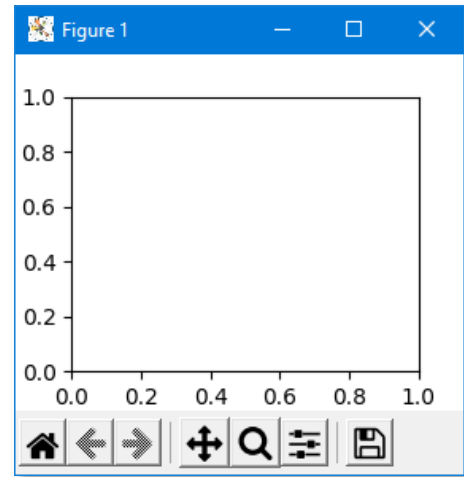


Рисунок 16.3 – Вікно візуалізації даних

На рисунку 16.4 наведено приклад візуалізації даних через побудову графіків відомих функцій, алгоритм реалізації яких наведено у лістингу 16.2.

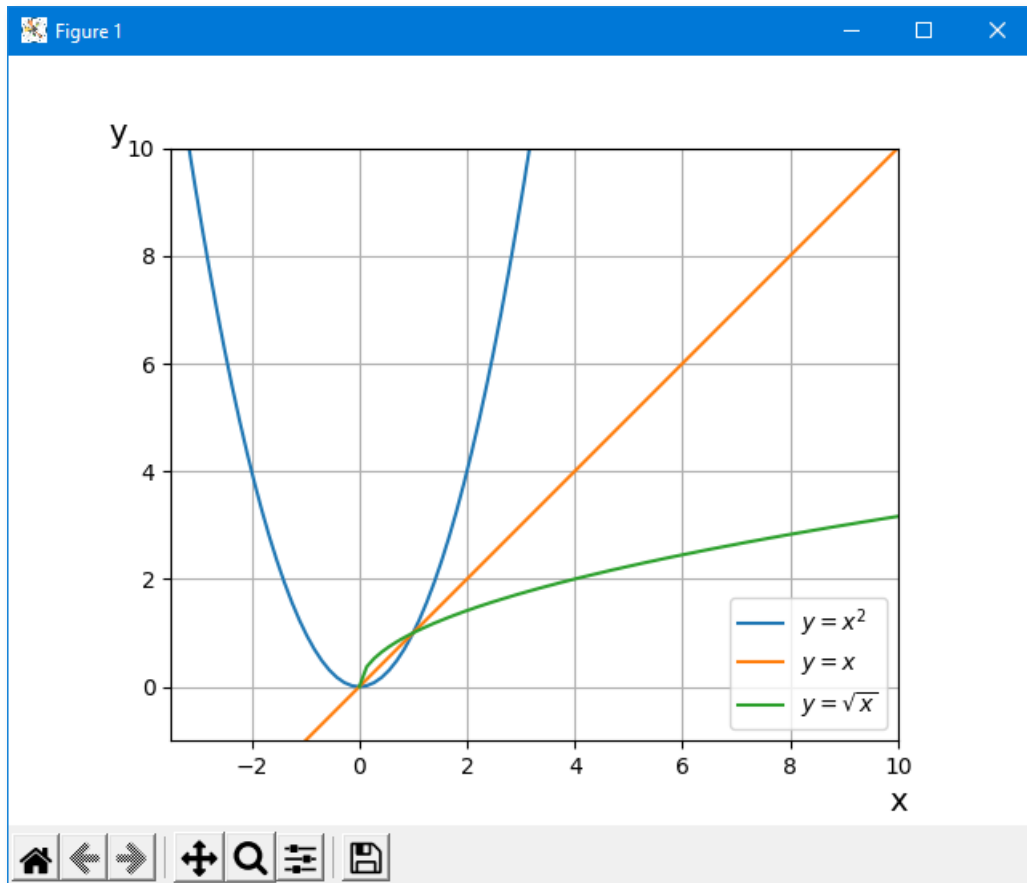


Рисунок 16.4 – Графіки відомих функцій

Лістинг 16.2 – Алгоритм побудови графіків функцій

```
import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(-10, 10, 145)

def parabola(arg):
    return pow(arg, 2)

def direct_proportion(arg):
    return arg

def sq_root(arg):
    with np.errstate(divide="ignore", invalid="ignore"):
        return np.sqrt(arg)

fig = plt.figure()

axes = plt.subplot()
axes.plot(x, parabola(x), label="line_1")
axes.plot(x, direct_proportion(x), label="line_2")
axes.plot(x, sq_root(x), label="line_3")

axes.legend([r"$y = x^2$", r"$y = x$", r"$y = \sqrt{x}$"], fontsize=10)

axes.set_xlim(-3.5, 10)
axes.set_ylim(-1, 10)

axes.set_xlabel("x", position=(1, 0), fontsize=14)
axes.set_ylabel("y", position=(0, 1), fontsize=14,
rotation="horizontal")

axes.set_aspect("equal")

axes.grid()
```

У вище показаному прикладі графіки функцій виконано в єдиному вікні візуалізації даних. Matplotlib має здатність створення графіків у окремих вікнах візуалізації даних, або здійснювати відображення кількох графіків в одному вікні візуалізації на окремих осях координат.

Створення окремих вікон візуалізації даних

Для створення окремих вікон у бібліотеці Matplotlib використовують об'єкти типу «Figure» та «Axes». Об'єкт «Figure» являє собою вікно, в якому відображається графік або діаграма. Об'єкт «Axes» являє собою ділянку вікна, в якій розміщується графік або діаграма.

Приклад використання бібліотеки Matplotlib для створення двох окремих вікон з графіками наведено у лістингу 16.3.

Лістинг 16.3 – Створення двох окремих вікон візуалізації даних

```
import matplotlib.pyplot as plt
import numpy as np

# Створюємо першу фігуру з графіком sin(x)
fig1 = plt.figure()
ax1 = fig1.add_subplot(111)
x = np.linspace(0, 2 * np.pi, 100)
y = np.sin(x)
ax1.plot(x, y)
ax1.set_title('Графік sin(x)')

# Створюємо другу фігуру з графіком cos(x)
fig2 = plt.figure()
ax2 = fig2.add_subplot(111)
y = np.cos(x)
ax2.plot(x, y)
ax2.set_title('Графік cos(x)')

# Показуємо обидві фігури
plt.show()
```

У цьому алгоритмі спочатку створюємо дві окремі фігури («fig1» та «fig2») за допомогою функції figure().

