

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ**  
**ІМЕНІ СЕМЕНА КУЗНЕЦЯ**



**"ЗАТВЕРДЖУЮ"**

Проректор з навчально-методичної роботи

Каріна НЕМАШКАЛО

**ПРОГРАМУВАННЯ**

**робоча програма навчальної дисципліни**

Галузь знань **12 «Інформаційні технології»**  
Спеціальність **124 «Системний аналіз»**  
Освітній рівень **перший (бакалаврський)**  
Освітня програма **«Управління складними системами»**

Статус дисципліни **обов'язкова**  
Мова викладання, навчання та оцінювання **українська**

Завідувач кафедри  
економічної кібернетики і системного аналізу

Лідія ГУР'ЯНОВА

Харків  
2022

**ЗАТВЕРДЖЕНО**

на засіданні кафедри економічної кібернетики і системного аналізу  
Протокол № 1 від 25.08.2022 р.

Розробник(и):

Яценко Роман Миколайович, к.е.н., доц. кафедри економічної кібернетики і системного аналізу.

**Лист оновлення та перезатвердження  
робочої програми навчальної дисципліни**

Навчальний рік	Дата засідання кафедри – розробника РПНД	Номер протоколу	Підпис завідувача кафедри

### Анотація навчальної дисципліни

Програма вивчення обов'язкової навчальної дисципліни професійного циклу «Програмування» складена відповідно до освітньо-професійної програми підготовки бакалавра зі спеціальності 124 «Системний аналіз», освітньої програми «Управління складними системами».

Предметом вивчення навчальної дисципліни є концепції та інструменти побудови програмних засобів для підвищення ефективності функціонування економічних систем в сучасних умовах розвитку ринкового середовища.

Програма навчальної дисципліни складається з таких змістових модулів:

1. Основи програмування на Python.
2. Структури даних та об'єктно-орієнтоване програмування.

Мета навчальної дисципліни є надання здобувачам вищої освіти системи теоретичних і практичних знань з програмування з метою побудови ефективних алгоритмів обробки даних та їх програмної реалізації. У якості головного інструменту обрано сучасну мову програмування Python.

Основним завданням вивчення дисципліни «Програмування» є вивчення теоретичних і практичних засад програмування, основ алгоритмізації та структур даних, формування у студентів навичок використання отриманих знань на практиці, створення власного програмних засобів.

### Характеристика навчальної дисципліни

Курс	1
Семестр	1,2
Кількість кредитів ECTS	10
Форма підсумкового контролю	екзамен

### Структурно-логічна схема вивчення навчальної дисципліни

Пререквізити	Постреквізити
Основи алгоритмізації	Системне програмування та операційні системи
	Web-технології
	Електронна комерція
	Бази даних
	Випадкові процеси

### Компетентності та результати навчання за дисципліною

Інтегральна компетентність: Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми системного аналізу у професійній діяльності або в процесі навчання, що передбачають застосування теоретичних положень та методів системного аналізу та інформаційних технологій і характеризуються комплексністю та невизначеністю умов

### Компетентності

Компетентності	Результати навчання
КЗ 2. Здатність застосовувати знання у практичних ситуаціях	РН8. Володіти сучасними методами розробки програм і програмних комплексів та прийняття оптимальних рішень щодо складу програмного забезпечення,
КЗ 13. Здатність працювати в міжнародному контексті	
КЗ 14. Здатність оцінювати та забезпечувати якість виконуваних робіт	
КЗ 16. Здатність зберігати та примножувати моральні,	

<p>культурні, наукові цінності і досягнення суспільства на основі розуміння історії та закономірностей розвитку предметної області, її місця у загальній системі знань про природу і суспільство та у розвитку суспільства, техніки і технологій, використовувати різні види та форми рухової активності для активного відпочинку та ведення здорового способу життя</p>	<p>алгоритмів процедур і операцій. РН9. Вміти створювати ефективні алгоритми для обчислювальних задач системного аналізу та систем підтримки прийняття рішень.</p>
<p>КФ 7. 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</p>	

## Програма навчальної дисципліни

### Змістовий модуль 1.

#### Основи програмування на Python

##### Тема 1. Основи мови програмування Python

- 1.1. Історія мови програмування Python.
- 1.2. Переваги Python.
- 1.3. Основні елементи мови.
- 1.4. Структурні елементи програми.
- 1.5. The Zen of Python.

##### Тема 2. Типи даних та змінні. Введення-виведення даних.

- 2.1. Оператори введення та виведення даних.
- 2.2. Типи даних.
- 2.3. Перетворення типів даних.
- 2.4. Складені оператори присвоювання.

