

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ПРИКАРПАТСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАСИЛЯ СТЕФАНІКА



Мазуренко В.В., Дмитришин М.І., Василичин П.Б.

**СТРУКТУРНО-ПРОЦЕДУРНЕ
ПРОГРАМУВАННЯ з PYTHON
ЛАБОРАТОРНИЙ ПРАКТИКУМ**

УДК 004.424
ББК 32.973-018.1
М13

Рекомендовано Вченою радою факультету математики та інформатики Прикарпатського національного університету імені Василя Стефаника (протокол № 10 від 21.11.2023 р.).

Рецензенти:

Олександр МАХНЕЙ, кандидат фізико-математичних наук, доцент (Прикарпатський національний університет імені Василя Стефаника, Івано-Франківськ);

Роман ДМИТРИШИН, доктор фізико-математичних наук, професор (Прикарпатський національний університет імені Василя Стефаника, Івано-Франківськ).

М13 Мазуренко В.В., Дмитришин М.І., Васишин П.Б. Структурно–процедурне програмування з Python: Лабораторний практикум. – Ів.-Фр.: Голіней, 2023. – 76 с.

У лабораторному практикумі наведено індивідуальні завдання і методичні рекомендації до виконання лабораторних робіт з основ програмування мовою Python.

Для студентів спеціальності «прикладна математика». Практикум буде корисним також для студентів галузей знань «математика і статистика» та «інформаційні технології».

ЗМІСТ

ПЕРЕДМОВА	4
Лабораторна робота 1. ЛІНІЙНІ ПРОГРАМИ	5
Лабораторна робота 2. РОЗГАЛУЖЕНІ ПРОГРАМИ	10
Лабораторна робота 3. ЦИКЛІЧНІ ПРОГРАМИ	16
Лабораторна робота 4. СПИСКИ І КОРТЕЖІ	27
Лабораторна робота 5. СИМВОЛИ І РЯДКИ	38
Лабораторна робота 6. СЛОВНИКИ І МНОЖИНИ	46
Лабораторна робота 7. ОБРОБКА ВИНЯТКІВ	58
Лабораторна робота 8. СТВОРЕННЯ ФУНКЦІЙ	66
Лабораторна робота 9. ТЕКСТОВІ І БІНАРНІ ФАЙЛИ	72
СПИСОК ЛІТЕРАТУРИ	76

ПЕРЕДМОВА

Програмування як процес написання (coding), відлагодження (debugging) і тестування (testing) комп'ютерних програм для автоматизації (на основі запропонованих алгоритмів) завдань у різних предметних областях, допускає різні підходи до його реалізації. Тривалий час домінуючим підходом у програмуванні була парадигма *структурно-процедурного програмування*, відповідно до якої комп'ютерна програма має вигляд ієрархічної структури блоків з потоком керування (control flow), що реалізується за допомогою таких керуючих конструкцій, як послідовне виконання, галуження, цикл, а також виклик підпрограм та/або функцій. Структуровані програми легше створювати, відстежувати логіку їх виконання і відлагоджувати; скорочується термін їх розробки; є можливість повторного використання одного і того ж коду з різних місць програми; зазвичай, є швидшими через низький рівень абстракції.

Лабораторний практикум зі структурно-процедурного програмування є важливим етапом у навчанні здобувачів освіти основам програмування та формуванні у них алгоритмічного мислення. Він зорієнтований на використання мови програмування Python. Це високорівнева, інтерпретована мова програмування, яка відзначається простотою синтаксису і читабельністю коду. Її універсальність у поєднанні з численними бібліотеками робить Python ідеальним вибором для широкого спектру завдань — від автоматизації задач і наукових обчислень, аж до інтелектуального аналізу даних і веб-розробки. Водночас варто зауважити, що завдання лабораторного практикуму є інваріантними стосовно мови програмування і за потреби можуть бути використані для вивчення інших мов.