Далі здійснюється ініціалізація об'єктів «Axes» для кожної фігури за допомогою методу add_subplot().

Нарешті, створюємо графіки на кожному з об'єктів «Axes» та задаємо їх назви за допомогою методу set_title().

На останок ми показуємо обидві фігури за допомогою функції show().

Компоновка кількох графіків у одному вікні візуалізації даних

Matplotlib має кілька методів для компоновання кількох графіків в одному вікні "Figure".

Застосування функції subplots()

Цей метод компоновки створює багато графіків в одному вікні за допомогою однієї команди. Він повертає кортеж з двох елементів — об'єкту "Figure" та об'єкту "Axes" або масиву об'єктів "Axes", залежно від того, скільки графіків ви створили. Приклад компоновки двох графіків в одному вікні наведено у лістингу 16.4.

Лістинг 16.4 – Компоновка із залучанням функції subplots()

```
fig, axs = plt.subplots(1, 2)
axs[0].plot(x, y1)
axs[1].plot(x, y2)
```

Застосування функції subplot()

Цей метод компоновки дозволяє створити один графік у вікні та розбити його на підграфіки з допомогою координат. Він приймає три аргументи: число рядків, число стовпців та номер підграфіка, який необхідно створити. Приклад, створення графіків в одному вікні, можете бути реалізований відповідно до лістингу 16.5.

Лістинг 16.5 – Компоновка із залучанням функції subplot()

```
plt.subplot(1, 2, 1)
plt.plot(x, y1)

plt.subplot(1, 2, 2)
plt.plot(x, y2)
```

На рисунку 16.5 наведено відображення графіків лінійної функції, компоновка яких реалізовано у вікно візуалізації із залучанням функції subplot().

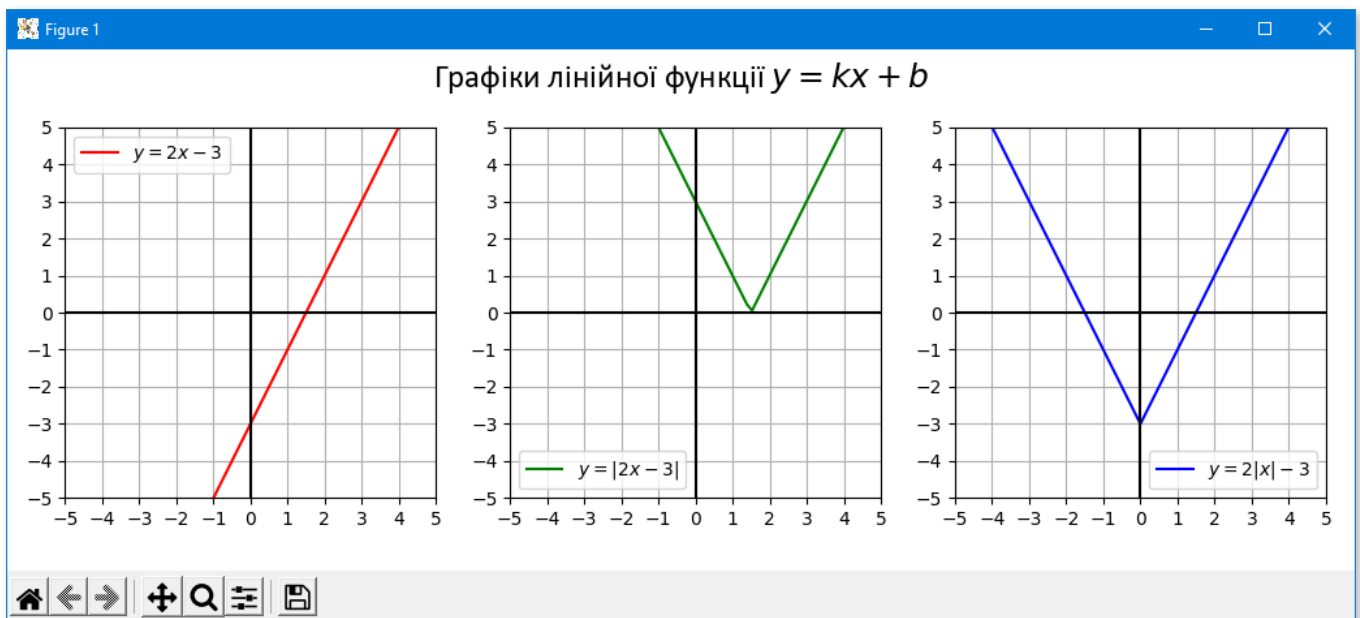


Рисунок 16.5 – Компоновка графіків із залучанням функції subplot()

Застосування GridSpec

Цей метод дозволяє створити складні макети з декількох графіків у вікні. Він дозволяє контролювати розташування та розміри кожного графіка окремо. Приклад по створенню двох графіків у вигляді одного вертикального стовпчика необхідно розробити інструкції відповідно до лістингу 16.6.

Лістинг 16.6 – Компоновка із залучанням функції subplot()

```
fig = plt.figure()
gs = fig.add_gridspec(2, 1)

ax1 = fig.add_subplot(gs[0, 0])
ax1.plot(x, y1)
ax2 = fig.add_subplot(gs[1, 0])
ax2.plot(x, y2)
```

На рисунку 16.5 показано компоновку вікна візуалізації із чотирма графіками.

