

Чернівецький національний університет імені Юрія Федьковича
(повне найменування закладу вищої освіти)

Факультет математики та інформатики
(назва факультету/навчально-наукового інституту)

Кафедра математичного моделювання
(назва кафедри, що забезпечує викладання)



“ЗАТВЕРДЖУЮ”
Декан факультету
математики та інформатики

Ольга МАРТИНЮК

_____ 2025 року

РОБОЧА ПРОГРАМА
навчальної дисципліни
Бібліотеки мови Python

(назва навчальної дисципліни)

вибіркова

(вказати: обов'язкова)

Освітньо-професійна програма «Системний аналіз»

(назва програми)

Спеціальність 124 Системний аналіз

(вказати: код, назва)

Галузь знань 12 Інформаційні технології

(вказати: шифр, назва)

Рівень вищої освіти перший (бакалаврський)

(вказати: перший (бакалаврський) / другий (магістерський) / третій (освітньо-науковий))

Факультет математики та інформатики

(назва факультету/ навчально-наукового інституту, на якому здійснюється підготовка фахівців за вказаною освітньою програмою)

Мова навчання українська

(вказати: на якій мові читається дисципліна)

Чернівці 2025 рік

Робоча програма навчальної дисципліни «Бібліотеки мови Python» складена відповідно до освітньо-професійної програми «Системний аналіз»

Розробник:

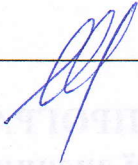
Юрченко І.В., доцент кафедри математичного моделювання, кандидат фіз.-мат. наук, доцент

Викладач, що забезпечує читання даної навчальної дисципліни:

Юрченко І.В., доцент кафедри математичного моделювання, кандидат фіз.-мат. наук, доцент

Затверджено на засіданні кафедри математичного моделювання
Протокол № 15 від «24» червня 2025 року

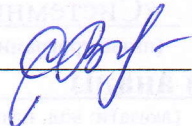
Завідувач кафедри _____



Ігор ЧЕРЕВКО

Схвалено методичною радою факультету математики та інформатики
Протокол № 12 від «25» червня 2025 року

Голова методичної ради _____



Віра СІКОРА

Мета навчальної дисципліни: ознайомити студентів з основними алгоритмами та структурами даних для розв'язання задач з використанням мови програмування Python, навчити студентів використовувати засоби об'єктно-зорієнтованого програмування та пакети (бібліотеки, модулі NumPy, Matplotlib, Tkinter, SciPy та ін.) мови Python для роботи з даними та побудови алгоритмів для розв'язування прикладних задач обробки даних.

Пререквізити. Навчальна дисципліна: “Програмування мовою Python”.

Результати навчання. У результаті вивчення навчальної дисципліни студент повинен

знати: основні концепції об'єктно-зорієнтованого програмування (ОЗП) в Python, призначення основних пакетів (NumPy, Matplotlib, Tkinter, SciPy та ін.) для роботи з даними в Python.

вміти: будувати алгоритми та структури даних з використанням основних концепцій ОЗП, застосовувати вміст пакетів (NumPy, Matplotlib, Tkinter, SciPy та ін.) для обробки даних та наукових обчислень з використанням Python для розв'язування прикладних задач.

Дисципліна формує такі **компетенції** у відповідності до стандарту вищої освіти спеціальності 124 – Системний аналіз та освітньої програми:

ЗК04. Знання та розуміння предметної області та розуміння професійної діяльності.

ЗК11. Здатність генерувати нові ідеї (креативність).

ЗК12. Здатність працювати в команді.

ФК6. Здатність до комп'ютерної реалізації математичних моделей реальних систем і процесів; проектувати, застосовувати і супроводжувати програмні засоби моделювання, прийняття рішень, оптимізації, обробки інформації, інтелектуального аналізу даних.

ФК7. Здатність використовувати сучасні інформаційні технології для комп'ютерної реалізації математичних моделей та прогнозування поведінки конкретних систем, а саме: об'єктно-орієнтований підхід при проектуванні складних систем різної природи, прикладні математичні пакети, застосування баз даних і знань.

ФК9. Здатність представляти математичні аргументи і висновки з них з ясністю і точністю і в таких формах, які підходять для аудиторії як усно так і в письмовій формі.

