



# ПРИКЛАДНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ МОВОЮ PYTHON

## Робоча програма навчальної дисципліни (Силабус)

### Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський)</i>
Галузь знань	<i>12 Інформаційні технології</i>
Спеціальність	<i>121 Інженерія програмного забезпечення</i>
Освітня програма	<i>Інженерія програмного забезпечення мультимедійних та інформаційно-пошукових систем</i>
Статус дисципліни	<i>Вибіркова</i>
Форма навчання	<i>Очна (денна)</i>
Рік підготовки, семестр	<i>3 рік підготовки, 6 семестр</i>
Обсяг дисципліни	<i>Лекції: 36 год., комп'ютерний практикум: 18 год., самостійна робота: 66 год.</i>
Семестровий контроль/ контрольні заходи	<i>Залік, модульна контрольна робота, календарний контроль</i>
Розклад занять	<i>Згідно розкладу на весняний семестр поточного навчального року (rozklad.kpi.ua)</i>
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	<i>Лектор: к.т.н., доцент, Новак Дмитро Сергійович, <a href="mailto:novak.knutd@gmail.com">novak.knutd@gmail.com</a> Комп'ютерний практикум: к.т.н., доцент, Новак Дмитро Сергійович, <a href="mailto:novak.knutd@gmail.com">novak.knutd@gmail.com</a></i>
Розміщення курсу	<i>Google classroom: <a href="https://classroom.google.com/u/0/c/NTYwMzQ4Nzc5NjI1">https://classroom.google.com/u/0/c/NTYwMzQ4Nzc5NjI1</a></i>

### Програма навчальної дисципліни

#### 1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

*Вивчення дисципліни «Прикладне програмне забезпечення мовою Python» дозволяє сформувати у здобувачів освіти компетенції, необхідні для розв'язання практичних задач професійної та наукової діяльності, пов'язаної з підготовкою та розробкою програмного забезпечення мовою Python.*

***Метою** вивчення дисципліни «Прикладне програмне забезпечення мовою Python» є надання студентам знань і практичних навичок в області розробки програмного забезпечення з використанням мови Python.*

***Предметом** дисципліни «Прикладне програмне забезпечення мовою Python» є вивчення основ програмування мовою Python та застосування її для створення різноманітних програмних продуктів і розв'язання практичних задач.*

*Вивчення дисципліни «Прикладне програмне забезпечення мовою Python» сприяє формуванню у студентів фахової компетентності (ФК) за освітньою програмою:*

***ФКО8** Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення.*

Вивчення дисципліни «Прикладне програмне забезпечення мовою Python» сприяє формуванню у студентів наступних програмних результатів навчання (ПРН) за освітньою програмою:

**ПРН01** Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки.

**ПРН07** Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.

**ПРН18** Знати та вміти застосовувати інформаційні технології обробки, зберігання та передачі даних.

## **2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)**

Успішному вивченню дисципліни «Прикладне програмне забезпечення мовою Python» передують вивчення дисциплін: «Основи програмування», «Програмування» та «Бази даних» навчального плану підготовки бакалаврів за спеціальністю 121 Інженерія програмного забезпечення.

Отримані при засвоєнні дисципліни «Прикладне програмне забезпечення мовою Python» теоретичні знання та практичні уміння сприяють успішному виконанню курсових проєктів та дипломної роботи бакалаврів.

## **3. Зміст навчальної дисципліни**

Дисципліна «Прикладне програмне забезпечення мовою Python» передбачає вивчення таких тем:

Тема 1. Основи програмування мовою Python.

Тема 2. Особливості розробки програмного забезпечення мовою Python.

Модульна контрольна робота.

Залік.

## **4. Навчальні матеріали та ресурси**

### **Базова література:**

1. Новак Д.С. Прикладне програмне забезпечення мовою Python. Навчально-методичний комплекс у Google класі. Доступ зареєстрованим студентам.

### **Додаткова література:**

2. Sutor R. S. *Dancing with python: Learn python software development from scratch and get started with quantum computing.* – Packt Pub. 2021, 744 p.

3. Slatkin B. *Effective Python.* – Addison-Wesley Professional. 2019, 480 p.

4. Flask. Електронний ресурс. Режим доступу: <https://www.tutorialspoint.com/flask/index.htm>

5. Pillai A. B. *Software architecture with Python.* – Packt Publishing Ltd. 2017, 619 p.

6. Bird A., Han L. C., Jiménez M. C., Lee G., Wade C. *The Python Workshop: Learn to code in Python and kickstart your career in software development or data science.* – Van Haren Publishing. 2019, 608 p.