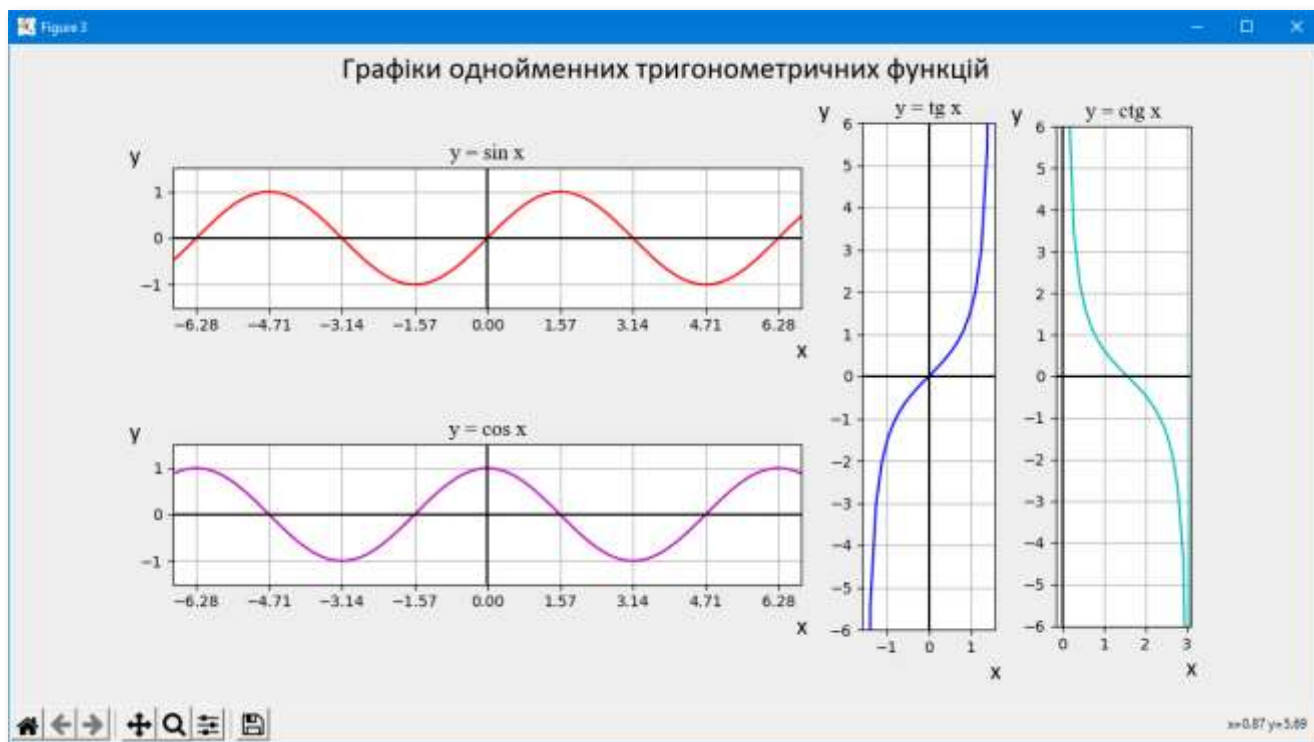


Рисунок 16.5 – Графіки однойменних тригонометричних функцій

На рисунку 16.6 наведено приклад компоновки показникових та логарифмічних функцій

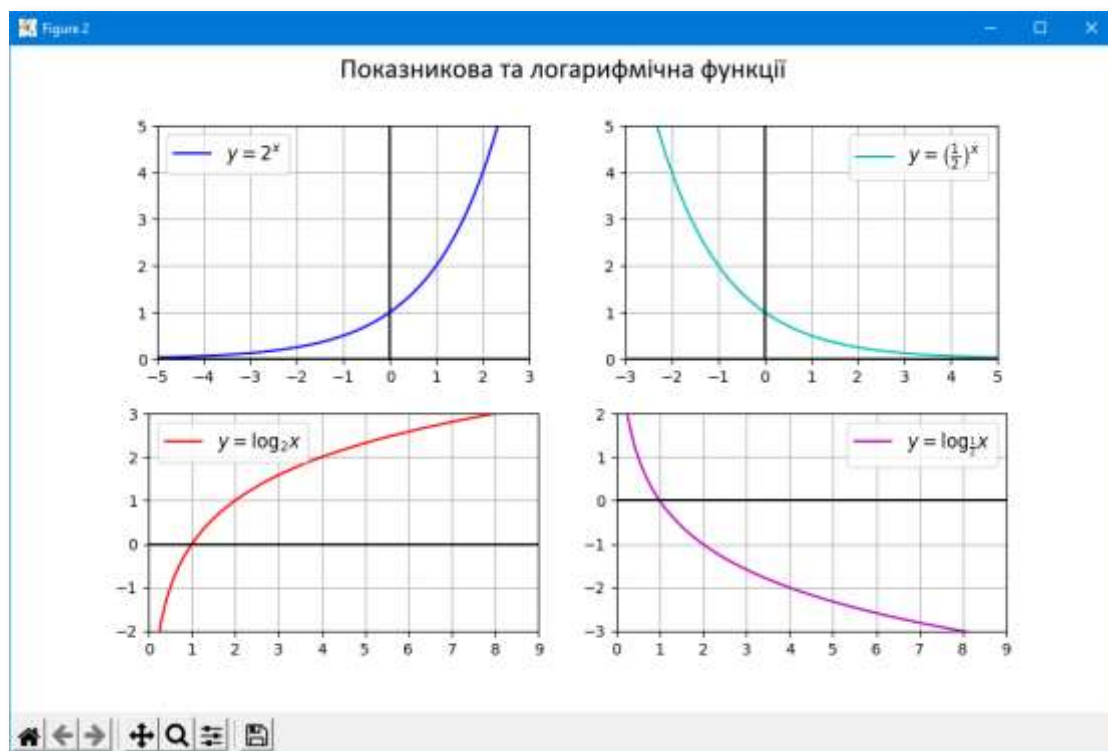


Рисунок 16.6 – Графіки показникових та логарифмічних функцій

Бібліотека Matplotlib оснащено засобами по організації візуалізації даних у вигляді діаграм. На рисунку 16.7 наведено приклад візуалізацію у вигляді стовбчастих діаграм.



Рисунок 16.7 – Візуалізація теоретичного розрахунку прибутку приватного підприємця

На рисунку 16.8 наведено приклад візуалізації даних у вигляді стовбчастих та кругових діаграм.

