

**ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД  
«УЖГОРОДСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ»  
ФАКУЛЬТЕТ МАТЕМАТИКИ ТА ЦИФРОВИХ ТЕХНОЛОГІЙ  
Кафедра системного аналізу та теорії оптимізації**

**«ЗАТВЕРДЖУЮ»**  
Декан факультету математики  
та цифрових технологій  
Микола МАЛЯР  
« 30 Месяц » 2023 року



**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

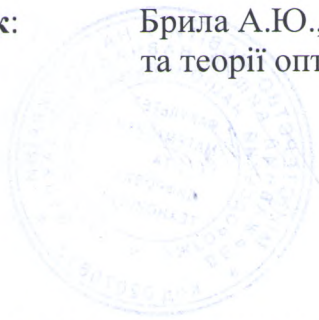
**ВСТУП ДО ПРОГРАМУВАННЯ. PYTHON**

Рівень вищої освіти	перший (бакалаврський)
Галузь знань	11 Математика та статистика
Спеціальність	113 Прикладна математика
Освітня програма	Системи штучного інтелекту
Статус дисципліни	обов'язкова
Мова навчання	українська

Ужгород 2023

Робоча програма навчальної дисципліни «Вступ до програмування. Python» для здобувачів вищої освіти галузі знань **11 Математика та статистика** спеціальності **113 Прикладна математика** освітньої програми **Системи штучного інтелекту**.

**Розробник:** Брила А.Ю., доцент, к. ф.-м. н., доцент кафедри системного аналізу та теорії оптимізації



Робочу програму розглянуто та затверджено на засіданні кафедри **системного аналізу та теорії оптимізації**

Протокол № 11 від « 16 » березня 2023 року

Завідувач кафедри [Signature] Мирослава ГЛЕБЕНА

Схвалено науково-методичною комісією **факультету математики та цифрових технологій**

протокол № 10 від « 20 » березня 2023 р.

Голова науково-методичної комісії [Signature] Наталія ЮРЧЕНКО

## 1. ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Найменування показників	Розподіл годин за навчальним планом
	Денна форма навчання
Кількість кредитів ЄКТС – <b>5</b>	Рік підготовки:
Загальна кількість годин – <b>150</b>	<b>1-й</b>
Кількість модулів – <b>2</b>	Семестр:
Тижневих годин для денної форми навчання: <b>1-й семестр</b> аудиторних – <b>4</b> самостійної роботи студента – <b>4</b>	<b>1-й</b>
	Лекції:
	<b>36</b>
	Практичні (семінарські):
	-
Вид підсумкового контролю: <b>1-й семестр</b> – екзамен.	Лабораторні:
	<b>38</b>
Форма підсумкового контролю: усна	Самостійна робота:
	<b>76</b>

## 2. МЕТА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

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 ЕОМ. Створення сучасних програмних продуктів вимагає досконалого знання мов програмування як високого, так і 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.

Відповідно до освітньої програми, вивчення дисципліни сприяє формуванню у здобувачів вищої освіти таких компетентностей:

- здатність учитися і оволодівати сучасними знаннями (ЗК01);
- здатність застосовувати знання у практичних ситуаціях (ЗК02);
- здатність генерувати нові ідеї (креативність) (ЗК03);
- здатність бути критичним і самокритичним (ЗК04);
- здатність проведення досліджень на відповідному рівні (ЗК05);
- здатність до абстрактного мислення, аналізу та синтезу (ЗК06);
- здатність до пошуку, оброблення та аналізу інформації з різних джерел (ЗК07);
- знання та розуміння предметної області та розуміння професійної діяльності (ЗК08);
- навички у використанні інформаційних і комунікаційних технологій (ЗК10);
- визначеність і наполегливість щодо поставлених завдань і взятих обов'язків (ЗК12);
- навички міжособистісної взаємодії (ЗК13);
- здатність до планування та розподілу часу (ЗК16);
- здатність приймати обґрунтовані рішення (ЗК17);
- здатність розробляти алгоритми та структури даних, програмні засоби та програмну документацію (ФК04);
- здатність розв'язувати професійні задачі за допомогою комп'ютерної техніки, комп'ютерних мереж та Інтернету, в середовищі сучасних операційних систем, з використанням стандартних офісних додатків (ФК06);
- здатність використовувати сучасні технології програмування та тестування програмного забезпечення (ФК08);
- здатність до проведення математичного і комп'ютерного моделювання, аналізу та обробки даних, обчислювального експерименту, розв'язання формалізованих задач за допомогою спеціалізованих програмних засобів (ФК09).