Наведені результати навчання за відповідною дисципліною співвідносяться із такими **програмними результатами навчання:**

ПР9. Вміти створювати ефективні алгоритми для обчислювальних задач системного аналізу та систем підтримки прийняття рішень.

ПР11. Знати і вміти застосовувати на практиці системи управління базами даних і знань та інформаційні системи.

ПР13. Проектувати, реалізовувати, тестувати, впроваджувати, супроводжувати, експлуатувати програмні засоби роботи з даними і знаннями в комп'ютерних системах і мережах.

Опис навчальної дисципліни

Загальна інформація

Форма навчання	Рік підготовки	Семестр	Кількість		Кількість годин						Вид підсумкового контролю
			кредитів	годин	лекції	практичні	семінарські	лабораторні	самостійна робота	індивідуальні завдання	
Денна	2	4	3	90	30	–	–	30	30	–	залік

Структура змісту навчальної дисципліни

Назви змістових модулів і тем	Кількість годин													
	денна форма							Заочна форма						
	усього	у тому числі					усього	у тому числі						
		л	пр	лаб	інд	срс		л	пр	лаб	інд	срс		
1	2	3	4	5	6	7	8	9	10	11	12	13		
Теми лекційних занять	Змістовий модуль 1. Об'єктно-зорієнтоване програмування													
Тема 1. Об'єктно-зорієнтоване програмування в Python.	13	4	–	4	–	5								
Тема 2. Рекурсивні структури даних у Python. Стеки, черги, деки. Списки (кільцеві). Дерев та графи. Перевизначення операцій.	13	4	–	4	–	5								
Тема 3. Множинне наслідування в Python.	13	4	–	4	–	5								

Тема 4. Метакласи та метапрограмування.	13	4	–	4	–	5							
Разом за ЗМ1	52	16	–	16	–	20							
Теми лекційних занять	Змістовий модуль 2. Модулі та пакети Python для роботи з даними												
Тема 5. Наукові обчислення. Пакети Numpy та Matplotlib.	11	4	–	4	–	3							
Тема 6. Регулярні вирази в Python. Модулі та пакети для використання операційної системи.	11	4	–	4	–	3							
Тема 7. Робота з даними в офісних документах. Пакет python-docx.	5	2	–	2	–	1							
Тема 8. Основи програмування в мережі у Python.	5	2	–	2	–	1							
Тема 9. Знайомство з бібліотекою scikit-learn (елементи штучного інтелекту)	6	2	–	2	–	2							
Разом за ЗМ 2	38	14	–	14	–	10							
Усього годин	90	30	–	30	–	30							

Тематика лекційних занять з переліком питань

№	Назва теми з основними питаннями
1	Об'єктно-зорієнтоване програмування в Python (класи, екземпляри класів, методи). Парадигма ОЗП (інкапсуляція, поліморфізм, успадкування). Приклади.
2	Рекурсивні структури даних у Python. Стеки, черги, деки. списки (кільцеві). Дерева та графи. Перевизначення операцій.
3	Множинне наслідування в Python. Приклади.
4	Метакласи та метапрограмування. Приклади.
5	Наукові обчислення. Пакети Numpy та Matplotlib. Приклади використання.
6	Регулярні вирази в Python. Модулі та пакети для роботи з операційною

	системою.
7	Робота з даними в офісних документах. Пакет python-docx.
8	Основи програмування в мережі у Python (побудова застосунків моделі "клієнт-сервер"). Приклади використання.
9	Знайомство з бібліотекою scikit-learn (огляд елементів застосування штучного інтелекту).

Тематика лабораторних занять з переліком питань

№	Назва теми (завдання)
1	Об'єктно-зорієнтоване програмування: створення класу та об'єктів-екземплярів класу.
2	Об'єктно-зорієнтоване програмування: наслідування атрибутів класу.
3	Об'єктно-зорієнтоване програмування: розробка сховища даних.
4	Розробка додатків з графічним інтерфейсом на основі бібліотеки tkinter.
5	Наукові обчислення. Пакети Numpy та Matplotlib.
6	Робота з даними в офісних документах. Пакет python-docx.
7	Вивчення бібліотеки scikit-learn мови Python (елементи штучного інтелекту)