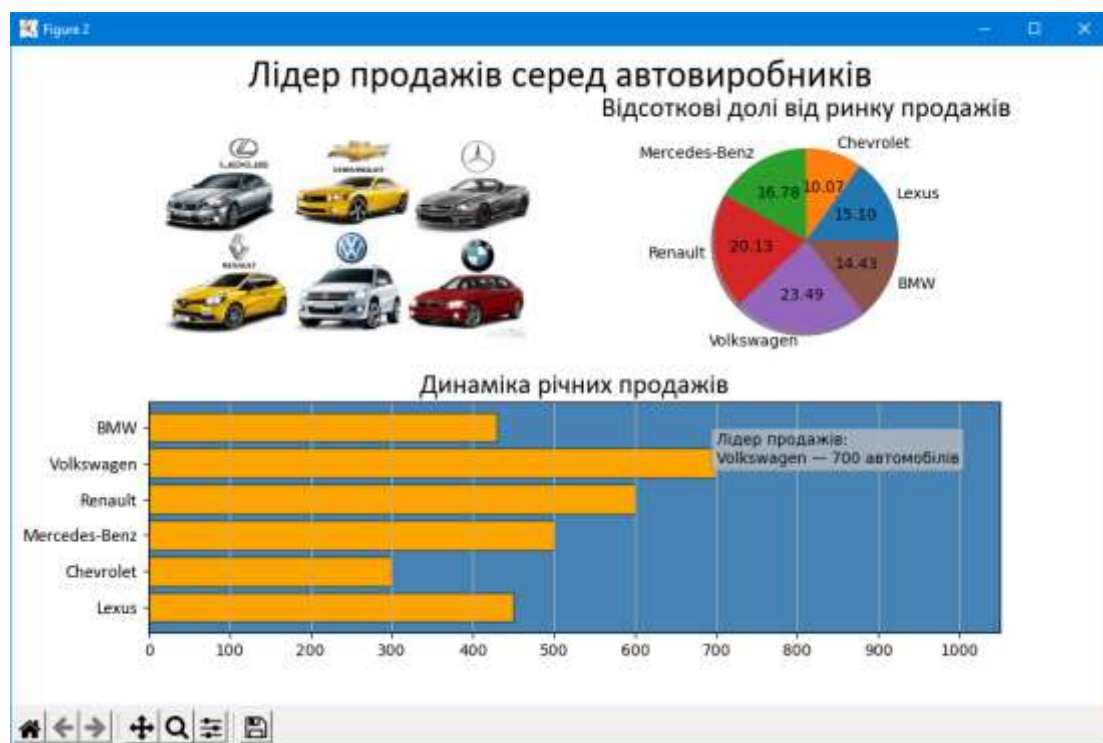


Рисунок 16.8 – Візуалізація даних для виявлення лідеру продажів

Висновки

Отже, Matplotlib — це потужний інструмент для відображення та аналізу даних у вигляді графічних візуалізацій, що робить його незамінним інструментом для дослідження даних та статистичного аналізу. Проте, для використання Matplotlib необхідний певний рівень досвіду, а також може знадобитися додатковий час на розробку складних графіків. Основною перевагою Matplotlib є його гнучкість та налаштовуваність. Користувач може контролювати кожен елемент графіка, від осей до тексту та кольорів. Крім того, бібліотека підтримує інтерактивну візуалізацію та вбудовану підтримку анімації. Однак, Matplotlib має деякі недоліки, зокрема, складність використання та дещо застарілий інтерфейс користувача, що може призвести до довгого часу розробки. Також, для виготовлення деяких складних графіків необхідно мати досить великий досвід у візуалізації даних.

Питання та завдання до контролю знань студентів

- 1 Які типи графіків можна створювати за допомогою Matplotlib?
- 2 Які інструменти Matplotlib підтримують інтерактивну візуалізацію даних?
- 3 Як налаштувати візуальний стиль графіків у Matplotlib?
- 4 Як використовувати Matplotlib для створення анімацій?
- 5 Які є недоліки Matplotlib та як їх можна уникнути?
- 6 Як створити кілька графіків у одному вікні "Figure" за допомогою Matplotlib?
- 7 Які методи Matplotlib дозволяють додавати елементи до графіків, такі як легенди, текст та анотації?
- 8 Як зберегти графіки, створені за допомогою Matplotlib, у різних форматах файлів?
- 9 Які основні структури даних в Numpy та як їх створювати?
- 10 Які функції в Numpy дозволяють виконувати математичні операції з масивами, такі як додавання, множення, віднімання та ділення?
- 11 Які методи Numpy дозволяють змінювати форму та розмір масивів?
- 12 Які функції Numpy дозволяють виконувати операції з масивами, такі як транспонування, вирівнювання, злиття та розбиття?
- 13 Які функції Numpy дозволяють обчислювати статистичні характеристики масивів, такі як середнє значення, медіана, мода та дисперсія?
- 14 Які методи Numpy дозволяють зчитувати та записувати масиви з файлів?
- 15 Які інструменти Numpy підтримують відлагодження та профілювання коду?

Лекція № 17

з навчальної дисципліни Інформатика

Тема Об'єктно-орієнтована парадигма у програмуванні

Мета заняття Ознайомити із принципами об'єктно-орієнтованого підходу у програмуванні, експериментально виявити доцільність застосування парадигми ООП, показати переваги спадкування, перевизначення методів

Матеріально-технічне забезпечення та дидактичні засоби, ТЗН

1 Інтерактивний вид діяльності, створений засобами Moodle

2 Мультимедійний презентаційний матеріал

Час – 2 години

План проведення лекції

Структура лекції	Відведений час	Методичні вказівки
1 Організаційна частина	5 хв.	Включає в себе: <i>привітання</i> , яке має на меті привернути увагу, забезпечити контакт лектора з аудиторією; <i>визначення присутності студентів на занятті</i>
2 Актуалізація опорних знань, перевірка вивченого матеріалу та мотивація навчальної діяльності студентів	10 хв.	Включає в себе: <i>вступне зауваження, мотивацію актуальності теми, перевірку попереднього матеріалу, формулювання мети лекції, огляд головних питань теми</i> (вказати питання та завдання для перевірки знань студентів)
3 Основна частина (викладення навчальних питань лекції)	60 хв.	Головне місце приділяється <i>викладенню основного змісту матеріалу теми, його аналізу, узагальненню висунутих положень</i> . Для успіху лекції важливе значення має поділ матеріалу на розділи, основні питання
4 Заключна частина Домашнє завдання: (відповідно до робочої програми)	5 хв.	Включає в себе: <i>узагальнення, теоретичні і фактичні висновки, закріплення вивченого на лекції матеріалу</i>

Література

1 Інформатика (профільний рівень) : підруч. для 11 кл. закл. заг. серед. освіти / В.Д. Руденко, Н.В. Речич, В.О. Потієнко. — Харків : Вид-во «Ранок», 2018.

2 Head First. Pythonб / Е. Фрімен, Е. Робсон, Б. Бейтс, П. Беррі, К. Сієрра (переклад укр. Г. Якубовська) — Харків : Вид-во «Фабула», 2021. – 624 с.

