

**Міністерство освіти і науки України  
Таврійський державний агротехнологічний університет  
імені Дмитра Моторного**

**СЦИЛІЩИН Ю.О., ШАРОВ С.В.**

**ПРИКЛАДНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ (PYTHON)**

**Лабораторний практикум**

Запоріжжя  
2024

УДК 004.9(076)

C-41

*Рекомендовано рішенням вченої ради факультету енергетики і комп'ютерних технологій Таврійського державного агротехнологічного університету імені Дмитра Моторного  
(Протокол №7 від 12 березня 2024 р.)*

**Рецензенти:**

**Вершков О.О.** – к.т.н., доцент, завідувач кафедри інженерної механіки та комп'ютерного проектування Таврійського державного агротехнологічного університету імені Дмитра Моторного.

**Медведєва М.О.** – к.пед.н., доцент, завідувач кафедри інформатики і інформаційно-комунікаційних технологій Уманського державного педагогічного університету імені Павла Тичини.

**Сіциліцин Ю.О., Шаров С.В.**

**C-41 Прикладне програмне забезпечення (Python):** лабораторний практикум. Запоріжжя, 2024. 147 с.

**ISBN**

Основне призначення лабораторного практикума полягає у відображенні теоретичної інформації, інструкції до виконання лабораторних робіт з дисципліни «Прикладне програмне забезпечення (Python)». В лабораторному практикумі розглянуті основні поняття та структури, що використовуються у програмуванні (змінні, типи даних, вирази, процедури введення/виведення, підпрограми, функції), а також підготовка програм із застосуванням згаданих елементів разом з розробкою алгоритмів для вирішення проблем і аналіз їхньої коректності. Курс також охоплює основи ООП на мові програмування Python. Лабораторний практикум призначений для здобувачів ступеня вищої освіти «Бакалавр» зі спеціальності 122 «Комп'ютерні науки», науково-педагогічних працівників, що викладають курс програмування на мові Python.

ISBN

УДК 004.9(076)

© Сіциліцин Ю.О., Шаров С.В.  
© Таврійський державний  
агротехнологічний університет  
імені Дмитра Моторного, 2024

## ЗМІСТ

ВСТУП .....	6
ЗМІСТОВИЙ МОДУЛЬ №1. ОСНОВИ РОБОТИ З МОВОЮ ПРОГРАМУВАННЯ PYTHON .....	7
ЛАБОРАТОРНА РОБОТА №1. Вступ в програмування мовою Python .....	8
1.1 Виконання роботи .....	8
1.2 Теоретичні відомості .....	8
1.3 Самостійна робота .....	16
1.4 Контрольні питання .....	22
1.5 Зміст звіту .....	22
1.6 Тестові завдання .....	23
ЛАБОРАТОРНА РОБОТА №2. Колекції у мові Python .....	25
2.1 Виконання роботи .....	25
2.2 Теоретичні відомості .....	25
2.3 Самостійна робота .....	40
2.4 Контрольні питання .....	45
2.5 Зміст звіту .....	45
2.6 Тестові завдання .....	46
ЛАБОРАТОРНА РОБОТА №3. Колекції у мові Python .....	48
3.1 Виконання роботи .....	48
3.2 Теоретичні відомості .....	48
3.3 Самостійна робота .....	52
3.4 Контрольні питання .....	53
3.5 Зміст звіту .....	53
3.6 Тестові завдання .....	53
ЛАБОРАТОРНА РОБОТА №4. Робота з файлами у Python .....	56
4.1 Виконання роботи .....	56
4.2 Теоретичні відомості .....	56
4.3 Самостійна робота .....	60
4.4 Контрольні питання .....	62
4.5 Зміст звіту .....	63
4.6 Тестові завдання .....	63
ЛАБОРАТОРНА РОБОТА №5. Класи і об'єкти у мові Python .....	65
5.1 Виконання роботи .....	65
5.2 Теоретичні відомості .....	65
5.3 Самостійна робота .....	72
5.4 Контрольні питання .....	73