Практикум охоплює широкий спектр тем, починаючи від лінійних програм та закінчуючи роботою з файлами. Особлива увага приділяється керуючим конструкціям, що дозволяє здобувачам освіти освоїти логіку виконання програми через галуження і цикли. Практичне використання списків, кортежів, рядків, словників і множин розширює їхні можливості у роботі з різними типами і структурами даних. Окрім того, важливий аспект практикуму — це вивчення обробки виняткових ситуацій, що забезпечує студентам розуміння важливості відлагодження програм. Навики створення функцій та використання файлів і модулів сприяють розробці більш структурованих, читабельних і гнучких у плані організації коду програм. Для кожної лабораторної роботи вказана тема, короткий перелік необхідних для виконання роботи теоретичних знань, 15 варіантів індивідуальних або групових (для командної роботи) завдань та вимоги до оформлення звіту.

I. МЕТА ЛАБОРАТОРНОЇ РОБОТИ

формування навичок роботи з файлами мовою Python

II. ДЛЯ ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ НЕОБХІДНО ЗНАТИ

- що таке файл і його ознаки
- три основні етапи роботи з файлами: відкриття, читання/запис, закриття
- інструкції для відкриття і закриття файлу та режими роботи з файлом
- основні операції з файлами
- особливості роботи з текстовими і бінарними файлами
- основні операції з файловою системою

РЕКОМЕНДОВАНА ЛІТЕРАТУРА: [2, п. 7.1-4], [3, § 9], [4, п. 8.1-5], [5]

ДОДАТКОВА ЛІТЕРАТУРА: [6–9]

III. ЗАВДАННЯ НА ЛАБОРАТОРНУ РОБОТУ

Працюючи у команді (номер варіанту виконує трійка студентів відповідно до таблиці нижче, перший зі студентів є TeamLeader, що розподіляє і контролює роботу, захист роботи від команди), написати програми з обробкою потенційних виняткових ситуацій для виконання наступних завдань.

Таблиця 1 Варіанти завдань

Варіант	Номер студента		
1	1	6	11
2	7	12	2
3	13	3	8
4	4	9	14
5	10	15	5

Варіант 1.

1. У вхідному файлі `data1.txt` міститься послідовність цілих чисел, які розділені пробілами. Дозаписати у цей файл такі рядки:
 - (a) від'ємні числа з послідовності;
 - (b) кількість нулів;
 - (c) додатні числа з послідовності;
 - (d) шлях до каталогу, у якому міститься файл, час створення і розмір файлу.

2. У вхідному файлі `data2.txt` містяться координати точок на площині. Дозаписати у цей файл рядки з інформацією про найвіддаленішу і найближчу пару точок серед заданих.

```
# Приклад файлу data2.txt
(-5, 2)
(1, -4)
(0, 0)
(-3, -7)
```

3. Задано словник студентів групи, у якому ключем є порядковий номер студента, а значенням список мов, які знає студент. Виконати такі завдання:

- (а) серіалізувати цей словник у бінарний файл `binary.data`;
- (б) десеріалізувати дані з файлу і вивести на екран мову(-и), яку(-і) знають усі студенти групи.

Варіант 2.

1. У вхідному файлі `data1.txt` міститься послідовність цілих чисел, які розділені пропусками. Дозаписати у цей файл такі рядки:

- (а) числа, що є менші за їх середнє арифметичне;
- (б) кількість чисел, що дорівнюють середньому арифметичному;
- (с) числа, що є більші за середнє арифметичне;
- (д) шлях до каталогу, у якому міститься файл, час створення і розмір файлу.

2. У вхідному файлі `data2.txt` містяться результати кожної з 4-х чвертей баскетбольних матчів. Дозаписати у цей файл рядки з інформацією про загальний рахунок матчу і різницю очок, набраних командами.

```
# Приклад файлу data2.txt
26:17 13:15 19:11 14:16
15:16 18:17 10:12 14:12
20:21 24:22 18:15 16:25
```

3. Задано словник з N англomовних слів з можливими повторами слів. Виконати такі завдання:

- (а) серіалізувати цей словник у бінарний файл `input.data`;
- (б) десеріалізувати дані з файлу і вивести на екран словник без повторів слів, впорядкованих лексикографічно, і кількість слів у ньому.