## 3. ПЕРЕДУМОВИ ДЛЯ ВИВЧЕННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Передумовами вивчення навчальної дисципліни «Вступ до програмування. Python» є базові знання з шкільного курсу інформатики.

## 4. ОЧІКУВАНІ РЕЗУЛЬТАТИ НАВЧАННЯ

Відповідно до освітньої програми «Системи штучного інтелекту», вивчення навчальної дисципліни повинно забезпечити досягнення здобувачами вищої освіти таких програмних результатів навчання (ПРН):

Програмні результати навчання	Шифр ПРН
Демонструвати знання й розуміння основних концепцій, принципів, теорій прикладної математики і використовувати їх на практиці.	ПРН 01

Виконувати математичний опис, аналіз та синтез дискретних об'єктів та систем, використовуючи поняття й методи дискретної математики та теорії алгоритмів.	ПРН 04
Будувати ефективні щодо точності обчислень, стійкості, швидкодії та витрат системних ресурсів алгоритми для чисельного дослідження математичних моделей та розв'язання практичних задач.	ПРН 09
Вміти застосовувати сучасні технології програмування та розроблення програмного забезпечення, програмної реалізації чисельних і символічних алгоритмів.	ПРН 11
Використовувати в практичній роботі спеціалізовані програмні продукти та програмні системи комп'ютерної математики.	ПРН 13
Виявляти здатність до самонавчання та продовження професійного розвитку.	ПРН 14
Уміти організувати власну діяльність та одержувати результат у рамках обмеженого часу.	ПРН 15
Демонструвати навички взаємодії з іншими людьми, уміння працювати в команді.	ПРН 16

Очікувані результати навчання, які повинні бути досягнуті після опанування навчальної дисципліни «Вступ до програмування. Python»:

<b>Очікувані результати навчання з дисципліни</b>	<b>Шифр ПРН</b>
Здатність класифікувати задачу і обирати відповідні методи щодо її розв'язання	ПРН 01, ПРН 04, ПРН 09
Демонструвати навички та здібності проектування та розробки прикладного програмного забезпечення	ПРН 01, ПРН 09, ПРН 11, ПРН 14,
Здатність до використання інформаційних і комунікаційних технологій	ПРН 11, ПРН 13,
Вміння працювати у команді при роботі з колективними проектами	ПРН 15, ПРН 16

## **5. ЗАСОБИ ДІАГНОСТИКИ ТА КРИТЕРІЇ ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ**

### **Методи навчання**

Метод проблемного викладення матеріалу, пояснювально-ілюстративний метод, метод проектів та кейсів, метод моделювання професійних ситуацій.

### **Засоби оцінювання та методи демонстрування результатів навчання**

Засобами оцінювання та методами демонстрування результатів навчання з навчальної дисципліни є: виконання індивідуальних завдань на комп'ютерах, виконання лабораторних робіт, презентація результатів виконаних завдань, модульні контрольні роботи, екзамен.

### **Форми (методи) контролю та критерії оцінювання результатів навчання**

Форми поточного контролю: усні відповіді на лабораторних заняттях, виконання практичних завдань, тестування, захист лабораторних робіт, виконання тестових завдань.

Форма модульного контролю: письмова контрольна робота.

Форми підсумкового семестрового контролю: екзамен.

## Розподіл балів, які отримують здобувачі вищої освіти (модуль 1)

Поточне оцінювання та самостійна робота																							Модульна контрольна робота	Сума
T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17	T18	T19	T20	T21	T22	T23	50	100
1	1	2	1	5	5	1	1	1	1	2	2	2	2	3	3	3	3	2	3	2	2	2		

## Розподіл балів, які отримують здобувачі вищої освіти (модуль 2)

Поточне оцінювання та самостійна робота													Модульна контрольна робота	Сума
T24	T25	T26	T27	T28	T29	T30	T31	T32	T33	T34	T35	50	100	
5	5	5	5	5	2	5	5	5	3	3	2			

## Оцінювання окремих видів навчальної роботи з дисципліни

Вид діяльності здобувача вищої освіти	Модуль 1		Модуль 2	
	Кількість	Максимальна кількість балів (сумарна)	Кількість	Максимальна кількість балів (сумарна)
Лабораторні заняття (допуск, виконання та захист)	11	50	8	50
Модульна контрольна робота	1	50	1	50
<b>Разом</b>		<b>100</b>		<b>100</b>

## Критерії оцінювання модульної контрольної роботи

**Методика оцінювання.** Матеріал кожного модуля, який здобувачі вищої освіти повинні засвоїти протягом семестру, виноситься на одну з двох модульних контрольних робіт, що проводяться в кожному семестрі.

