

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет імені В. Н. Каразіна
Бахмутський навчально-науковий професійно-педагогічний інститут
Кафедра електромеханічних та комп'ютерних систем

До захисту допущено

Завідувач кафедри


(підпис)

Інна НЕФЬОДОВА
(ім'я, прізвище)

« 07 » _____ грудня 2024 року

КВАЛІФІКАЦІЙНА РОБОТА (ПРОЄКТ)

рівень вищої освіти _____ другий (магістерський) _____

спеціальність _____ 015.39 Професійна освіта (Цифрові технології) _____

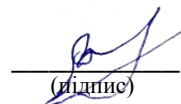
освітньо-професійна програма Професійна освіта. Комп'ютерні технології в управлінні та навчанні _____

тема Професійна підготовка фахівців у галузі цифрових технологій до викладання освітнього модулю «Основи програмування мовою Python» _____

Виконав(ла)

здобувач(ка) групи БЗ-К23мг
(шифр групи)

Ілля ТОЛКАЧОВ
(ім'я, прізвище)


(підпис)

Керівник роботи

к.ф.-м.н., доц. Інна НЕФЬОДОВА
(науковий ступінь, вчене звання, ім'я, прізвище)


(підпис)

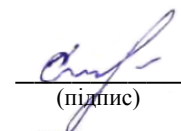
Рецензент роботи


д.ф.-м.н., проф. Олег ЛИТВИН
(науковий ступінь, вчене звання, ім'я, прізвище)


(підпис)

Консультант

д.пед.н., проф. Вікторія КУЛЕШОВА
(науковий ступінь, вчене звання, ім'я, прізвище)


(підпис)

Засвідчую, що у цій роботі немає цитат та вилучень з праць інших авторів без відповідних посилань
здобувач (ка) _____ 
(підпис)

Харківський національний університет імені В. Н. Каразіна

Факультет/ІНІ Бахмутський навчально-науковий професійно-педагогічний інститут

Кафедра Електромеханічних та комп'ютерних систем

Рівень вищої освіти другий (магістерський)

Спеціальність 015.39 Професійна освіта (Цифрові технології)

Освітньо-професійна програма Професійна освіта. Комп'ютерні технології в управлінні та навчанні

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інна НЕФЬОДОВА

(ім'я, прізвище)



(підпис)

« 08 » жовтня 2024 року

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ (ПРОЄКТ)

Толкачов Ілля Ігорович

(прізвище, ім'я, по батькові здобувача)

1. Тема роботи Професійна підготовка фахівців у галузі цифрових технологій до викладання освітнього модулю «Основи програмування мовою Python»

керівник роботи Нефьодова Інна Віталіївна, к. ф.-м. н., доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «08» жовтня 2024 року № 5101-5/3263

2. Строк подання здобувачем роботи «02» грудня 2024 р.

3. Перелік питань, які потрібно розробити: Актуальність професійної підготовки фахівців у галузі цифрових технологій. Характеристика об'єктів галузі: стан і стратегії розвитку. Вимоги до кадрового забезпечення об'єкту галузі. Методика професійної підготовки фахівців з цифрових технологій. Дидактичний проект консультативного заняття з теми «Основи програмування мовою Python» дисципліни «Інформатика та комп'ютерні технології» для здобувачів освіти спеціальності Професійна освіта (Цифрові технології).

4. План роботи

№ з/п	Назви етапів роботи
1	Огляд літературних джерел, нових розробок, опублікованих даних та іншої інформації, пов'язаної з темою роботи
2	Дослідження теоретичних підходів до актуальності професійної підготовки фахівців у галузі цифрових технологій до викладання освітнього модулю «Основи програмування мовою Python»
3	Характеристика об'єктів галузі: стан і стратегії розвитку
4	Розробка методики професійної підготовки фахівців у галузі цифрових технологій до викладання освітнього модулю «Основи програмування мовою Python»
5	Розробка вимог до кадрового забезпечення об'єкту галузі
6	Оформлення першого варіанту тексту, подання його на ознайомлення науковому керівнику
7	Усунення недоліків, написання остаточного варіанту тексту, оформлення дипломної роботи
8	Подання роботи на кафедру, перевірка на плагіат та зовнішнє рецензування роботи
9	Захист дипломної роботи у ЕК

5. Дата видачі завдання «08» жовтня 2024 р.

Здобувач(ка)



(підпис)

Ілля ТОЛКАЧОВ

(ім'я, прізвище)

Керівник роботи



(підпис)

Інна НЕФЬОДОВА

(ім'я, прізвище)

РЕФЕРАТ

Об'єкт дослідження є процес професійної підготовки здобувачів освіти.

Предметом дослідження є методика професійної підготовки фахівців з цифрових технологій для викладання освітнього модуля «Основи програмування мовою Python».

Мета дослідження – теоретично обґрунтувати та експериментально перевірити методику професійної підготовки фахівців з цифрових технологій для викладання освітнього модуля «Основи програмування мовою Python» у закладах вищої освіти.

В результаті виконання дослідження три лабораторні роботи «Програмування алгоритмів циклічної структури», «Проектування алгоритмів з використанням одновимірних масивів даних», «Опрацювання двовимірних масивів даних» та підібрано варіанти завдань для самостійного виконання.

За основними результатами дослідження виконана публікація тез доповіді на VIII Міжнародній науково-практичній конференції здобувачів вищої освіти та молодих учених «Студенти та молодь – для майбутнього країни» (м. Харків, 14-15 листопада 2024 р.).

Обсяг дипломної роботи становить: пояснювальна записка, презентація доповіді. Пояснювальна записка складається зі вступу, чотирьох розділів, висновків, списку використаних джерел, додатків. Загальний обсяг роботи 76 сторінок, з яких 64 сторінки основного тексту. Список використаних джерел становить 30 найменування, 5 таблиць, 40 рисунків.

ЦИФРОВІ ТЕХНОЛОГІЇ, ПРОФЕСІЙНА ПІДГОТОВКА, ОСВІТНІЙ МОДУЛЬ, ПРОГРАМУВАННЯ PУTHON, АЛГОРИТМИ, МАСИВИ ДАНИХ.

ABSTRACT

The object of the study is the process of professional training of students.

The subject of the study is the methodology of professional training of specialists in digital technologies for teaching the educational module «Fundamentals of Programming in Python».

The purpose of the study is to theoretically substantiate and experimentally verify the methodology of professional training of digital technology specialists for teaching the educational module «Fundamentals of Programming in Python» in higher education institutions.

As a result of the research, three laboratory works «Programming Algorithms of Cyclic Structure», «Designing Algorithms Using One-Dimensional Data Arrays», «Processing Two-Dimensional Data Arrays» were completed and options for independent tasks were selected.

Based on the main results of the research, the abstracts of the report were published at the VIII International Scientific and Practical Conference of Higher Education Students and Young Scientists «Students and Youth – for the Future of the Country» (Kharkiv, November 14-15, 2024).

The thesis consists of an explanatory note and a presentation of the report. The explanatory note consists of an introduction, four chapters, conclusions, a list of references, and appendices. The total volume of the work is 76 pages, of which 64 pages are the main text. The list of sources used consists of 30 items, 5 tables, and 40 figures.

**DIGITAL TECHNOLOGIES, PROFESSIONAL TRAINING,
EDUCATIONAL MODULE, PYTHON PROGRAMMING, ALGORITHMS,
DATA ARRAYS.**

ЗМІСТ

Вступ.....	7
Розділ 1 Актуальність професійної підготовки фахівців у галузі цифрових технологій	12
Розділ 2 Характеристика об'єктів галузі: стан і стратегії розвитку.....	16
2.1 Характеристика та аналіз нормативно-методичної документації з дисципліни «Інформатика та комп'ютерні технології».....	16
2.2 Презентації до лекційних занять	18
2.3 Лабораторна робота на тему «Програмування алгоритмів циклічної структури».....	24
2.4 Лабораторна робота на тему «Проектування алгоритмів з використанням одновимірних масивів даних».....	33
2.5 Лабораторна робота на тему «Опрацювання двовимірних масивів даних».....	40
Розділ 3 Вимоги до кадрового забезпечення об'єкту галузі	0
Розділ 4 Методика професійної підготовки фахівців з цифрових технологій. Дидактичний проект консультативного заняття з теми «Основи програмування мовою Python» дисципліни «Інформатика та комп'ютерні технології» для здобувачів освіти спеціальності Професійна освіта (Цифрові технології).....	53
Висновки.....	62
Список використаних джерел.....	65
Додаток А	69
Додаток Б	74

ВСТУП

В даний час в Україні відбувається інтенсивне реформування вищої освіти, що дозволяє забезпечити формування готовності здобувачів освіти до майбутньої професійної діяльності.

Реформування освіти стає ключовим завданням української освітньої політики, особливо в умовах воєнного стану, де опинилась наша держава.

Змінюються пріоритети, переосмислюються основні концепти ЗВО.

Все більше вітчизняних дослідників приходять до висновку, що необхідно змінювати освітні стандарти, пред'являти нові вимоги до змісту підготовки здобувачів освіти, проводити переорієнтацію оцінки результатів, посилити потребу в інженерах-педагогах з цифрових технологій, які здатні вирішувати професійні завдання.

А тому як основна мета, що нами була поставлена: реалізація діяльності закладу вищої освіти та підготовка висококваліфікованого, компетентного, затребуваного на ринку праці фахівця, готового до самовдосконалення, застосування навичок комунікації, відповідальності, дисциплінованості та професійного зростання в галузі інженерно-педагогічної діяльності. Особливо це є затребуваним в умовах цифровізації, необхідності майбутньому інженеру-педагогу з цифрових технологій реалізовувати професійні завдання у закладах професійно-технічної освіти.

Зазначені обставини підтверджують важливість проблеми формування готовності майбутніх фахівців з цифрових технологій до майбутньої професійної діяльності; висувають високі вимоги до компетентності та майстерності випускників Бахмутського навчально-наукового професійно-педагогічного інституту Харківського національного університету імені В. Н. Каразіна.

Ця мета як актуальна проблема встановила, що компетентність фахівців з цифрових технологій залежить від інженерів-педагогів з цифрових технологій.

Виявлено, що професійна підготовка інженерів-педагогів з цифрових технологій включає всі сторони розвитку особистості: здібності, цілеспрямованість, наполегливість у досягненні результатів навчання; завдяки формуванню культури формується соціальний досвід, розвиваються моральні сторони особистості.

Отже, нашій державі вкрай потрібні фахівці з цифрових технологій для викладання освітніх модулів. І це дійсно є нагальною потребою. Саме такими є інженери-педагоги зі спеціальності 015.39 Професійна освіта (Цифрові технології).

Аналіз наукових праць поставив перед країною важливу проблему: підготовки фахівців, майбутніх інженерів-педагогів з цифрових технологій, що володіють високим професіоналізмом.

Відповідно в процесі всього терміну навчання у ЗВО необхідно здійснювати формування суб'єкта, який буде готовий у майбутньому займатися професійною діяльністю з реалізації цифрових технологій.

Незважаючи на значну увагу дослідників до формування готовності студентів з цифрових технологій до майбутньої професійної діяльності, дана проблема залишається недостатньо вивченою.

Розглядають застосування освітнього процесу до нових цифрових реалій В.Биков, О. Беспарточна, І. Богданов, Р.Горбатюк, Р. Гуревич, Дж. Ендерсон, О. Коношевський, В. Хоменко, М. Хорн. Учені досліджують, як цифрові технології можуть впливати на когнітивні, емоційні та соціальні аспекти навчання.

О. Глазунова, М. Жалдак, Н. Морзе, Ю.Триус, О. Щербак акцентують увагу на методичних аспектах використання цифрових інструментів у професійній діяльності, зокрема у підготовці майбутніх педагогів до використання технологій у своїй роботі. Ці дослідження виявляють основний до розвитку цифрової компетентності.

Аналіз психолого-педагогічної літератури переконав, що проблема професійної підготовки фахівців з цифрових технологій для викладання

освітнього модуля «Основи програмування мовою Python» у закладах вищої освіти, а також вирішення низки суперечностей, а саме між:

– необхідністю аналізу, реформування навчального процесу та забезпечення якості формування готовності здобувачів освіти БНППІ та недостатньою розробленістю у ЗВО теорії та методики здійснення цього процесу;

– потребою студентів до набуття знань, умінь, навичок з формування професійно важливих і особистісно-значущих якостей, що необхідні у професійної діяльності в галузі інженерно-педагогічної освіти та недостатнім рівнем вирішення цієї проблеми у навчально-виховному процесі ЗВО.

Важливість вирішення зазначених суперечностей і зумовили вибір теми кваліфікаційної роботи.

Отже, актуальність, об'єктивна потреба закладів вищої освіти в пошуках нових підходів до змісту навчання в інженерно-педагогічному закладі вищої освіти, недостатня розробленість проблеми зумовили вибір теми дослідження: «Професійна підготовка фахівців у галузі цифрових технологій до викладання освітнього модуля «Основи програмування мовою Python»».

Мета дослідження – теоретично обґрунтувати та експериментально перевірити методику професійної підготовки фахівців з цифрових технологій для викладання освітнього модуля «Основи програмування мовою Python» у закладах вищої освіти.

Завдання дослідження:

1. Визначити ступінь актуальності проблеми професійної підготовки здобувачів освіти зі спеціальності 015.39 Професійна освіта (Цифрові технології).

2. Проаналізувати навчально-методичне забезпечення для викладання освітнього модуля «Основи програмування мовою Python» дисципліни «Інформатика та комп'ютерні технології».

3. Підготувати презентації та лабораторні роботи для викладання освітнього модуля «Основи програмування мовою Python».

4. Теоретично обґрунтувати, розробити перевірити методику професійної підготовки фахівців з цифрових технологій для викладання освітнього модуля «Основи програмування мовою Python».

Об'єкт дослідження: процес професійної підготовки здобувачів освіти.

Предмет дослідження: методика професійної підготовки фахівців з цифрових технологій для викладання освітнього модуля «Основи програмування мовою Python».

Методи дослідження:

– загальнонаукові (аналіз, синтез, систематизація, зіставлення, узагальнення) з метою професійної підготовки фахівців з цифрових технологій.

– емпіричні (тестування, опитування);

– педагогічний експеримент з метою перевірки методики професійної підготовки фахівців з цифрових технологій.

Наукова новизна одержаних результатів дослідження:

вперше: – уточнено поняття «підготовка інженерів-педагогів у галузі цифрових технологій до реалізації функцій професійної діяльності», яке розглядається як безперервний, цілеспрямований процес з виявлення відповідності сформованої педагогічної практики сучасним вимогам, усунення виявлених професійних дефіцитів, результатом якого є підвищення рівня готовності до реалізації професійної діяльності у сучасних умовах. Під професійними дефіцитами інженерів-педагогів ми розуміємо неузгодженість між існуючим і необхідним рівнем знань, умінь і способів діяльності фахівців з цифрових технологій, що призводить до невідповідності сформованого педагогічного досвіду сучасних вимог;

– розроблено діагностичний інструментарій, у тому числі рефлексивний, що включає авторську методику, що дозволяє вивчити рівень готовності інженерів-педагогів до реалізації професійної діяльності в

сучасних умовах (низький, середній та високий) за компонентами готовності до реалізації професійної діяльності ціннісно-змістовий (усвідомлюю значущість, цінність інженерно-педагогічної діяльності в сучасних умовах), когнітивний (знаю, вмю), операційно-діяльнісний (застосовую, роблю)) та виявити професійні дефіцити інженерів-педагогів як основу адресної підготовки.

