

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ
КАФЕДРА ІНТЕЛЕКТУАЛЬНИХ ПРОГРАМНИХ СИСТЕМ**

«ЗАТВЕРДЖУЮ»

Заступник декана
з навчальної роботи

_____ Кашпур О.Ф.

«__» _____ 2019 року

**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
ПРОГРАМУВАННЯ МОВОЮ PYTHON
для студентів**

галузь знань	12 Інформаційні технології
спеціальність	121 Інженерія програмного забезпечення
освітній рівень	бакалавр
освітня програма	Програмна інженерія
спеціалізація	Інтелектуальні системи
вид дисципліни	вибіркова

Форма навчання	денна
Навчальний рік	2021/2022
Семестр	6
Кількість кредитів ECTS	4
Мова викладання, навчання та оцінювання	українська
Форма заключного контролю	залік

Викладач: **к. ф.-м. н., доц. Верес М.М.** (лекції, лабораторні заняття).

Пролонговано: на 20__/20__ н. р. _____ (_____) «__» 20__ р.

на 20__/20__ н. р. _____ (_____) «__» 20__ р.

Розробник: Верес Максим Миколайович, к.ф.-м.н., доцент кафедри інтелектуальних програмних систем.

ЗАТВЕРДЖЕНО

Завідувач кафедри інтелектуальних програмних систем

_____ О.І. Провотар

Протокол №__ від «__»_____2019 р.

Схвалено науково-методичною комісією факультету комп'ютерних наук та кібернетики

Протокол від «__»_____2019 року №__

Голова науково-методичної комісії _____ Л.Л. Омельчук

«__»_____2019 року

Затверджено вченою радою факультету комп'ютерних наук та кібернетики

Протокол від «__»_____2019 року №__

Голова вченої ради факультету _____ А.В. Анісімов

1. Мета дисципліни – вивчення базових концепцій, механізмів та технік процедурного, об'єктно-орієнтованого, паралельного та web-орієнтованого програмування мовою Python 3, здобуття базових навичок проектування, розробки та тестування програмного забезпечення.

2. Попередні вимоги до опанування або вибору навчальної дисципліни. Для успішного вивчення дисципліни «Програмування мовою Python» студенти повинні відповідати наступним вимогам:

1. Успішне опанування курсів:
 1. Основи програмування.
 2. Програмування.
 3. Основи об'єктно-орієнтованого програмування.
 4. Розробка WEB-орієнтованих систем.
 5. Організація баз даних та знань.
 6. Об'єктно-орієнтоване програмування.
2. Знання:
 1. Основних концепцій процедурного та об'єктно-орієнтованого програмування.
 2. Основ WEB-програмування.
 3. Основ програмування мовами C++, JavaScript та Java.
 4. Основ проектування реляційних баз даних.
 5. Основ мови структурних запитів SQL.
 6. Основ тестування програмного забезпечення.
3. Вміння:
 1. Програмувати у процедурному та об'єктно-орієнтованому стилях.
 2. Програмувати мовами C++/ JavaScript/ Java.
 3. Формувати запити до реляційних баз даних на мові SQL.
 4. Проектувати класові ієрархії мовами C++/ JavaScript/ Java.
 5. Розробляти тестове покриття з використанням фреймворків Google Test/ Catch2/ Boost Test/ JUnit.
4. Володіння:
 1. Базовими навичками використання інтегрованих середовищ розробки програмного забезпечення CLion/ WebStorm/ IntelliJ IDEA/ Eclipse/ NetBeans/ Microsoft Visual Studio/ Microsoft Visual Studio Code.
 2. Базовими навичками роботи з системами управління реляційними базами даних SQLite/ MySQL/ MariaDB/ PostgreSQL.
 3. Англійською мовою на рівні не нижче Intermediate.

4. Анотація навчальної дисципліни. Навчальна дисципліна «Програмування мовою Python» є складовою освітньо-професійної програми підготовки фахівців за першим (бакалаврським) рівнем вищої освіти у галузі знань 12 Інформаційні технології спеціальністю 121 Інженерія програмного забезпечення в рамках освітньо-професійної програми Програмна інженерія.

Дана дисципліна належить до переліку дисциплін вільного вибору студента, блок вибору «Інтелектуальні системи». Викладається у **6 семестрі в обсязі – 120 год., (4 кредити ECTS)**, зокрема: лекції – 28 год., лабораторні заняття – 14 год., консультації – 2 год., самостійна робота – 76 год. У курсі передбачено 2 змістовні частини, 2 контрольні роботи та 3 лабораторні роботи. Завершується дисципліна – **заліком**.