5.5 Зміст звіту .....	73
5.6 Тестові завдання .....	74
<b>ЗМІСТОВИЙ МОДУЛЬ №2. РОБОТА З ДАНИМИ У МОВІ PYTHON</b> .....	<b>76</b>
<b>ЛАБОРАТОРНА РОБОТА №6. Регулярні вирази. Функції. Лямбди у Python</b> .....	<b>77</b>
6.1 Виконання роботи .....	77
6.2 Теоретичні відомості .....	77
6.3 Самостійна робота .....	86
6.4 Контрольні питання .....	86
6.5 Зміст звіту .....	87
6.6 Тестові завдання .....	87
<b>ЛАБОРАТОРНА РОБОТА №7. Побудова графіків у Python</b> .....	<b>89</b>
7.1 Виконання роботи .....	89
7.2 Теоретичні відомості .....	89
7.3 Самостійна робота .....	98
7.4 Контрольні питання .....	100
7.5 Зміст звіту .....	100
7.6 Тестові завдання .....	100
<b>ЛАБОРАТОРНА РОБОТА №8. Розроблення програмного забезпечення з графічним інтерфейсом мовою Python</b> .....	<b>103</b>
8.1 Виконання роботи .....	103
8.2 Теоретичні відомості .....	103
8.3 Самостійна робота .....	109
8.4 Контрольні питання .....	110
8.5 Зміст звіту .....	110
8.6 Тестові завдання .....	110
<b>ЛАБОРАТОРНА РОБОТА №9. Python. Робота з базами даних. MySQL</b> .....	<b>112</b>
9.1 Виконання роботи .....	112
9.2 Теоретичні відомості .....	119
9.3 Самостійна робота .....	118
9.4 Контрольні питання .....	118
9.5 Зміст звіту .....	118
9.6 Тестові завдання .....	119
<b>ЛАБОРАТОРНА РОБОТА №10. Регулярні вирази. Робота з форматом XML</b> .....	<b>121</b>
10.1 Виконання роботи .....	121
10.2 Теоретичні відомості .....	121
10.3 Самостійна робота .....	125

10.4 Контрольні питання .....	125
10.5 Зміст звіту .....	126
10.6 Тестові завдання .....	126
ЛАБОРАТОРНА РОБОТА №11. Розробка вебсервісів SOAP з використанням Python та бібліотеки Srupe .....	129
11.1 Виконання роботи .....	129
11.2 Теоретичні відомості .....	129
11.3 Самостійна робота .....	131
11.4 Контрольні питання .....	131
11.5 Зміст звіту .....	132
11.6 Тестові завдання .....	132
ЛАБОРАТОРНА РОБОТА №12. Процеси і потоки на Python. Динамічне програмування.....	134
12.1 Виконання роботи .....	134
12.2 Теоретичні відомості .....	134
12.3 Самостійна робота .....	140
12.4 Контрольні питання .....	141
12.5 Зміст звіту .....	142
12.6 Тестові завдання .....	142
КРИТЕРІЇ ОЦІНЮВАННЯ ПРАКТИЧНИХ РОБІТ .....	144
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	145

## ВСТУП

Програмування стало одним із шляхів кар'єрного зростання в ІТ-сфері, а професія програміста на сьогодні є доволі високооплачуваною та престижною. Зараз актуальними є мови програмування для створення web-додатків, функціонування серверної частини, аналізу великих даних тощо.

Python є багатоцільовою мовою програмування, яка дозволяє створювати програмний код, що добре читається. Відносний лаконізм мови Python дозволяє створити програму, яка буде набагато коротше свого аналога, написаного на іншій мові. Програми на Python можна запускати в різних операційних системах без будь-яких змін. Ще однією перевагою Python є його стандартна бібліотека, яка встановлюється разом з Python і містить готові інструменти для роботи з операційною системою, веб-сторінками, базами даних, різними форматами даних, для побудови графічного інтерфейсу програм тощо. Завдяки поєднанню простоти синтаксису та потужних можливостей, Python є підходящим вибором для новачка-програміста та досвідченого професіонала.