подальшого розвитку набули зміст та засоби професійної перепідготовки інженера-педагога з цифрових технологій.

Теоретичне та практичне значення одержаних результатів полягає в обґрунтуванні методики професійної підготовки фахівців з цифрових технологій для викладання освітнього модуля «Основи програмування мовою Python» у закладах вищої освіти засобами інноваційних технологій.

Практична значущість дослідження полягає у розробці та реалізації освітнього модулю «Основи програмування мовою Python» дисципліни «Інформатика та комп'ютерні технології», що забезпечує підвищення рівня професіоналізму майбутніх фахівців в галузі інформаційних технологій та викладання у ЗВО.

Матеріали дослідження використовувались при підготовці здобувачів вищої освіти Бахмутського Навчально-наукового професійно-педагогічного інституту Харківського національного університету імені В. Н. Каразіна зі спеціальності 015.39 Професійна освіта (Цифрові технології); удосконалено зміст дисциплін «Методика професійного навчання», «Інформатика та комп'ютерні технології».

Апробація результатів дослідження: за основними результатами дослідження виконана публікація тез доповіді на VIII Міжнародній науково-практичній конференції здобувачів вищої освіти та молодих учених «Студенти та молодь – для майбутнього країни» (м. Харків, 14-15 листопада 2024 р.).

Структура роботи. Робота складається із вступу, чотирьох розділів, висновків, списку використаних джерел та додатків.

РОЗДІЛ 1

АКТУАЛЬНІСТЬ ПРОФЕСІЙНОЇ ПІДГОТОВКИ ФАХІВЦІВ У ГАЛУЗІ ЦИФРОВИХ ТЕХНОЛОГІЙ

Аналіз психолого-педагогічної літератури з проблем підготовки інженерів-педагогів до реалізації професійної діяльності, відповідності практики підготовки інженерів-педагогів з цифрових технологій до реалізації професійної діяльності сучасним вимогам до педагогічної діяльності, визначені проблеми даного процесу; уточнено поняття «підготовка інженерів-педагогів з цифрових технологій до реалізації професійної діяльності в сучасних умовах», «професійні дефіцити інженерів-педагогів», визначено та обґрунтовано компоненти та показники готовності інженерів-педагогів до реалізації професійної діяльності в сучасній змінних умовах; виявлено та обґрунтовано критеріально-оцінювальні процедури вивчення готовності інженерів-педагогів до реалізації професійної діяльності у сучасних умовах, визначено рівні їхньої готовності.

Аналіз наукових публікацій показав, що працюючі інженери-педагоги, як і молоді фахівці зазнають серйозних труднощів через зміни в професійній діяльності та організації педагогічної діяльності в сучасних умовах.

Труднощі виражаються у невідповідності між сформованим педагогічним досвідом і сучасними вимогами до реалізації професійних функцій діяльності інженера-педагога.

Підтримуємо погляди, де акцентується на підготовці вже працюючих інженерів-педагогів до виконання нових професійних функцій, розглядаючи її як пошукову систему, спрямовану на забезпечення цілісності теоретичної та практичної готовності до професійної педагогічної діяльності, визначальну потребу коригування на основі виявлення професійних труднощів (дефіцитів) у педагогічній практиці [11, С. 7].

Таким чином, аналіз та узагальнення праць, присвячених розгляду поняття «підготовка інженерів-педагогів з цифрових технологій до реалізації

функцій професійної діяльності», дозволяє нам уявити його як безперервний, цілеспрямований процес щодо виявлення відповідності сформованої педагогічної практики сучасної ним вимогам, усунення виявлених професійних дефіцитів, результатом якого є підвищення рівня готовності до реалізації професійної діяльності у сучасних умовах. При цьому професійні дефіцити інженерів-педагогів ми розглядаємо як неузгодженість між існуючим і необхідним рівнем знань, умінь і способів діяльності інженера-педагога з цифрових технологій, що призводить до її відповідності сформованого педагогічного досвіду сучасних вимог [29, С. 174].

Незважаючи на те, що питання підготовки інженерів-педагогів з цифрових технологій до професійної діяльності висвітлюються у дослідженнях, проблема їхньої готовності до роботи в сучасних умовах у закладах професійно-технічної освіти залишається досить актуальною.

Так, одні вчені розуміють «готовність» як психологічну установку, на якій ґрунтується діяльність, інші вважають, що поняття «готовність» збігається з соціальною установкою, що визначає поведінку особистості, треті вважають, що поняття готовність тотожні підготовленості. При цьому всі дослідники сходяться на тому, що в змісті готовності особистості до професійної діяльності виділяються як діяльнісна, і психологічна складові. У зв'язку з цим можна виділити діяльнісний та особистісний підходи визначення готовності до професійної діяльності [21, С. 113].

У межах функціонального підходу готовність розуміється як сукупність певних знань, що необхідні для ефективного здійснення діяльності.

Відносно інженерно-педагогічної діяльності, готовність - сформовані на основі професійних знань умінь і навичок, що сприяють успішному включенню в дану діяльність.

У межах особистісного аспекту готовність визначаємо як складну, стійку інтегративну якість особистості, до структури якої входять мотиваційний, змістовий та операційний компоненти.

Враховуючи особистісний підхід, ми визначили, що вона розглядатиметься нами як інтегральна якість особистості, що характеризується рівнем сформованості ціннісно-змістового (цінності та змісти), когнітивного (знання, уміння), операційно-діяльнісного (способи діяльності) компонентів, що сприяє результативному виконанню трудових функцій навчання, виховання та розвитку студентів.

Ми дійшли висновку, що підвищення рівня готовності інженера-педагога з цифрових технологій до професійної діяльності багато в чому буде залежати від ефективності системи безперервної освіти.

У зв'язку з цим ми можемо стверджувати, що провідним методологічним орієнтиром у підготовці інженерів-педагогів до професійної діяльності є персоніфікований підхід.

Дослідження практики застосування персоніфікованого підходу показало, що, незважаючи на живий інтерес до нього, використання персоніфікації в практичній діяльності досі утруднено. Це пов'язано з відсутністю врахування професійних дефіцитів інженерів-педагогів при організації їх навчання, замовлення освітньої організації на формування конкретних компетентностей, що необхідних інженеру-педагогу, контрольно-рефлексивного етапу після проходження навчання, коли інженер-педагог з цифрових технологій демонструє набуті компетенції через практичні заходи [7, С. 171].

Розглядаємо персоніфікацію як принцип, елементи освітнього процесу, які мають виходити з інтересів, потреб та прагнень осіб, що залучені до навчальної діяльності, було визначено, що реалізація даного підходу буде результативнішою за умов:

- навчання на основі індивідуального освітнього маршруту, що побудовано відповідно до можливостей, професійних дефіцитів та потреб інженера-педагога з цифрових технологій;

- урахування потреби освітньої організації в певних компетентностях інженера-педагога;

– варіативності освітніх програм, змісту та методів підготовки, часу та місця надання освітньої послуги, інтеграції можливостей системи підвищення кваліфікації та методичної роботи на всіх її рівнях;

– рефлексії реалізації індивідуального освітнього маршруту та вивчення динаміки рівня готовності інженера-педагога з цифрових технологій до реалізації професійної діяльності у сучасних умовах.

Аналіз реалізації даних умов у професійній підготовці майбутніх інженерів-педагогів з цифрових технологій на всіх її рівнях показав, що найбільш значущою складністю є відсутність критеріально-оцінних процедур, що дозволяє виявити професійні дефіцити.

З метою виявлення професійних дефіцитів у практичній діяльності інженера-педагога, а також оцінки результативності здійснення професійної підготовки інженерів-педагогів з цифрових технологій нами були обґрунтовані та представлені критеріально-оцінні процедури, що дозволяють визначити рівень готовності інженерів-педагогів до реалізації професійної діяльності в сучасних умовах, виходячи з виділених нами компонентів готовності.

РОЗДІЛ 2

ХАРАКТЕРИСТИКА ОБ'ЄКТІВ ГАЛУЗІ: СТАН І СТРАТЕГІЇ РОЗВИТКУ

2.1 Характеристика та аналіз нормативно-методичної документації з дисципліни «Інформатика та комп'ютерні технології»

Курс «Інформатика та комп'ютерні технології» розроблений відповідно до Стандарту вищої освіти України спеціальності 015 Професійна освіта (за спеціалізаціями) для підготовки бакалаврів, та є обов'язковим освітнім компонентом. Метою курсу є набуття здобувачами освіти професійних та особистісних компетенцій, які дадуть змогу використовувати сучасні інформаційні і комунікаційні технології в різних галузях професійної діяльності, науковій та практичній роботі, для самоосвіти та інших цілей. Поряд з практичною метою, курс реалізує освітні та виховні цілі, сприяє розширенню світогляду здобувачів, підвищенню їхньої загальної культури й освіченості.

Особливо важливо забезпечити педагогам обґрунтовану підготовку у сфері сучасних інформаційних і телекомунікаційних технологій. Викладачі нового покоління повинні володіти навичками аналізу, вибору та впровадження таких технологій, які відносяться до цілей і змісту навчання, сприяють розвитку особистості та розкриттю її потенціалу.

Застосування інформаційно-комунікаційних технологій у навчальному процесі сприятиме досягненню ключових завдань модернізації освіти: підвищенню якості професійної підготовки, розширенню доступності навчання, задоволенню потреби гармонійного розвитку особистості та інтеграції до інформаційного суспільства.

Особливу роль у цьому процесі мають педагоги, які є ключовими учасниками створення й наповнення інформаційно-освітнього середовища навчальних закладів на основі веб-технологій. Завдання такого середовища –

зробити доступним національний, культурний і освітній капітал, забезпечуючи рівний доступ до знань для всіх учасників освітнього процесу на рівні закладу, регіону та країни загалом.

Дисципліна «Інформатика та комп'ютерні технології» викладається на першому курсі навчання першого (бакалаврського) рівня освіти. Для здобувачів освіти денної форми навчання для вивчення дисципліни надано 6 кредитів (180 годин), з яких 26 годин лекційних занять, 64 години лабораторних занять, 90 години самостійної роботи.

Курс складається із трьох змістовних розділів:

1. Інформаційні технології обробки даних.

Загальні відомості про інформацію, інформаційні системи та обчислювальні машини. Апаратне забезпечення персональних комп'ютерів. Програмне забезпечення персональних комп'ютерних систем. Інформаційні технології обробки текстової інформації. Введення та редагування тексту, операції з фрагментами тексту. Форматування тексту. Технологія створення, редагування та форматування документів складної структури: таблиць, виразів у вигляді формул, графічних об'єктів, організаційних діаграм. Системи опрацювання табличних даних. Основи роботи у середовищі табличного процесора MS Excel. Створення, редагування та форматування електронних таблиць. Робота з функціями і формулами. Створення, редагування і форматування графіків та діаграм. Робота з базами даних у середовищі MS Excel. Мультимедійні технології. Сучасні комп'ютерні презентації. Правила оформлення комп'ютерних презентацій. Робота з організатором презентацій MS PowerPoint.

2. Хмарні технології.

Ресурси Інтернету. Хмарні технології та можливості їх використання. Пошук інформації і робота в середовищі Google Docs: Writely (Document), Spreadsheets. Створення й обробка анкет в середовищі Google. Робота з сервісом Google як інструментом створення презентацій.

3. Основи алгоритмізації та програмування.

Етапи розв'язування задач на ЕОМ. Поняття алгоритму. Загальні властивості. Способи описування. Загальна характеристика й класифікація мов програмування. Початкові відомості про середовище проектування. Основні конструкції мови програмування: числа, змінні, функції вирази. Оператори присвоєння, введення, виведення. Структура програми. Лінійні програми. Умовні оператори. Оператори вибору варіанту та переходу. Програмування розгалужених обчислювальних процесів. Програмування циклічних процесів. Структуровані типи даних: поняття про масиви, індекси, основні операції над масивами. Одномірні та двовимірні масиви даних, типові завдання їх обробки. Модульне програмування. Поняття про процедури та функції користувача, особливості їх використання.

Проведений аналіз змісту дисципліни та існуючого навчально-методичного забезпечення, а також враховуючи побажання викладача, що веде цю дисципліну, було запропоновано підготувати 3 презентації до лекцій з основ програмування та розробити 3 лабораторні роботи.

2.2 Презентації до лекційних занять

Мною було проаналізовано лекційний та навчальний матеріал [17-22] за наступними темами:

- циклічні оператори, реалізація циклічних алгоритмів;
- складні структури даних: списки, масиви.

Відповідно до цих тем було підготовлено три презентації до лекційних занять. Розмір кожної презентації складає 15-20 слайдів.

Уривки презентації на тему «Циклічні оператори» представлено на рисунках 2.1-2.4. В презентації надана інформація про різні типи циклів і відповідні їм оператори; розглянуто синтаксис операторів циклу, наведені приклади їх застосування.

Цикл for

Цикл **for** (цикл з параметром) в мові програмування Python призначений для *перебору елементів структур даних і деяких інших об'єктів*. Це не цикл з лічильником, яким є **for** в багатьох інших мовах.

Що значить перебір елементів? Наприклад, у нас є список, що складається з ряду елементів. Спочатку беремо з нього перший елемент, потім другий, потім третій і так далі. З кожним елементом ми виконуємо одні й ті ж дії в тілі **for**. Нам не треба витягати елементи по їх індексах і піклуватися про те, на якому з них список закінчується. Цикл **for** сам перебере і визначить кінець.

```
spisok = [12, 44, 28, 30]
for element in spisok:
    print(element/2)
результат: 6 22 14 15
```

Читається як для кожного елемента в списку робити наступне (то, що в тілі циклу).

Сам список не змінювався.

Рис. 2.1 Слайд 3 з презентації на тему «Циклічні оператори»

Функція range ()

"Range" перекладається як "діапазон". Вона може приймати один, два або три аргументи. Їх призначення таке ж як у функції `randrange ()` з модуля `random`. Якщо заданий тільки один аргумент, то генеруються числа від 0 до зазначеного числа, не включаючи його. Якщо задані два, то числа генеруються від першого до другого, не включаючи його. Якщо задані три, то третє число - це крок. Загальна структура функції `range()` така:

```
range([<початок>], <кінець> [,<крок>])
```

Обов'язковим є лише параметр кінець

Розглянемо окремі випадки:

range(n) – шкала чисел від 0 до n-1;

range(k, n) – шкала чисел від k до n-1;

range(k, n, m) – шкала чисел від k до n-1 з кроком m (причому m може бути від'ємним).

