

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ІРПІНСЬКИЙ ФАХОВИЙ КОЛЕДЖ ЕКОНОМІКИ ТА ПРАВА

ЗАТВЕРДЖУЮ

Заступник директора з навчальної роботи

 Олена ПУСТОВА

«01» 09 2023 р.

РОБОЧА ПРОГРАМА

навчальної дисципліни «Алгоритмізація та програмування»

для підготовки фахового молодшого бакалавра

за галуззю знань 12 Інформаційні технології

спеціальності 122 «Комп'ютерні науки»

Робоча програма навчальної дисципліни «Алгоритмізація та програмування» складена на основі програми навчальної дисципліни, затвердженої у 2023 році.

Розробник:

Анастасія ІЩУК

Розглянуто і схвалено на засіданні циклової комісії інформаційних технологій, математичних та природничих дисциплін, протокол №02 від «31» серпня 2023 р.

Голова циклової комісії



Анастасія ІЩУК

Завідувач навчально-методичного
кабінету коледжу



Людмила РОМАНЕНКО

ЗМІСТ

ПЕРЕДМОВА.....	4
КРИТЕРІЇ ОЦІНЮВАННЯ	5
ЗМІСТ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ.....	7
СПИСОК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ ІНФОРМАЦІЇ	16

ПЕРЕДМОВА

Спеціаліста будь-якого фаху в сучасних умовах навіть неможливо уявити без уміння використання комп'ютерної техніки. Особливо це стосується спеціалістів, чия діяльність безпосередньо пов'язана з використанням ПК. Створення сучасних програмних продуктів вимагає досконалого знання мов програмування як високого так низького рівня, а також уміння ефективної організації структур даних та розробки алгоритмів їх обробки. Даний курс призначений для навчання студентів основ алгоритмізації та здобуття навиків у реалізації алгоритмів на мові програмування високого рівня Python.

Мета: формування у студентів алгоритмічного мислення та базових знань з алгоритмізації опрацювання фундаментальних структур даних; засвоєння теоретичних концепцій сучасних парадигм програмування та методологій проектування програм; вивчення принципів програмування на мові Python, отримання практичних навичок створення прикладних програм і реалізації алгоритмів обробки інформації на мові Python. Особлива увага приділяється розвитку практичних навиків програмування з обробки різних типів даних та проведення їх аналізу.

Вивчення навчальної дисципліни «Алгоритмізація та програмування» передбачає набуття здобувачами освіти програмних компетентностей, а саме:

Загальні компетентності (ЗК):

ЗК 3. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК 4. Здатність застосовувати знання у практичних ситуаціях.

ЗК 5. Знання та розуміння предметної області та розуміння професійної діяльності.

ЗК 8. Здатність вчитися і оволодівати сучасними знаннями.

Спеціальні компетентності (СК):

СК 2. Здатність використовувати теоретичні та фундаментальні знання в галузі комп'ютерних наук та інформаційних технологій для вирішення різноманітних проблем.

СК 3. Здатність розробляти, аналізувати та застосовувати ефективні алгоритми для розв'язання конкретних професійних задач залежно від предметного середовища

СК 8. Здатність застосовувати сучасні методи, технології та інструментальні засоби проектування й створення програмних систем та їх супроводження.

Очікувані результати навчання

Результати навчання (РН):

РН 4. Застосовувати сучасні методи математичного та комп'ютерного моделювання і будувати ефективні алгоритми для чисельного дослідження та розв'язання прикладних задач.

РН 9. Застосовувати сучасний інструментарій комп'ютерної графіки та анімації під час вирішення практичних задач професійної діяльності.

РН 10. Знати методології, методи, моделі, процеси і технології життєвого циклу розробки та тестування програмного забезпечення.

РН 11. Застосовувати сучасні мови програмування та технології для розробки програмного забезпечення розподілених систем.

PH 12. Знати основні принципи функціонування системного та прикладного програмного забезпечення.