Індивідуальні науково-дослідні завдання (ІНДЗ)

№	Завдання до тем
1	Студент може індивідуально виконувати додаткові завдання навчально-дослідницької спрямованості за завданнями, наданими викладачем.
2	Студенти можуть отримати до 10 балів в рахунок ІНДЗ, якщо самостійно зареєструються на безкоштовних курсах платформи Prometheus з програмування мовою Python або на курсах з Python платформи Coursera, пройдуть навчання, отримають відповідний сертифікат і надішлють його на сайт дистанційного навчання викладачу разом з детальним звітом з практичних завдань пройденого курсу (постановки задач, коди виконаних програм, пояснення коду) та скріншотом успішності на курсі. Кількість балів буде виставлена пропорційно до навчальних результатів студента (згідно зі статистикою сайту Prometheus або Coursera).

* ІНДЗ – для змістового модуля, або в цілому для навчальної дисципліни визначається викладачем, з урахуванням специфіки дисципліни.

Завдання для самостійної роботи студентів

Самостійна робота складається з повторення матеріалу, засвоєного на лекціях, самостійного опанування частини теоретичного матеріалу, роботи з контрольними запитаннями та завданнями.

Студенти можуть отримувати до 1 бала в рахунок виконання завдань СРС під час кожного лекційного заняття за правильні відповіді на запитання лектора, активне обговорення багатоваріантних підходів до рішення представленої лектором проблеми (для активізації пошукової та дослідної діяльності студентів).

№	Назва теми	Завдання для самостійної роботи	К-сть год.
1	Теми 1-9	підготовка до лекційних занять	10
2	Теми 1-9	підготовка до лабораторних занять	15
3	Теми 1-9	підготовка до підсумкового модуль-контролю	5

Методи навчання

Методи навчання та викладання: лекції, лабораторні заняття, електронне навчання з використанням системи Moodle, тестування, виконання завдань ІНДЗ.

Система контролю та оцінювання

Критерієм підсумкового оцінювання є досягнення студентом мінімальних порогових рівнів оцінок (балів) за кожним передбаченим результатом навчання.

Мінімальний пороговий рівень оцінки варто визначати за допомогою якісних критеріїв і трансформувати його в мінімальну позитивну оцінку використовуваної числової (рейтингової) шкали.

Система оцінювання рівня навчальних досягнень ґрунтується на принципах ECTS та є накопичувальною. Протягом семестру студенти виконують 7 лабораторних робіт. Кожна лабораторна робота оцінюється в 10 балів.

Виконуючи завдання лабораторної роботи, студент повинен оформити і завантажити для подальшої перевірки на сайт електронного навчання звіт разом із працездатними файлами програмної реалізації завдань ЛР (правила оформлення наведені на сторінці навчальної дисципліни на сайті).

50% балів, відведених на оцінювання ЛР, студент отримує за працюючий програмний продукт, в якому реалізовано всі завдання ЛР та оформлений звіт. Решта 50% балів виставляється після захисту студентом виконаного звіту. На захисті звіту з ЛР студент має відповісти на питання щодо постановки задачі та розробленого ним алгоритму реалізації кожного із завдань ЛР. При відповіді на теоретичні питання та питання щодо програмної реалізації алгоритму у випадку неістотної помилки знімається 10-20% балів, а у випадку істотної 20-40% балів, якщо ж студент не опанував теоретичний матеріал, плутається в означеннях,

наводить логічно невірні твердження, то знімається до 50% балів від усієї суми балів за ЛР.

Максимальна кількість, яку можна набрати на підсумковому модулі (тестування) – 30 балів.

Підсумкова оцінка виставляється за результатами суми балів, набраних на змістових модулях під час семестру та підсумковому модулі згідно з нижченаведеною таблицею.

Методи формування професійної компетентності: розповідь, пояснення, бесіда, демонстрація, візуалізація, дискусія тощо. Методи формування практичних умінь та навичок: розв'язування задач лабораторних робіт, виконання завдань, розробка та аналіз алгоритмів і програмного коду, захист звітів з лабораторних робіт.

Засобами оцінювання та демонстрування результатів навчання є: стандартизовані тести; аналітичні звіти з лабораторних робіт; презентації результатів виконаних завдань та досліджень ІНДЗ, усний контроль у вигляді індивідуального та фронтального опитування на лекціях та лабораторних заняттях.