Рис. 2.2 Слайд 5 з презентації на тему «Циклічні оператори»

Цикл `while`

Цикл `while`

Можна сказати, `while` є універсальним циклом. Він присутній у всіх мовах, які підтримують структурне програмування, в тому числі в Python. Його синтаксис узагальнено для всіх мов можна висловити так:

while логічний_вираз:

вираз 1

...

вираз n

While, після того як його тіло виконано, потік повертається до заголовка циклу і знову перевіряє умова. Якщо логічний вираз повертає істину, то тіло знову виконується. Потім знову повертаємося до заголовку і так далі.

Цикл завершує свою роботу тільки тоді, коли логічний вираз в заголовку повертає хибу, тобто умова виконання циклу більше не дотримується. Після цього потік виконання переміщується до виразів, розташованих нижче всього циклу.

Рис. 2.3 Слайд 9 з презентації на тему «Циклічні оператори»

Оператори `continue` і `break`

Оператор **continue**.

Оператор **continue** починає наступний прохід циклу, минаючи тіло циклу (`for` або `while`)

Приклад.

Виведемо усі парні числа від 6 до 30, окрім чисел у діапазоні від 14 до 22

```
for i in range(6, 31, 2):
```

```
    if i>12 and i<26:
```

```
        # якщо число 12<i<26
```

```
        continue
```

```
        # перехід на нову ітерацію циклу
```

```
    print(i, end=" ")
```

Результат:

6 8 10 12 26 28 30

Рис. 2.4 Слайд 13 з презентації на тему «Циклічні оператори»

На рисунках 2.5-2.8 представлено слайди з презентації на тему «Робота зі списками». В презентації розглянуто способи створення списків, методи і

функції опрацювання списків, приклади їх використання.

1. Введення в списки.

Список у Python – це вбудований тип (клас) даних, що є одним із різновидів структур даних. Структуру даних можна як складну одиницю, що об'єднує у собі групу простіших. Кожен різновид структур даних має свої особливості.

Список – це змінна послідовність довільних елементів.

Щоб скористатися списками, їх потрібно створити. Створити список можна кількома способами. Наприклад, можна обробити будь-який об'єкт, що ітерується (наприклад, рядок) вбудованою функцією **list**:

```
some_list[START:STOP:STEP]
```

Рис. 2.5 Слайд 2 з презентації на тему «Робота зі списками»

1. Метод `append ()`. Додавання елемента до списку

Метод **append ()** використовується для додавання елемента до списку. Метод може отримувати тільки один параметр. Параметром методу може бути будь-який об'єкт: число, рядок, список і т.д.

Приклад використання методу **append ()**

```
# Приклад 1
# Метод append () - додавання елементів до списку
# заданий список
A = [2, 3.78, 'abcde', True]

# Додати 1 елемент до списку
A.append(7) # A = [2, 3.78, 'abcde', True, 7]
print( "A =", A)

# Сформуванати список квадратів чисел від 1 до 10
# За допомогою методу append ()
B = []
i = 1;
while i <= 10:
    B.append(i * i)
    i = i + 1
print ("B =", B)
```

Результат виконання програми

```
A = [2, 3.78, 'abcde', True, 7]
B = [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```

Рис. 2.6 Слайд 8 з презентації на тему «Робота зі списками»

Приклад 1. Використання методу `sort ()` для сортування списку в порядку зростання.

```
# Приклад 6 # Метод sort () - сортування списку
# Заданий список
A = [ 'a', 'f', 'v', 'd', 'n', 'b' ]

# Сортування списку
A.sort()

B = [1, 3, 5, 10, 2, 8]
B.sort()

print("A =", A)
print("B =", B)
```

Результат виконання програми
A = ['a', 'b', 'd', 'f', 'n', 'v']
B = [1, 2, 3, 5, 8, 10]

Для того, щоб використовувати метод `sort ()` всі елементи списку повинні бути або числовими або малими.
Наприклад, наступний код

```
# Помилка!
C = ["Hello", "ABC", 7]
C.sort()
згенерує помилку
TypeError: '<' not supported between instances of 'int' and 'str'
```

Рис. 2.7 Слайд 14 з презентації на тему «Робота зі списками»

Функції

Функції для роботи зі списками, на відміну від методів, не змінюють сам список, а повертають певне значення. З функціями `len()` і `list()` ви вже знайомі.

```
# max(list) Повертає найбільше значення елемента
>>> a = [1, 5, 7, 31, -5]
>>> max(a)
31

# min(list) Повертає найменше значення елемента
>>> min(a)
-5

# sum(list) Повертає значення суми елементів
>>> sum(a)
39

# del(list[n]) Видаляє елемент із індексом n
>>> del(a[2])
>>> a
[1, 5, 31, -5]
```

Знайдемо індекс найбільшого елемента у списку **a = [1, 5, 7, 31, -5]**:

```
>>> a = [1, 5, 7, 31, -5]
>>> m = max(a)           # m = 31
>>> n = a.index(m)      # n = 3
```

Комбінуючи функції та методи роботи зі списками, ми можемо, як із цеглинок, скласти алгоритм розв'язування складної задачі.

Рис. 2.8 Слайд 20 з презентації на тему «Робота зі списками»

На рисунках 2.9-2.12 представлено слайди презентації на тему «Масиви в Python». В презентації розглянуті питання створення масивів та методів

2. Як працює масив у Python?

Масив зберігається в суміжних місцях пам'яті, де індекс - це послідовність чисел, яка представляє значення, що зберігаються в кожному конкретному індексі. Щоб отримати доступ до або вказати значення для певного індексу в масиві, ми використовуємо набір квадратних дужок (), також ми можемо використовувати цикл для введення для ітерації через масив. Масив має індекси (множинна форма індексу) та значення.

Рис. 2.9 Слайд 5 з презентації на тему «Масиви в Python»

3. Як створити масиви в Python?

- Для створення масиву в Python нам потрібно спочатку імпортувати модуль масиву.

```
import array as arr
```

де,
arr => - псевдонім

Інший спосіб імпорту модуля полягає в наступному:

```
from array import *
```

Синтаксис для створення масиву:

```
array(typecode (, initializer))
```

де, typecode => int, float або double або тип значення, яке має масив.
ініціалізатор => - необов'язкове значення і може мати будь-який тип, наприклад список, рядок або будь-які ітерабельні елементи певного типу.

Рис. 2.10 Слайд 8 з презентації на тему «Масиви в Python»

2. Вставити ()

Синтаксис: `array.insert(index, element)`
Він додає елемент до масиву перед індексом

Код:

```
#importing array module
import array as arr
#creating array
a = arr.array('i', (100, 200, 300))
#inserting a value of 400 to after index 2
a.insert(3, 400);
#looping through array a
for i in a:
    print(i)
```

Вихід:

```
100
200
300
400
```

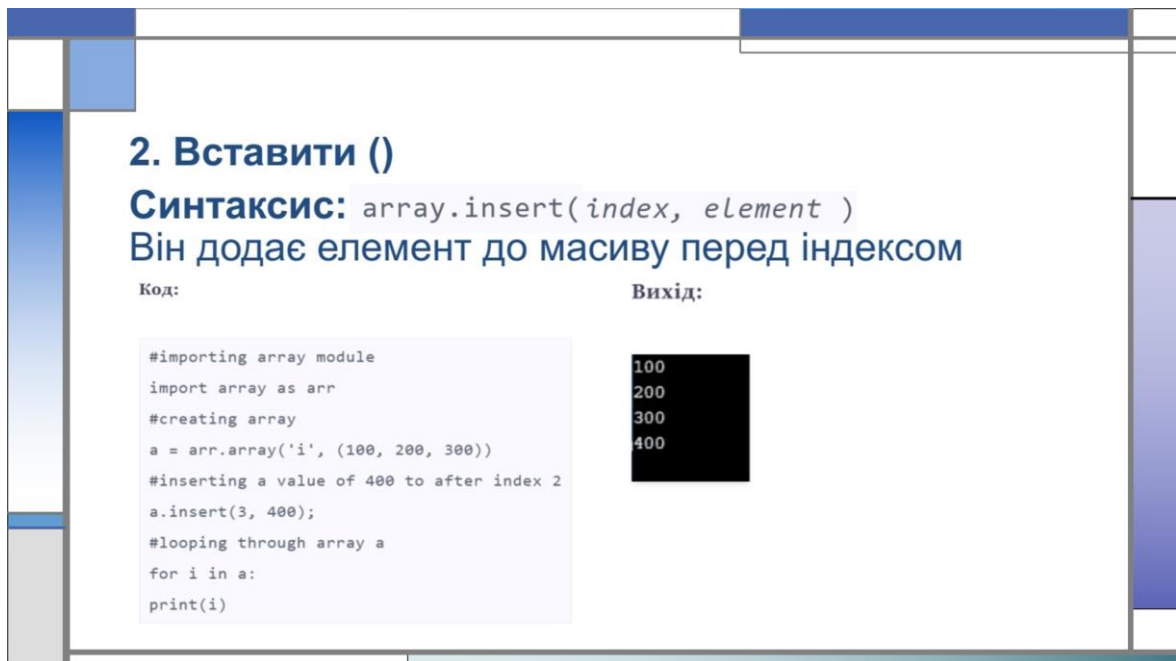


Рис. 2.11 Слайд 13 з презентації на тему «Масиви в Python»

7. count ()

Синтаксис: `array.count(element)`
Ця функція повертає, скільки разів елемент траплявся в даному масиві.

Код:

```
#importing array module
import array as arr
#creating array
a3 = arr.array('i', (100, 200, 300, 100, 400, 100, 500) )
#printing the array count
print(a3.count(100))
```

Вихід:

```
3
```

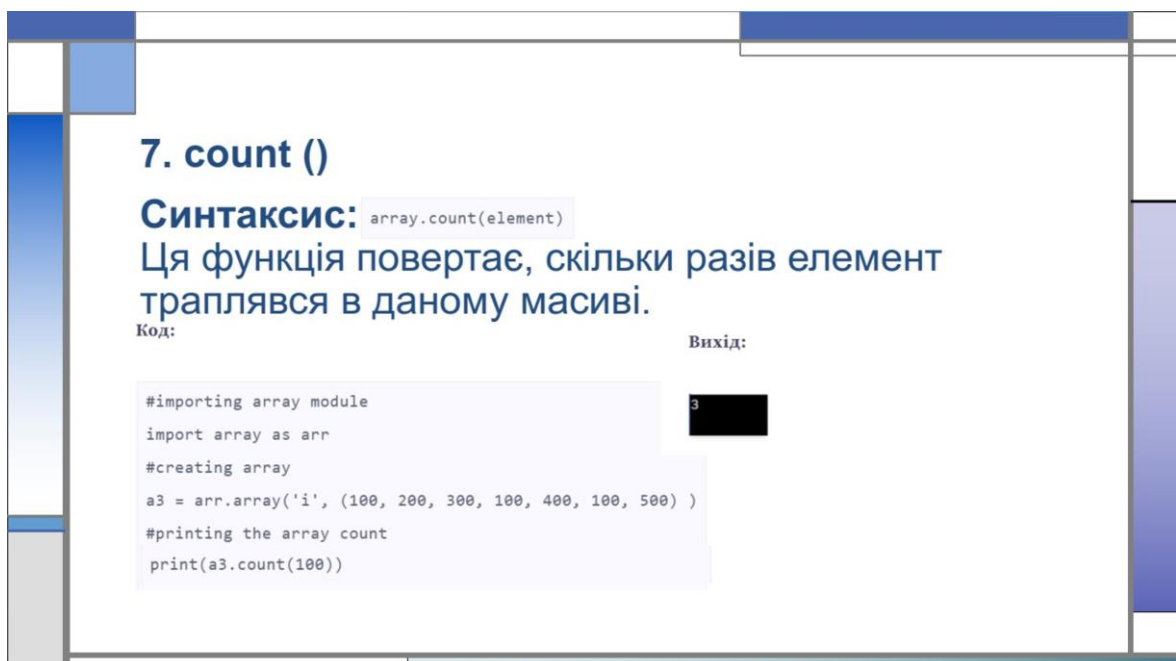


Рис. 2.12 Слайд 18 з презентації на тему «Масиви в Python»

2.3 Лабораторна робота на тему «Програмування алгоритмів циклічної структури»

Мета роботи: знайомство з циклічними алгоритмічними конструкціями мови Python, отримання практичних навичок програмування алгоритмів циклічної структури.

Порядок виконання роботи

1. Виконайте початковий етап написання звіту, який складається у вказівці теми і мети лабораторної роботи.
2. Повторіть основний теоретичний матеріал по темі лабораторної роботи.
3. Ознайомтеся з прикладами програмування алгоритмів циклічної структури.
4. Виконайте завдання лабораторної роботи відповідно до свого варіанту.
5. Оформіть звіт, зазначивши в ньому всі необхідні етапи лабораторної роботи для кожного з виконаних Вами завдань; пред'явіть звіт на перевірку викладачеві.

Приклади програмування алгоритмів циклічної структури

Приклад 1. Визначити добуток ряду $P = \prod_{n=1}^{\infty} \frac{x}{n^3} = \frac{x}{1^3} \cdot \frac{x}{2^3} \cdot \frac{x}{3^3} \cdot \frac{x}{4^3} \cdot \dots$,

обчислення закінчити при виконанні умови $\frac{1}{n^3} < 10^{-3}$

Алгоритм розв'язання задачі представлено на рисунку 2.13

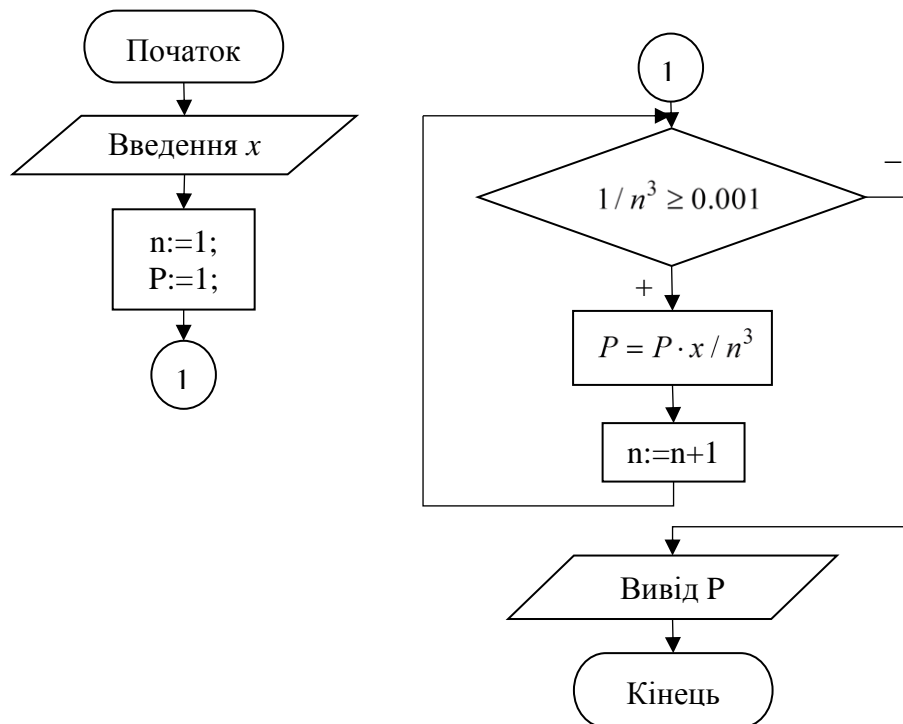
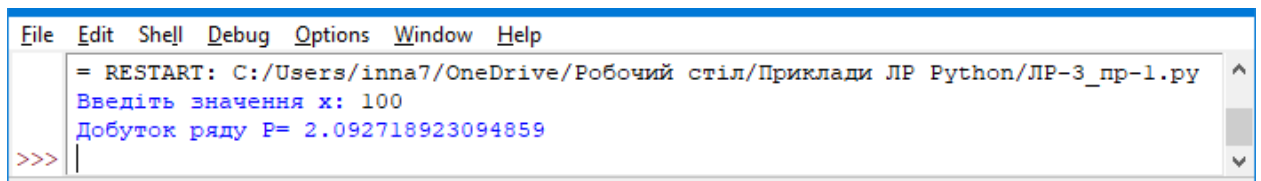


Рис. 2.13 Схема алгоритму розв'язання прикладу 1

Код програми представлений на рисунку 2.14. Результат виконання програми на рисунку 2.15.