PH 15. Розробляти супровідну документацію на різних етапах процесу життєвого циклу розробки програмного забезпечення.

Передумовами вивчення даної дисципліни є вивчення шкільного курсу «Інформатика». Дисципліна «Алгоритмізація та програмування» є необхідною для подальшого вивчення дисциплін «Теорія алгоритмів», «Об'єктивно-орієнтоване програмування», «Крос-платформне програмування», «Технологія створення програмних продуктів», «Технології захисту інформації» та ін..

В процесі викладання дисципліни використовується **5-бальна система оцінювання**.

У навчальному процесі використовуються такі методи навчання:

- лекція;
- лабораторна робота;
- самостійна робота.

Оцінювання знань студентів здійснюється у формі усного опитування, тестів, оцінювання лабораторних робіт. Після вивчення навчальної дисципліни «Алгоритмізація та програмування» студенти складають екзамен.

Форми та засоби поточного і підсумкового контролю

Об'єктивність оцінки знань студентів залежить від форм та засобів поточного й підсумкового контролю. Процес вивчення курсу передбачає здійснення поточного та підсумкового контролю. Форми проведення поточного контролю, їх періоди визначаються робочим планом викладача. Поточний контроль проводиться викладачем на лекційних, контрольних, лабораторних заняттях, під час перевірки самостійних робіт. Формою підсумкового семестрового контролю є екзамен, що визначається навчальним планом. Підсумковий контроль здійснюється після вивчення курсу під час екзамену. До екзамену допускаються студенти, які засвоїли всі теми програми, виконали лабораторні, рубіжну контрольну роботу.

КРИТЕРІЇ ОЦІНЮВАННЯ

(5-бальна шкала)

Критерії оцінки знань і вмінь

"відмінно" – «5» (високий рівень) – якщо студент вільно, глибоко й у повному обсязі засвоїв програмний матеріал; вичерпно, логічно викладає теоретичний матеріал в усній і письмовій формі; швидко і впевнено приймає правильні рішення при виконанні практичних завдань; має стійкі навички рішення різних задач;

"добре" – «4» (достатній рівень) – якщо студент вільно володіє матеріалом; у повному обсязі засвоїв програмний матеріал; здатний самостійно приймати правильні рішення при виконанні практичних завдань; має добрі навички рішення практичних задач;

"задовільно" – «3» (середній рівень) якщо студент може самостійно виконати більшу частину навчального матеріалу, дати основні поняття та визначення; допускає неточності при прийнятті рішень; при вирішенні практичних завдань потребує додаткових вказівок.

"незадовільно" – «2» (початковий рівень) – якщо студент засвоїв і може відтворити лише окремі питання програми; допускає суттєві помилки при прийнятті рішень; не має стійких навичок при вирішенні практичних завдань.

ЗМІСТ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Тема №1 Сучасні мови програмування. Середовище програмування. Типи даних. Величини. Змінні. Лінійні алгоритми.

План лекційного заняття №1:

1. Огляд сучасних мов програмування, їх особливості.
2. Основні можливості мови Python. Особливості мови Python.
3. Переваги мови Python, її використання.
4. Програмне забезпечення.
5. Інтерпретатор з інтерактивною оболонкою IDLE.
6. Інтегроване середовище програмування PyCharm.
7. Встановлення і запуск середовища програмування.
8. Тип даних. Величини. Змінні. Вказівка присвоювання.
9. Модулі. Функції.
10. Створення лінійних алгоритмів зі змінними.

Самостійна робота:

- ✓ Сфери використання Python, реалізації, місце у сучасній ІТ-індустрії.

Перелік рекомендованих джерел інформації

Основний: 1, 2, 3, 5, 6.

Додатковий: 1, 2, 3, 4.

ЛАБОРАТОРНА РОБОТА №1. Основи мови Python.