Модульна контрольна робота (при застосуванні комп'ютерів) може виконуватись в два етапи: перший – теоретичний матеріал (письмово); другий – практичні завдання (із застосуванням комп'ютерів). Модульна контрольна робота складається із 3-ох завдань (одного теоретичного питання (10 балів) та 2-ох практичних завдань по 20 балів). Сумарна максимальна кількість балів, що виставляється здобувачу вищої освіти за виконання всіх завдань однієї контрольної роботи залежить від складності матеріалу, який виноситься на модульний контроль.

За виконання лабораторних робіт, що стосуються даних модулів здобувачу вищої освіти нараховується різна кількість балів, в залежності від складності матеріалу.

Лабораторна робота зараховується, якщо за кожне із завдань лабораторної роботи здобувач вищої освіти досягнув мінімального порогового рівня, визначеного за кожним запланованим результатом навчання для навчальної дисципліни. Критерієм успішного проходження здобувачем освіти поточного оцінювання (включно із захистом лабораторної роботи) є досягнення здобувачем освіти не менше 50% балів від загальної кількості запланованої за конкретною темою. Конкретна максимальна кількість балів подається у таблицях розподілу балів, які отримують здобувачі за модуль та за окремі види навчальної роботи.

Невиконані та незахищені лабораторні роботи, а також неявка на модульну контрольну роботу оцінюються в 0 балів незалежно від причини невиконання (неявки).

Сумарна оцінка (від 0 до 100 балів) виставляється у відомість модульного контролю. Модуль зараховується, якщо сумарний бал складає не менше 60 балів, і зараховано всі лабораторні роботи, які є складовими даного модуля.

Здобувач вищої освіти, який не з'явився на модульну контрольну роботу, або ж його модульна оцінка складає від 0 до 34 балів, зобов'язаний скласти (перескласти) модуль до початку підсумкового контролю у строки, визначені викладачем дисципліни та погоджені деканатом факультету.

### **Визнання результатів навчання, отриманих у неформальній освіті**

Визнання результатів навчання, отриманих у неформальній освіті здійснюється на добровільній основі та передбачає підтвердження того, що здобувач досяг необхідних результатів навчання наявним іменним сертифікатом успішного проходження безоплатного онлайн-курсу «Основи програмування» ([https://prometheus.org.ua/course/course-v1:KPI+Programming101+2015\\_T1](https://prometheus.org.ua/course/course-v1:KPI+Programming101+2015_T1)) української освітньої платформи Прометеус (<https://courses.prometheus.org.ua/>). Таким чином, отриманий сертифікат може бути перезарахований та оцінений, на основі рейтингів успішності за даним онлайн-курсом наданим здобувачем, але не більше ніж «60» балів.

### **Критерії оцінювання підсумкового контролю**

**Екзаменаційна методика оцінювання.** До складання екзамену допускаються здобувачі вищої освіти, які мають підсумковий до екзаменаційний рейтинговий бал не менше 35 і, яким зараховано всі лабораторні роботи.

Підсумкова рейтингова до екзаменаційна оцінка визначається як середньоарифметичне значення двох модулів.

Здобувач вищої освіти, до екзаменаційний рейтинговий бал якого складає від 0 до 34 балів, зобов'язаний покращити його до початку екзамену під час чергування викладачів на кафедрі у строки, визначені викладачем дисципліни та погоджені деканатом факультету. В протилежному випадку, здобувач не допускається до екзамену, і у нього виникає академічна заборгованість.

Екзамен з навчальної дисципліни здобувач вищої освіти може не складати, якщо він успішно пройшов усі модульні контролю та його влаштовує підсумкова до екзаменаційна рейтингова оцінка за навчальний рік. Здобувачі вищої освіти, рейтинговий бал яких становить від 35 до 59, екзамен складають обов'язково. Здобувач освіти може підвищити на екзамені рейтинговий бал, при цьому, за результатами складання екзамену оцінка не може бути менша за до екзаменаційний рейтинговий бал.

Екзамен проводиться в усній формі. Екзаменаційний білет складається з двох теоретичних питань та одного практичного завдання. Оцінювання результатів навчання на екзамені здійснюється за 100-бальною шкалою. Оцінка за екзамен вноситься у відомість обліку успішності.