В результаті вивчення навчальної дисципліни студенти повинні:

Знати:

1. Основні поняття, концепції та техніки процедурного програмування мовою Python 3.

2. Основні поняття, концепції та техніки об'єктно-орієнтованого програмування мовою Python 3.
3. Основні поняття, концепції та техніки паралельного програмування мовою Python 3.
4. Основні поняття, концепції та техніки web-орієнтованого програмування мовою Python 3.
5. Базові підходи до проектувати та розробки тестового покриття програмного коду на мові Python 3.

Вміги:

1. Проектувати та розробляти програмне забезпечення з використанням технік процедурного, об'єктно-орієнтованого та паралельного програмування мовою Python 3.
2. Проектувати та розробляти графічні користувацькі інтерфейси з використанням мови Python 3 та бібліотеки PyQt.
3. Проектувати та розробляти web-орієнтовані програмні системи з використанням мови Python 3 та фреймворку Django.
4. Проектувати та розробляти тестове покриття програмного коду на мові Python3 з використанням фреймворку PyTest.

Дисципліна «Програмування мовою Python» є логічним продовженням, доповненням та розширенням дисципліни «Об'єктно-орієнтоване програмування» та є базовою для дисциплін «Системи штучного інтелекту» та «Метапрограмування» першого (бакалаврського) рівня вищої освіти у галузі знань 12 Інформаційні технології за спеціальністю 121 Інженерія програмного забезпечення, в рамках освітньо-професійної програми «Програмна інженерія».

4. Завдання (навчальні цілі). Основними завданнями дисципліни «Програмування мовою Python» є набуття знань, умінь та навичок (компетентностей) на рівні новітніх досягнень в області програмування мовою Python відповідно до освітньої кваліфікації бакалавр з інженерії програмного забезпечення. Зокрема, розвивати:

- Здатність до абстрактного мислення, аналізу та синтезу (ЗК01).
- Здатність застосовувати знання у практичних ситуаціях (ЗК02).
- Здатність спілкуватися державною мовою як усно, так і письмово (ЗК03).
- Здатність спілкуватися іноземною мовою як усно, так і письмово (ЗК04).
- Здатність вчитися і оволодівати сучасними знаннями (ЗК05).
- Здатність до пошуку, оброблення та аналізу інформації з різних джерел (ЗК06).
- Здатність розробляти архітектури, модулі та компоненти програмних систем (СК03).
- Володіння знаннями про інформаційні моделі даних та системи, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних (СК07).
- Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення (СК13).
- Здатність до алгоритмічного та логічного мислення (СК14).
- Здатність реалізувати сучасні методи побудови та аналізу ефективних алгоритмів в конкретних застосуваннях (СК15.1).

5. Результати навчання за дисципліною.

Результат навчання (1. знати; 2. вміти; 3. комунікація; 4. автономність та відповідальність)		Форми (та/або методи і технології) викладання і навчання	Методи оцінювання та пороговий критерій оцінювання (за необхідності)	Відсоток у підсумковій оцінці з дисципліни	
Код	Результат навчання				
PH1.1	Знати основні особливості синтаксису, базові типи та структури даних в рамках мови Python 3.	Лекції, самостійна робота,	Контрольна робота № 1.	3%	
	Знати основи процедурного програмування мовою Python 3.				
	Знати основи об'єктно-орієнтованого програмування мовою Python 3.				
PH1.4	Знати основні підходи до проектування та розробки програмних модулів та пакетів мовою Python 3.				3%
PH1.5	Знати основи паралельного програмування мовою Python 3.				3%
PH1.6	Знати основи проектування графічних користувацьких інтерфейсів мовою Python 3.				3%
PH1.7	Знати загальну архітектуру та конфігураційні можливості Django-проектів.				3%
PH1.8	Знати призначення та основні можливості шаблонів та форм в рамках фреймворку Django.				4%
PH1.9	Знати призначення та основні можливості контролерів та моделей в рамках фреймворку Django.				4%
PH1.10	Знати основні методи обробки URL-запитів та можливості конфігурування системи управління контентом в рамках фреймворку Django.				4%
PH1.11	Знати основні можливості фреймворку PyTest для тестування Python 3-програм та Django-проектів.				4%
PH2.1	Вміти проектувати архітектуру програмних модулів та пакетів мовою Python 3.			3%	
	Вміти проектувати та розробляти програмні модулі та пакети мовою				