Мета: ознайомитися з алгоритмами послідовної (лінійної) структури, з процедурами запуску програм, які реалізують ці алгоритми на мові Python; знайомство з інтегрованим середовищем розробки – *integrated development environment (IDLE)*.

ЛАБОРАТОРНА РОБОТА №2. Модулі, функції і методи опрацювання числових даних.

Мета: навчитися розробляти алгоритми обчислення математичних виразів, написання відповідних програм на мові програмування.

Тема №2 Величини в алгоритмах з розгалуженням.

План лекційного заняття №2:

1. Поняття алгоритму з розгалуженням.
2. Повне та неповне розгалуження.
3. Логічні операції.

Самостійна робота:

- ✓ Операції у логічних виразах.

Перелік рекомендованих джерел інформації

Основний: 1, 3, 4, 5.

Додатковий: 1, 2, 3, 4.

ЛАБОРАТОРНА РОБОТА №3. Алгоритми з розгалуженням.

Мета: навчитися розробляти алгоритми структурою розгалуження (*if, if-else, if-elif-else*).

Тема №3 Величини в алгоритмах із циклами. Випадкові числа.

План лекційного заняття №3:

1. Величини в циклах з лічильником
2. Величини в циклах з передумовою
3. Цикл з параметром.
4. Шкала чисел.
5. Випадкові числа. Модуль випадкових чисел.

Самостійна робота:

- ✓ Випадкові числа та розподіли: модуль `random` та його застосування.
- ✓ Особливості генерації випадкових чисел.

Перелік рекомендованих джерел інформації

Основний: 1, 2, 3, 4, 5.

Додатковий: 1, 2.

ЛАБОРАТОРНА РОБОТА №4. Проектування алгоритмів і програм циклічної структури. Оператор циклу з параметром.

Мета: ознайомитися з циклічною конструкцією *for* і її використанням в мові *Python*; навчитися працювати з числами і рядками використовуючи дані структури.

ЛАБОРАТОРНА РОБОТА №5. Проектування алгоритмів і програм циклічної структури. Оператор циклу з передумовою.

Мета: ознайомитися з циклічною конструкцією *while* і її використанням в мові *Python*; навчитися працювати з числами і рядками використовуючи дані структури.

Тема №4 Рядки. Підрядок.

План лекційного заняття №4:

1. Створення і заповнення рядка.
2. Підстановка значень у рядки.
3. Методи рядків.
4. Експрес-послідовності.
5. Зрізи.
6. Операції з рядками.
7. Пошук в рядках.
8. Індексція символів у рядку. Зріз рядка (підрядок).

Самостійна робота:

- ✓ Метод `format`.
- ✓ Порівняння рядків.

Перелік рекомендованих джерел інформації

Основний: 1, 2, 3, 4, 5.

Додатковий: 1, 2, 3, 4.

ЛАБОРАТОРНА РОБОТА №6. Рядки.

Мета: ознайомитися з рядками в мові Python, діями над ними.

ЛАБОРАТОРНА РОБОТА №7. Рядки. Методи.

Мета: Робота з рядками у Python. Методи.

Тема №5 Списки.

План лекційного заняття №5:

1. Операції над списками.
2. Використання вбудованих методів списку: додавання, видалення, заміна та інші.
3. Створення пустого списку з наступним заповненням його випадковими числами.

4. Генератор списків.

Самостійна робота:

- ✓ Створення списків.
- ✓ Comprehension.
- ✓ Порядок.
- ✓ Індексція.
- ✓ Slicing .
- ✓ Ітерованість.
- ✓ Змінюваність списків.
- ✓ Функції над списками.
- ✓ Інкрементальні операції над списками.
- ✓ Порівняння списків.
- ✓ Список як параметр функції.

Перелік рекомендованих джерел інформації

Основний: 1, 2, 3, 4, 5, 6.

Додатковий: 1, 2, 3, 4.

ЛАБОРАТОРНА РОБОТА №8. Списки.