##### Тема 3. Умовний оператор та розгалужений обчислювальний процес.

- 3.1. Синтаксис умовного оператора.
- 3.2. Вкладені умовні інструкції.
- 3.3. Оператори порівняння.
- 3.4. Логічні оператори.
- 3.5. Каскадні умовні конструкції.
- 3.6. Тернарний умовний оператор.

##### Тема 4. Цикл з визначеною кількістю ітерацій.

- 4.1. Цикл for.
- 4.2. Функція range.
- 4.3. Генерація псевдовипадкових чисел.

##### Тема 5. Обробка символьних даних.

- 5.1. Загальні підходи до обробки символьних даних.
- 5.2. Створення зрізів.
- 5.3. Методи роботи з рядками.

##### Тема 6. Цикли з перевіркою умов.

- 6.1. Цикл while.
- 6.2. Оператори управління циклом.

6.3. Множинне присвоювання.

### **Тема 7. Структурне програмування за допомогою функцій.**

7.1. Принципи структурного програмування.

7.2. Оголошення функцій.

7.3. Локальні та глобальні змінні.

## **Змістовий модуль 2. Структури даних Python**

### **Тема 8. Обробка даних у послідовностях та списках.**

8.1. Поняття списку.

8.2. Методи split та join.

8.3. Операції зі списками.

8.4. Генератори списків.

### **Тема 9. Двовимірні структури даних.**

9.1. Вкладені списки.

9.2. Створення та введення вкладених списків.

9.3. Приклади програм з двовимірними структурами даних.

### **Тема 10. Множинні та асоціативні структури даних.**

10.1. Призначення та створення словників.

10.2. Робота з елементами словника.

10.3. Призначення множини як структури даних.

10.4. Операції над множинами.

### **Тема 11. Робота з файлами та файловою системою.**

11.1. Загальні підходи до роботи з текстовими файлами.

11.2. Операції відкриття та закриття файлів.

11.3. Запис даних до файлу.

11.4. Зчитування даних з файлу.

11.5. Методи роботи зі складеними даними.

### **Тема 12. Стандартна бібліотека Python**

Перелік лабораторних занять, а також питань та завдань до самостійної роботи наведено у таблиці "Рейтинг-план навчальної дисципліни".

### **Методи навчання та викладання**

При викладанні навчальної дисципліни "Програмування" для активізації навчального процесу передбачено застосування сучасних навчальних технологій, таких, як: проблемні лекції; міні-лекції; презентації, виконання індивідуальних творчих завдань.

Проблемна лекція «Огляд популярних Python-фреймворків» в темі 1 спрямована на розвиток логічного мислення студентів. Коло питань теми обмежується двома-трьома ключовими моментами, увага студентів концентрується на матеріалі, що не знайшов відображення в підручниках, використовується досвід закордонних навчальних закладів з роздачею студентам під час лекцій друкованого матеріалу та виділенням головних висновків з питань, що розглядаються. При викладанні лекційного матеріалу студентам пропонуються питання для самостійного розмірковування. При цьому лектор задає запитання які спонукають студента шукати розв'язання проблемної ситуації. Така система примушує студентів сконцентруватися і почати активно мислити в пошуках правильної відповіді.

На початку проведення проблемної лекції необхідно чітко сформулювати проблему, яку необхідно вирішити студентам. При викладанні лекційного матеріалу слід уникати прямої відповіді на поставлені запитання, а висвітлювати матеріал таким чином, щоб отриману інформацію студент міг використовувати при розв'язанні проблеми.

Міні-лекція «Сховище репозиторіїв GitHub» в темі 7 передбачає викладення навчального матеріалу за короткий проміжок часу й характеризуються значною ємністю, складністю логічних побудов, доказів та узагальнень. Міні-лекції проводяться, як правило, як частина заняття-дослідження. На початку проведення міні-лекції за вказаною вище темою лектор акцентує увагу студентів на необхідності представити викладений лекційний матеріал у так званому структурно-логічному вигляді. На розгляд виносяться питання, які зафіксовані у плані лекцій, але викладаються вони стисло. Лекційне заняття, проведене у такий спосіб, побуджує у студента активність та увагу при сприйнятті матеріалу, а також спрямовує його на використання системного підходу при відтворенні інформації, яку він одержав від викладача.

Презентації результатів виконання самостійного завдання щодо стандартної бібліотеки Python за темою 12 – це виступи перед аудиторією, що використовуються для представлення звіту про виконання індивідуальних завдань. Однією з позитивних рис презентації та її переваг при використанні в навчальному процесі є обмін досвідом, який здобули студенти при роботі над індивідуальним завданням або у певній малій групі.

### Порядок оцінювання результатів навчання

ХНЕУ ім. С. Кузнеця використовує накопичувальну (100-бальну) систему оцінювання.