7. Wilkes M. *Advanced Python Development: Using Powerful Language Features in Real-World Applications (1st ed.).* – Apress. 2020, 628 p

8. Shaw B., Badhwar S., Bird A. *Web Development with Django.* Van Haren Publishing. 2021, 826 p.

Матеріали знаходяться у вільному доступі в Інтернеті.

## Навчальний контент

### 5. Методика опанування навчальної дисципліни (освітнього компонента)

№ з/п	Тип навчального заняття	Опис навчального заняття
<i>Тема 1. Основи програмування мовою Python</i>		
1	<i>Лекція 1. Основи Python.</i>	<i>Історія та особливості мови Python. Основні поняття мови Python.</i>
2	<i>Лекція 2. Типи та структури даних.</i>	<i>Логічні оператори. Цикли. Списки. Кортежі. Рядки. Словники.</i>
3	<i>Лекція 3. Робота з винятками та логування даних.</i>	<i>Винятки. Генерація винятків в Python. Власні винятки. Логування. Логування даних. Логування винятків.</i>
4	<i>Лекція 4. Робота з функціями.</i>	<i>Функції. Аргументи функцій. Перевірки функцій. Функції як об'єкти. Область видимості функції. Локальні і глобальні змінні. Анонімні функції. Замикання. Декоратори.</i>
5	<i>Комп'ютерний практикум 1. Робота зі списками, словниками, циклами та вбудованими функціями.</i>	<i>Завдання: продемонструвати свої навички роботи зі списками та словниками, освоїти основні методи для роботи з ними; вивчити види циклів в Python; розглянути основні вбудовані функції мови програмування Python і навчитися з ними працювати.</i>
6	<i>Лекція 5. Модулі і пакети.</i>	<i>Python-код в нормальному і інтерактивному режимі. Інтерпретатор CPython. Модулі в Python. Використання пакетів в Python.</i>
7	<i>Комп'ютерний практикум 2. Робота з ітераторами та генераторами.</i>	<i>Завдання: вивчити поняття ітератора та генератора в Python, а також їх переваги; ознайомитися з їх користуванням.</i>
8	<i>Лекція 6. Робота з файлами.</i>	<i>Запис даних у файл. Створення та видалення каталогів. Інструкція with. Структуровані текстові файли. Характеристики форматів XML та JSON.</i>
9	<i>Комп'ютерний практикум 3. Робота з файлами та форматування даних у формат JSON.</i>	<i>Завдання: вивчити роботу з файлами з допомогою функцій із стандартної бібліотеки; розглянути поняття синтаксичного аналізу тексту та ознайомитися з його застосуванням в Python; вивчити можливості взаємодії Python з форматом зберігання даних JSON.</i>
<i>Тема 2. Особливості розробки програмного забезпечення мовою Python</i>		
10	<i>Лекція 7. Вступ в об'єктно-орієнтоване програмування в мові Python</i>	<i>Абстракція, інкапсуляція, спадкування та поліморфізм. Створення класу та його атрибути.</i>
11	<i>Лекція 8. Основи об'єктно-орієнтованого програмування в мові Python</i>	<i>Методи об'єкта і методи класу. Інкапсуляція (Public, Protected, Private). Спадкування і поліморфізм. Множинне спадкування. Проектування ієрархії класів і клас object.</i>
12	<i>Лекція 9. Поглиблене вивчення об'єктно-орієнтованого програмування в мові Python</i>	<i>Міксини в Python. Агрегація. Ітератори та генератори. Корутини. Сінглтон. Метакласи. Динамічне створення класів.</i>

13	Комп'ютерний практикум 4. Розробка програми з використанням ООП.	Завдання: ознайомитися з методологією об'єктно-орієнтованого програмування, вивчити реалізацію даної методології мовою Python
14	Лекція 10. Unit-тестування в мові Python.	Фреймворки для проведення автономного тестування в Python. Основні структурні елементи unit-test. Інтерфейс командного рядка. Завантаження і запуск тестів. Тестування за категоріями. Комбінування тестів.
15	Лекція 11. Вступ у Flask	Контексти у Flask. Обробка запиту. Відповідь сервера. Перехоплення запитів. Шаблони.
16	Лекція 12. Основи Flask	Основи шаблонізатора Jinja. Коментарі. Оголошення змінних. Фільтри. Макроси. Екранування. Створення URL у Flask. Робота зі статичними файлами у Flask. Автоматичний імпорт об'єктів.
17	Лекція 13. Поглиблене вивчення Flask	Робота з формами у Flask. Форми в консолі. Візуалізація форми. Робота з підтвердженням форми. Налаштування cookies у Flask. Створення моделей. Розгортання Flask-додатку.
18	Лекція 14 Бібліотека NumPy	Масиви. Операції над масивами. Модифікація масивів. Приклади роботи з NumPy.
19	Лекція 15. Бібліотека Matplotlib	Основні графічні команди. Текстові параметри графіків. Архітектура вікна графіку. Розміщення графіків. Способи відображення легенди. Збереження графіків у файл.
20	Комп'ютерний практикум 5. Візуалізація результатів роботи математичних алгоритмів з використанням NumPy та Matplotlib	Завдання: розглянути можливості пакету NumPy, його основні складники і можливості для обробки багатовимірних масивів даних; вивчити бібліотеку Matplotlib і її основні інструменти для створення графіків і діаграм.
21	Лекція 16. Написання ботів для Telegram	Telegram API і Telegram Bot API. Обробники команд та кнопок. Створення telegram-бота для отримання інформації про погоду.
22	Лекція 17. Підсумкове заняття. Залік.	Повторення вивченого матеріалу. Залікова контрольна робота.