Навчальні матеріали лекції

ВСТУП

Основний дефект процедурного стилю програмування полягає в тому, що він слабо пристосований для розроблення великих програм. Із часом програми стають настільки складними, що розібратися в них непросто. Одним із напрямків розв'язування цієї проблеми є об'єктно-орієнтоване програмування (ООП).

ПРИЧИНИ ВИКОРИСТАННЯ ООП

Існує кілька причин, чому стало необхідним використовувати об'єктно-орієнтоване програмування.

Модульність та повторне використання коду

ООП дозволяє розділити програму на окремі модулі або класи, які можуть бути легко перевикористані в інших програмах. Це зменшує кількість коду, який потрібно написати, і зменшує час розробки.

ООП дозволяє приховати деталі реалізації класу від користувача. Це забезпечує безпеку програми і забезпечує більш простий інтерфейс користувача.

ООП дозволяє створювати нові класи, які наслідують властивості та методи інших класів. Це дозволяє зменшити кількість дублювання коду та спростити його розширення.

ООП дозволяє використовувати різні класи з однаковим інтерфейсом, що дозволяє розширювати функціональність програми, не змінюючи вже існуючий код.

ООП дозволяє більш легко відлагоджувати та підтримувати код, оскільки функціональність програми може бути розділена на окремі класи та модулі.

ОСНОВНІ ТЕРМІНИ ООП

Клас (Class) — це шаблон, який визначає структуру об'єктів. Клас містить дані (властивості) та функції (методи), які відображають поведінку цих об'єктів.

Класи є основними будівельними блоками ООП. Вони дозволяють визначити нові типи об'єктів, які можуть мати властивості та поведінку, що відрізняються від типів, які вже визначені в мові програмування.

Один клас може мати багато екземплярів, які представляють конкретні об'єкти цього класу. Кожен екземпляр класу має свої власні значення властивостей, які відрізняють його від інших екземплярів цього класу.

Наприклад, якщо ми розглядаємо клас "Автомобіль", то він може мати властивості, такі як "марка", "модель", "рік випуску", "колір" тощо, і методи, такі як "рухатися", "зупинятися", "бути ремонтним" тощо. Кожен екземпляр класу "Автомобіль" буде мати свої унікальні значення для цих властивостей.

Класи дозволяють створювати більш складні програми, оскільки вони дозволяють зібрати в одному місці дані та функції, які відносяться до конкретного

типу об'єкта. Крім того, класи забезпечують можливість наслідування, що дозволяє створювати нові класи на основі вже існуючих з певними змінами та доповненнями.

Метод (Method) — це функція, яка пов'язана з об'єктом певного класу. Методи визначають поведінку об'єктів класу і можуть виконувати різні дії з властивостями об'єкта.

Методи можуть бути публічними, приватними або захищеними. Публічні методи доступні для використання ззовні класу і можуть бути викликані будь-яким кодом. Приватні методи доступні тільки в межах класу і не можуть бути викликані ззовні. Захищені методи доступні тільки в межах класу та його нащадків і не можуть бути викликані ззовні.

Методи допомагають реалізувати концепцію інкапсуляції в ООП, оскільки вони забезпечують контроль над тим, як дані використовуються та модифікуються в межах класу. Крім того, методи забезпечують повторне використання коду, оскільки вони можуть бути використані в багатьох об'єктах класу.

Інтерфейс — це набір методів та властивостей, який використовується для взаємодії об'єктів між собою. Інтерфейс дозволяє розділити реалізацію та використання об'єктів.

Змінна класу (Class variable) — це змінна, яка визначена в класі і доступна всім екземплярам класу. Це означає, що змінна класу зберігає значення, яке спільне для всіх екземплярів класу.

Змінні класу використовуються для зберігання даних, які відносяться до всього класу, а не до окремих екземплярів. Наприклад, якщо ми створюємо клас "Автомобіль", то ми можемо визначити змінну класу "кількість автомобілів", яка буде зберігати загальну кількість автомобілів, які були створені з цим класом.

Змінні класу визначаються в тілі класу, зазвичай перед оголошенням методів. Вони можуть бути викликані в будь-якому методі класу, а також в зовнішньому коді за допомогою назви класу. Для звернення до змінної класу використовується синтаксис, подібний до звернення до атрибутів екземплярів класу, але з використанням імені класу замість імені екземпляру.

Наприклад, якщо змінна класу "кількість автомобілів" визначена в класі "Автомобіль", то до неї можна звернутися зовнішнім кодом за допомогою синтаксису "Auto.count_cars". Звернення до змінної класу через ім'я класу дозволяє змінювати її значення для всіх екземплярів класу, які мають доступ до цієї змінної.

Екземпляр класу (Instance) — це конкретний об'єкт, створений на основі класу. Він містить дані, що відповідають властивостям (атрибутам) класу, та поведінку, що відповідає методам класу.

Іншими словами, екземпляр класу — це об'єкт, який має властивості та методи, визначені в класі, з якого він був створений. Наприклад, якщо ми маємо клас "Книга", то екземпляр цього класу може мати властивості, такі як назва книги, автор, рік видання та інші. Крім того, він може мати методи, такі як "отримати інформацію про книгу" або "змінити рік видання книги".

Створення екземпляру класу відбувається за допомогою ключового слова "new" у більшості мов програмування. При створенні екземпляру класу виділяється пам'ять для збереження всіх властивостей та методів, визначених у класі. Кожен екземпляр класу має свій власний набір значень для всіх атрибутів класу, що дозволяє створювати багато об'єктів на основі одного класу з різними значеннями властивостей.

До екземплярів класу можна звертатися за допомогою змінних, які містять посилання на них. Звернення до атрибутів та методів екземпляру класу здійснюється за допомогою крапки (".") та назви властивості чи методу, наприклад "book.title" або "book.get_info()".

Об'єкт (Object) — це екземпляр класу, який містить дані та методи, що працюють з цими даними. Іншими словами, об'єкт є конкретним представником класу з унікальним набором властивостей та методів.

Кожен об'єкт має свій власний стан, що визначається значеннями його властивостей або змінними екземпляра класу. Об'єкти можуть взаємодіяти один з одним та зовнішніми системами шляхом виклику методів, що містяться в класі. Це дозволяє створювати програми, що описують поведінку та взаємодію об'єктів.