**Таблиця відповідності оцінок за різними шкалами оцінювання**

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену	для заліку
90-100	A	відмінно	зараховано
82-89	B	добре	
74-81	C		
64-73	D	задовільно	
60-63	E		

35-59	FX	незадовільно з можливістю повторного складання	незараховано з можливістю повторного складання
0-34	F	незадовільно з обов'язковим повторним вивченням дисципліни	незараховано з обов'язковим повторним вивченням дисципліни

### Критерій оцінювання підсумкового контролю з дисципліни

- **«відмінно» (90-100 балів, А)** заслуговує студент, який виявив всебічне і глибоке знання програмового матеріалу, вміння вільно виконувати завдання, передбачені програмою, засвоїв основну і ознайомився з додатковою літературою, розуміє взаємозв'язок головних понять дисципліни та їх значення для майбутньої професії;
- **«добре» (82-89 балів, В)** заслуговує студент, який виявив повне знання програмного матеріалу, успішно виконує передбачені програмою завдання, засвоїв основну літературу рекомендовану програмою, виявив систематичний характер знань з дисциплін і здатний до самостійного доповнення, але під час відповіді допустив деякі неточності;
- **«добре» (74-81 бал, С)** заслуговує студент, що виявив не цілком повне знання програмного матеріалу, не завжди успішно виконує передбачені програмою завдання, частково засвоїв основну літературу, рекомендовану програмою, виявив не систематичний характер знань з дисциплін і не завжди здатний до їх самостійного доповнення і під час відповіді допускає деякі неточності;
- **«задовільно» (64-73 бали, D)** заслуговує студент, що виявив знання основного програмного матеріалу в обсязі, необхідному для подальшого навчання та майбутньої роботи за професією, вміє виконувати завдання, передбачені програмою, знайомий з основною рекомендованою літературою. Як правило, дана оцінка виставляється студентам, що допустили помилки у відповіді на заліку чи екзамені та при виконанні залікових або екзаменаційних завдань, але які володіють необхідними знаннями для їх усунення за допомогою викладача;
- **«задовільно» (60-63 балів, E)** заслуговує студент, що виявив часткове знання основного програмового матеріалу в обсязі, необхідному для подальшого навчання та майбутньої роботи за професією, не завжди вміє виконувати завдання, передбачені програмою, знайомий лише частково з основною рекомендованою літературою. Як правило, дана оцінка виставляється студентам, що допустили грубі помилки у відповіді на заліку чи екзамені та при виконанні залікових або екзаменаційних завдань, але які частково володіють необхідними знаннями для їх усунення за допомогою викладача.
- **«незадовільно» (35-59 балів, FX)** виставляється студенту, який виявив суттєві прогалини в знаннях основного програмового матеріалу, допустив принципові помилки у виконанні передбачених програмою завдань.
- **«незадовільно» (0-34 балів, F)** виставляється студенту коли протягом семестру він допустив грубі помилки у виконанні передбачених програмою завдань.

## 6. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

### 6.1. Зміст навчальної дисципліни

#### Модуль 1

#### Змістовний модуль 1. Основи інформатики та алгоритмізації

Тема 1. **Поняття інформації.** Неперервна та дискретна інформація. Властивості, характерні риси, вимірювання інформації.

- Тема 2. **Інформаційна та обчислювальна системи.** Поняття інформаційної та обчислювальної систем. Структура ПЕОМ. Технологія виконання команд процесором. Абстрактні обчислювальні машини.
- Тема 3. **Кодування інформації в пам'яті ПЕОМ.** Представлення даних в ОП. Системи числення. Кодування числової символної, графічної інформації. Кодування команд та їх формати. Скалярні та структурні типи даних.
- Тема 4. **Програмне забезпечення ПЕОМ.** Системне і прикладне програмне забезпечення. Операційна система. Сервісні та інструментальні системи. Системи програмування.
- Тема 5. **Вступ в алгоритмізацію.** Поняття алгоритму. Властивості алгоритмів. Способи запису алгоритмів. Типи алгоритмів. Виконавець алгоритму. Формальне виконання алгоритму. Аргументи, результати, проміжні величини.
- Тема 6. **Технологія програмування.** Алгоритмічні основи програмування. Етапи розв'язання задач на ПЕОМ. Структурне програмування. Структурний підхід до побудови алгоритмів за методикою “зверху-вниз”.