## 6. Самостійна робота студента

Дисципліна «Прикладне програмне забезпечення мовою Python» ґрунтується на самостійній підготовці до аудиторних занять на теоретичні та практичні теми.

№ з/п	Назва теми, що виноситься на самостійне опрацювання	Кількість годин	Література
1	Підготовка до лекції 1	2	1-3
3	Підготовка до лекції 2	2	1-3
4	Підготовка до лекції 3	2	1-3
5	Підготовка до лекції 4	2	1-3
6	Підготовка до комп'ютерного практикуму 1	2	1-3
7	Підготовка до лекції 5	2	1-3

8	Підготовка до комп'ютерного практикуму 2	3	1-3
9	Підготовка до лекції 6	2	1-3
10	Підготовка до комп'ютерного практикуму 3	3	1-3
11	Підготовка до лекції 7	2	1-3
12	Підготовка до лекції 8	2	1-3
13	Підготовка до лекції 9	2	1-3
14	Підготовка до комп'ютерного практикуму 4	3	1-3
15	Підготовка до лекції 10	2	1-3
16	Підготовка до лекції 11	2	1-3
17	Підготовка до лекції 12	2	1-3
18	Підготовка до лекції 13	2	1-3
19	Підготовка до лекції 14	2	1-3
20	Підготовка до лекції 15	2	1-3
21	Підготовка до комп'ютерного практикуму 5	3	1-3
22	Підготовка до лекції 16	2	1-3
23	Підготовка до модульної контрольної роботи	20	1-3

## Політика та контроль

### 7. Політика навчальної дисципліни (освітнього компонента)

**Відвідування занять.** Відсутність на аудиторному занятті не передбачає нарахування штрафних балів, оскільки фінальний рейтинговий бал студента формується виключно на основі оцінювання результатів навчання. Разом з тим, обговорення результатів виконання тематичних завдань, а також презентація / публічний виступ та участь у обговореннях та доповнення на семінарах оцінюватимуться під час аудиторних занять. Для активної участі у роботі семінару студент готується за рекомендованою викладачем до певного семінарського заняття літературою. Участь у роботі семінару також передбачає підготування доповідей та співдоповідей у межах усіх занять.

**Пропущені контрольні заходи оцінювання.** Кожен студент має право відпрацювати пропущені з поважної причини (лікарняний, мобільність тощо) заняття за рахунок самостійної роботи. Детальніше за посиланням: <https://kpi.ua/files/n3277.pdf>.

**Процедура оскарження результатів контрольних заходів оцінювання.** Студент може підняти будь-яке питання, яке стосується процедури контрольних заходів та очікувати, що воно буде розглянуто згідно із наперед визначеними процедурами. Студенти мають право аргументовано оскаржити результати контрольних заходів, пояснивши з яким критерієм не погоджуються відповідно до оціночного. Календарний контроль проводиться з метою підвищення якості навчання студентів та моніторингу виконання студентом вимог силабусу.

**Академічна доброчесність.** Політика та принципи академічної доброчесності визначені у розділі 3 Кодексу честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського». Детальніше: <https://kpi.ua/code>.



**Норми етичної поведінки.** Норми етичної поведінки студентів і працівників визначені у розділі 2 Кодексу честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського». Детальніше: <https://kpi.ua/code>.

**Інклюзивне навчання.** Засвоєння знань та умінь в ході вивчення дисципліни «Науково-дослідна діяльність у комп'ютерній інженерії» може бути доступним для більшості осіб з особливими освітніми потребами, окрім здобувачів з серйозними вадами зору, які не дозволяють виконувати завдання за допомогою персональних комп'ютерів, ноутбуків та/або інших технічних засобів.

