

Міністерство освіти і науки України
Чернівецький національний університет
імені Юрія Федьковича

ПРОГРАМУВАННЯ

**Методичні рекомендації та
завдання для лабораторних робіт**

Частина 2

Чернівці
Чернівецький національний університет
2024

УДК 044.438(076.5)
П-784

Рекомендовано до друку Вченою радою факультету математики та інформатики
Чернівецького національного університету імені Юрія Федьковича
(протокол № 10 від « 24 » квітня 2024 року)

Укладачі:

Фратавчан Тоня Михайлівна, кандидат фізико-математичних наук,
доцент кафедри математичного моделювання;

Івасюк Галина Петрівна, кандидат фізико-математичних наук, доцент
кафедри математичного моделювання;

Фратавчан Валерій Григорович, кандидат фізико-математичних наук,
доцент кафедри МПУіК.

П-784 Програмування: методичні рекомендації та завдання для
лабораторних робіт. Частина 2. Укл.: Т.М.Фратавчан, Г.П.Івасюк,
В.Г.Фратавчан – Чернівці: Чернівецький національний університет,
2024. – 84 с.

Методичні рекомендації містять завдання для лабораторних робіт та
теоретичні відомості, необхідні для їх виконання, з дисципліни
«Програмування».

Для студентів, що здобувають освіту в галузі знань „Середня освіта
(інформатика)”, „Середня освіта (математика)”, „Математика” та інших
спеціальностей з подібною програмою вивчення дисципліни.

УДК 044.438(076.5)

Зміст

Вступ	4
Мета та програмні результати навчання	5
Теоретичний зміст програми навчальної дисципліни у другому семестрі ...	7
Лабораторна робота №1. Множини та обробка виключень у Python	8
Лабораторна робота №2. Функції у Python	17
Лабораторна робота №3. Робота з файлами у Python	29
Лабораторна робота №4. Рекурсивні функції. Робота з бібліотекою Numpy	37
Лабораторна робота №5. Робота з бібліотекою Matplotlib	46
Лабораторна робота №6. Робота з бібліотекою Tkinter	62
Лабораторна робота №7. Елементи ООП. Створення класів та об'єктів у Python	73
Рекомендована література	81

Вступ

Дисципліна «Програмування» призначена для вивчення сучасних методів опрацювання інформації, здобуття навиків алгоритмізації та програмування алгоритмічною мовою високого рівня Python та їх застосування у процесі навчання і майбутній професійній діяльності.

Мова програмування Python швидко розвивається. Цьому сприяє не тільки досить вдала концепція мови, а й згуртоване співтовариство, сформоване з розробників і популяризаторів мови. Необхідне програмне забезпечення, включаючи середовища розробки, в основному безкоштовне. Усе це дає підстави розглядати Python як одну з найперспективніших мов програмування.

На сьогодні Python використовується при реалізації найрізноманітніших проектів, серед яких:

- розробка сценаріїв для роботи з Web та Internet-програмами;
- мережеве програмування;
- засоби підтримки технологій HTML і XML;
- програми для роботи з електронною поштою й підтримки Internet-протоколів;
- програми для обслуговування найрізноманітніших баз даних;
- програми для наукових розрахунків;
- програми з графічним інтерфейсом;
- створення ігор і комп'ютерної графіки та багато іншого.

Методичні вказівки містять завдання до практичних та лабораторних робіт з програмування мовою Python, а також теоретичні відомості, необхідні для їх виконання, які дають змогу студентам самостійно засвоїти здобуті знання. Кожне з лабораторних завдань розраховано на 2 години аудиторної роботи та роботи у позааудиторний час.

Матеріали методичних вказівок можуть бути корисними студентам різних спеціальностей всіх форм навчання як для розв'язання типових задач, так і для самостійного вивчення теоретичного матеріалу, а також як допоміжний засіб при організації дистанційного навчання.

Мета та програмні результати навчання

Мета навчальної дисципліни: надання студентам базових знань з теорії програмування, формування у студентів здатностей:

- застосовувати професійні знання й уміння на практиці ;
- використовувати інформаційні і комунікаційні технології;
- адаптуватися до різних професійних ситуацій, проявляти творчий підхід, ініціативу;
- аналізувати проблеми, ставити постановку цілей і завдань, виконувати вибір способу й методів дослідження, а також оцінку його якості;
- вирішувати проблеми в професійній діяльності на основі аналізу й синтезу;
- застосовувати сучасні парадигми програмування під час програмної реалізації професійних задач ;
- працювати автономно;
- розробляти і досліджувати математичні моделі явищ, процесів та систем;
- застосовувати чисельні методи для дослідження математичних моделей ;
- використовувати програмні засоби загального та спеціального призначення для розв'язання прикладних задач з математики та інформатики;
- використовувати обчислювальні інструменти для чисельних і символьних розрахунків .