```
x = float(input('Введіть значення x: '))
n=1
p=1
while 1/n**3>=0.001:
    p=p*x/n**3
    n=n+1
print('Добуток ряду P=', p)
```

Рис. 2.14 Код програми до прикладу 1



```
= RESTART: C:/Users/inna7/OneDrive/Робочий стіл/Приклади ЛР Python/ЛР-3_пр-1.py
Введіть значення x: 100
Добуток ряду P= 2.092718923094859
>>>
```

Рис. 2.15 Вікно результатів розв'язання задачі

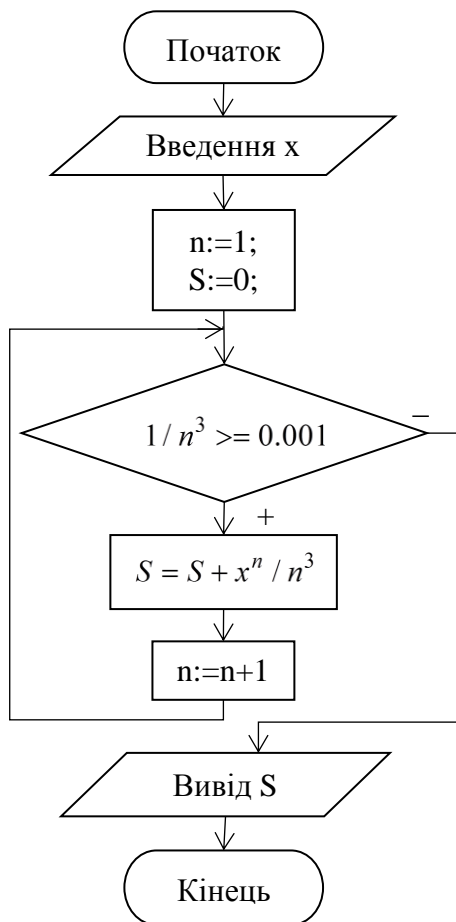
Приклад 2. Визначити суму ряду $S = \sum_{n=1}^{\infty} \frac{x^n}{n^3}$, обчислення закінчити при

виконанні умови $\frac{1}{n^3} < 10^{-3}$

Алгоритм розв'язання задачі

Код програми

```
x = float(input('Введіть значення x: '))
n=1
s=0
k=1
while k>=0.001:
    s=s+x**n*k
    n+=1
    k=1/n**3
print('Сума ряду S=%0.4f %s')
```



Результат виконання програми на рисунку 2.16.

```

File Edit Shell Debug Options Window Help
>>>
= RESTART: C:/Users/inna7/OneDrive/Робочий стіл/Приклади ЛР Python/ЛР-3_1
Введіть значення x: 1
Сума ряду S=1.1975
>>>
  
```

Рис. 2.16 Вікно результатів розв'язання задачі прикладу 2

Приклад 3. Надрукуйте таблицю залежності $Z = \begin{cases} \sin x + 1, & \text{при } x \leq 1 \\ \ln^2 x, & \text{при } 1 < x < 4, \\ 2 \cdot \cos x, & \text{при } x \geq 4 \end{cases}$,

якщо x змінюється від -1 до 6 з кроком 0,5.

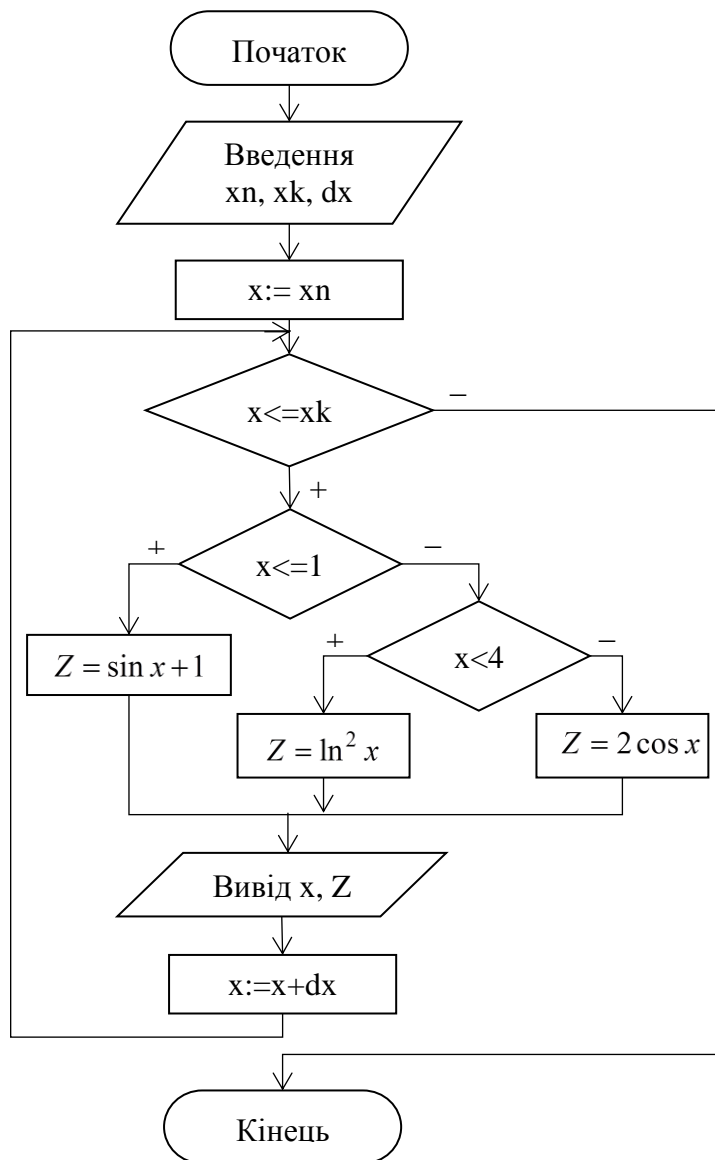
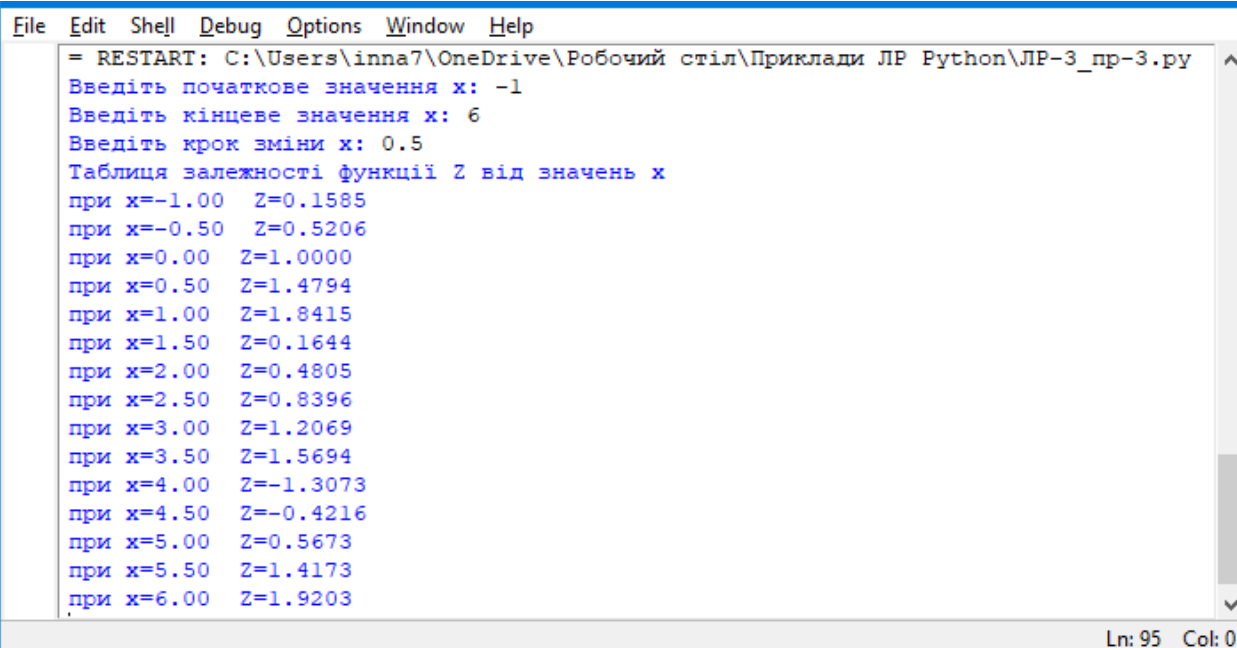


Рис. 2.17 Схема алгоритму розв'язання прикладу 3
Код програми представлений на рисунку 2.18. Результат виконання програми на рисунку 2.19.

```

File Edit Format Run Options Window Help
import math
xp = float(input('Введіть початкове значення x: '))
xk = float(input('Введіть кінцеве значення x: '))
dx = float(input('Введіть крок зміни x: '))
x=xp
print('Таблиця залежності функції Z від значень x')
while x<=xk:
    if x<=1: z=math.sin(x)+1
    elif x<4: z=math.log(x)**2
    else: z=2*math.cos(x)
    print('при x=%.2f Z=%.4f' %(x,z))
    x=x+dx
  
```

Рис. 2.18 Код програми до прикладу 3



```
File Edit Shell Debug Options Window Help
= RESTART: C:\Users\inna7\OneDrive\Робочий стіл\Приклади ЛР Python\ЛР-3_пр-3.py
Введіть початкове значення x: -1
Введіть кінцеве значення x: 6
Введіть крок зміни x: 0.5
Таблиця залежності функції Z від значень x
при x=-1.00 Z=0.1585
при x=-0.50 Z=0.5206
при x=0.00 Z=1.0000
при x=0.50 Z=1.4794
при x=1.00 Z=1.8415
при x=1.50 Z=0.1644
при x=2.00 Z=0.4805
при x=2.50 Z=0.8396
при x=3.00 Z=1.2069
при x=3.50 Z=1.5694
при x=4.00 Z=-1.3073
при x=4.50 Z=-0.4216
при x=5.00 Z=0.5673
при x=5.50 Z=1.4173
при x=6.00 Z=1.9203
Ln: 95 Col: 0
```

Рис. 2.19 Вікно результатів розв'язання задачі з прикладу 3

Приклад 4. Клієнт банку зробив внесок в 10000 грн. на термін 5 років. Банк гарантує 12% річних. Для показу динаміки зростання рахунку клієнта рік від року необхідно обчислити значення, які приймає накопичена сума на кінець кожного року.

Математична модель задачі

Позначимо внесок літерою – V , річний відсоток – p , суму накопичення – S , поточний термін – i , термін внеску – n років.

Накопичення за n років визначається формулою $S = V(1 + p/100)^n$.

Таким чином задача зводиться до табулювання функції $S(i) = V(1 + p/100)^i$ для $i=1, 2, \dots, n$. Після обчислення значення функції відразу ж будемо виводити його разом з відповідним поточним значенням i на екран.

Структура даних

Ім'я	Значення	Тип змінних
v	Початковий внесок (грн.)	float
p	Річний відсоток на внески (%)	float
n	Термін внеску (роки)	int
s	Сума накопичення (грн.)	float
i	Поточний термін (рік)	int

Алгоритм розв'язання задачі представлений на рисунку 2.20

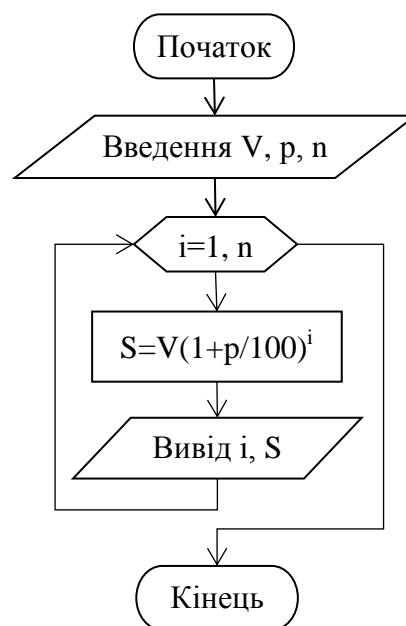


Рис. 2.20 Схема алгоритму розв'язання прикладу 4

Код програми представлений на рисунку 2.21. Результат виконання програми на рисунку 2.22.