Крім того, кожен об'єкт має свій власний ідентифікатор, що дозволяє програмісту відрізнити один об'єкт від іншого. Об'єкти можуть бути створені динамічно під час виконання програми, що дозволяє програмістам створювати програми, які можуть адаптуватися до різних умов та змінювати свою поведінку на основі введення користувача або змінних оточення.

У загальному випадку, об'єкти є основними елементами ООП та вони дозволяють створювати програми з більшою чіткістю та зрозумілістю коду.

Змінна екземпляра класу (Instance variable) — це змінна, яка зберігає значення атрибуту для конкретного екземпляру класу. Кожен екземпляр класу має свій власний набір значень для всіх атрибутів класу, що дозволяє створювати багато об'єктів на основі одного класу з різними значеннями властивостей.

Наприклад, якщо ми маємо клас "Книга", то кожен екземпляр цього класу може мати власні значення атрибутів, таких як назва книги, автор, рік видання та інші. Для збереження цих значень використовуються змінні екземпляра класу.

Змінні екземпляра класу ініціалізуються під час створення нового екземпляру класу, за допомогою конструктора. У конструкторі можна вказати значення по замовчуванню для атрибутів класу, а також передати значення з зовнішнього середовища до змінної екземпляра. Після ініціалізації змінних екземпляра класу, до них можна звертатися з інших методів класу та зовнішнього коду.

Звернення до змінних екземпляра класу здійснюється за допомогою крапки (".") та назви змінної, наприклад "book.title" або "book.author". Змінні екземпляра класу можна змінювати під час роботи програми, що дозволяє змінювати значення властивостей об'єктів.

Наслідування — це процес створення нового класу на основі існуючого. Наслідуваний клас називається батьківським класом, а новий клас — нащадком. Нащадок успадковує всі властивості та методи батьківського класу, а також може додавати власні.

Поліморфізм — це можливість об'єктів різних класів використовувати однакові методи. Це дозволяє зменшити кількість коду та спростити програму, оскільки різні класи можуть виконувати однакову функцію за допомогою одного методу.

Абстракція — це процес виділення важливих характеристик об'єкту та ігнорування неважливих. Абстракція дозволяє зменшити складність програми та полегшити її розуміння.

СТВОРЕННЯ КЛАСІВ І ОБ'ЄКТІВ

Для створення класу мовою програмування Python використовується ключове слово `class`, за яким слідує назва класу та двокрапка. Після цього можуть бути описані атрибути та методи класу. Приклад створення простого класу `Person` з одним атрибутом `name` та методом `introduce()` наведено у лістингу 17.1.

Лістинг 17.1 – Створення класу `Person`

```
class Person:
    def __init__(self, name):
        self.name = name

    def introduce(self):
        print("Привіт, моє ім'я:", self.name)
```

У цьому прикладі метод `__init__()` є конструктором класу, який встановлює значення атрибута `name`. Ключове слово `self` вказує на поточний об'єкт класу, який буде створений при виклику конструктора.

Щоб створити об'єкт класу, достатньо викликати конструктор класу за допомогою імені класу. Наприклад, для створення об'єкту класу `Person` з ім'ям "Геннадій" можна використати наступний код показаний у лістингу 17.2.

Лістинг 17.2 – Створення об'єкту класу `Person`

```
person = Person("Геннадій")
person.introduce() # виведе "Привіт, моє ім'я: Геннадій "
print(person.name) # виведе "Геннадій"
```

КОНСТРУКТОР КЛАСУ

У мові Python є спеціальний метод `__init__` (зліва і справа по два знаки підкреслення), який називають **конструктором класу**.

Конструктор є спеціальним методом, який викликається автоматично при створенні нового об'єкту класу. Він дозволяє ініціалізувати атрибути об'єкта, задати початкові значення, а також виконати будь-які додаткові дії, які потрібні при створенні об'єкта. Метод `__init__` має параметр `self`, який вказує на об'єкт, який створюється. Він також може приймати додаткові параметри, які використовуються для ініціалізації атрибутів об'єкта. Основною функцією методу `__init__` є ініціалізація властивостей об'єкта за замовчуванням.

Основні функції методу `__init__`

Ініціалізація властивостей

Метод `__init__` може приймати аргументи та встановлювати початкові значення для атрибутів об'єкта.

Валідація даних

Метод `__init__` може перевіряти вхідні дані на валідність та відкидати невірні значення.

Виклик батьківського конструктора

При наслідуванні класу, метод `__init__` дочірнього класу може викликати конструктор батьківського класу за допомогою функції `super()`.

Виконання додаткових операцій

Метод `__init__` може виконувати будь-які додаткові операції, які потрібні для коректної ініціалізації об'єкта.

Наприклад, якщо в класі є атрибут `name`, то в конструкторі можна задати початкове значення цього атрибуту відповідно до лістингу 17.3.

Лістинг 17.3 – Впровадження початкового значення атрибуту

```
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age
```

В цьому прикладі конструктор класу `Person` приймає аргумент `name` і ініціалізує атрибут `name` значенням цього аргументу.

Конструктор може одночасно містити як параметри, яким не присвоюються значення, так і параметри, яким присвоюються відповідні значення. У такому разі параметри, яким не присвоюються значення, вказуються першими, а параметри, яким присвоюються значення — після них.

Крім ініціалізації атрибутів, метод `__init__` може виконувати будь-які додаткові дії, які потрібні при створенні об'єкта. Наприклад, можна вивести повідомлення про створення нового об'єкта відповідно до лістингу 17.4.

Лістинг 17.4 – Виведення повідомлення про створення нового об'єкта

```
class Person:
    def __init__(self, name):
        self.name = name
        print("New person created with name", name)
```

При створенні об'єкта цього класу буде виведене повідомлення про його створення.

НАСЛІДУВАННЯ В МОВІ PYTHON

Наслідування — це механізм в ООП, що дозволяє створювати нові класи на основі існуючих. Клас, який використовується для створення нового класу, називається базовим класом або батьківським класом, а клас, що створюється, називається похідним класом або дочірнім класом.

У мові Python наслідування реалізується за допомогою ключового слова `class`. Для того, щоб клас успадковувався від іншого класу, потрібно вказати базовий клас у визначенні похідного класу відповідно до лістингу 17.5.