Мета: Робота зі списками у Python.

ЛАБОРАТОРНА РОБОТА №9. Списки. Методи.

Мета: Робота з методами у списках Python.

Тема №6 Кортеж. Діапазони. Множини. Словник.

План лекційного заняття №6:

1. Кортеж. Доступ до елементів кортежу.
2. Перетворення кортежу в список і назад.
3. Діапазон. Перетворити діапазон у список і кортеж.
4. Дії з діапазонами.
5. Множина. Типи множин.
6. Оператори, функції і методи операцій над множинами.
7. Генератор множин.
8. Що таке словник.

9. Поняття хеш-таблиці.
10. Синтаксис словника.
11. Вбудовані методи словників.

Самостійна робота:

- ✓ Створення словників.
- ✓ Змінюваність словників.
- ✓ Невпорядкованість.
- ✓ Ітерованість.
- ✓ Функції над словниками. Порівняння словників.

Перелік рекомендованих джерел інформації

Основний: 1, 2, 3, 4, 5, 6.

Додатковий: 1, 2, 3, 4.

ЛАБОРАТОРНА РОБОТА №10. Кортежі.

Мета: Робота з кортежами Python.

ЛАБОРАТОРНА РОБОТА №11. Словники.

Мета: Робота зі словниками Python.

ЛАБОРАТОРНА РОБОТА №12. Діапазони. Множини.

Мета: Робота з діапазонами та множинами Python.

Тема №7. Масиви. Вказівники.

План лекційного заняття №7:

1. Одновимірний масив.
2. Структура одновимірного масиву.
3. Створення масиву в процесі його оголошення
4. Створення масиву шляхом обчислення значень його елементів
5. Уведення масиву з клавіатури та операції з елементами масиву.

Самостійна робота:

- ✓ Методи впорядкування масивів.

Перелік рекомендованих джерел інформації

Основний: 1, 2, 3, 4, 5, 6.

Додатковий: 1, 2, 3, 4.

ЛАБОРАТОРНА РОБОТА №13. Одновимірні масиви.

Мета: Робота з одновимірними масивами Python.

Тема №8 Двовимірні масиви. Робота з масивами.

План лекційного заняття №8:

1. Двовимірні масиви.
2. Операції з масивами.
3. Перебір елементів матриці.

4. Квадратні матриці. Головна та побічна діагоналі.

Самостійна робота:

- ✓ Елементи програмування. Обробка двовимірних масивів.
- ✓ Впорядкування масивів.
- ✓ Бінарний пошук.

Перелік рекомендованих джерел інформації

Основний: 1, 2, 3, 4, 5, 6.

Додатковий: 1, 2, 3, 4.

ЛАБОРАТОРНА РОБОТА №14. Двовимірні масиви.

Мета: *Робота з двовимірними масивами Python.*

Тема №9 Файли. Робота з файлами в середовищі програмування Python.

План лекційного заняття №9:

1. Відкриття текстового файлу.
2. Режими відкриття файлу.
3. Запис та дописування даних у файл.
4. Порядкова робота з файлами.
5. Створення та видалення каталогів.
6. Отримання списку вмісту каталогу.
7. Отримання метаданих файлу.
8. Перейменування файлу.
9. Видалення файлу.
10. Копіювання даних у файлі. Копіювання файлу. Копіювання папок.
11. Зчитування даних з текстових файлів.

Самостійна робота:

- ✓ Бінарний і текстовий режими.

Перелік рекомендованих джерел інформації

Основний: 1, 2, 3, 4, 5, 6.

Додатковий: 1, 2, 3, 4.

ЛАБОРАТОРНА РОБОТА №15. Робота з файлами у Python.

Мета: *ознайомитися з засобами роботи з файлами в мові Python, діями над ними.*

Тема №10 Графічні примітиви.

План лекційного заняття №10:

1. Система координат на полотні в Python
2. Основні графічні команди.