```

File Edit Format Run Options Window Help
v = float(input('Введіть початковий внесок (грн.): '))
p = float(input('Введіть річний відсоток на внески (%): '))
n = int(input('Введіть термін внеску (роки): '))
print('Динаміка зростання рахунку клієнта')
for i in range(1,n+1):
    s=v*(1+p/100)**i
    print('В %d році сума виплат складе %.2f грн.' %(i, s))
  
```

Рис. 2.21 Код програми до прикладу 4

```

File Edit Shell Debug Options Window Help
Введіть початковий внесок (грн.): 10000
Введіть річний відсоток на внески (%): 12
Введіть термін внеску (роки): 5
Динаміка зростання рахунку клієнта
В 1 році сума виплат складе 11200.00 грн.
В 2 році сума виплат складе 12544.00 грн.
В 3 році сума виплат складе 14049.28 грн.
В 4 році сума виплат складе 15735.19 грн.
В 5 році сума виплат складе 17623.42 грн.
>>> |

```

Рис. 2.22 Вікно результатів розв'язання задачі з прикладу 4

Завдання до лабораторної роботи

Завдання 1. Скласти алгоритм розв'язування задачі відповідно до варіанта та зобразити його графічно у вигляді блок-схеми. Записати алгоритм у вигляді коду програми на Python. Виконати складену програму для деяких вхідних даних. У програмі цієї задачі використати оператор умовного циклу.

Варіанти завдання

1. Визначити суму ряду $S = \sum_{n=1}^{\infty} \frac{x}{2n+3}$, обчислення закінчити при виконанні

умови $\frac{1}{2n+3} < 10^{-3}$.

2. Надрукуйте таблицю залежності $K = \frac{ax^2 + b^3x}{\sqrt{x} + a^2bx}$, якщо x змінюється від 0,2 до 1,6 з кроком 0,2.

3. Знайдіть добуток ряду $P = \prod_{n=1}^{\infty} \frac{x^3}{1-n^2}$, обчислення закінчити при виконанні

умови $\frac{1}{n^2} < 10^{-3}$.

4. Визначити суму ряду $S = \sum_{n=1}^{\infty} \frac{x}{n^2}$, обчислення закінчити при виконанні

умови $\frac{1}{n^2} < 10^{-3}$.

5. Надрукуйте таблицю залежності $y = \frac{2x^2 + 3}{x^2 - 4x + 1}$, якщо x змінюється від -1 до 6 з кроком 0,5.
6. Скільки доданків має бути в сумі $1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$, щоб ця сума виявилася більшою 5.
7. Обчисліть $Y = \begin{cases} x - 6, & \text{якщо } \cos x \geq 0.2 \\ 1 + \sin x, & \text{якщо } \cos x < 0. \end{cases}$ при зміні аргументу X від -2 до 4.
8. Обчисліть суму $\sin 1 + \sin 2 + \dots + \sin M$ і число доданків до першого від'ємного доданка. ($M=7$)
9. Знайдіть добуток ряду $P = \prod_{n=1}^{\infty} \frac{x^2}{1 + n^3}$, обчислення закінчити при виконанні умови $\frac{1}{n^3} < 10^{-3}$.
10. Надрукуйте таблицю залежності $y = 7\sqrt{\cos(\pi x)^2 + 2}$, якщо x змінюється від 0 до 2,4 з кроком 0,4.
11. Визначити суму ряду $S = \sum \frac{\cos x}{3x - 5}$, якщо x змінюється від x_n до x_k з кроком h .
12. Визначити суму ряду $S = \sum_{n=1}^{\infty} \frac{x}{2 + n^2}$, обчислення закінчити при виконанні умови $\frac{1}{n^2} < 10^{-3}$.
13. Обчисліть $Y = \begin{cases} 5 - x, & \text{якщо } \operatorname{tg} x < 1 \\ 3 + \cos x, & \text{якщо } \operatorname{tg} x \geq 1 \end{cases}$ при зміні аргументу X від -3 до 5.

14. Визначити суму ряду $S = \sum_{n=1}^{\infty} \frac{x \cdot (n+1)}{n^3}$, обчислення закінчити при

виконанні умови $\frac{n+1}{n^3} < 10^{-4}$.

15. Обчисліть $Y = \begin{cases} 1 + 7x, & \text{якщо } \sin x < -0.4 \\ \operatorname{tg} x + 3, & \text{якщо } \sin x \geq -0.4 \end{cases}$ при зміні аргументу X від 0 до 5.

2.4 Лабораторна робота на тему «Проектування алгоритмів з використанням одновимірних масивів даних»

Для підготовки лабораторної роботи використані матеріали з лабораторного практикуму [24]

Мета роботи: вивчення основних прийомів роботи зі списками даних та застосування циклів до опрацювання їх елементів.

Порядок виконання роботи

1. Виконайте початковий етап написання звіту, який складається у вказівці теми і мети лабораторної роботи.
2. Повторіть основний теоретичний матеріал по темі лабораторної роботи.
3. Ознайомтеся з прикладами розробки програмних проектів, які здійснюють обробку одновимірних масивів даних.
4. Виконайте завдання лабораторної роботи відповідно до свого варіанту.
5. Оформіть звіт, зазначивши в ньому всі необхідні етапи лабораторної роботи для кожного з виконаних Вами завдань; пред'явіть звіт на перевірку викладачеві.

Короткі теоретичні відомості

Масив є індексованою сукупністю однотипних даних, що позначаються спільним іменем змінної. Доступ до кожного окремого значення у цих даних здійснюється за іменем масиву та номером елемента в ньому. За допомогою циклу можна перебирати усі елементи масиву з метою виконання з ними вводу, виводу даних, виконання обчислень, тощо. [24]

У мові програмування Python така форми структурування даних, як масив відсутня. Тобто, в класичному розумінні, як у прикладах на інших мовах програмування вище, коли пам'ять виділяється одразу для зберігання усіх елементів масиву. Тут для зберігання індексованих сукупностей використовують списки. Основна відмінність між списком і масивом у можливості «дописувати» елементи до списку в ході виконання програми (або вилучати). Проте сам механізм доступу до елемента в списку за його індексом практично нічим не відрізняється від масиву. [24]

В програмах на Python часто використовують списки, проініціалізовані попередньо деяким набором елементів. Для цього використовують синтаксичну конструкцію вигляду:

$$\langle \text{list-name} \rangle = [\langle \text{value1} \rangle, \langle \text{value2} \rangle, \dots, \langle \text{valueN} \rangle]$$

яка створює список з N елементів і заповнює його значеннями, вказаними в квадратних дужках. Часто зручніше створювати порожній (без жодного елемента) список і додавати в нього елементи по мірі вводу даних функцією *append*. [24]

Оголосити порожній список можна, залишивши в розглянутій вище конструкції квадратні дужки порожніми. В кодї далі для цього використано функцію *array* з модуля *array*, це дозволяє при оголошенні списку зафіксувати тип елементів. При цьому типи даних можна, зокрема, позначати такими символами: 'b', 'h', 'i', 'l', 'q', якщо елементи масиву – цілі числа, 'f', 'd', якщо елементи масиву дійсні числа, 'u' – символні дані в Unicode. [24]

Приклади вводу та обробки одновимірних масивів даних

Приклад 1. Знайти мінімальний елемент масиву A(10) та його порядковий номер.

Структура даних

Ім'я	Значення	Тип змінних
a	Масив дійсних чисел	float
i	Номер елементу масиву	int
a_min	Мінімальний елемент масиву	float
n	Номер мінімального елементу	int

Алгоритм розв'язання задачі представлений на рисунку 2.23

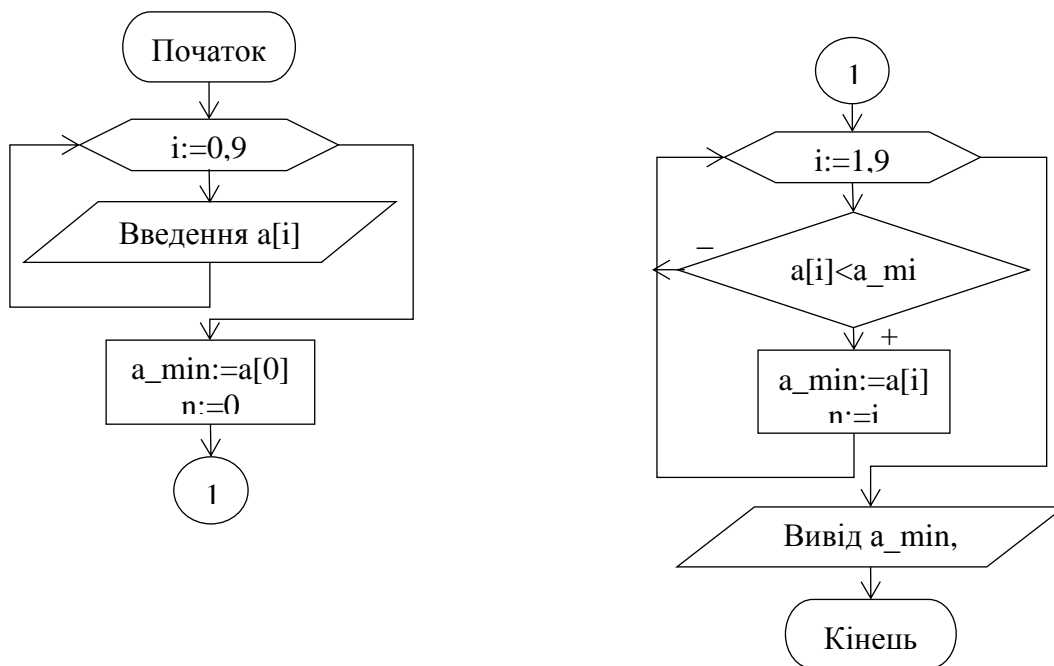


Рис. 2.23 Схема алгоритму розв'язання прикладу 1

Код програми представлений на рисунку 2.24. Результат виконання програми на рисунку 2.25.

```

File Edit Format Run Options Window Help
a=[1.3,2.5,-6,4,0,-12.3,0,1,1,-9]
print('Елементи масиву A=', a)
a_min=a[0]
n=0
for i in range(1,10):
    if a[i]<a_min:
        a_min=a[i]
        n=i
print('Мінімальний елемент =',a_min, ' Його індекс =',n+1)
    
```

Рис. 2.24 Код програми до прикладу 1

```

File Edit Shell Debug Options Window Help
Елементи масиву A= [1.3, 2.5, -6, 4, 0, -12.3, 0, 1, 1, -9]
Мінімальний елемент = -12.3 його індекс = 6
>>> |

```

Рис. 2.25 Вікно результатів розв'язання задачі з прикладу 1

Приклад 2. Обчислити суму елементів масиву $C(15)$, які менше b , і вивести номери цих елементів.

Алгоритм розв'язання задачі представлений на рисунку 2.26

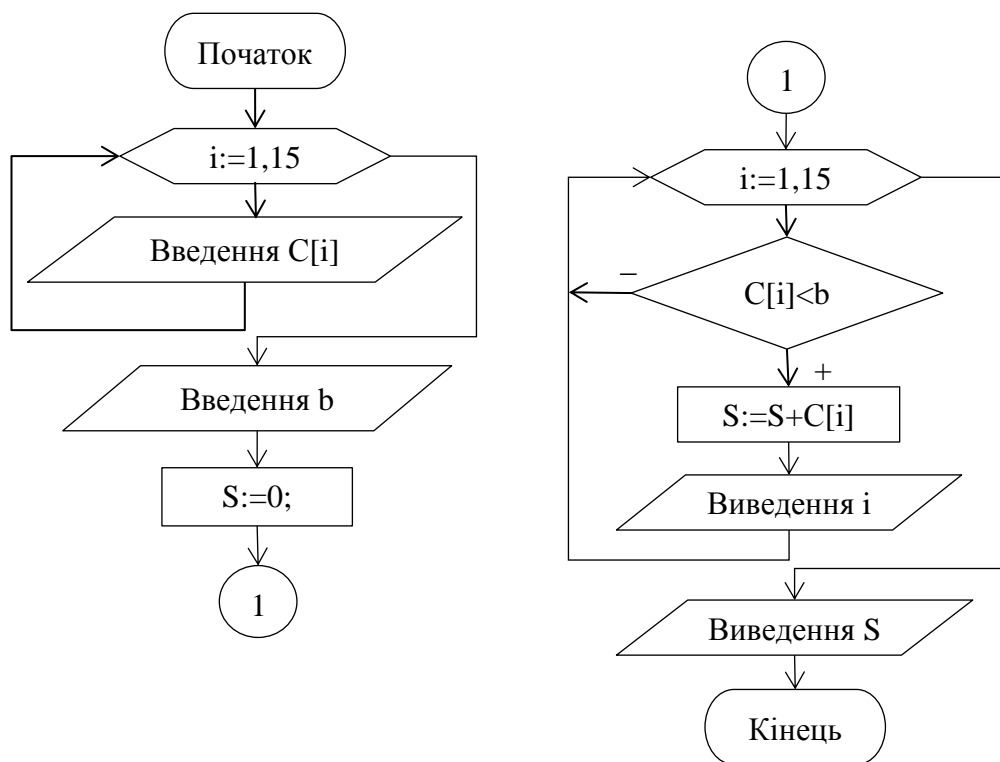


Рис. 2.26 Схема алгоритму розв'язання прикладу 2

Код програми представлений на рисунку 2.27. Результат виконання програми на рисунку 2.28.

```

File Edit Format Run Options Window Help
import random
c=[round(random.uniform(-100,100),2) for i in range(15)]
print('Елементи масиву C =', c)
b = float(input('Введіть значення b: '))
s=0
print('Номера елементів масиву, які менше b: ')
for i in range(15):
    if c[i]<b:
        s+=c[i]
        print(i+1,end=' ')
print()
print('Сума елементів масиву, які менше b =%.2f' %s)

```

Рис. 2.27 Код програми до прикладу 2

```

File Edit Shell Debug Options Window Help
Елементи масиву C = [-30.55, 77.04, 63.07, -55.03, 90.62, -89.05, 66.55, -16.19,
97.81, 14.04, -91.82, 68.52, 65.26, -96.75, -54.85]
Введіть значення b: 0
Номера елементів масиву, які менше b:
1 4 6 8 11 14 15
Сума елементів масиву, які менше b =-434.24
>>> |

```

Рис. 2.28 Вікно результатів розв'язання задачі з прикладу 2

Приклад 3. Обчислення добутку та кількості парних елементів одновимірного масиву.

Структура даних

n	Кількість елементів в масиві	int
d	Масив цілих чисел	int
i	Номер елементу масиву	int
P	Добуток парних елементів масиву	int
k	Кількість парних елементів масиву	int

Алгоритм розв'язання задачі представлений на рисунку 2.29.

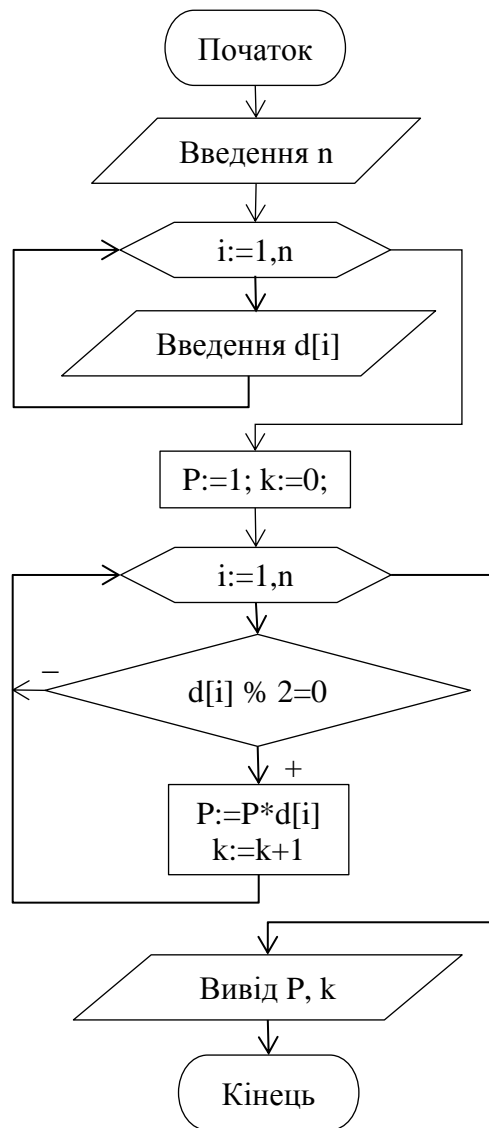


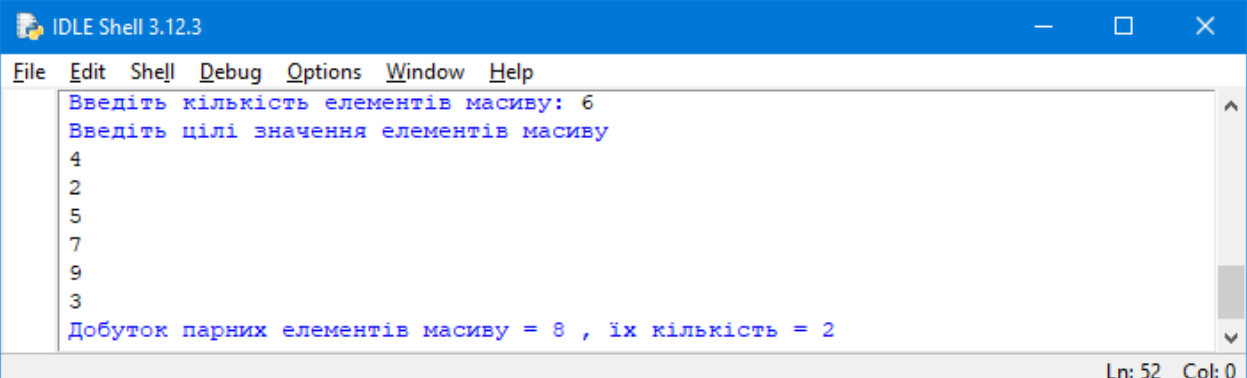
Рис. 2.29 Схема алгоритму розв'язання прикладу 3

Код програми представлений на рисунку 2.30. Результат виконання програми на рисунку 2.31.