Лістинг 17.5 – Реалізація успадкування класів

```
class BaseClass:
    # визначення базового класу

class DerivedClass(BaseClass):
    # визначення похідного класу
```

У цьому прикладі клас `DerivedClass` успадковує властивості та методи класу `BaseClass`.

При наслідуванні класу, похідний клас може змінювати або доповнювати властивості та методи батьківського класу. Також можна визначити власні методи та атрибути в похідному класі. У лістингу 17.6 наведено приклад успадкування властивостей та методів, а також визначання власних атрибутів та методів створеними класами.

Лістинг 17.6 – Реалізація успадкування властивостей та методів, а також визначення власних атрибутів та методів

```
class Animal:
    def __init__(self, name, species):
        self.name = name
        self.species = species

    def make_sound(self):
        pass

class Dog(Animal):
    def __init__(self, name, breed):
        super().__init__(name, species="Dog")
        self.breed = breed

    def make_sound(self):
        return "Woof!"

class Cat(Animal):
    def __init__(self, name):
        super().__init__(name, species="Cat")

    def make_sound(self):
        return "Meow"
```

У цьому прикладі клас `Animal` містить два атрибути `name` та `species`, а також метод `make_sound()`. Класи `Dog` та `Cat` успадковують ці властивості та методи від `Animal`, але також визначають власні атрибути та методи. В результаті можна створювати об'єкти з різними властивостями, в залежності від того, який клас використовується відповідно до лістингу 17.7.

Лістинг 17.6 – Створення об'єктів з різними властивостями

```
dog = Dog("Buddy", "Golden Retriever")
cat = Cat("Fluffy")
print(dog.name, dog.breed)
```

ПОЛІМОРФІЗМ

У мові Python поліморфізм можна реалізувати за допомогою двох основних методів: перевизначення методів і використання поліморфних функцій.

Перевизначення методів полягає в тому, що дочірні класи можуть перевизначити методи, що були у батьківському класі, і надати їм іншу реалізацію. Приклад реалізації пере визначення методів показаний у лістингу 17.7.

Лістинг 17.7 – Реалізація пере визначення методів

```
class Animal:
    def sound(self):
        print("The animal makes a sound.")

class Dog(Animal):
    def sound(self):
        print("The dog barks.")

class Cat(Animal):
    def sound(self):
        print("The cat meows.")

animal = Animal()
animal.sound() # виведе "The animal makes a sound."

dog = Dog()
dog.sound() # виведе "The dog barks."

cat = Cat()
cat.sound() # виведе "The cat meows."
```

У цьому прикладі клас `Animal` містить метод `sound()`, який виводить повідомлення про те, що тварина видає звук. Класи `Dog` і `Cat` успадковують властивості та методи класу `Animal`, але перевизначають метод `sound()`, щоб надати свою власну реалізацію звуку. При виклику методу `sound()` з об'єктів `Dog` і `Cat` будуть виводитись різні повідомлення.

Поліморфні функції — це функції, які можуть приймати об'єкти різних класів та виконувати різні операції, в залежності від типу об'єкта.

У лістингу 17.8 наведено приклад реалізації поліморфної функції.

Лістинг 17.7 – Реалізація поліморфної функції

```
class Square:
    def __init__(self, side):
        self.side = side

    def area(self):
        return self.side * self.side

class Circle:
    def __init__(self, radius):
        self.radius = radius

    def area(self):
        return 3.14 * self.radius * self.radius

def get_area(shape):
    return shape.area()

square = Square(5)
circle = Circle(3)

print(get_area(square)) # виведе 25
print(get_area(circle)) # виведе 28.26
```

У цьому прикладі функція `get_area()` може приймати об'єкти класів `Square` та `Circle`, тому що обидва класи мають метод `area()`, який повертає площу фігури.

ПЕРЕЗАВАНТАЖЕННЯ ОПЕРАТОРІВ

Сутність перезавантаження операторів полягає в тому, що за допомогою спеціальних методів, які інколи називають магічними, одні й ті самі оператори виконують різні дії над об'єктами. Ці методи, так само, як і метод `__init__`, на початку й наприкінці мають два знаки підкреслення. За аналогією з методом `__init__`, вони викликаються автоматично, коли екземпляр об'єкта бере участь у відповідній операції, наприклад, у операції додавання, множення тощо.

Магічні методи (іноді їх називають "dunder-методами") в мові Python — це спеціальні методи, які починаються та закінчуються двома підкресленнями. Вони дозволяють перевизначати стандартні оператори, такі як `+`, `-`, `*`, `/`, а також дозволяють виконувати різні дії під час створення, ініціалізації, знищення та інших подій пов'язаних з класами.

Ось кілька прикладів магічних методів, які можуть бути перевизначені у класах:

- `__init__` конструктор класу, який викликається при створенні нового об'єкта класу;
- `__str__` метод, який повертає рядок, який представляє об'єкт;
- `__repr__` метод, який повертає рядок, який може бути використаний для відтворення об'єкта;

- `__add__` метод, який виконує додавання двох об'єктів;
- `__sub__` метод, який виконує віднімання двох об'єктів;
- `__mul__` метод, який виконує множення двох об'єктів;
- `__div__` метод, який виконує ділення двох об'єктів;
- `__len__` метод, який повертає довжину об'єкту;
- `__eq__` метод, який перевіряє, чи рівні два об'єкти.

Приклад, який використовує декілька магічних методів наведено у лістингу 17.8.

Лістинг 17.8 — Реалізація класу `Vector`, який представляє вектор у двовимірному просторі

```
class Vector:
    def __init__(self, x, y):
        self.x = x
        self.y = y

    def __str__(self):
        return f"({self.x}, {self.y})"

    def __add__(self, other):
        return Vector(self.x + other.x, self.y + other.y)

    def __sub__(self, other):
        return Vector(self.x - other.x, self.y - other.y)

    def __mul__(self, scalar):
        return Vector(self.x * scalar, self.y * scalar)

    def __rmul__(self, scalar):
        return Vector(self.x * scalar, self.y * scalar)

    def __eq__(self, other):
        return self.x == other.x and self.y == other.y

v1 = Vector(1, 2)
v2 = Vector(3, 4)
print(v1) # виведе "(1, 2)"
print(v2) # виведе "(3, 4)"
v3 = v1 + v2
print(v3) # виведе "(4, 6)"
v4 = v2 - v1
print(v4) # виведе "(2, 2)"
v5 = v1 * 2
print(v5) # виведе "(2, 4)"
v6 = 3 * v2
print(v6) # виведе "(9, 12)"
print(v1 == v2) # виведе "False"
print(v1 == Vector(1, 2)) # виведе "True"
```

У цьому прикладі клас `Vector` представляє вектор у двовимірному просторі. Він містить магічні методи, які дозволяють виконувати операції додавання (`__add__`), віднімання (`__sub__`) та множення на скаляр (`__mul__`). Крім того, клас містить метод

`__rmul__`, який дозволяє множення вектора на скаляр зліва (тобто вираз $2 * v1$ буде коректно оброблений).