Самостійна робота:

- ✓ Створення зображення і анімації.
- ✓ Заливка зображень.

Перелік рекомендованих джерел інформації

Основний: 1, 2, 3, 4, 5, 6.

Додатковий: 1, 2, 3, 4.

ЛАБОРАТОРНА РОБОТА №16. Робота з графічними примітивами у Python. Створення зображень.

Мета: *ознайомитися з методами роботи в мові Python з графічними примітивами.*

Тема №11 Бібліотека Matplotlib.

План лекційного заняття №11:

1. Встановлення Matplotlib.
2. Основні елементи графіку.
3. Основні графічні команди бібліотеки Matplotlib.
4. Робота з текстом.
5. Структура рисунка в matplotlib.
6. Кординатні вісі.
7. Легенди.
8. Кольорова шкала.
9. Побудова графіків та діаграм.
10. Робота з лінійним графіком. Тип лінії.
11. Кілька графіків (діаграм) на одному полі. Розміщення графіків (діаграм) на різних полях.
12. Збереження графіку як зображення в Matplotlib. Збереження графіку у різних форматах.

Самостійна робота:

- ✓ Візуалізація даних. Ступінчастий, стековий, точковий графік.
- ✓ Побудова гістограм, кругових діаграм, графіків розсіювання та полярних діаграм у Python.
- ✓ Побудова кольорової сітки.
- ✓ Побудова 3D графіків.

Перелік рекомендованих джерел інформації

Основний: 1, 2, 3, 4, 5, 7, 8, 9, 10.

Додатковий: 1, 2, 3, 4.

ЛАБОРАТОРНА РОБОТА №17. Побудова графіків візуалізації даних в Python за допомогою бібліотеки Matplotlib.

Мета: *ознайомитись з пакетом matplotlib, освоїти основні принципи програмування 2D "наукової графіки".*

ЛАБОРАТОРНА РОБОТА №18. Побудова діаграм в Python за допомогою бібліотеки Matplotlib.

Мета: *ознайомитись з пакетом matplotlib, освоїти основні принципи програмування 2D "наукової графіки".*

Тема №12 Бібліотеки NUMPY, SciPy.

План лекційного заняття №12:

1. Створення масиву.
2. Матриці. Операції над матрицями.
3. Доступ до елементів, зрізи.
4. Форма масиву і її зміна.
5. Об'єднання масивів.
6. Клонування даних.
7. Матричне множення.
8. Агрегатори.

Самостійна робота:

- ✓ SciPy Sub-пакети.
- ✓ Інтегрування.
- ✓ Інтерполяція.
- ✓ Сплайни.
- ✓ Лінійна алгебра в SciPy – Linalg.

Перелік рекомендованих джерел інформації

Основний: 1, 2, 3, 4, 5, 6.

Додатковий: 1, 2, 3, 4.

ЛАБОРАТОРНА РОБОТА №19. Пакет NumPy. Основні принципи роботи.

Мета: *Навчитися виконувати операції над масивами і матрицями, користуючись модулем NumPy (Numeric Python).*

Тема №13 Введення в ООП.

План лекційного заняття №13:

1. Self. Класи.
2. Методи об'єктів. init .
3. Змінні класу і об'єктів.
4. Спадковість.
5. Основи написання класів.
6. Деталі реалізації класів.

Самостійна робота:

- ✓ Метакласи.
- ✓ Проектування з використанням класів.

Перелік рекомендованих джерел інформації

Основний: 1, 2, 3, 4, 5,6.

Додатковий: 1, 2, 3, 4.

ЛАБОРАТОРНА РОБОТА №20. Python ООП. Класи.

Мета: *Отримати практичні навички використання класів Python.*

ЛАБОРАТОРНА РОБОТА №21. Отримання практичних навичок написання програм на мові Python.