Метою навчальної дисципліни «Прикладне програмне забезпечення (Python)» є ознайомлення студентів з основними поняттями програмування мовою високого рівня Python. Тематика цього курсу - основні поняття та структури, що використовуються у програмуванні (змінні, типи даних, вирази, процедури введення/виведення, підпрограми, функції), а також підготовка програм із застосуванням згаданих елементів разом з розробкою алгоритмів для вирішення проблем і аналіз їхньої коректності.

Завданням дисципліни «Прикладне програмне забезпечення (Python)» є формування у здобувачів вищої освіти базової підготовки в області програмування; опанування будови та принципів роботи Python скриптів; освоєння методів створення, розміщення та завантаження програм мовою Python.

Викладений матеріал у лабораторному практикумі структурований відповідно до робочої програми дисципліни «Прикладне програмне забезпечення (Python)», що викладається здобувачам вищої освіти першого (бакалавського) рівня спеціальності 122 «Комп'ютерні науки». Структура лабораторного практикуму складається зі вступу, завдань до практичних робіт, переліку використаних джерел. Кожна практична робота містить тему та мету роботи, пояснення до виконання, теоретичні відомості до теми практичної роботи, практичну частину, інструкції до виконання завдань, поданих на самостійне опрацювання, контрольні питання та тестові завдання.

## **Змістовий модуль №1**

**Тема: ОСНОВИ РОБОТИ З МОВОЮ ПРОГРАМУВАННЯ  
PYTHON**

## ЛАБОРАТОРНА РОБОТА №1

**Тема:** Вступ в програмування мовою Python

**Мета:** Вивчити базові програмування мовою Python. Придбання навиків програмування мовою Python

**Час:** 2 години

**Література:** 1, с. 39-73; 3, с. 22-69; 4, с. 10-27; 14, с. 41-64; 22, с. 17-20.

### 1.1 Виконання роботи

- Вивчити теоретичні відомості (п. 1.2).
- Виконати самостійну роботу (п. 1.3).
- Відповісти на контрольні питання (п. 1.4).
- Оформити звіт з виконання практичної роботи (п. 1.5).
- Опрацювати тестові завдання (п. 1.6)

### 1.2 Теоретичні відомості

Будь-яка Python-програма складається з послідовності допустимих символів, записаних в певному порядку і за певними правилами. Програма включає в себе:

- коментарі;
- символи;
- знаки пунктуації;
- ідентифікатори;
- ключові слова.

#### **Коментарі (Не обов'язково)**

Коментарі призначені для пояснення коду для розробника і починаються з символу #, діючи до кінця рядка.

Python коментарі пропускає

#### **Знаки пунктуації**

Наприклад: знак коми "," – для поділу параметрів функцій, знак двокрапка ":" – використовується в циклах.

#### **Ідентифікатори (Ім'я)**

Ідентифікатором називають ім'я існуючого об'єкта в програмі, яке є його унікальним ознакою, що дозволяє відрізнити його від інших.

Тобто іншими словами, значення слова "ідентифікатор" це те ж саме що і значення слова "ім'я".

#### **Дані**

Існують постійні та змінні дані.

Приклад.

```
def func():
for i in range(5):
print(f"from child thread: {i}")
sleep(0.5)
th = Thread(target=func)
th.start()
print("App stop")
```

Висновок програми:

```
from child thread: 0
App stop
from child thread: 1
from child thread: 2
from child thread: 3
from child thread: 4
```

Як можна побачити, програма продовжує працювати, навіть після того, як головний потік завершився (повідомлення: *"App stop"*).

Для того, щоб потоки не заважали зупинці програми (тобто, щоб вони зупинялися разом з завершенням роботи програми), необхідно при створенні об'єкту *Thread* аргументу *daemon* привласнити значення *True*, або після створення потоку, перед його запуском привласнити властивості *daemon* значення *True*.

Змінимо процес створення потоку у приведеній вище програмі:

```
th = Thread(target=func, daemon=True)
```

Запустимо її, отримаємо наступний результат:

```
from child thread: 0
```

```
App stop
```

Потік зупинився разом з зупинкою програми.