Також клас має метод `__eq__`, який перевіряє, чи рівні два вектори. Це дозволяє порівнювати об'єкти класу `Vector` за допомогою оператора `==`.

У лістингу 17.9 наведено приклад, у якому клас дозволяє виконувати операції додавання, віднімання, множення та ділення комплексних чисел, а також множення на скаляр та ділення на скаляр.

Лістинг 17.9 — Реалізація класу `ComplexNumber` для виконання алгебраїчних дій із комплексними числами

```
class ComplexNumber:
    def __init__(self, real=0, imag=0):
        self.real = real
        self.imag = imag

    def __str__(self):
        if self.imag >= 0:
            return f"{self.real}+{self.imag}j"
        else:
            return f"{self.real}{self.imag}j"

    def __add__(self, other):
        return ComplexNumber(self.real + other.real, self.imag +
                               other.imag)

    def __sub__(self, other):
        return ComplexNumber(self.real - other.real, self.imag -
                               other.imag)

    def __mul__(self, other):
        if isinstance(other, (int, float)):
            return ComplexNumber(self.real * other, self.imag * other)
        elif isinstance(other, ComplexNumber):
            real = self.real * other.real - self.imag * other.imag
            imag = self.real * other.imag + self.imag * other.real
            return ComplexNumber(real, imag)

    def __truediv__(self, other):
        if isinstance(other, (int, float)):
            return ComplexNumber(self.real / other, self.imag / other)
        elif isinstance(other, ComplexNumber):
            denominator = other.real**2 + other.imag**2
            real = (self.real * other.real + self.imag * other.imag) /
denominator
            imag = (self.imag * other.real - self.real * other.imag) /
denominator
            return ComplexNumber(real, imag)

    def __eq__(self, other):
        return self.real == other.real and self.imag == other.imag

a = ComplexNumber(1, 2)
b = ComplexNumber(3, 4)
```

```

c = ComplexNumber(1, -2)
d = ComplexNumber(0, 5)

print(a) # виведе "1+2j"
print(b) # виведе "3+4j"
print(c) # виведе "1-2j"
print(d) # виведе "5j"

e = a + b
print(e) # виведе "4+6j"

f = b - a
print(f) # виведе "2+2j"

g = a * b
print(g) # виведе "-5+10j"

h = b / c
print(h) # виведе "-0.4+1.8j"
print(a == b) # виведе "False"
print(a == c) # виведе "False"
print(a == ComplexNumber(1, 2)) # виведе "True"

```

У цьому прикладі клас `ComplexNumber` представляє комплексне число. Він містить магічні методи, які дозволяють виконувати операції додавання (`__add__`), віднімання (`__sub__`), множення (`__mul__`) та ділення (`__truediv__`) комплексних чисел, а також множення на скаляр (`__mul__`) та ділення на скаляр (`__truediv__`).

У лістингу 17.10 наведено приклад реалізації класу `Car` містяться магічні методи, які дозволяють виконувати порівняння.

Лістинг 17.10 — Реалізація класу `Car` які дозволяють виконувати порівняння

```

def __init__(self, make, model, year):
    self.make = make
    self.model = model
    self.year = year

def __str__(self):
    return f"{self.make} {self.model} ({self.year})"

def __eq__(self, other):
    return self.make == other.make and self.model == other.model and
self.year == other.year

def __lt__(self, other):
    if self.year < other.year:
        return True
    elif self.year > other.year:
        return False
    else:
        return self.make < other.make

car1 = Car("Toyota", "Corolla", 2022)
car2 = Car("Honda", "Civic", 2021)
car3 = Car("Toyota", "Camry", 2022)

```

```
print(car1) # виведе "Toyota Corolla (2022)"
print(car2) # виведе "Honda Civic (2021)"
print(car1 == car2) # виведе "False"
print(car1 == car3) # виведе "True"

cars = [car1, car2, car3]
sorted_cars = sorted(cars)
for car in sorted_cars:
    print(car)
```

У цьому прикладі клас Car представляє автомобіль. Він містить магічні методи, які дозволяють виконувати порівняння за допомогою операторів `==` (`__eq__`) та `<` (`__lt__`). Метод `__str__` використовується для отримання рядкового представлення об'єкту класу Car. Також показано використання функції `sorted`, яка сортує список автомобілів за роком випуску та маркою. Для цього потрібно визначити порядок відносності за допомогою методу `__lt__`. Автомобілі порівнюються за роком випуску. Якщо рік випуску однаковий, то автомобілі порівнюються за маркою.

Висновки

Застосування об'єктно-орієнтованого програмування (ООП) дозволяє створювати більш складні програми шляхом розділення коду на окремі об'єкти, які мають власні властивості та методи. ООП дозволяє створювати більш чистий та читабельний код. Класи та об'єкти можна створювати з такими властивостями та методами, які відображають сутність предметної області, що спрощує розуміння коду.

Загалом, застосування ООП на Python є дуже корисним та ефективним, особливо для створення більш складних програм. Це дає можливість покращити якість коду, спростити його розуміння та збільшити ефективність програми в цілому.

Питання та завдання до контролю знань студентів

- 1 Які основні принципи ООП використовуються в Python?
- 2 Які переваги має використання ООП в Python порівняно з іншими підходами?
- 3 Як визначити класи та об'єкти в Python?
- 4 Як використовувати наслідування класів у Python?
- 5 Які основні властивості та методи класів в Python?
- 6 Як використовувати поліморфізм у Python?
- 7 Які особливості використання ООП у Python потрібно враховувати при роботі з бібліотеками та фреймворками?
- 8 Як можна використовувати ООП для розв'язання конкретних задач у Python?
- 9 Як можна зберігати та обробляти дані у класах в Python?
- 10 Які основні виклики можуть виникати при розробці програм на Python з використанням ООП та як їх можна вирішувати?