## **Змістовний модуль 2. Базові програмні структури мови Python**

- Тема 7. **Мова програмування.** Поняття мови програмування. Алфавіт, синтаксис, семантика. Огляд основних можливостей мови та встановлення Python.
- Тема 8. **Типи величин.** Поняття величини і типу даних. Типи даних в Python.
- Тема 9. **Змінні у Python.** Опис змінних. Оператор присвоювання. Перетворення типів.
- Тема 10. **Вирази.** Арифметичні та логічні вирази.
- Тема 11. **Реалізація лінійних алгоритмів.** Введення даних з клавіатури та виведення на екран. Реалізація лінійних алгоритмів.
- Тема 12. **Реалізація алгоритмів з розгалуженням.** Оператор умовного переходу.
- Тема 13. **Реалізація циклічних алгоритмів.** Оператор циклу while. Вкладені цикли. Оператори break, continue.
- Тема 14. **Рядки.** Рядки в Python. Операції з рядками. Використання циклу for. Регулярні вирази.
- Тема 15. **Списки.** Опис списків. Операції над списками. Використання циклу for.
- Тема 16. **Кортежі.** Опис кортежів. Операції над списками.
- Тема 17. **Множини.** Опис множин. Операції над множинами.
- Тема 18. **Словники.** Опис словників. Операції над словниками.
- Тема 19. **Алгоритми пошуку і сортування.**
- Тема 20. **Функції в Python.** Параметри і аргументи функцій. Локальні і глобальні змінні. Значення None. Лямбда-форми. Використання предикатів.

Тема 21. **Рекурсивні функції.** Опис рекурсивних функцій. Переваги та недоліки їх використання.

Тема 22. **Модулі в Python.** Опис та підключення модулів

Тема 23. **Пакети в Python.** Опис та підключення пакетів

## **Модуль 2**

### **Змістовний модуль 3. Об'єктно-орієнтоване програмування**

Тема 24. **Класи і об'єкти.** Опис класів. Поля, методи, властивості. Створення та використання об'єктів

Тема 25. **Успадкування в Python.** Опис класів нащадків. Перевантаження методів. Конструктори.

Тема 26. **Поліморфізм.** Принципи реалізації та приклади використання.

Тема 27. **Ітератори. Генератори.** Опис та приклади використання.

Тема 28. **Перевантаження операцій.** Опис та приклади використання.

Тема 29. **Обробка виключних ситуацій.** Загальні принципи обробки виключних ситуацій. Класи виключних ситуацій.

### **Змістовний модуль 4. Розширені можливості**

Тема 30. **Використання стандартних модулів.** Взаємодія з операційною системою, стиснення даних та ін.. Встановлення та підключення модулів з віддалених репозитаріїв.

Тема 31. **Час і дата в Python.** Виконання операцій з датами.

Тема 32. **Використання файлів.** Створення та маніпуляції з текстовими та бінарними файлами.

Тема 33. **Побудова графічного інтерфейсу користувача.** Основи побудови додатків з графічним інтерфейсом.

Тема 34. **Аналіз даних. Побудова графіків.** Графічне відображення даних з використанням Matplotlib.

Тема 35. **Розробка тестів.** Розробка тестів засобами unittest.

## 6.2. Структура навчальної дисципліни

Назви змістових модулів і тем	Кількість годин					
	Денна форма: денна					
	Усього	у тому числі				
		л	п	лаб	інд	ср
1	2	3	4	5	6	7
<b>1 семестр</b>						
<b>Модуль 1</b>						
<b>Змістовний модуль 1. Основи інформатики та алгоритмізації</b>						
Тема 1. <b>Поняття інформації.</b> Інформація. Властивості, характерні риси, вимірювання інформації.	1,5	0,5				1
Тема 2. <b>Інформаційна та обчислювальна системи.</b> Інформаційна та обчислювальна системи. Структура ПЕОМ. Технологія виконання команд процесором. Абстрактні обчислювальні машини.	1,5	0,5				1
Тема 3. <b>Кодування інформації в пам'яті ПЕОМ.</b> Представлення даних в ОП. Системи числення. Кодування числової символної, графічної інформації. Кодування команд та їх формати. Скалярні та структурні типи даних.	4	1		1		2
Тема 4. <b>Програмне забезпечення.</b> Системне і прикладне програмне забезпечення. Операційна система. Сервісні та інструментальні системи. Системи програмування.	2					2
Тема 5. <b>Вступ в алгоритмізацію.</b> Поняття алгоритму. Способи запису алгоритмів. Типи алгоритмів. Властивості алгоритмів. Виконавець алгоритму. Формальне виконання алгоритму. Аргументи, результати, проміжні величини.	7	1		2		4
Тема 6. <b>Технології програмування.</b> Алгоритмічні основи програмування. Етапи розв'язання задач на ПЕОМ. Структурне програмування. Структурний підхід до побудови алгоритмів за методикою “зверху-вниз”.	4	1		1		2
<b>Разом за змістовним модулем 1</b>	20	4		4		12
<b>Змістовний модуль 2. Базові програмні структури мови Python</b>						
Тема 7. <b>Мова програмування.</b> Поняття мови програмування. Алфавіт, синтаксис, семантика. Огляд	1,5	0,5				1