Формами поточного контролю є усна чи письмова (тестування, лабораторна робота, ІНДЗ) відповідь студента.

Формою підсумкового контролю є залік.

Критерії оцінювання поточного та підсумкового контролю

Розподіл балів, які отримують студенти

Поточне оцінювання (аудиторна та самостійна робота)							Кількість балів (залікова робота)	Сумарна к-ть балів
Змістовий модуль №1			Змістовий модуль № 2					
T1	T2	T3	T6	T7	T8	T9	30	100
10	10	10	10	10	10	10		

Шкала оцінювання: національна та ЄКТС

100-бальна шкала	Оцінка за національною шкалою		Оцінка за шкалою ЄКТС	
			Оцінка	Пояснення за розширеною шкалою
90-100	Зараховано	Відмінно	A	відмінно
80-89			Добре	B
70-79		C		добре
60-69		Задовільно		D
50-59			E	достатньо

35-49			FX	(незадовільно) з можливістю повторного складання
1-34	Незараховано	Незадовільно	F	(незадовільно) з обов'язковим самостійним повторним опрацюванням освітнього компонента до перескладання

На оцінку "відмінно" заслуговує студент, який виявив всебічні, систематичні та глибокі знання, здатність самостійно виконувати завдання, передбачені програмою, ознайомлений з основною і додатковою літературою, рекомендованою програмою. Така оцінка передбачає також засвоєння студентом взаємозв'язку основних понять дисципліни та їх значення для набутої професії.

Оцінку "добре" ставлять студентів, який засвоїв навчально-програмовий матеріал у повному обсязі, успішно виконує передбачені програмою завдання, опрацював основну літературу, рекомендовану програмою, тобто студентів, який засвідчив систематичний характер знань із дисципліни і здатний до їх самостійного поповнення й оновлення у процесі подальшої навчальної роботи і професійної діяльності.

На оцінку "задовільно" заслуговує студент, який виявив знання основного навчального матеріалу в обсязі, необхідному для подальшого навчання і майбутньої роботи за професією, здатний виконувати завдання, передбачені програмою, ознайомлений з основною літературою, рекомендованою програмою. Як правило, цю оцінку виставляють студентам, які припустилися огріхів у відповіді на іспиті та при виконанні екзаменаційних завдань, але продемонстрували спроможність усунути їх.

Оцінку "незадовільно" ставлять студентів, у знаннях якого є прогалини, який припустився принципових помилок у виконанні передбачених програмою завдань, тобто студентів, який неспроможний продовжити навчання чи приступити до професійної діяльності після закінчення вищого навчального закладу без додаткових занять з відповідної дисципліни.

Перелік питань для самоконтролю та підсумкового контролю навчальних досягнень студентів

1. Об'єктно-зорієнтоване програмування в Python (класи, екземпляри класів, методи). Приклади побудови класу.
2. Парадигма ОЗП (інкапсуляція, поліморфізм, успадкування (наслідування)). Приклади реалізації.
3. Рекурсивні структури даних у Python. Стеки, черги, деки. Приклади реалізації.
4. Рекурсивні структури даних у Python. Списки (кільцеві). Приклади реалізації.
5. Рекурсивні структури даних у Python. Дерева та графи. Приклади реалізації.

6. Перевизначення операцій. Приклади.
7. Множинне наслідування в Python. Приклади реалізації.
8. Метакласи та метапрограмування. Приклади реалізації.
9. Наукові обчислення. Пакети Numpy та Matplotlib. Приклади використання.
10. Регулярні вирази в Python. Приклади використання.
11. Модулі та пакети для роботи з операційною системою.
12. Робота з даними в офісних документах. Пакет python-docx.
13. Основи програмування в мережі у Python (побудова застосунків моделі "клієнт-сервер").

Зарахування результатів неформальної освіти

Здобувачі вищої освіти має право на участь у неформальній/інформальній освіті.