```

File Edit Format Run Options Window Help
n = int(input('Введіть кількість елементів масиву: '))
print('Введіть цілі значення елементів масиву')
d=[int(input()) for i in range(n)]
p=1
k=0
for i in range(n):
    if d[i]%2==0:
        p*=d[i]
        k+=1
print('Добуток парних елементів масиву =',p, ', їх кількість =',k)
  
```

Рис. 2.30 Код програми до прикладу 3



```
File Edit Shell Debug Options Window Help
Введіть кількість елементів масиву: 6
Введіть цілі значення елементів масиву
4
2
5
7
9
3
Добуток парних елементів масиву = 8 , їх кількість = 2
Ln: 52 Col: 0
```

Рис. 2.31 Вікно результатів розв’язання задачі з прикладу 3

Завдання до лабораторної роботи

Завдання 1. Скласти алгоритм розв’язування задачі відповідно до варіанта та зобразити його графічно у вигляді блок-схеми. Записати алгоритм у вигляді коду програми на Python. Виконати складену програму для деяких вхідних даних. Передбачити виведення на екран початкового масиву даних та результату роботи програми.

У звіт подати:

- умову задачі варіанта;
- блок-схему алгоритму розв’язування задачі;
- складений Python-код;
- результат виконання програми для деяких вхідних даних.

Варіанти завдання

1. Обчислити добуток елементів масиву $A(14)$, більших за b , і вивести на друк їх номери.
2. Обчислити середнє арифметичне парних елементів масиву $C(10)$.
3. Знайти максимальний елемент масиву $X(10)$ і відняти його від кожного елемента масиву. Виведіть значення максимального елемента і новий масив.
4. Знайти кількість елементів масиву $P(15)$ кратних 4 і не більших заданого числа A .

5. Записати 100 замість мінімального елемента масиву $H(13)$. Вивести новий масив.
6. У масиві $A(15)$ усі парні значення зменшите на 2, всі непарні значення збільште на 2. Виведіть новий масив.
7. Знайти кількість елементів масиву $K(14)$, що належать інтервалу $(7, 20)$.
8. Обчислити добуток та кількість від'ємних елементів масиву $A(9)$.
9. Обчислити суму і кількість елементів масиву $C(12)$, які більше A і менші B .
10. Поміняти місцями перший і максимальний елемент масиву $A(14)$. Вивести новий масив.
11. Знайти суму і кількість елементів масиву $B(12)$, що кратні 7 та належать інтервалу $[7, 70]$.
12. Обчислити середнє арифметичне додатних елементів масиву $F(14)$, що стоять на парних місцях.
13. Поміняти місцями перший і мінімальний елемент масиву $A(10)$. Вивести новий масив.
14. Підрахувати кількість парних і кількість непарних елементів в масиві $C(12)$.
15. Обчислити суму квадратів всіх додатних елементів масиву $T(12)$.

2.5 Лабораторна робота на тему «Опрацювання двовимірних масивів даних»

Для підготовки лабораторної роботи використані матеріали з лабораторного практикуму. [24]

Мета роботи: оволодіння практичними навичками програмування алгоритмів циклічної структури, зберігання та опрацювання даних у формі двовимірних масивів, а також використання функцій форматного вводу та виводу числових даних.

Порядок виконання роботи

1. Виконайте початковий етап написання звіту, який складається у вказівці теми і мети лабораторної роботи.
2. Повторіть основний теоретичний матеріал по темі лабораторної роботи.
3. Ознайомтеся з прикладами розробки програмних проектів, які здійснюють обробку двовимірних масивів даних.
4. Виконайте завдання лабораторної роботи відповідно до свого варіанту.
5. Оформіть звіт, зазначивши в ньому всі необхідні етапи лабораторної роботи для кожного з виконаних Вами завдань; пред'явіть звіт на перевірку викладачеві.

Короткі теоретичні відомості

Під двовимірним масивом розуміють сукупність елементів одного типу, яка індексується за двома напрямками двома цілими числами та має спільну назву. Така сукупність даних має зручне візуальне подання у вигляді прямокутної таблиці, що відобразилося у вживанні термінів «рядки» чи «стовпці» до елементів з однаковим значенням одного з індексів (див. рисунок). [24]

Стовпці 1...k

Рядки 1...n	елемент(1,1)	елемент(1,2)	...	елемент(1, k)
	елемент(2,1)	елемент(2,2)	...	елемент(2, k)

	елемент(n, 1)	елемент(n, 2)	...	елемент(n, k)

Доступ до конкретного елемента двовимірного масиву здійснюється вказанням двох чисел, – номера «рядка», та номера «стовпця». За значенням лише одного з індексів тут можна встановити лише сукупність елементів, яка сама по собі також є масивом, але вже одновимірним. [24]

З точки зору збереження даних пам'яті комп'ютера така структура є дещо «штучною», бо адресація комірок пам'яті є послідовною. Тому з двовимірним масивом працюють послідовно як із сукупністю одновимірних

масивів, а в літературі з програмування часто можна зустріти визначення двовимірного масиву як масиву масивів або списку масивів. [24]

Необхідність зберігання та програмного опрацювання двовимірних масивів обумовлюється, насамперед, їх прикладним значенням. В різних задачах фізики та техніки, економіки, нейромережових технологіях систем штучного інтелекту і т. п. використовується математичний апарат матриць, які за своєю конструкцією і є двовимірними числовими масивами. Терміни матриця та двовимірний масив часто ототожнюються, насправді ж, поняття масиву є ширшим, оскільки позначає сукупність елементів довільного типу, а матриця утворюється лише з чисел. [24]

Використання в Python списків замість масивів в двовимірному випадку завдає багато клопоту. Якщо в Java чи C++ «масив масивів» – лише концепція, за якою система працює з багатовимірними масивами, то список списків в Python, – це реальний об'єкт, який потрібно створити в програмі для того, щоб зберігати та опрацьовувати двовимірний масив. [24]

Система оголошення змінних в Python, яка визначає тип змінної за присвоєним їй значенням, хоча і є динамічною, проте строго контролює тип об'єкта записаного в змінну та дозволяє виконувати зі значенням змінної лише дії, які передбачені її типом. Інколи програмісти намагаються оголосити двовимірний список відповідних розмірів інструкцією вигляду:

$$\langle \text{variable} \rangle = [[\langle \text{value} \rangle] * \langle \text{colCount} \rangle] * \langle \text{rowCount} \rangle$$

ініціалізуючи всі його елементи деяким значенням. [24]

Слушність такого підходу частково оправдана тим, що значення value має певний тип, і цей тип буде задавати тип елементів списку. Хоча динамічна типізація не буде перешкоджати запису в елемент такого списку значень будь- якого іншого типу. Оскільки списки в Python можуть містити об'єкти будь- якого типу, то для двовимірного масиву можна оголосити звичайний (одновимірний) список, а потім, в циклі, для кожного рядка масиву створити окремий список, та заповнити його елементами. У підсумку

звертання до елемента матриці за індексами $p[i][j]$ насправді конвертуватиметься у вибір j -го елемента в i -му списку-рядку. [24]

Приклади вводу та обробки двовимірних масивів даних

Приклад 1. Для кожного рядка двовимірного масиву $D(3, 4)$ визначити суму елементів стовпців з парними номерами.

Алгоритм розв'язання задачі представлений на рисунку 2.32.

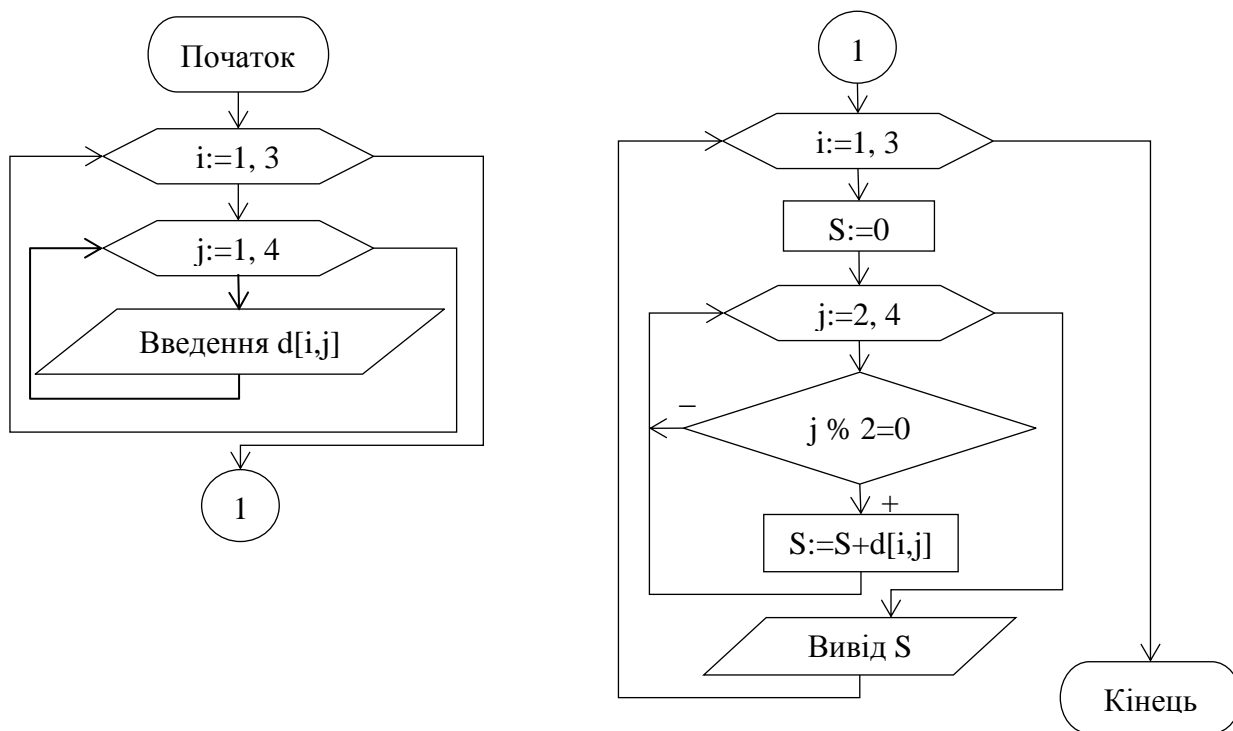


Рис. 2.32 Схема алгоритму розв'язання прикладу 1

Код програми представлений на рисунку 2.33. Результат виконання програми на рисунку 2.34.

```
File Edit Format Run Options Window Help
print('Для кожного рядка двовимірного масиву D(3, 4) визначити
D=[[2.3, 1, -5, 4], [8, 5.2, -1, 0.4], [7, 0.3, 5, -1]]
print('Заданий масив')
for i in D:
    for j in i:
        print('%4.1f'%j, end=' ')
    print()
for i in range(3):
    s=0
    for j in range(1,4,2): s+=D[i][j]
    print('Сума елементів в %d рядку = %.1f'%(i+1,s))
```

Рис. 2.33 Код програми до прикладу 1

```

File Edit Shell Debug Options Window Help
Для кожного рядка двовимірного масиву D(3, 4) визначити суму елементів стовпців
в парних номерах
Заданий масив
2.3 1.0 -5.0 4.0
8.0 5.2 -1.0 0.4
7.0 0.3 5.0 -1.0
Сума елементів в 1 рядку = 5.0
Сума елементів в 2 рядку = 5.6
Сума елементів в 3 рядку = -0.7

```

Рис. 2.34 Вікно результатів розв'язання задачі з прикладу 1

Приклад 2. Визначити кількість від'ємних елементів в кожному стовпці двовимірного масиву B(4, 3).

Алгоритм розв'язання задачі представлений на рисунку 2.35.

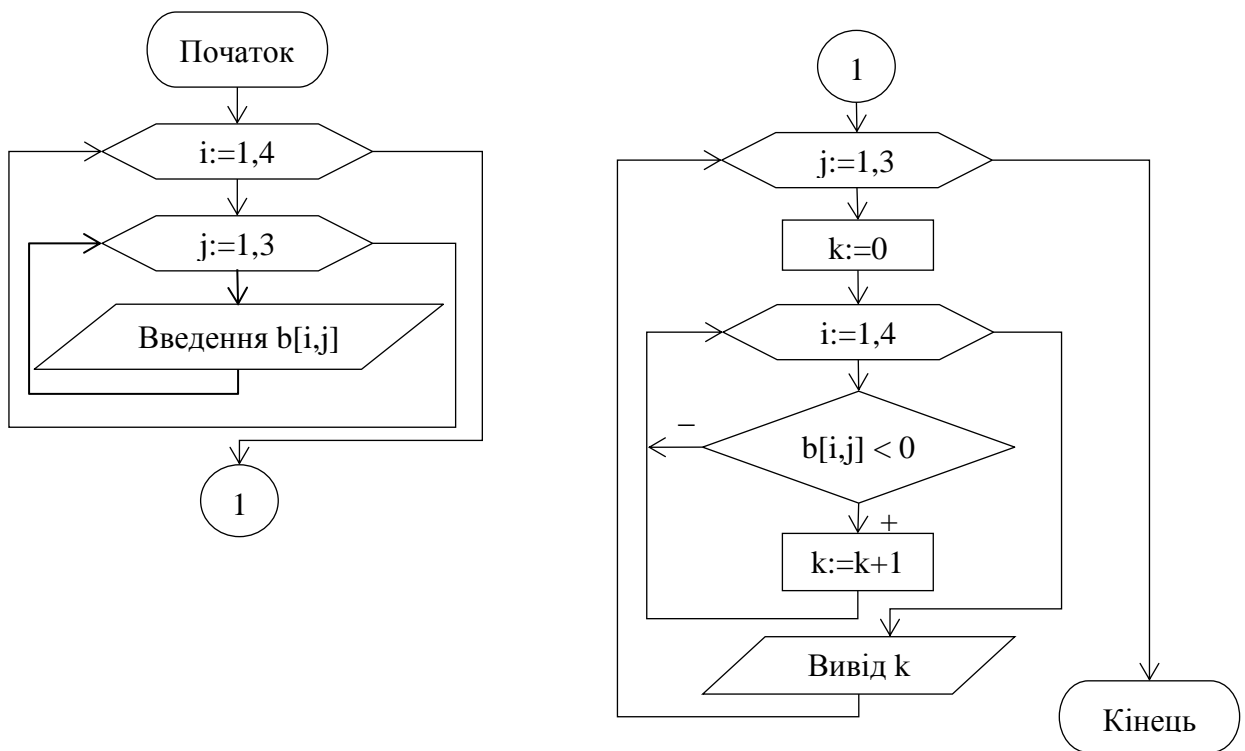


Рис. 2.35 Схема алгоритму розв'язання прикладу 2

Код програми представлений на рисунку 2.36. Результат виконання програми на рисунку 2.37.