	Python 3 з використанням процедурного програмування.	опрацювання рекомендованої літератури, виконання рекомендованих домашніх завдань.	
PH2.3	Вміти проектувати та розробляти програмні модулі та пакети мовою Python 3 з використанням об'єктно-орієнтованого програмування.		5%
PH2.4	Вміти проектувати та розробляти програмні модулі та пакети мовою Python 3 з використанням паралельного програмування.		4%
PH2.5	Вміти проектувати та розробляти графічні користувацькі інтерфейси мовою Python 3.		3%
PH2.6	Вміти проектувати та розробляти шаблони користувацьких інтерфейсів за допомогою фреймворку Django.		7%
PH2.7	Вміти проектувати та розробляти контролери та моделі за допомогою фреймворку Django.		7%
PH2.8	Вміти проектувати та розробляти модулі обробки URL-запитів.		7%
PH2.9	Вміти конфігурувати систему управління контентом в рамках фреймворку Django.		7%
PH2.10	Вміти проектувати та розробляти тестове покриття Python 3-програм та Django-проектів з використанням фреймворку PyTest.		7%
PH3.1	Консультуватися з викладачем стосовно питань що виникають у ході опанування теоретичного матеріалу.		
PH3.2	Обговорювати з колегами та викладачем проблеми питання що виникають у ході виконання лабораторних робіт.	2%	
PH4.1	Закріплення та поглиблення набутих на лекціях теоретичних знань щодо проектування та розробки програмних систем з використанням мови програмування Python 3 та фреймворку Django.		3%
PH4.2	Закріплення та поглиблення набутих під час виконання лабораторних робіт практичних навичок проектування та розробки програмних систем з використанням мови програмування Python 3 та фреймворку Django.		3%

даних і знань.																									
ПРН14. Застосовувати на практиці інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення.																									
ПРН15. Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення.																									
ПРН17. Вміти застосовувати методи компонентної розробки програмного забезпечення.																									
ПРН18. Знати та вміти застосовувати інформаційні технології обробки, зберігання та передачі даних.																									
ПРН25.1. Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.																									
ПРН27.1. Аналізувати, оцінювати і вибирати інструментальні та обчислювальні засоби, технології, алгоритмічні і програмні рішення при проектуванні та розробці програмних систем.																									

7. Схема формування оцінки.

7.1 Форми оцінювання студентів.

Семестрове оцінювання. Максимальна/мінімальна кількість балів, які можуть бути отримані студентами протягом семестру, становить **100/60 балів** за яких:

1. Контрольна робота № 1: РН1.1 – РН1.6, РН2.1 – РН 2.5 – **15/9 балів**.
2. Контрольна робота № 2: РН1.7 – РН1.11, РН2.6 – РН 2.10 – **15/9 балів**.
3. Лабораторна робота № 1: РН2.1 – РН 2.5, РН3.1, РН3.2, РН4.1, РН4.2 – **30/18 балів**.
4. Лабораторна робота № 2: РН2.6 – РН 2.10, РН3.1, РН3.2, РН4.1, РН4.2 – **40/24 бали**.

Підсумкове оцінювання (у формі заліку):

- Залікові бали визначаються як сума оцінок/балів за всіма успішно оціненими результатами навчання передбачених даною програмою.
- Оцінки нижче від мінімального порогового рівня не додаються.
- Мінімальний пороговий рівень для сумарної оцінки за всіма компонентами становить 60% від максимально можливої кількості балів.

7.2 Організація оцінювання.

Терміни проведення форм оцінювання:

1. Контрольна робота № 1: до 9 тижня семестру.
2. Контрольна робота № 2: до 19 тижня семестру.
3. Лабораторна робота № 1: до 10 тижня семестру.
4. Лабораторна робота № 2: до 18 тижня семестру.

У випадку отримання незадовільної оцінки (менше 50% від встановленого максимуму) за контрольну роботу, студенти мають право на одне перескладання цієї роботи у визначений викладачем термін із можливістю отримати не більше 80% балів від встановленої для неї максимальної кількості балів.

У випадку відсутності студентів з поважних причин відпрацювання та перездачі контрольних робіт здійснюються у відповідності до «Положення про порядок оцінювання знань студентів при кредитно-модульній системі організації навчального процесу» від 1 жовтня 2010 року.