Система оцінювання сформованих компетентностей у студентів враховує види занять, які згідно з програмою навчальної дисципліни передбачають лекційні, лабораторні заняття, а також виконання самостійної роботи. Контрольні заходи включають:

поточний контроль, що здійснюється протягом семестру під час проведення лекційних, лабораторних занять і оцінюється сумою набраних балів (максимальна сума – 60 балів; мінімальна сума, що дозволяє студенту скласти іспит, – 35 балів);

модульний контроль, що проводиться у формі поточних контрольних робіт за змістові модулі та має на меті *інтегровану* оцінку результатів навчання студента після вивчення матеріалу з логічно завершеної частини дисципліни – змістового модуля 1 та 2;

підсумковий контроль, що проводиться у формі екзамену під час сесії другого семестру, відповідно до графіку навчального процесу.

**Поточний та модульний контроль** оцінювання знань протягом змістових модулів включає:

виконання самостійних індивідуальних завдань. Загальна кількість балів – 40;

презентація результатів виконання самостійного завдання щодо бібліотеки Python – 5 балів;

поточні контрольні роботи – 15 балів.

**Підсумковий контроль** знань та компетентностей студентів з навчальної дисципліни здійснюється на підставі проведення семестрового екзамену, завданням якого є перевірка розуміння студентом програмного матеріалу в цілому, логіки та взаємозв'язків між окремими розділами, здатності творчого використання накопичених знань, вміння формулювати своє ставлення до певної проблеми навчальної дисципліни тощо.

Екзаменаційний білет охоплює програму дисципліни і передбачає визначення рівня знань та ступеня опанування студентами компетентностей. Кожен екзаменаційний білет складається із 20 стереотипних тестових завдань та 2 практичних завдань (діагностичне та евристичне завдання), які передбачають вирішення типових професійних завдань фахівця на робочому місці та дозволяють діагностувати рівень теоретичної підготовки студента і рівень його компетентності з навчальної дисципліни.

Екзаменаційний білет включає:

Стереотипні тестові завдання: максимальна кількість балів – 20.

Діагностичне завдання: максимальна кількість балів – 10.

Евристичне завдання: максимальна кількість балів – 10.

Результат семестрового екзамену оцінюється в балах (максимальна кількість – 40 балів, мінімальна кількість, що зараховується, – 25 балів) і проставляється у відповідній графі екзаменаційної "Відомості обліку успішності".

Студента слід **вважати атестованим**, якщо сума балів, одержаних за результатами підсумкової/семестрової перевірки успішності, дорівнює або перевищує 60. Мінімумально можлива кількість балів за поточний і модульний контроль упродовж семестру – 35 та мінімумально можлива кількість балів, набраних на екзамені, – 25.

Підсумкова оцінка з навчальної дисципліни розраховується з урахуванням балів, отриманих під час екзамену, та балів, отриманих під час поточного контролю за накопичувальною системою. Сумарний результат у балах за семестр складає: "60 і більше балів – зараховано", "59 і менше балів – не зараховано" та заноситься у залікову "Відомість обліку успішності" навчальної дисципліни.

Форми оцінювання та розподіл балів наведено у таблиці "Рейтинг-план навчальної дисципліни".

### Рейтинг-план навчальної дисципліни

Тема	Форми та види навчання	Форми оцінювання	Мах бал
1	2	3	4
<b>Змістовий модуль 1. Основи програмування на Python</b>			
ТЕМА 1	<i>Аудиторна робота</i>		
	Лекція 1 за питаннями Теми 1		
	<i>Самостійна робота</i>		
	Вивчення лекційного матеріалу		
ТЕМА 2	<i>Аудиторна робота</i>		
	Лекція 2 за питаннями Теми 2	Перевірка індивідуального завдання	4
	Лабораторна робота 1. Типи даних та змінні. Введення-виведення даних		
	<i>Самостійна робота</i>		
	Вивчення лекційного матеріалу, підготовка до лабораторного заняття		

ТЕМА 3	<i>Аудиторна робота</i>		
	Лекція 3 за питаннями Теми 3	Перевірка індивідуального завдання	4
	Лабораторна робота 2. Умовний оператор та розгалужений обчислювальний процес		
<i>Самостійна робота</i>			
	Вивчення лекційного матеріалу, підготовка до лабораторного заняття		
ТЕМА 4	<i>Аудиторна робота</i>		
	Лекція 4 за питаннями Теми 4	Перевірка індивідуального завдання	4
	Лабораторна робота 3. Цикл з визначеною кількістю ітерацій		
	<i>Самостійна робота</i>		
	Вивчення лекційного матеріалу, підготовка до лабораторного заняття		
ТЕМА 5	<i>Аудиторна робота</i>		
	Лекція 5 за питаннями Теми 5	Перевірка індивідуального завдання	4
	Лабораторна робота 4. Обробка символічних даних		
	<i>Самостійна робота</i>		
	Вивчення лекційного матеріалу, підготовка до лабораторного заняття		
ТЕМА 6	<i>Аудиторна робота</i>		