**Навчання іноземною мовою.** У ході виконання завдань студентам може бути рекомендовано звернутися до англomовних джерел. Призначення заохочувальних та штрафних балів Відповідно до Положення про систему оцінювання результатів навчання сума всіх заохочувальних балів не може перевищувати 10% рейтингової шкали оцінювання.

Всі студенти повинні відвідувати лекційні та практичні заняття, на яких потрібно активно працювати над засвоєнням навчального матеріалу. За об'єктивних причин (наприклад - хвороба, міжнародне стажування) навчання може відбуватись в он-лайн формі індивідуально за погодженням із керівником курсу.

**Політика щодо дедлайнів та перекладання:**

Роботи, які здаються із порушенням термінів без поважних причин, оцінюються на нижчу оцінку. Перекладання модулів відбувається із дозволу деканату за наявності поважних причин (наприклад, лікарняний).

**Політика щодо академічної доброчесності:**

Усі письмові роботи перевіряються на наявність плагиату і допускаються до захисту із коректними текстовими запозиченнями не більше 20%. Списування під час контрольних робіт заборонені (в т. ч. із використанням мобільних пристроїв).

## **8. Види контролю та рейтингова система оцінювання результатів навчання (PCO)**

Протягом семестру студенти виконують 5 комп'ютерних практикумів. Максимальна кількість балів за кожний комп'ютерний практикум: 15 балів.

Бали нараховуються за:

- якість виконання комп'ютерного практикуму: 0-6 бали;
- відповідь під час захисту комп'ютерного практикуму: 0-6 бали;
- своєчасне представлення роботи до захисту: 0-3 бали.

Критерії оцінювання якості виконання:

- 3-6 бали – робота виконана якісно, в повному обсязі;
- 1-4 бали – робота виконана якісно, в повному обсязі, але має недоліки;
- 0 балів – робота виконана не в повному обсязі, або містить суттєві помилки.

Критерії оцінювання відповіді:

- 3-6 бали – відповідь повна, добре аргументована;
- 1-4 бал – у відповіді є суттєві помилки;
- 0 балів – немає відповіді або відповідь невірна.

Критерії оцінювання своєчасності представлення роботи до захисту:

- 1-3 бали – робота представлена до захисту не пізніше вказаного терміну;
- 0 балів – робота представлена до захисту пізніше вказаного терміну.

Максимальна кількість балів за виконання та захист комп'ютерних практикумів:  
15 балів × 5 комп. практ. = 75 балів.

Завдання на **модульну контрольну роботу** складається з 5 питань – 3 теоретичних та 2 практичних. Відповідь на кожне теоретичне/практичне запитання оцінюється 5 балами.

Критерії оцінювання кожного теоретичного/практичного запитання модульної контрольної роботи:

4-5 балів – відповідь вірна, повна, добре аргументована;

3-4 балів – відповідь вірна, але неповна або погано аргументована;

2-3 бали – у відповіді є незначні помилки;

1-2 бали – у відповіді є суттєві помилки;

0 балів – немає відповіді або відповідь невірна.

Максимальна кількість балів за модульну контрольну роботу:

5 балів × 3 теоретичні запитання + 5 балів × 2 практичні запитання = 25 балів.

Рейтингова шкала з дисципліни дорівнює:

$R = R_c = 75 \text{ балів} + 25 \text{ балів} = 100 \text{ балів}$ .

За описом:  $R = R_{\text{комп.практ}} + R_{\text{МКР}} = 75 + 25 \text{ балів} = 100 \text{ балів}$

Календарний контроль: провадиться двічі на семестр як моніторинг поточного стану виконання вимог силабусу.

На першій атестації (8-й тиждень) студент отримує «зараховано», якщо його поточний рейтинг не менше 50 % від максимальної кількості балів, яку може отримати студент до першої атестації (20 балів).

На другій атестації (14-й тиждень) студент отримує «зараховано», якщо його поточний рейтинг не менше 50 % від максимальної кількості балів, яку може отримати студент до другої атестації (30 балів).

Семестровий контроль: **залік**.

Умови допуску до семестрового контролю:

При семестровому рейтингу ( $r_c$ ) не менше 60 % (60 балів) та зарахуванні усіх робіт комп'ютерного практикуму.

Необхідною умовою допуску до заліку є виконання і захист комп'ютерного практикуму.

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

Кількість балів	Оцінка
100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно
64-60	Достатньо
Менше 60	Незадовільно
Не виконані умови допуску	Не допущено

Складено к.т.н., доц. Новак Д.С.

Ухвалено кафедрою ПЗКС (протокол №8 від 25.01.2023)

Погоджено Методичною комісією факультету прикладної математики (протокол №6 від 27.01.2023)