Мета: *Відпрацювати практичні навички написання програм на мові Python.*

ПЕРЕЛІК ПИТАНЬ ДЛЯ РУБІЖНОГО КОНТРОЛЮ

з курсу «Алгоритмізація та програмування»

1. Огляд сучасних мов програмування, їх особливості.
2. Основні можливості мови Python. Особливості мови Python.
3. Переваги мови Python, її використання.
4. Програмне забезпечення.
5. Інтерпретатор з інтерактивною оболонкою IDLE.
6. Інтегроване середовище програмування PyCharm.
7. Тип даних. Величини. Змінні. Вказівка присвоювання.
8. Модулі. Функції.
9. Математичні оператори.
10. Створення лінійних алгоритмів зі змінними.
11. Сфери використання Python, реалізації, місце у сучасній IT-індустрії.
12. Поняття алгоритму з розгалуженням.
13. Повне та неповне розгалуження.
14. Логічні операції.
15. Операції у логічних виразах.
16. Оператори порівняння.
17. Величини в циклах з лічильником
18. Величини в циклах з передумовою
19. Цикл з параметром.
20. Шкала чисел.
21. Випадкові числа. Модуль випадкових чисел.
22. Випадкові числа та розподіли: модуль random та його застосування.
23. Особливості генерації випадкових чисел.
24. Створення і заповнення рядка.
25. Підстановка значень у рядки.
26. Методи рядків.
27. Експрес-послідовності.
28. Зрізи.
29. Операції з рядками.
30. Пошук в рядках.
31. Індексція символів у рядку. Зріз рядка (підрядок).
32. Методи рядків.
33. Порівняння рядків.

ПЕРЕЛІК ПИТАНЬ ДО ЕКЗАМЕНУ

1. Сучасні мови програмування. Середовище програмування. Структура програми.
2. Що таке алгоритм послідовної (лінійної) структури, програма послідовної (лінійної) структури?
3. Лінійні алгоритми.
4. Які основні типи даних визначені в мові Python.
5. Числові типи даних у Python.
6. Цілочисельне ділення у Python.
7. Що таке алгоритм розгалуженої структури?
8. Який синтаксис має оператор розгалуження (множинного розгалуження)?
9. Декрементация з range (). Поняття, приклади.
- 10.Способи виклику range()
- 11.Синтаксис мови програмування Python.
- 12.Основні найпростіші команди: інструкція, присвоєння, введення та виведення.
- 13.Логічний тип даних у Python.
- 14.Розгалуження у Python: умовний оператор, каскадне розгалуження, тернарний умовний оператор.
- 15.Цикл з умовою продовження: синтаксис та правило виконання.
- 16.Що таке цикл, для чого його використовують?
- 17.Як описується та виконується циклічна інструкція while?
- 18.Як описується та виконується циклічна інструкція for?
- 19.Переривання та продовження циклів: спеціальні команди break та continue.
- 20.Списки: означення та правила роботи. Створення. Індексация та зрізи.
- 21.Методи і функції по роботі зі списками.
- 22.Кортежі. Означення та правила роботи. Створення. Індексация та зрізи.
- 23.Операції над кортежами.
- 24.Рядки. Означення та правила роботи. Створення. Індексация та зрізи.
Операції над рядками. Форматування рядків.
- 25.Рядки. Методи split () і join ()
- 26.Словники. Створення словників. Звернення до елементів.
- 27.Множини. Поняття про множини. Створення множин. Ключі та значення.
Звернення до елементів.
- 28.Масиви. Одновимірні масиви.
- 29.Двовимірні масиви. Робота з масивами.
- 30.Робота з файлами в середовищі програмування Python. Основні особливості роботи з файлами.
- 31.Графічні примітиви. Методи роботи в мові Python з графічними примітивами.
- 32.Matplotlib. Основні графічні команди. Робота з текстом. Структура рисунка в matplotlib. Координатні вісі. Легенди. Кольорова шкала.