### 12.3 Самостійна робота

За круглим столом розставлено  $N$  стільців, на кожному з яких сидить філософ. У центрі столу розміщено страву з макаронами. На столі лежать  $N$  виделок, кожна з яких знаходиться між двома сусідніми тарілками. Кожен філософ може бути у двох станах: розмірковувати чи є макарони. Для того, щоб почати їсти, філософу необхідні дві виделки: одна у правій руці, а друга у лівій. Закінчивши їжу, філософ кладе виделки на місце і починає розмірковувати доти, доки знову не зголодніє.

У цьому є дві небезпечні ситуації: <змова сусідів> і <голодна смерть>.

<Змова сусідів> має місце, коли сусіди ліворуч і праворуч від філософа будують підступи. Змовники по черзі забирають вилки то ліворуч, то праворуч від «жертви». Такі узгоджені дії зловмисників призводять до вимушеного голодування, оскільки він ніколи не може скористатися обома виделками.

<Голодна смерть> виникає, коли філософи одночасно зголодніють і одночасно спробують взяти, наприклад, свою ліву вилку. При цьому виникає тупикова ситуація, тому що ніхто з них не може почати їсти, не маючи другої вилки.

Поведінка кожного філософа має моделюватися окремим процесом чи потоком. При використанні процесів взаємодія між процесами - філософами може здійснюватися через додатковий процес - обідній стіл, який знає інформацію про наявність виделок на столі. У кожному процесі-філософі має бути реалізовано кілька алгоритмів взяття вилок, причому номер алгоритму повинен вказуватись при запуску процесу-філософа:

1. Філософ спочатку намагається взяти ліву вилку, а потім, як тільки ліва вилка виявилася у нього, намагається взяти праву. При цьому він продовжує утримувати ліву вилку, якщо права недоступна. Якщо всі філософи діють за таким алгоритмом, може виникнути ситуація <голодна смерть>. Тобто процеси, що моделюють філософів опиняться в тупиковій ситуації (зависнуть)

2. Філософ вибирає першу вилку випадковим чином, в іншому - все те ж саме, що й у першому алгоритмі, в т.ч. можливе виникнення тупикової ситуації, але з меншою ймовірністю, ніж у першому випадку.

3. У програмі потрібно запропонувати та реалізувати такий алгоритм поведінки філософів, який гарантовано не призведе до ситуацій <змова сусідів> та <голодна смерть>.

## 12.4 Контрольні питання

1. Використання методу join().
2. Скільки конструкторів похідних типів реалізовано у MPI? Перерахуйте та охарактеризуйте їх.
3. Чим відрізняються топології «тор» та «решітка»?
4. Що таке вид файлу? Навіщо використовується це поняття?
5. Які методи синхронізації використовуються при віддаленому доступі до пам'яті?
6. Що таке інтеркомунікатор? Чим це поняття відрізняється від поняття інтракомунікатора?
7. Охарактеризувати поняття «Паралельність» та «Конкурентність».
8. В чому різниця між синхронним та асинхронним виконанням програми?

9. Способи створення потоків
10. Особливості використання потоків-демонів?

### 12.5 Зміст звіту

Звіт з виконання практичної роботи повинен складатися з наступних складових:

1. Прізвище, ім'я, по-батькові здобувача вищої освіти, номер академічної групи.
2. Номер практичної роботи.
3. Тема та мета практичної роботи.
4. Завдання до практичної роботи.
5. Текст розробленого програмного забезпечення.
6. Результати тестування: вхідні дані, скрипти та результати роботи програми.
7. Інтерфейс роботи з програмою в декількох режимах.

### 12.6 Тестові завдання

**Запитання 1.** Штучна мова, створена для передачі команд машинам, зокрема комп'ютерам – це ... ?

- а. мова програмування
- б. транслятор
- в. інтерпретатор
- г. компілятор

**Запитання 2.** Набір символів, з яких утворюють слова та речення мови програмування – це ... ?

- а. алфавіт
- б. семантика
- в. синтаксис
- г. словник

**Запитання 3.** Сукупність спеціальних слів, які мають однозначне пояснення та застосування в мові програмування – це ... ?