основних можливостей мови та встановлення Python.						
Тема 8. <b>Типи величин.</b> Поняття величини і типу даних. Типи даних в Python.	1,5	0,5				1
Тема 9. <b>Змінні у Python.</b> Опис змінних. Оператор присвоювання. Перетворення типів.	1,5	0,5				1
Тема 10. <b>Вирази.</b> Арифметичні та логічні вирази.	1,5	0,5				1
Тема 11. <b>Реалізація лінійних алгоритмів.</b> Введення даних з клавіатури та виведення на екран. Реалізація лінійних алгоритмів.	2,5	0,5		1		1
Тема 12. <b>Реалізація алгоритмів з розгалуженням.</b> Оператор умовного переходу.	2,5	0,5		1		1
Тема 13. <b>Реалізація циклічних алгоритмів.</b> Оператор циклу while. Вкладені цикли. Оператори break, continue.	5	1		2		2
Тема 14. <b>Рядки.</b> Рядки в Python. Операції з рядками. Використання циклу for. Регулярні вирази.	6	2		2		2
Тема 15. <b>Списки.</b> Опис списків. Операції над списками. Використання циклу for.	4	1		1		2
Тема 16. <b>Кортежі.</b> Опис кортежів. Операції з кортежами.	4	1		1		2
Тема 17. <b>Множини.</b> Опис множин. Операції над множинами.	5	1		2		2
Тема 18. <b>Словники.</b> Опис словників. Операції над словниками.	5	1		2		2
Тема 19. <b>Алгоритми пошуку і сортування.</b>	3	1				2
Тема 20. <b>Функції в Python.</b> Параметри і аргументи функцій. Локальні і глобальні змінні. Значення None. Лямбда-форми. Використання предикатів.	5	1		2		2
Тема 21. <b>Рекурсивні функції.</b> Опис рекурсивних функцій. Переваги та недоліки їх використання.	5	1		2		2
Тема 22. <b>Модулі в Python.</b> Опис та підключення модулів	3,5	0,5		1		2
Тема 23. <b>Пакети в Python.</b> Опис та підключення пакетів.	3,5	0,5		1		2

<b>Разом за змістовним модулем 2</b>	60	14		18		28
<b>Разом за модуль</b>	80	18		22		40
<b>Модуль 2</b>						
<b>Змістовний модуль 3. Об'єктно-орієнтоване програмування</b>						
Тема 24. <b>Класи і об'єкти.</b> Розробка тестів засобами unittest.	12	2		4		6
Тема 25. <b>Успадкування в Python.</b> Опис класів нащадків. Перевантаження методів. Конструктори.	5	2		1		2
Тема 26. <b>Поліморфізм.</b> Принципи реалізації та приклади використання.	4	1		1		2
Тема 27. <b>Ітератори. Генератори.</b> Опис та приклади використання.	4	1		1		2
Тема 28. <b>Перевантаження операцій.</b> Опис та приклади використання.	5	2		1		2
Тема 29. <b>Обробка виключних ситуацій.</b> Загальні принципи обробки виключних ситуацій. Класи виключних ситуацій.	6	1		1		4
<b>Разом за змістовним модулем 3</b>	36	8		9		18
<b>Змістовний модуль 4. Розширені можливості</b>						
Тема 30. <b>Використання стандартних модулів.</b> Взаємодія з операційною системою, стиснення даних та ін.. Встановлення та підключення модулів з віддалених репозитаріїв.	5	2		1		2
Тема 31. <b>Час і дата в Python.</b> Виконання операцій з датами.	4	1		1		2
Тема 32. <b>Використання файлів.</b> Створення та маніпуляції з текстовими та бінарними файлами.	6	2		2		2
Тема 33. <b>Побудова графічного інтерфейсу користувача.</b> Основи побудови додатків з графічним інтерфейсом.	8	1		1		6
Тема 34. <b>Аналіз даних. Побудова графіків.</b> Графічне відображення даних з використанням Matplotlib.	5	2		1		2
Тема 35. <b>Розробка тестів.</b> Розробка тестів засобами unittest.	6	1		1		4

<b>Разом за змістовним модулем 4</b>	34	8		7		18
<b>Разом за модуль</b>	70	20		16		36
<b>Разом за семестр</b>	150	36		38		76