	Лекція 6 за питаннями Теми 6	Перевірка індивідуального завдання	4	
	Лабораторна робота 5. Цикли з перевіркою умов			
	<b>Самостійна робота</b>			
	Вивчення лекційного матеріалу, підготовка до лабораторного заняття			
ТЕМА 7.	<b>Аудиторна робота</b>			
	Лекція 7 за питаннями Теми 7	Перевірка індивідуального завдання Поточна КР	4 7	
	Лабораторна робота 6. Структурне програмування за допомогою функцій Поточна контрольна робота за змістовим модулем 1			
	<b>Самостійна робота</b>			
	Вивчення лекційного матеріалу, підготовка до лабораторного заняття			

1	2	3	4
---	---	---	---

Змістовий модуль 2. Структури даних Python			
ТЕМА 8	<b>Аудиторна робота</b>		
	Лекція 8 за питаннями Теми 8	Перевірка індивідуально го завдання	4
	Лабораторна робота 7. Обробка даних у послідовностях та списках		
	<b>Самостійна робота</b>		
Вивчення лекційного матеріалу, підготовка до лабораторного заняття			

ТЕМА 9	<b>Аудиторна робота</b>		
	Лекція 9 за питаннями Теми 9	Перевірка індивідуального завдання	4
	Лабораторна робота 8. Двовимірні структури даних		
	<b>Самостійна робота</b>		
Вивчення лекційного матеріалу, підготовка до лабораторного заняття			
ТЕМА 10	<b>Аудиторна робота</b>		
	Лекція 10 за питаннями Теми 10	Перевірка індивідуального завдання	4
	Лабораторна робота 9. Асоціативні структури даних		
	<b>Самостійна робота</b>		
Вивчення лекційного матеріалу, підготовка до лабораторного заняття			
ТЕМА 11	<b>Аудиторна робота</b>		
	Лекція 11 за питаннями Теми 11	Перевірка індивідуального завдання	4
	Лабораторна робота 11. Робота з файлами та файловою системою		
	<b>Самостійна робота</b>		
Вивчення лекційного матеріалу, підготовка до лабораторного заняття			
ТЕМА 12	<b>Аудиторна робота</b>		
	Лекція 12 за питаннями Теми 12	Презентація Поточна КР	5 8
	Поточна контрольна робота за змістовим модулем 2		
	<b>Самостійна робота</b>		
Вивчення лекційного матеріалу, підготовка до лабораторного заняття			

Іспит	40
<b>Загальна кількість балів</b>	<b>100</b>



## Рекомендована література

### Основна

1. Васильєв О.М. Програмування мовою Python. – Л.: Bohdan Books, 2022. – 504 с.
2. Беррі П. Head First. Python. – Х.: ФАБУЛА, 2021. – 624 с.
3. Маттес Е. Пришвидшений курс Python. – Л.: Видавництво Старого Лева, 2021. – 600 с.
4. Яценко Р.М. Опорний конспект лекцій «Програмування» [Електронний ресурс]. – Режим доступу : <https://pns.hneu.edu.ua/course/view.php?id=8027>.

### Додаткова

5. Фрімен Е., Робсон Е. Head First. Патерни проектування. – Х.: ФАБУЛА, 2020. – 672 с.
6. Мартін Р. Чистий кодер. – Х.: ФАБУЛА, 2023. – 256 с.
7. Крєневич А.П. Python у прикладах і задачах. Частина 1. Структурне програмування Навчальний посібник із дисципліни "Інформатика та програмування" – К.: ВПЦ "Київський Університет", 2017. – 206 с.
8. Крєневич А.П. Python у прикладах і задачах. Частина 2. Об'єктно-орієнтоване програмування. Навчальний посібник – К.: ВПЦ "Київський Університет", 2020. – 152 с.
9. 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. – Київ : КПІ ім. Ігоря Сікорського, 2018. – 195 с.

### Інформаційні ресурси в Інтернеті

10. W3Schools Online Web Tutorials Distribution [Електронний ресурс]. – Режим доступу : <https://www.w3schools.com/>.
11. Heroku: Cloud Application Platform [Електронний ресурс]. – Режим доступу : <https://www.heroku.com/>.
12. Сайт Національної бібліотеки України ім. Вернадського [Електронний ресурс]. – Режим доступу : [www.nbuv.gov.ua](http://www.nbuv.gov.ua).
13. Сайт Python Programming Language [Електронний ресурс]. – Режим доступу : <https://www.python.org/>.
14. Teach Python 3 and web design with 200+ exercises [Електронний ресурс]. – Режим доступу : <https://snakify.org/en/>