```
File Edit Format Run Options Window Help
import random
print('Визначити кількість від'ємних елементів в кожному стовпці двовимірного ма
B = [[round(random.uniform(-100,100),2) for j in range(3)] for i in range(4)]
print('Згенерований масив:')
for i in B:
    for j in i:
        print('%6.2f'%j, end=' ')
    print()
for j in range(3):
    k=0
    for i in range(4):
        if B[i][j]<0: k+=1
    print('Кількість від'ємних елементів у %d стовпці = %d'%(j+1,k))
```

Рис. 2.36 Код програми до прикладу 2

```
Визначити кількість від'ємних елементів в кожному стовпці двовимірного
масиву B(4,3)
Згенерований масив:
 70.22 -24.67 -56.14
 -0.15 -99.22  11.69
 87.96 -19.70  -7.24
 24.38  26.82 -63.01
Кількість від'ємних елементів у 1 стовпці = 1
Кількість від'ємних елементів у 2 стовпці = 3
Кількість від'ємних елементів у 3 стовпці = 3
```

Рис. 2.37 Вікно результатів розв'язання задачі з прикладу 2

Приклад 3. Поміняти місцями перший і максимальний елемент масиву A(3, 3). Вивести новий масив.

Алгоритм розв'язання задачі представлений на рисунку 2.38.

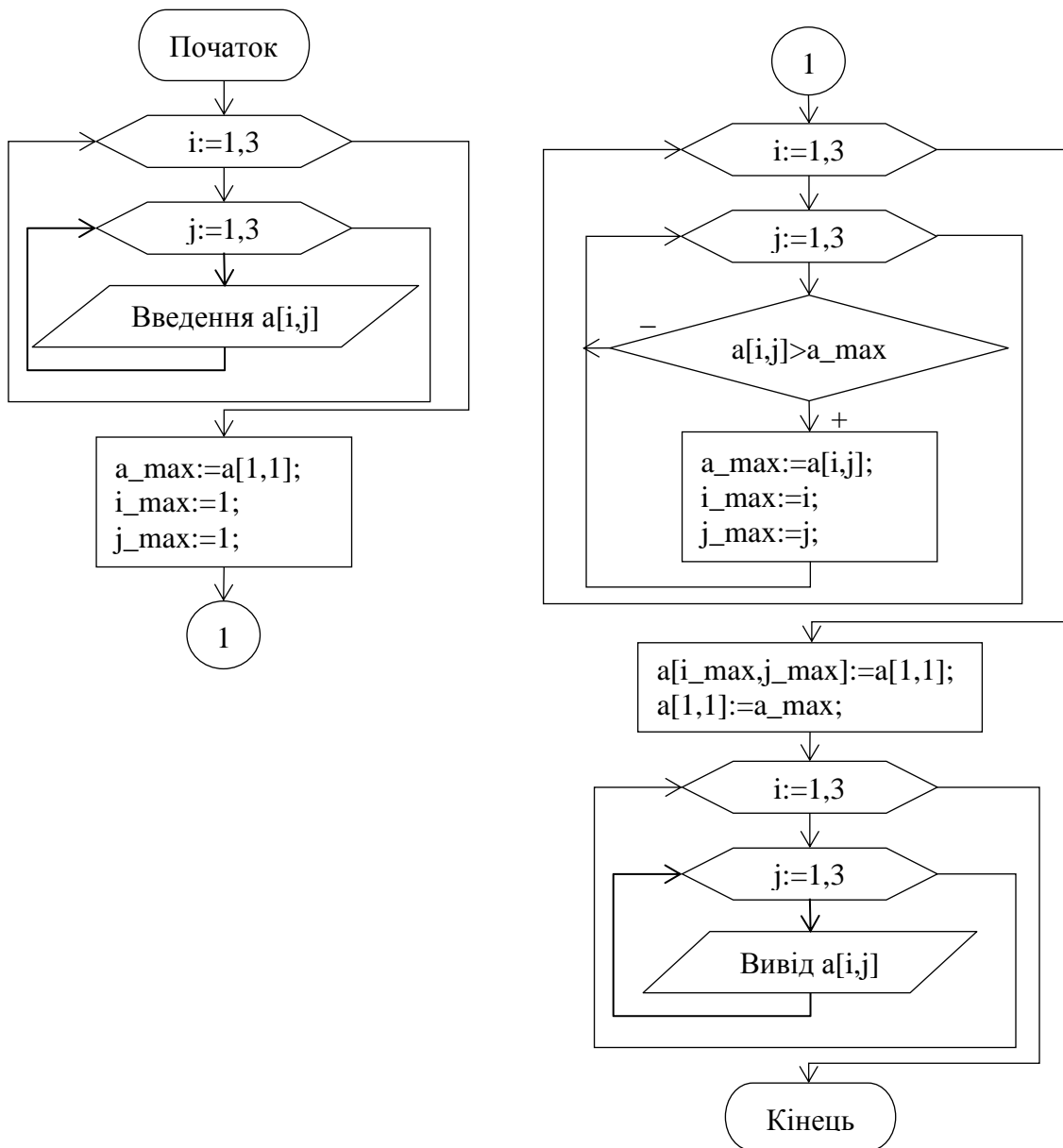


Рис. 2.38 Схема алгоритму розв'язання прикладу 3

Код програми представлений на рисунку 2.39. Результат виконання програми на рисунку 2.40.

```
File Edit Format Run Options Window Help
print('Поміняти місцями перший і максимальний елемент масиву A(3, 3).
print('Введіть 9 елементів масиву A:')
a = [[float(input()) for j in range(3)] for i in range(3)]
print('Заданий масив:')
for i in a:
    for j in i:
        print('%6.1f'%j, end=' ')
    print()
a_max=a[0][0]
i_max=0
j_max=0
for i in range(3):
    for j in range(3):
        if a[i][j]>a_max:
            a_max=a[i][j]
            i_max=i
            j_max=j
a[i_max][j_max]=a[0][0]
a[0][0]=a_max
print('Новий масив:')
for i in a:
    for j in i:
        print('%6.1f'%j, end=' ')
    print()
```

Рис. 2.39 Код програми до прикладу 3