### 6.3. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
1.	Системи числення. Перевід чисел. Виконання арифметичних операцій.	1
2.	Способи запису алгоритмів. Розробка лінійних алгоритмів. Розробка алгоритмів з розгалуженням	1
3.	Розробка циклічних алгоритмів.	1
4.	Технології програмування	1
5.	Введення даних з клавіатури та виведення на екран. Реалізація лінійних алгоритмів.	1
6.	Оператор умовного переходу.	1
7.	Оператор циклу while. Вкладені цикли. Оператори break, continue.	2
8.	Рядки в Python. Операції з рядками. Використання циклу for. Регулярні вирази.	2
9.	Опис списків. Операції над списками. Використання циклу for.	1
10.	Опис кортежів. Операції з кортежами.	1
11.	Опис множин. Операції над множинами.	2
12.	Опис словників. Операції над словниками.	2
13.	Параметри і аргументи функцій. Локальні і глобальні змінні. Значення None.	1
14.	Лямбда-форми. Використання предикатів.	1
15.	Рекурсивні функції.	2
16.	Модулі в Python.	1
17.	Пакети в Python.	1
18.	Класи і об'єкти.	4
19.	Успадкування в Python.	1
20.	Поліморфізм.	1
21.	Ітератори. Генератори.	1
22.	Перевантаження операцій.	1

23.	Обробка виключних ситуацій.	1
24.	Використання стандартних модулів.	1
25.	Час і дата в Python.	1
26.	Використання файлів.	2
27.	Побудова графічного інтерфейсу користувача.	1
28.	Аналіз даних. Побудова графіків.	1
29.	Розробка тестів.	1
<b>Разом</b>		<b>38</b>

#### 6.4. Самостійна робота

№ з/п	Назва теми	Кількість годин
1.	Поняття інформації. Інформація. Властивості, характерні риси, вимірювання інформації.	1
2.	Інформаційна та обчислювальна системи. Інформаційна та обчислювальна системи. Структура ПЕОМ. Технологія виконання команд процесором. Абстрактні обчислювальні машини.	1
3.	Кодування інформації в пам'яті ПЕОМ. Представлення даних в ОП. Системи числення. Кодування числової символної, графічної інформації. Кодування команд та їх формати. Скалярні та структурні типи даних.	2
4.	Програмне забезпечення. Системне і прикладне програмне забезпечення. Операційна система. Сервісні та інструментальні системи. Системи програмування.	2
5.	Вступ в алгоритмізацію. Поняття алгоритму. Способи запису алгоритмів. Типи алгоритмів. Властивості алгоритмів. Виконавець алгоритму. Формальне виконання алгоритму. Аргументи, результати, проміжні величини.	4
6.	Технології програмування. Алгоритмічні основи програмування. Етапи розв'язання задач на ПЕОМ. Структурне програмування. Структурний підхід до побудови алгоритмів за методикою “зверху-вниз”.	2
7.	Мова програмування. Поняття мови програмування. Алфавіт, синтаксис, семантика. Огляд основних можливостей мови та встановлення Python.	1

8.	Типи величин. Поняття величини і типу даних. Типи даних в Python.	1
9.	Змінні у Python. Опис змінних. Оператор присвоювання. Перетворення типів.	1
10.	Вирази. Арифметичні та логічні вирази.	1
11.	Реалізація лінійних алгоритмів. Введення даних з клавіатури та виведення на екран. Реалізація лінійних алгоритмів.	1
12.	Реалізація алгоритмів з розгалуженням. Оператор умовного переходу.	1
13.	Реалізація циклічних алгоритмів. Оператор циклу while. Вкладені цикли. Оператори break, continue.	2
14.	Рядки. Рядки в Python. Операції з рядками. Використання циклу for. Регулярні вирази.	2
15.	Списки. Опис списків. Операції над списками. Використання циклу for.	2
16.	Кортежі. Опис кортежів. Операції з кортежами.	2
17.	Множини. Опис множин. Операції над множинами.	2
18.	Словники. Опис словників. Операції над словниками.	2
19.	Алгоритми пошуку і сортування.	2
20.	Функції в Python. Параметри і аргументи функцій. Локальні і глобальні змінні. Значення None. Лямбда-форми. Використання предикатів.	2
21.	Рекурсивні функції.	2
22.	Модулі в Python.	2
23.	Пакети в Python.	2
24.	Класи і об'єкти.	6
25.	Успадкування в Python.	2
26.	Поліморфізм.	2
27.	Ітератори. Генератори.	2
28.	Перевантаження операцій.	2
29.	Обробка виключних ситуацій.	4
30.	Використання стандартних модулів.	2
31.	Час і дата в Python.	2
32.	Використання файлів.	2

33.	Побудова графічного інтерфейсу користувача.	6
34.	Аналіз даних. Побудова графіків.	2
35.	Розробка тестів.	4
<b>Разом</b>		<b>76</b>

## **7. ІНСТРУМЕНТИ, ОБЛАДНАННЯ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ВИКОРИСТАННЯ ЯКИХ ПЕРЕДБАЧАЄ НАВЧАЛЬНА ДИСЦИПЛІНА**

**Технічні засоби** – персональні комп’ютери, мультимедійний проектор.

**Програмне забезпечення** – операційна система, PyCharm, бібліотеки та середовище програмування Python, сервіс Google Meet, система електронного навчання Moodle.