У випадку встановлення фактів порушення студентами академічної доброчесності передбачених пунктом 9.8.2 «Положення про організацію освітнього процесу у Київському національному університеті імені Тараса Шевченка» що діє від 07.05.2018, вони будуть притягнуті до відповідальності передбаченої пунктом 9.8.3 цього положення.

7.3 Шкала відповідності оцінок.

Зараховано / Passed	60-100
Не зараховано / Fail	0-59

7.4 Структура навчальної дисципліни. Тематичний план лекцій і лабораторних занять.

№ лекції	Назва лекції	Кількість годин		
		Лекції	Лабораторні заняття	Самостійна робота
Частина 1. Основи програмування мовою Python 3.				
1	Вступ до курсу. Мета курсу. Зв'язок з іншими	2		2

	курсами. Структура курсу. Рекомендована література. Необхідний інструментарій. Лабораторні роботи. Вимоги до студентів. Форми контролю.			
	Тема 1. Загальні особливості синтаксису. Базові оператори. Умовні оператори. Циклічні оператори.			
2	Тема 2. Базові типи та структури даних.	2		7
3	Тема 3. Функції. Області видимості та простори імен. Аргументи функцій. Розширені можливості функцій.	2		4
4	Тема 4. Ітератори. Генератори. Функції-генератори. Вирази-генератори.	2		4
5	Тема 5. Об'єкти. Класи. Класові ієрархії. Абстрактні класи.	2	2	7
6	Тема 6. Обробка винятків. Модулі та пакети. Документування коду.	2		4
7	Тема 7. Процеси. Потoki. Підпрограми. Паралелізм. Конкурування. Синхронізація.	2	2	5
8	Тема 8. Графічні користувацькі інтерфейси: Бібліотека PyQt.	2		5
Контрольна робота 1				2
Всього по частині 1		16	4	40
Частина 2. Основи Web-програмування мовою Python 3.				
9	Тема 9. Встановлення та конфігурування фреймворку Django. Створення Django-проектів.	2		4
10	Тема 10. Шаблони користувацьких інтерфейсів. Генерація та обробка форм. Валідація даних.	2	2	6
11	Тема 11. Контролери та обробка запитів користувачів. Робота з динамічними даними. Куки. Сесії. Фільтри.	2	2	7
12	Тема 12. Моделі та активні записи. Представлення, збереження та обробка даних. Міграції. Асоціації та відношення між типами.	2	2	6
	Тема 13. Маршрутизація та обробка URL-запитів. CRUD. REST.			
	Тема 14. Конфігурування система управління контентом.			
14	Тема 15. Тестування та налагодження Django-проектів. Фреймворк PyTest.	2	2	5
Контрольна робота 2				2

Всього по частині 2	12	10	36
Консультація		2	
ВСЬОГО	28	14	76

Загальний обсяг – **120** год., в тому числі:

Лекції – **28** год.

Лабораторні заняття – **14** год.

Консультації – **2** год.

Самостійна робота – **76** год.

Лабораторні роботи.

Лабораторна робота № 1: Проектування та розробка спеціалізованих алгоритмічних програмних пакетів.

Лабораторна робота № 2: Проектування та розробка спеціалізованих веб-орієнтованих програмних систем.

9. Рекомендовані джерела.

Основні:

1. M. Lutz: Learning Python: Powerful Object-Oriented Programming, 5th ed. // O'Reilly Media, Inc., 2013.
2. D. Beazley, B.K. Jones: Python Cookbook: Recipes for Mastering Python 3, 3rd ed. // O'Reilly Media, Inc., 2013.
3. A. Martelli, A. Ravenscroft, S. Holden: Python in a Nutshell: The Definitive Reference, 3rd ed. // O'Reilly Media, Inc., 2017.
4. B. Lubanovic: Introducing Python: Modern Computing in Simple Packages. // O'Reilly Media, Inc., 2015.
5. J. Hunt: A Beginners Guide to Python 3 Programming. // Springer, 2019.
6. J. Hunt: Advanced Guide to Python 3 Programming. // Springer, 2019.
7. N. Ceder: The Quick Python Book, 3rd ed. // Manning Publications Co., 2018.
8. D. Hellmann: The Python 3 Standard Library by Example, 2nd ed. // Pearson Education, Inc., 2017.
9. C. Hattingh: Using Asyncio in Python 3: Understanding Python's Asynchronous Programming Features. // O'Reilly Media, Inc., 2018.
10. F. Romano, D. Phillips, R. van Hattem: Python: Journey from Novice to Expert. // Packt Publishing, 2016.
11. M. Jaworski, T. Ziade: Expert Python Programming: Become an ace Python programmer by learning best coding practices and advance-level concepts with Python 3.5, 2nd ed. // Packt Publishing, 2016.
12. H.J.W. Percival: Test-Driven Development with Python: Obey the Testing Goat: Using Django, Selenium and JavaScript. // O'Reilly Media, Inc., 2014.
13. J. Kronika, A. Bendoraitis: Django 2 Web Development Cookbook: 100 practical recipes on building scalable Python web apps with Django 2, 3rd ed. // Packt Publishing, 2018.