Варіант 3.

1. У вхідному файлі `data1.txt` міститься послідовність цілих чисел, які розділені пропусками. Дозаписати у цей файл такі рядки:

- (а) кількість членів послідовності, модулі яких є простими числами : вивести такі члени послідовності;
- (б) кількість членів послідовності, модулі яких є складеними числами : вивести такі члени послідовності;
- (с) кількість членів послідовності, модулі яких не є ні простими, ні складеними числами : вивести такі члени послідовності.
- (д) шлях до каталогу, у якому міститься файл, час створення і розмір файлу.

2. У вхідному файлі `data2.txt` міститься перелік наявних у шкільній бібліотеці книг. Дозаписати у цей файл рядки з інформацією про книги в одному примірнику та кількість таких книг у списку.

```
# Приклад файлу data2.txt
```

Тарас Шевченко : “Кобзар” — 1 прим.

Юрій Винничук : “Легенди Львова” — 3 прим.

Ліна Костенко : “Гіацинтове сонце” — 2 прим.

3. Задано словник міст, у якому ключем є країна, а значенням список окремих міст цієї країни. Виконати такі завдання:
 - (a) серіалізувати цей словник у бінарний файл `binary.data`;
 - (b) десеріалізувати дані з файлу і для введеного з клавіатури міста, вивести на екран країну, де воно знаходиться.

Варіант 4.

1. У вхідному файлі `data1.txt` міститься непорожня послідовність цифр (0 – 11) дванадцяткової системи числення, які розділені пропусками. Дозаписати у цей файл такі рядки:
 - (a) вивести a разів цифру a , де a — перша цифра з послідовності;
 - (b) вивести b разів цифру b , де b — друга цифра з послідовності;
 - (c) і т.д. аж до останньої цифри з послідовності;
 - (d) шлях до каталогу, у якому міститься файл, час створення і розмір файлу.
2. У вхідному файлі `data2.txt` міститься перелік телефонних розмов особи з абонентами різних операторів. Дозаписати у цей файл рядки з інформацією про розмови з абонентами від вибраного оператора (введеного з клавіатури) і середню тривалість розмов.
Приклад файлу `data2.txt`
1.12.2021 — 58 с (Vodafone)
1.12.2021 — 81 с (Kyivstar)
1.12.2018 — 36 с (Lifecell)
3. Задано словник студентів групи, у якому ключем є порядковий номер студента, а значенням список мов, які знає студент. Виконати такі завдання:
 - (a) серіалізувати цей словник у бінарний файл `binary.data`;
 - (b) десеріалізувати дані з файлу і вивести на екран усі мови, які знають студенти групи.

Варіант 5.

1. У вхідному файлі `data1.txt` міститься непорожня послідовність цілих чисел, які розділені пропусками. Дозаписати у цей файл такі рядки:
 - (a) кількість чисел з послідовності, які діляться націло на 3 : вивести такі числа;
 - (b) кількість чисел з послідовності, які діляться на 3 з остачею 1 : вивести такі числа;
 - (c) кількість чисел з послідовності, які діляться на 3 з остачею 2 : вивести такі числа;
 - (d) шлях до каталогу, у якому міститься файл, час створення і розмір файлу.
2. У вхідному файлі `data2.txt` міститься графік руху потягів між двома станціями. Дозаписати у цей файл рядки з впорядкованим за часом відправлення графіком руху потягів та доданою інформацією для кожного потягу про його час у дорозі.
Приклад файлу `data2.txt`
1 11:26 17:22
2 19:04 01:36
3 16:12 23:08
3. Задано словник з N англійських слів без повторів слів. Виконати такі завдання:
 - (a) серіалізувати цей словник у бінарний файл `binary.data`;
 - (b) десеріалізувати дані з файлу і вивести на екран кількість слів у словнику, що починаються введеною буквою, і самі слова.