- а. словник
- б. алфавіт
- в. семантика
- г. синтаксис

**Запитання 4.** Система правил складання базових конструкцій мови програмування – це ... ?

- а. словник
- б. алфавіт

- в. семантика
- г. синтаксис

**Запитання 5.** За технологією програмування розрізняють:

- а. процедурні мови
- б. об'єктно-орієнтовні
- в. декларативні
- г. мова скриптів

**Запитання 6.** Комп'ютерна програма (або набір к. програм), що перетворює вихідний код, написаний певною мовою програмування, на семантично еквівалентний код в іншій мові програмування, який, як правило, необхідний для виконання програми машиною, наприклад, комп'ютером – це ... ?

- а. компілятор
- б. інтерпретатор
- в. транслятор
- г. середовище програмування

**Запитання 7.** Перетворюють частину програми в машинний код, виконують і після цього переходять до наступної частини це...

- а. інтерпретатор
- б. компілятор
- в. транслятор
- г. середовище програмування

**Запитання 8.** Який рядок присвоює змінній D значення 58?

- а. d=58
- б. D=58
- в. "d"=58
- г. D:58

**Запитання 9.** Як потрібно вводити назву змінної?

- а. Назва змінної може складатися лише з малих (a..z) та великих (A..Z) літер англійського алфавіту
- б. Імена можуть починатися з цифр
- в. Ім'я змінної залежить від регістра
- г. Можна використовувати "-"

**Запитання 10.** Які є типи даних в Python?

- а. integer
- б. float
- в. point
- г. string

## КРИТЕРІЇ ОЦІНЮВАННЯ ПРАКТИЧНИХ РОБІТ

Виконання практичних робіт оцінюється за результатами захисту відповідних звітів. Звіт з практичної роботи має бути представлений викладачу до початку виконання наступної практичної роботи. Звіт має містити наступні складові:

- Прізвище, ім'я, по-батькові здобувача вищої освіти, номер академічної групи.
- Номер практичної роботи.
- Тему та мету практичної роботи.
- Короткий конспект основних теоретичних відомостей.
- Результати виконання самостійної роботи.

Результати виконання практичних робіт оцінюються за такою шкалою (у відсотках до максимального балу):

90-100% «відмінно»: виконано всі завдання практичного заняття, здобувач освіти чітко та без помилок відповів на всі контрольні питання.

75-89% «добре»: виконано всі завдання практичного заняття; здобувач освіти відповів на всі контрольні питання із незначними зауваженнями.

60-74% «задовільно»: виконані майже всі завдання практичного заняття з зауваженнями; здобувач освіти відповів на всі контрольні питання із зауваженнями.

35-59% «незадовільно»: здобувач освіти не виконав чи виконав неправильно запропоновані завдання практичної роботи, відповів на контрольні питання з помилками чи не відповів на контрольні питання.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Васильєв О. М. Програмування мовою Python. Тернопіль : Навчальна книга – Богдан, 2019. 504 с.
2. Дегтярєва Н. та ін. Навчання програмуванню мовою python учнів старших класів при роботі з графічними об'єктами. Наука і техніка сьогодні. 2023. № 3(17). С. 316-326.
3. Костюченко А. О. Основи програмування мовою Python : навч. посіб. Чернігів : ФОП Баликіна С. М., 2020. 180 с.
4. Крєневич А. П. Python у прикладах і задачах. Частина 1. Структурне програмування : навч. посіб. Київ : ВПЦ «Київський Університет», 2017. 206 с.
5. Крєневич А. П. Python у прикладах і задачах. Частина 2. Об'єктно-орієнтоване програмування. Навчальний посібник. К.: ВПЦ «Київський Університет», 2020. – 152 с.
6. Об'єктно-орієнтоване програмування мовою PYTHON : Конспект лекцій [Електронний ресурс] : навч. посіб. для студ. спеціальності 153 «Мікро- та наносистемна техніка» освітньої програми «Мікро- та наноелектроніка» / КПІ ім. Ігоря Сікорського ; уклад.: Д. Д. Татарчук, Ю. В. Діденко. Київ : КПІ ім. Ігоря Сікорського, 2021. 129 с.
7. Основи програмування алгоритмічною мовою Python: навч. посіб. для студ. освітньої програми «Комп'ютерні системи та мережі» спеціальності 123 «Комп'ютерна інженерія» / М.А. Новотарський. Київ : КПІ ім. Ігоря Сікорського, 2022. 701с.
8. Петренко С., Дегтярєва Н. Мова програмування Python: Основи програмування. Навчальний посібник для студентів спеціальності 014 Середня освіта (Інформатика) Суми: СумДПУ імені А.С.Макаренка, 2023. 101 с.
9. Розробка примітивного месенджера на Python за допомогою бібліотеки socket. URL: [https://drukarnia.com.ua/articles/rozrobka-primitivnogo-mesendzhera-na-python-za-dopomogoyu-biblioteki-socket-of5\\_P](https://drukarnia.com.ua/articles/rozrobka-primitivnogo-mesendzhera-na-python-za-dopomogoyu-biblioteki-socket-of5_P).
10. Сіциліцин Ю. О. Вибір онлайн засобів для розробки паралельних програм з використанням бібліотеки MPI. Сучасні комп'ютерні та інформаційні системи і технології: матеріали III Всеукраїнської наук.-практ. інтернет-конф. (Запоріжжя, 12-19 грудня 2022 р.) Запоріжжя: ТДАТУ, 2022. С. 223–225.
11. Толстікова О. В., Мирошніченко І. С., Коцюр А. Б. Ефективність використання програмування асинхронних додатків мовою Python. Проблеми інформатизації та управління. Т. 1. №. 53. С. 72–77.
12. Шаров С.В. Лубко Д.В. Використання рекурсії при вивченні мови логічного програмування Turbo Prolog. Наукові записки. Серія:

- Проблеми методики фізико-математичної і технологічної освіти. 2015.Т. 3. №8. С. 68–73.
13. Шаров С. В., Осадчий В. В. Бази даних та інформаційні системи. Навчальний посібник. Мелітополь : Вид-во МДПУ ім. Б. Хмельницького, 2014. 352 с.
  14. Яковенко А. В. Основи програмування. Python. Частина 1 : підручник. Київ : КПІ ім. Ігоря Сікорського, 2018. 195 с.
  15. Browning J.B., Alchin M.: Pro Python 3: Features and Tools for Professional Development, 3 rd ed. Apress, 2019. 468 p.
  16. Downey A.B.: Think Python: How to Think Like a Computer Scientist, 2nd ed. O'Reilly Media, Inc., 2015. 244 p.
  17. HOWTO з програмування сокетів. URL: <https://docs.python.org/uk/3/howto/sockets.html>.
  18. Kalb I.: Learn to Program with Python 3: A Step-by-Step Guide to Programming, 2nd ed. Apress, 2018. 382 p.
  19. Matplotlib.pyplot. URL: [https://matplotlib.org/3.5.3/api/\\_as\\_gen/matplotlib.pyplot.html](https://matplotlib.org/3.5.3/api/_as_gen/matplotlib.pyplot.html).
  20. Melnyk V., Nahorniuk A., Bortnik K. Дослідження швидкодії виконання запиту в СУБД MySQL та MariaDB засобами Python та PHP. Computer-integrated technologies: education, science, production. 2019. № 36. С. 123–127.
  21. Phillips D.: Python 3 Object-oriented Programming: Unleash the power of Python 3 objects, 2 nd ed. Packt Publishing, 2015. 460 p.
  22. Sharov S., Tereshchuk S., Tereshchuk A., Kolmakova V., Yankova N.. Using MOOC to Learn the Python Programming Language. International Journal of Emerging Technologies in Learning (iJET). 2023. № 18(2). P. 17–23.
  23. Spyne Documentation. URL: <https://spyne.io/docs/2.11>.
  24. SOAP. URL: <https://en.wikipedia.org/wiki/SOAP>.
  25. Web Services Description Language. URL: [https://en.wikipedia.org/wiki/Web\\_Services\\_Description\\_Language](https://en.wikipedia.org/wiki/Web_Services_Description_Language).

Навчально-методичне видання

**Сіциліцин Юрій Олександрович  
Шаров Сергій Володимирович**

# **Прикладне програмне забезпечення (Python)**

Лабораторний практикум