33. Matplotlib . Побудова графіків та діаграм. Робота з лінійним графіком. Тип лінії. Кілька графіків (діаграм) на одному полі. Розміщення графіків (діаграм) на різних полях.
34. Matplotlib . Збереження графіку як зображення в Matplotlib. Збереження графіку у різних форматах.
35. Matplotlib . Побудова 3D графіків.
36. Бібліотеки NUMPY, SciPy. Основні принципи роботи.
37. Введення в ООП. Self. Класи. Методи об'єктів. init . Змінні класу і об'єктів. Спадковість. Основи написання класів. Деталі реалізації класів.

СПИСОК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ ІНФОРМАЦІЇ

Основний:

1. 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.
2. Костюченко А.О. Основи програмування мовою Python: навчальний посібник. Ч.: ФОП Баликіна С.М., 2020. -180 с.
3. Основи програмування. Python. Частина 1 [Електронний ресурс]: підручник для студ. спеціальності 122 "Комп'ютерні науки", спеціалізації "Інформаційні технології в біології та медицині"/А.В. Яковенко; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 1,59 Мбайт). – Київ: КПІ ім. Ігоря Сікорського, 2018. – 195 с.
4. Козак Л. І. Основи програмування: навчальний посібник / Л. І. Козак, І. В. Костюк, С. П. Стасевич. – Львів: "Новий Світ-2000", 2019. – 326 с.
5. Ришковець Ю. В. Алгоритмізація та програмування. Ч. 1 : навчальний посібник / Ю. В. Ришковець, В. А. Висоцька. – Львів : "Новий Світ-2000", 2020. – 337 с.
6. Ришковець Ю. В. Алгоритмізація та програмування. Ч. 2 : навчальний посібник / Ю. В. Ришковець, В. А. Висоцька. – Львів : "Новий Світ-2000", 2020. – 314 с.
7. Duncan M. McGreggor. Mastering matplotlib. – Packt Publishing, 2015. – 292 р.
8. Alexandre Devert. Matplotlib Plotting Cookbook. – Packt Publishing, 2014. – 222 р.
9. Sandro Tosi. Matplotlib for Python Developers. – Packt Publishing, 2009. – 308 р.
10. Shai Vaingast. Beginning Python Visualization: Crafting Visual Transformation Scripts. – Springer, 2009. — 384 с.

Додатковий:

1. Васильєв О. М. Програмування мовою Python. Тернопіль: Навчальна книга – Богдан, 2019. – 504с.
2. Івашко В.В. Конспект лекцій з навчальної дисципліни «Основи програмування». Чернівці : Чернівецький національний університет імені Юрія Федьковича 2021. – 177 с.
3. Руденко В., Жугастров О. Інформатика. Основи алгоритмізації та програмування мовою Python. Харків: Ранок, 2019. – 192 с.
4. Копей В. Б. Мова програмування Python для інженерів і науковців : навч. посіб. / В. Б. Копей. - Івано-Франківськ : ІФНТУНГ, 2019. - 272 с.

Інформаційні ресурси в Інтернеті

1. <https://www.python.org/>
2. <https://repl.it>
3. <https://wingware.com/>
4. <https://www.jetbrains.com/pycharm-edu/?fromMenu>
5. <https://www.scipy.org/>
6. <https://python-patterns.guide/>
7. <https://docs.python.org/3/library/sys.html>
8. <https://docs.python.org/3/library/os.html>
9. <https://www.djangoproject.com/>
10. <https://realpython.com/python-sockets/>
11. https://www.w3schools.com/python/python_regex.asp
12. <https://realpython.com/python-web-scraping-practical-introduction/>
13. <https://jupyter.org/>
14. <https://realpython.com/pandas-dataframe>
15. <https://matplotlib.org>
16. www.numpy.org
17. <https://docs.scipy-lectures.org/intro/numpy/numpy.html>
18. <https://www.tutorialspoint.com/numpy/index.htm>