IV. БОНУСНЕ ЗАВДАННЯ

У кожній кредитній картці є надрукований на ній номер, а також номер, вбудований у магнітну стрічку зі зворотного боку картки. Останній зберігається також у базі даних для того, щоб коли карткою розраховуються за покупку, банк знав, з якого рахунку списувати гроші. У світі багато людей з кредитними картками, тому ці номери доволі великі: American Express використовує номери з 15 розрядів, MasterCard — з 16, Visa — з 13 або 16. Ці розряди — десяткові (від 0 до 9). Номери кредитних карток мають певну структуру. Номери карток American Express починаються з 34 чи 37; номери карток MasterCard починаються з 51, 52, 53, 54 чи 55; номери карток Visa починаються з 4. Також номери карток мають вбудовану у них контрольну суму, що обчислюється за допомогою математичної формули, використовуючи алгоритм Ганса Луна з ІВМ. Відповідно до цього алгоритму, можна перевірити, чи номер кредитної картки є синтаксично коректним, наступним чином: рядки довжиною 1 або менше не є дійсними; пропуски дозволені для введення, але їх необхідно видалити перед перевіркою; усі інші нецифрові символи заборонені.

Приклад 1: дійсний номер кредитної картки 4539148803436467

Перший крок алгоритму Луна — це подвоєння кожної другої цифри, починаючи справа:
4_3_1_8_0_4_6_6_

Якщо подвоєння числа є числом, що перевищує 9, необхідно відняти 9 від добутку.

Результати подвоєння розставлені на свої місця: 8569 2478 0383 3437

Далі необхідно підсумувати всі цифри: $8+5+6+9+2+4+7+8+0+3+8+3+3+4+3+7 = 80$

Якщо сума націло ділиться на 10, то номер картки дійсний. Цей номер дійсний!

Приклад 2: недійсний номер кредитної картки 8273123273520569

Подвоєння кожної другої цифри, починаючи справа: 7253226253120539

Сума цифр: $7 + 2 + 5 + 3 + 2 + 2 + 6 + 2 + 5 + 3 + 1 + 2 + 0 + 5 + 3 + 9 = 57$

57 націло не ділиться на 10, тому цей номер картки недійсний!

Кожен рядок вхідного файлу numbers.txt містить номер кредитної картки. Напишіть функцію(i) для валідації номерів кредитних карток за алгоритмом Луна та виведення приналежності картки до трьох компаній (American Express, MasterCard, Visa), або повідомлення INVALID — в іншому разі.

ТЕСТ ВХІД:

4003600000000014

5105105105105100

378282246310005

4111111111111113

ВИХІД:

Visa

MasterCard

American Express

INVALID

V. ЗВІТ ПО ЛАБОРАТОРНІЙ РОБОТІ

Зміст звіту:

1. Титульний аркуш зі стандартними атрибутами
2. Коди програм з коментарями
3. Тестування програм
4. Висновки

СПИСОК ЛІТЕРАТУРИ

- [1] Васильєв О.М. Програмування мовою Python. – Тернопіль: Навчальна книга — Богдан, 2019.
- [2] Висоцька В. А., Оборська О. В. Python: алгоритмізація та програмування: навчальний посібник — Львів: Новий Світ — 2000, 2021.
- [3] Крєневич А. П. Python у прикладах і задачах. Частина 1. Структурне програмування: Навчальний посібник. – К.: ВПЦ "Київський університет", 2017.
- [4] Мізюк О. Путівник мовою програмування Python. – Ел. вид., 2020. – Режим доступу: <https://pythonguide.rozh2sch.org.ua>
- [5] Селіверстов Р., Мельничин А. Основи програмування мовою Python: навч. посібник. — Львів: ЛНУ імені Івана Франка, 2020.
- [6] Severance Ch. R. Python for Everybody: Exploring Data Using Python 3, 2023. – Access mode: https://do1.dr-chuck.com/pythonlearn/EN_us/pythonlearn.pdf
- [7] Swaroop С. Н. A byte of Python, 2005. – Access mode: <https://python.swaroopch.com>
- [8] PEP 8 — Style Guide for Python Code. – Access mode: <https://peps.python.org/pep-0008/#source-file-encoding>
- [9] The Python Tutorial. – Access mode: <https://docs.python.org/3/tutorial/index.html>