## **8. РЕКОМЕНДОВАНІ ДЖЕРЕЛА ІНФОРМАЦІЇ**

### **Основна література**

1. Брила А.Ю., Ломага М.М., Вощепинець А.С. Арифметичні вирази у Python. Лінійні алгоритми. (Методичні вказівки до лабораторних робіт з дисципліни «Програмування»). Ужгород, 2023. 16 с.
2. Брила А.Ю., Ломага М.М., Вощепинець А.С. Логічні вирази у Python. Алгоритми з розгалуженням. (Методичні вказівки до лабораторних робіт з дисципліни «Програмування»). Ужгород, 2023. 16 с.
3. Брила А.Ю., Ломага М.М., Вощепинець А.С. Цикли у Python. (Методичні вказівки до лабораторних робіт з дисципліни «Програмування»). Ужгород, 2023. 16 с.
4. Брила А.Ю., Ломага М.М., Вощепинець А.С. Функції у Python. (Методичні вказівки до лабораторних робіт з дисципліни «Програмування»). Ужгород, 2023. 16 с.
5. Семйон І.В., Чупов С.В., Брила А.Ю., Апшай Н.І. Основи інформатики. Методичні матеріали з організації самостійної роботи для студентів математичного факультету з дисципліни “програмування”. Ужгород, 2010. 41 с.
6. Брила А.Ю., Ломага М.М., Змикало М.М. Методичні вказівки із завданнями до лабораторних робіт з курсу “Програмування” для студентів математичного факультету. Ч.І. Лінійні та розгалужені алгоритми. Ужгород, 2015. 41 с.
7. Семйон І.В., Чупов С.В. Брила А.Ю., Антосяк П.П., Дудла М.В. Основи алгоритмізації. Методичні вказівки до лабораторних робіт для студентів І-го курсу математичного факультету спеціальності “прикладна математика”. Ужгород, 2015. 63 с.
8. Брила А.Ю. Антосяк П.П. Сортування та пошук. Методичні вказівки до практичних занять з дисципліни “Алгоритми і структури даних” . Ужгород, 2015. 63 с
9. Основи програмування мовою Python: навчальний посібник. Ч.: ФОП Баликіна С.М., 2020. 180 с.
10. Юрченко І.В., Сікора В.С. Програмування мовою Python: навчальний посібник.– Чернівці: Чернівецький національний університет, 2022. 104 с
11. Семйон І.В., Чупов С.В., Брила А.Ю., Апшай Н.І. Основи інформатики. Методичні матеріали з організації самостійної роботи для студентів математичного факультету з дисципліни “програмування”. Ужгород, 2010. 41 с.
12. Програмування. Структурний підхід. Методичні вказівки до комп’ютерного практикуму. Для студентів І курсу Фізико-технічного інституту НТУУ “КП”. / Уклад.: Н.М. Куцуль, А.Ю. Шелестов, А.М. Лавренюк, С.В. Скакун, О.М. Куцуль, А.В. Колотій. К.: НТУУ “КП”, 2011. 120 с.
13. Luciano Ramalho: Fluent Python: Clear, Concise, and Effective Programming. O’Reilly Media, Inc., 2015.

14. Alex Martelli, Anna Ravenscroft, Steve Holden: Python in a Nutshell: The Definitive Reference, 3rd ed. O'Reilly Media, Inc., 2017.
15. Doug Hellmann: The Python 3 Standard Library by Example. Pearson Education, Inc., 2017.
16. Python : офіц. сайт. URL : <https://www.python.org/doc/>
17. SciPy: офіц. сайт. URL : <https://scipy.org/>

## **Допоміжна**

1. Mark Lutz: Learning Python: Powerful Object-Oriented Programming, 5th ed. O'Reilly Media, Inc., 2013.
2. David Beazley, Brian K. Jones: Python Cookbook: Recipes for Mastering Python 3, 3rd ed. O'Reilly Media, Inc., 2013.
3. Mark Lutz: Python Pocket Reference: Python in Your Pocket, 5th ed. O'Reilly Media, Inc., 2014.
4. Mark Summerfield: Python in Practice: Create Better Programs Using Concurrency, Libraries, and Patterns. Addison-Wesley, 2014.
5. Bill Lubanovic: Introducing Python: Modern Computing in Simple Packages. O'Reilly Media, Inc., 2015.