У межах поточного контролю можуть визнаватися результати неформальної/інформальної освіти за умови наявності сертифікату або освітньої декларації про результати неформальної /інформальної освіти з питань, що відповідає тематиці курсу («Порядок визнання у Чернівецькому національному університеті імені Юрія Федьковича результатів навчання, здобутих шляхом неформальної та/або інформальної освіти»), <https://www.chnu.edu.ua/media/4g5fzssb/poriadok-vyznannia-rezultativ-navchannia-zdobutykh-shliakhom-neformalnoi-ta-abo-informalnoi-osvity.pdf>).

Студентам можуть бути зараховані додаткові бали, отримані через неформальну освіту, до загальної суми балів, набраної з освітньої компоненти, за умови, що результати з проблеми, за якою відбувалося навчання, відповідають тематиці курсу та отримані на ліцензійних платформах.

Рекомендована література

Основна

1. Крєневич А.П. Алгоритми і структури даних. Підручник.– К.: ВПЦ "Київський Університет", 2021.– 200 с.
<https://mechmat.knu.ua/wp-content/uploads/2021/09/pidruchnyk-alhorytmy-i-struktury-danykh.pdf>
2. Юрченко І.В., Сікора В.С. Програмування мовою Python: Навч. посібник.– Чернівці: Чернівецький національний університет, 2022.– 104 с. <https://archer.chnu.edu.ua/xmlui/handle/123456789/3666>
3. The Python Tutorial [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.python.org/3/tutorial/index.html>.
4. Навчальні матеріали: Python [Електронний ресурс] – Режим доступу до ресурсу: <https://mp.mechmat.knu.ua/library>.

5. ОСНОВИ ПРОГРАМУВАННЯ: методичні вказівки до виконання комп'ютерних практикумів на PYTHON з навчальної дисципліни «Основи програмування» для студентів спеціальності 122 «Комп'ютерні науки» / Уклад. Л.М. Добровська. – К.: НТУУ «КПІ ім. Ігоря Сікорського», 2017. – 254 с.
6. Обвінцев О.В. Об'єктно-орієнтоване програмування. Курс на основі Python. Матеріали лекцій.– Київ: Основа, 2017.
7. Доля П.Г. Вступ до наукового Python'a.– Харків: Харківський національний університет ім. В.Н. Каразіна, 2016.– 265 с.

Допоміжна

8. Джейсон Р. Бріггс. Python для дітей. Веселий вступ до програмування.– Львів: Видавництво Старого Лева, 2017.– 400 с.
9. Ерік Маттес. Пришвидшений курс Python.– Львів: Видавництво Старого Лева, 2021.– 600 с.
10. Bradley N. Miller and David L. Ranum. Problem solving with algorithms and data structures using Python.– Luther College, 2014.
<http://interactivepython.org/runestone/static/pythonds/index.html>
<https://aliev.me/runestone/> (переклад).
11. Mark Summerfield. Programming in Python 3. A Complete Introduction to the Python Language.- Pearson Education, Inc., 2010.– 636 p.
<https://www.amazon.com/Programming-Python-Complete-Introduction-Language/dp/0321680561>

Інформаційні ресурси

<https://moodle.chnu.edu.ua/course/view.php?id=232>
<https://github.com/krenevych/informatics/>
<http://www.python.org>
<https://mp.mechmat.knu.ua/library>
<https://www.w3schools.com/python/default.asp>
<https://matplotlib.org/>
<https://docs.python.org/uk/3/library/tkinter.html>
<https://numpy.org/>
<https://python-docx.readthedocs.io/en/latest/>
<https://scipy-lectures.org/packages/scikit-learn/index.html>
<https://machinelearningmastery.com/how-to-develop-a-convolutional-neural-network-to-classify-photos-of-dogs-and-cats/>

Політика академічної доброчесності

Дотримання політики щодо академічної доброчесності учасниками освітнього процесу при вивченні навчальної дисципліни регламентовано такими документами:

✓ «Етичний кодекс Чернівецького національного університету імені Юрія Федьковича» https://www.chnu.edu.ua/media/jxdfs0zb/etychnyi-kodeks-chemivets_koho-natsionalnoho-universytetu.pdf

✓ «Положенням про виявлення та запобігання академічного плагіату у Чернівецькому національному університету імені Юрія Федьковича» <https://www.chnu.edu.ua/media/n5nbzwgb/polozhennia-chnu-pro-plahi-at-2023plusdodatky-31102023.pdf>