```
File Edit Shell Debug Options Window Help
Поміняти місцями перший і максимальний елемент масиву A(3, 3). Вивести
новий масив.
Введіть 9 елементів масиву A:
-7
6.2
3
4
100
8.2
0.1
3
5
Заданий масив:
-7.0    6.2    3.0
 4.0  100.0  8.2
 0.1    3.0    5.0
Новий масив:
100.0    6.2    3.0
 4.0   -7.0    8.2
 0.1    3.0    5.0
```

Рис. 2.40 Вікно результатів розв'язання задачі з прикладу 3

Завдання до лабораторної роботи

Скласти алгоритм розв'язування задачі відповідно до варіанта та зобразити його графічно у вигляді блок-схеми. Записати алгоритм у вигляді коду програми на Python. Виконати складену програму для деяких вхідних даних. Передбачити виведення на екран початкового масиву даних та результату роботи програми.

У звіт подати:

- умову задачі варіанта;
- блок-схему алгоритму розв'язування задачі;
- складений Python-код;
- результат виконання програми для деяких вхідних даних.

Варіанти завдання

1. Визначити суму від'ємних елементів в кожному стовпці двовимірного масиву $Y(6, 4)$.
2. Визначити суму та кількість додатних елементів двовимірного масиву $A(4, 4)$.
3. Визначити кількість ненульових елементів в кожному стовпці двовимірного масиву $B(6, 4)$.
4. Визначити номери від'ємних елементів в кожному рядку двовимірного масиву $C(4, 5)$.
5. Визначити мінімальний елемент і його індекси у двовимірному масиві $E(4, 3)$.
6. Визначити суму додатних елементів в кожному рядку двовимірного масиву $D(5, 4)$.
7. Визначити для кожного стовпця двовимірного масиву $F(4, 6)$ суму елементів непарних рядків.
8. Перетворити двовимірний масив $G(6, 5)$, розділивши елементи кожного рядка на перший елемент цього рядка.
9. Визначити максимальний елемент і його індекси в двовимірному масиві $H(4, 5)$.

10. Визначити кількість додатних елементів в кожному рядку двовимірного масиву $P(3, 5)$.
11. У кожному стовпчику двовимірного масиву $Q(6, 3)$ визначити номери рядків, що містять нульові елементи.
12. Визначити суму та кількість від'ємних елементів двовимірного масиву $R(5, 4)$.
13. Перетворити двовимірний масив $Z(4, 3)$, розділивши елементи кожного стовпця на останній елемент цього стовпця.
14. Визначити кількість додатних елементів в кожному стовпці двовимірного масиву $X(6, 5)$.
15. Визначити номери стовпців, що містять нульові елементи для кожного рядка двовимірного масиву $Y(4, 4)$.

РОЗДІЛ 3

ВИМОГИ ДО КАДРОВОГО ЗАБЕЗПЕЧЕННЯ ОБ'ЄКТУ ГАЛУЗІ

Попит на Python-розробників стрімко зростає в усьому світі, адже ця мова програмування використовується у багатьох галузях розробки. Серед основних напрямків – створення програмних додатків, веброзробка, геймдев, розробка IoT-рішень («інтернет речей»), машинне навчання, аналіз даних та багато інших інноваційних сфер.

Популярність Python обумовлена його універсальністю, простотою та ефективністю. Особливо значущою є роль Python у сферах, пов'язаних із обробкою великих даних, розробкою штучного інтелекту та машинного навчання. Галузі, такі як охорона здоров'я, освіта, фінанси, активно впроваджують рішення на основі Python для оптимізації процесів, автоматизації та впровадження передових технологій.

Хто такий розробник Python і що він робить?

Розробник Python – це фахівець, який використовує мову програмування Python для створення програмного забезпечення, веб-додатків, автоматизації процесів, роботи з даними, розробки штучного інтелекту та інших технічних задач. Python є однією з найбільш популярних мов програмування завдяки її простоті, гнучкості та широкому спектру застосувань.

Основні обов'язки розробника Python.

– Розробка програмного забезпечення: написання коду для додатків або систем; використання Python для створення десктопних, серверних чи мобільних додатків.

– Веб-розробка: розробка веб-додатків із використанням фреймворків, таких як Django, Flask або FastAPI; робота з бекендом для забезпечення взаємодії між сервером і користувачем.

– Аналіз даних та машинне навчання: обробка, аналіз та візуалізація даних за допомогою бібліотек (наприклад, Pandas, NumPy, Matplotlib,

Seaborn); реалізація моделей машинного навчання за допомогою TensorFlow, PyTorch або Scikit-learn.

– Автоматизація та написання скриптів: створення скриптів для автоматизації рутинних процесів (обробка файлів, веб-скрейпінг тощо); використання Python для інтеграції систем і автоматизації бізнес-процесів.

– Робота з базами даних: збереження, оновлення та отримання даних із баз за допомогою ORM (Object-Relational Mapping) або SQL-запитів.

– Розробка API: створення RESTful або GraphQL API для взаємодії між різними компонентами системи.

– Тестування та налагодження коду: пошук і виправлення помилок у коді; написання автоматизованих тестів для забезпечення якості.

– Участь у проєктному плануванні: співпраця з командою для визначення вимог до програмного забезпечення; Пропонування технічних рішень.

Навички, які потрібні розробнику Python:

– Глибоке знання Python та популярних бібліотек.

– Розуміння принципів ООП (об'єктно-орієнтованого програмування).

– Досвід роботи з базами даних (PostgreSQL, MySQL, MongoDB).

– Знання фреймворків для веб-розробки (Django, Flask тощо).

– Досвід роботи з системами контролю версій (наприклад, Git).

– Розуміння основ роботи мереж та протоколів (HTTP, HTTPS).

– Базові знання front-end технологій (HTML, CSS, JavaScript) будуть

плюсом.

Сфери, де працюють розробники Python:

– IT-компанії (стартапи та великі корпорації).

– Наукові установи (робота з даними та дослідження).

– Банківська сфера (розробка аналітичних систем і безпеки).

– Геймдев (скрипти для ігор і автоматизація процесів).

– Автоматизація бізнесу.

Розробник Python – це універсальний спеціаліст, який може працювати

над проєктами у найрізноманітніших галузях, від фінансів до розробки штучного інтелекту. Майбутнє Python-розробників виглядає надзвичайно перспективним. Ця мова відкриває безліч кар'єрних можливостей у таких сферах, як машинне навчання, штучний інтелект, аналіз даних, автоматизація та веброзробка. Python – це інструмент, який дозволяє не лише досягти професійного успіху, а й зробити внесок у розвиток сучасних технологій.

РОЗДІЛ 4
МЕТОДИКА ПРОФЕСІЙНОЇ ПІДГОТОВКИ ФАХІВЦІВ З ЦИФРОВИХ
ТЕХНОЛОГІЙ

ДИДАКТИЧНИЙ ПРОЕКТ КОНСУЛЬТАТИВНОГО ЗАНЯТТЯ З
ТЕМИ «ОСНОВИ ПРОГРАМУВАННЯ МОВОЮ PYTHON»
ДИСЦИПЛІНИ «ІНФОРМАТИКА ТА КОМП'ЮТЕРНІ ТЕХНОЛОГІЇ»
ДЛЯ ЗДОБУВАЧІВ ОСВІТИ СПЕЦІАЛЬНОСТІ
ПРОФЕСІЙНА ОСВІТА (ЦИФРОВІ ТЕХНОЛОГІЇ)

4.1 Вихідні дані:

Навчальний заклад: Бахмутський навчально-науковий професійно-педагогічний інститут Харківського національного університету імені В. Н. Каразіна.

Галузь знань: 01 Освіта / Педагогіка.

Спеціальність: 015 Професійна освіта (Цифрові технології).

Рівень вищої освіти: перший (бакалаврський).

Дисципліна: «Інформатика та комп'ютерні технології».

Тема консультативного заняття: «Основи програмування мовою Python».

Отже, дисципліна містить такі характеристики як:

кількість кредитів – 6 (денна та заочна форма навчання);

змістових модулів – 3;

загальна кількість годин для вивчення дисципліни – для денної форми навчання 180 навчальних годин, з яких: 90 годин самостійної роботи та 90 годин аудиторних занять (26 годин лекційних занять та 64 годин лабораторних занять).

Форми контролю: Іспит.

Великий обсяг навчального матеріалу, обширні, складні цілі навчання та великий відсоток часу, що відведено на самостійну роботу, обумовлюють

необхідність в проведенні консультативних занять для уточнення та пояснення навчального матеріалу з дисципліни «Інформатика та комп'ютерні технології».

Проектування цілей консультативного заняття представлено в таблиці 4.1 [3].

Таблиця 4.1

Цілі консультативного заняття

Цілі консультативного заняття	Цілі формування різних рівнів засвоєння навчального матеріалу	Умови досягнення	Результат у вигляді дій здобувачів освіти
1	2	3	4
1	З переліку визначень впізнавати основні поняття теми «Основи програмування мовою Python» такі, як «кінець рядка», «знак дорівнює», «вираз», «основна інструкція», «вкладений блок інструкції», «коментар», код.	Знати визначення понять «кінець рядка», «знак дорівнює», «вираз», «основна інструкція», «вкладений блок інструкції», «коментар», код.	Правильно названі з переліку основні поняття теми «Основи програмування мовою Python» такі, як «кінець рядка», «знак дорівнює», «вираз», «основна інструкція», «вкладений блок інструкції», «коментар», код.
2	Уміти розрізняти відмінності між поняттями «кінець рядка», «знак дорівнює», «вираз», «основна інструкція», «вкладений блок інструкції», «коментар», код.	Виконання дій першого рівня: правильно названі з переліку основні поняття теми «Основи програмування мовою Python» такі, як «кінець рядка», «знак дорівнює», «вираз», «основна інструкція», «вкладений блок інструкції», «коментар», код.	Вміло розпізнано відмінності між поняттями такими як «кінець рядка», «знак дорівнює», «вираз», «основна інструкція», «вкладений блок інструкції», «коментар», код.

1	2	3	4
3	Уміти аналізувати поняття кінець рядка», «знак дорівнює», «вираз», «основна інструкція», «вкладений блок інструкції», «коментар», код.	Виконання дій першого і другого рівнів: вміло розпізнано відмінності між поняттями кінець рядка», «знак дорівнює», «вираз», «основна інструкція», «вкладений блок інструкції», «коментар», код.	Правильно проаналізовано поняття кінець рядка», «знак дорівнює», «вираз», «основна інструкція», «вкладений блок інструкції», «коментар», код.
4	Уміти досліджені переваги та недоліки поняття кінець рядка», «основна інструкція», «вкладений блок інструкції», «коментар», код.	Виконання дій першого, другого і третього рівнів.	Правильно досліджено «основна інструкція», «вкладений блок інструкції», «коментар», код.

Наведемо перелік джерел інформації для підготовки студентів до консультації згідно з робочою програмою дисципліни «Інформатика та комп'ютерні технології». Нижче представлено перелік основної та допоміжної літератури, а також інформаційні ресурси для вивчення теми «Основи програмування мовою Python» та підготовки до консультативного заняття:

Рекомендована література та інформаційні ресурси

1. Васильєв О.М. Програмування мовою Python. Тернопіль : Навчальна книга – Богдан, 2019. 504 с.
2. Костюченко А. О. Основи програмування мовою Python: навчальний посібник. Чернігів : ФОП Баликіна С. М., 2020. 180 с.
3. Яковенко А. В. Основи програмування. Python. Частина 1 [Електронний ресурс]: підручник. Київ : КПІ ім. Ігоря Сікорського, 2018. 195 с.

3. Путівник мовою програмування Python: електронний підручник.
URL: <https://pythonguide.rozh2sch.org.ua/>.

4. Практикум з програмування мовою Python. URL:
<https://pythonexercises.rozh2sch.org.ua/>.

Визначимо найбільш складні для розуміння та засвоєння питання (табл. 4.2) [2].

Таблиця 4.2

Обрання питань для консультування та формулювання відповідей
на можливі питання

Теми (або тема) дисципліни	Зміст програми за кожною темою	Найбільш складні питання за темою	Відповіді на питання
1	2	3	4
Основи програмування мовою Python	Базові поняття мови Python	1. Синтаксис мови Python, сутність «виразів»?	1. Для надання значень змінним оператором присвоєння є знак дорівнює «=». Формат оператора: ім'я_змінної=вираз.
		2. Охарактеризуйте сутність «вкладеного блоку інструкцій»	2. Вкладений блок інструкцій Тобто вкладені інструкції об'єднуються в блоки за величиною відступів. Відступ може бути будь-яким, головне, щоб в межах одного вкладеного блоку відступ був однаковий. Не варто забувати про читабельність коду, так, відступ в 1 пропуск є малочитабельним. Рекомендується використовувати 4 пропуски. Можна бачити, що синтаксис оформлення основної інструкції та вкладеного блоку інструкцій істотно відрізняється від синтаксису більшості мов, в яких використовуються операторні

			дужки для виділення вкладеного блоку інструкції (наприклад, begin ... end в Паскалі або { ... } в Сі).
--	--	--	--

Оберемо методи активізації навчальної діяльності студентів на консультації (табл. 4.3) [3].

Таблиця 4.3

Методи активізації навчальної діяльності здобувачів вищої освіти на консультації

Дидактичні методи	Реалізація методів при проведенні консультаційного заняття
1	2
Методи підвищення наочності	Використання інтерактивної дошки для демонстрації слайдів з теми «Основи програмування мовою Python»
Мотиваційні методи	Для реалізації мотивації використаємо: тип: внутрішня мотивація; вид: вступна мотивація; метод: мотивуючий вступ; прийом: віднесення до особистості. Повідомлення важливості вивчення даної теми: «Тема «Основи програмування мовою Python». Вивчення цієї теми для вас, як майбутніх інженерів-педагогів з цифрових технологій, є на сьогоднішній час дуже актуальною. Відповідно до вашої професійної діяльності знання цієї теми знадобляться вам у Вашій професійній діяльності. Це питання особливо набуває значення для програмістів та фахівців з цифрових технологій..
Проблемні методи	Використання проблемного питання. Проблемні питання: «Яким чином використовуємо коментарі? Обґрунтуйте відповідь.
Комунікативні методи	Імітація ситуацій з реального життя. «Яким чином ви будете використовувати мову Python, наведіть приклади»

Далі необхідно здійснити вибір способів організації консультативного заняття. Він здійснюється з урахуванням даних, наведених в таблиці 4.4.

Варіанти організації консультативного заняття [1]

№ варіанта	Етапи організації заняття	Характеристика варіанта
1	2	3
1	<ul style="list-style-type: none"> - вступне слово лектора, - відповіді на питання студентів і обговорення їх, - заключне слово викладача 	Недоліком цього варіанту проведення лекції-консультації є відсутність послідовності, системи в питаннях, на які доводиться викладачу давати відповіді. Питання поступають хаотично, що знижує якість консультації.
2	<ul style="list-style-type: none"> - збір питань в письмовій формі до лекції, їх систематизація, - відповіді на питання, що поступили, - відповіді на додаткові питання, - обмін думками, - висновки 	Цей варіант, на відміну від попереднього, дозволяє викладачу групувати відповіді, що сприяє кращому засвоєнню навчального матеріалу студентами.
3	<ul style="list-style-type: none"> - видача завдань на самостійне вивчення матеріалу теми. - підготовка питань лектору. - відповіді і їх обговорення 	В цьому випадку консультування грає функцію додаткового інформування зі складних питань і пояснення незрозумілого навчального матеріалу.
4	<ul style="list-style-type: none"> - повідомлення теми, - консультування декількома фахівцями в певній області науки і техніка з актуальних питань науки і нової техніки 	Цей варіант лекції-консультації проводиться, як правило, зі спеціальних дисциплін, іноді для цієї мети використовуються наукові семінари. Такі заняття дають можливість зіставити думки різних учених на одну і ту ж проблему і є чудовою школою ведення дискусії.

Згідно представленої таблиці обираємо 1 варіант організації консультативного заняття, на якому викладач пояснює питання, які здалися незрозумілими студентам.

Наводимо розробку сценарію проведення консультативного заняття у відповідності до обраного варіанту його організації (табл. 4.5) [2].

Таблиця 4.5

Сценарій консультативного заняття

Етапи проведення консультативного заняття	Дії викладача	Дії здобувачів вищої освіти
1	2	3
Організаційний момент	Викладач вітає студентів, робить перекличку, пропонує студентам розпочати роботу на консультації.	Студенти вітають викладача, беруть участь у перекличці, налаштовуються на роботу на консультації.
Повідомлення теми і мети консультації	Повідомлення теми заняття «Основи програмування мовою Python»	Фіксація теми, сприйняття цілей, представлення результатів засвоєння матеріалу теми даного заняття
Мотивація мети	Повідомлення важливості вивчення даної теми: «Основи програмування мовою Python». Вивчення цієї теми для вас, як майбутніх фахівців з цифрових технологій, є на сьогоднішній час дуже актуальною. Відповідно до вашої професійної діяльності знання цієї теми знадобляться вам для ефективного реалізації професійних завдань.	Сприйняття важливості і актуальності вивчення теми, прояв інтересу до неї.
Актуалізація знань	Викладач проводить фронтальне усне опитування з метою перевірки базових знань: 1. Як ви розумієте значення поняття «синтаксис мови» 2. Схарактеризуйте вкладений блок інструкції?	Здобувачі освіти беруть участь у опитуванні

1	2	3
Формування ООД	<p>Викладач проводить консультацію згідно плану, за допомогою методу - пояснення</p> <p>План</p> <ol style="list-style-type: none"> 1. Синтаксис мови. 2. Основні твердження. 	Слухають пояснення, конспектують.
Визначення проблемних моментів під час вивчення питань теми та формування ВД	<p>Викладач запитує здобувачів освіти про недоречності, які виникли у них під час самостійного вивчення теми. Викладач відповідає на поставлені запитання.</p> <p>Коментар – це довільний текст у будь-якому місці програми, що пишеться після символу #, і представляє інтерес лише як замітка для того, хто буде переглядати код програми.</p> <p>Наприклад:</p> <pre>- print('Привіт, Світ!') # print - це функція або: # print - це функція - print('Привіт, Світ!')</pre> <p>Коментарі містять пояснювальні тексти і полегшують читання і розуміння програм, тому намагайтеся в своїх програмах писати якомога більше корисних коментарів, які містять: припущення, важливі рішення, важливі деталі, проблеми, які ви намагаєтеся вирішити, проблеми, яких ви намагаєтеся уникнути і т.д.</p>	Студенти запитують: 1. Назвіть, що означає «Коментар, мовою Python»?

1	2	3
Підведення підсумків	Викладач підводить підсумки проведення консультації: «Сьогодні ми розглянули незрозумілі вам питання теми.	Здобувачі освіти слухають, відповідають.

Отже, на даному етапі ми підготували детальний сценарій для проведення консультативного заняття, орієнтуючись на вибраний варіант організації цього процесу.

Контурний конспект заняття з теми «Основи програмування мовою Python» представлено у Додатку А.

ВИСНОВКИ

У кваліфікаційній роботі відповідно до мети і завдань розкрито стан наукової проблеми.

Установлено, що питання, що порушено у кваліфікаційній роботі до цього часу не були предметом вивчення науковців.

Виявлено, що професійна підготовка інженерів-педагогів з цифрових технологій включає всі сторони розвитку особистості: здібності, цілеспрямованість, наполегливість у досягненні результатів навчання; завдяки формуванню культури формується соціальний досвід, розвиваються моральні сторони особистості.

Автором уточнено поняття «підготовка інженерів-педагогів у галузі цифрових технологій до реалізації функцій професійної діяльності», що розглядається як безперервний, цілеспрямований процес з виявлення відповідності сформованої педагогічної практики сучасним вимогам, усунення виявлених професійних дефіцитів, результатом якого є підвищення рівня готовності до реалізації професійної діяльності у сучасних умовах. Під професійними дефіцитами інженерів-педагогів ми розуміємо неузгодженість між існуючим і необхідним рівнем знань, умінь і способів діяльності фахівців з цифрових технологій, що призводить до невідповідності сформованого педагогічного досвіду сучасним вимогам.

Розроблено діагностичний інструментарій, у тому числі рефлексивний, що включає авторську методичку, що дозволяє вивчити рівень готовності інженерів-педагогів до реалізації професійної діяльності в сучасних умовах (низький, середній та високий) за компонентами готовності до реалізації професійної діяльності ціннісно-змістовий (усвідомлюю значущість, цінність інженерно-педагогічної діяльності в сучасних умовах), когнітивний (знаю, вмю), операційно-діяльнісний (застосовую, роблю) та виявити професійні дефіцити інженерів-педагогів як основу адресної підготовки.

У другому розділі магістерської роботи надана характеристика та

аналіз нормативно-методичної документації з дисципліни «Інформатика та комп'ютерні технології» зокрема розділу «Основи алгоритмізації та програмування». Зроблено висновок про необхідність підготовки презентацій до лекцій з основ програмування та поставки лабораторних робіт.

В ході виконання магістерської роботи підготовлено три презентації до лекційних занять за темами:

- циклічні оператори;
- робота зі списками;
- масиви в Python.

Розмір кожної презентації складає 15-20 слайдів.

Поставлено три лабораторні роботи «Програмування алгоритмів циклічної структури», «Проектування алгоритмів з використанням одновимірних масивів даних» та «Опрацювання двовимірних масивів даних». В інструкціях до лабораторних робіт наведено: мета та порядок виконання роботи; короткі теоретичні відомості, приклади розв'язання задач з побудовою блок-схем алгоритмів та програм на мові Python; підібрано варіанти завдань для самостійного виконання.

У третьому розділі магістерської роботи розглянуто питання перспективного розвитку кар'єри для фахівця, який володіє мовою програмування Python.

У четвертому розділі розроблено дидактичний проект консультативного заняття з теми «Основи програмування мовою Python» дисципліни «Інформатика та комп'ютерні технології» для здобувачів інженерно-педагогічної спеціальності «Професійна освіта.(Цифрові технології)».

Сформульовано цілі консультативного заняття. Обрано методи активізації навчальної діяльності студентів на консультації. Здійснено вибір способів організації консультативного заняття.

Розроблено сценарій проведення консультативного заняття у відповідності до обраного варіанту його організації.

Проаналізовано джерела інформації для підготовки студентів до консультації згідно з робочою програмою дисципліни. Подано список використаних джерел та відповідні посилання.

За основними результатами дослідження виконана публікація тез доповіді на VIII Міжнародній науково-практичній конференції здобувачів вищої освіти та молодих учених «Студенти та молодь – для майбутнього країни» (м. Харків, 14-15 листопада 2024 р.). Текст тез доповіді наведено у додатку Б.