14. B. Okken: Python Testing with pytest: Simple, Rapid, Effective, and Scalable. // The Pragmatic Programmers, LLC, 2017.
15. G.L. Turnquist, B.N. Das: Python Testing Cookbook: Easy solutions to test your Python projects using test-driven development and Selenium, 2nd ed. // Packt Publishing, 2018.
16. B.M. Harwani: Qt5 Python GUI Programming Cookbook: Building responsive and powerful cross-platform applications with PyQt. // Packt Publishing, 2018.

Додаткові:

1. L. Ramalho: Fluent Python: Clear, Concise, and Effective Programming. // O'Reilly Media, Inc., 2015.
2. K. Reitz, T. Schlusser: The Hitchhiker's Guide to Python: Best Practices for Development. // O'Reilly Media, Inc., 2016.
3. B. Stephenson: The Python Workbook: A Brief Introduction with Exercises and Solutions, 2nd ed. // Springer, 2019.
4. A.B. Downey: Think Python: How to Think Like a Computer Scientist, 2nd ed. // O'Reilly Media, Inc., 2016.
5. D. Phillips: Python 3 Object-oriented Programming: Unleash the power of Python 3 objects, 2nd ed. // Packt Publishing, 2015.
6. I. Kalb: Learn to Program with Python 3: A Step-by-Step Guide to Programming, 2nd ed. // Apress, 2018.
7. J.B. Browning, M. Alchin: Pro Python 3: Features and Tools for Professional Development, 3rd ed. // Apress, 2019.
8. M.L. Hetland: Beginning Python: From Novice to Professional, 3rd ed. // Apress, 2017.
9. P. Gries, J. Campbell, J. Montojo: Practical Programming, Third Edition: An Introduction to Computer Science Using Python 3.6. // The Pragmatic Bookshelf, 2017.
10. M. Gorelick, I. Ozsvald: High Performance Python: Practical Performant Programming for Humans. // O'Reilly Media, Inc., 2014.
11. M. Lutz: Python Pocket Reference: Python in Your Pocket, 5th ed. // O'Reilly Media, Inc., 2014.
12. M. Summerfield: Python in Practice: Create Better Programs Using Concurrency, Libraries, and Patterns. // Addison-Wesley, 2014.
13. A. Mele: Django 2 by Example: Build powerful and reliable Python web applications from scratch. // Packt Publishing, 2018.
14. T. Aratyn: Building Django 2.0 Web Applications: Create enterprise-grade, scalable Python web applications easily with Django 2.0. // Packt Publishing, 2018.
15. F. Marani: Practical Django 2 and Channels 2: Building Projects and Applications with Real-Time Capabilities. // Apress, 2019.
16. J. Elman, M. Lavin: Lightweight Django: Using Rest, Websockets & Backbone. // O'Reilly Media, Inc., 2015.
17. A. Pajankar: Python Unit Test Automation: Practical Techniques for Python Developers and Testers. // Apress, 2017.
18. B. Oliveira: Pytest Quick Start Guide: Write better Python code with simple and maintainable tests. // Packt Publishing, 2018.

10 Електронні ресурси.

1. The Official Home of the Python Programming Language (<https://www.python.org/>)
2. Python 3/8 Documentation (<https://docs.python.org/3.8/>)
3. PEP 0 – Index of Python Enhancement Proposals (<https://www.python.org/dev/peps/>)
4. Repl.it – Online Python Editor and IDE (<https://repl.it/languages/python3>)
5. PyQt (<https://riverbankcomputing.com/software/pyqt/intro>)
6. PyTest Framework (<https://docs.pytest.org/>)
7. Django Web framework (<https://www.djangoproject.com/>)
8. PostgreSQL (<https://www.postgresql.org/>)