Після вивчення даної дисципліни студент повинен

знати:

- основні поняття та твердження з програмного матеріалу даного курсу;
- основні поняття інформатики – виконавець, алгоритм, програма;
- синтаксис, семантику та властивості структур керування (ланцюг, розгалуження, цикли);
- синтаксичні конструкції мови програмування Python; формальні методи побудови алгоритмів та програм за допомогою рекурентних співвідношень;
- основні властивості програм; будову простих та складених типів даних;
- опис та використання підпрограм;
- поняття про виключні ситуації;

- роботу з файлами;
- модульне програмування;
- спеціалізовані програмні засоби комп'ютерної та прикладної математики;

вміти:

– будувати лінійні, розгалужені та циклічні алгоритми з використанням підпрограм, модулів у мові програмування Python.

– аналізувати задачі шкільних курсів математики та інформатики різних рівнів

складності, демонструвати здатність їх розв'язувати ;

– застосовувати знання і розуміння для ідентифікації, формулювання і вирішення технічних задач спеціальності використовуючи відомі методи;

– застосовувати сучасні інформаційно-комунікаційні та цифрові технології, спеціалізовані програмні засоби комп'ютерної математики та інтернет-ресурси у професійній діяльності та пошуку наукової інформації для самоосвіти;

– застосовувати базові загальні математичні моделі для специфічних ситуацій, мати навички управління інформацією і застосування комп'ютерних засобів статистичного аналізу даних;

– описувати сучасні тенденції в математиці та інформатиці ;

– системно мислити та застосовувати творчі здібності до формування принципово нових ідей;

– здійснювати пошук інформації в різних джерелах для розв'язання задач спеціальності;

– визначати та застосовувати методи розроблення та дослідження алгоритмів розв'язування задач з інформатики, описувати і застосовувати методи оцінювання ефективності алгоритмів ;

– ефективно працювати як індивідуально, так і у складі команди;

– поєднувати теорію і практику.

Теоретичний зміст програми навчальної дисципліни у другому семестрі

Тема 1. Множини в Python.

Тема 2. Обробка виключень в Python.

Тема 3. Функції в Python.

Тема 4. Робота з файлами.

Тема 5. Бібліотека numpy.

Тема 6. Бібліотека matplotlib.

Тема 7. Бібліотека tkinter.

Тема 8. Елементи об'єктно-орієнтованого програмування в Python.

Лабораторна робота №1. Множини та обробка виключень у Python

Мета: Ознайомитися з основними поняттями та методами роботи з множинами, навчитися використовувати множини при написанні програм; навчитися здійснювати обробку виключень (помилки) в Python.

Короткі теоретичні відомості

Множини

Мова програмування Python має засоби для роботи з множинами. У Python **множина** – це неупорядкована колекція елементів, яка має наступні особливості:

- усі елементи множини є унікальні. Іншими словами, у множині не може бути двох однакових елементів;
- елементи множини є незмінюваними об'єктами;
- елементи множини можуть бути різних типів.

У Python множина представлена окремим типом даних. Множини підтримують операції, що існують в теорії множин з курсу математики. Множина може бути пустою (не містити елементів).

Щоб створити об'єкт типу “множина” можна використати один з наступних способів:

- з допомогою присвоєння цьому об'єкту множини елементів фігурних дужках { };
- з допомогою присвоєння об'єкту деякого результату, що був повернутий функцією set(). Функція set() дозволяє перетворити в множину об'єкти інших типів;
- з допомогою присвоєння об'єкту результату функції frozenset(). Функція frozenset() є конструктором класу frozenset. Ця функція дозволяє отримати незмінну (immutable) множину;
- з допомогою присвоєння об'єкту результату деякої операції над множинами (наприклад, операції & перетину множин);
- з допомогою присвоєння об'єкту результату функції, яка повертає деяку множину.

Мова Python підтримує наступні операції над множинами:

in – перевірка елемента на входження в множину;

– (мінус) – різниця множин;

| – об'єднання множин;

& – перетин множин;

^ – симетрична різниця.

Методи для роботи з множинами в Python

Функції	Опис
add()	Додає елемент до множини.
all()	Повертає True, якщо всі елементи множини дорівнюють True (або якщо множина порожня).
any()	Повертає True, якщо хоч один з елементів множини дорівнює True. Якщо множина порожня, повертається False.
clear()	Видаляє всі елементи з множини.
copy()	Повертає копію множини.
difference()	Повертає різницю двох або більше множин у вигляді нової множини.
difference_update()	Видаляє всі елементи іншої множини з поточної множини.
discard()	Видаляє елемент із множини, якщо він є її частиною. (Нічого не робить, якщо елемент не знаходиться у множині)
enumerate()	Повертає об'єкт переліку, який містить індекс і значення для всіх елементів множини у вигляді пари.
intersection()	Повертає перетин двох множин у вигляді нової множини.
intersection_update()	Оновлює множину перетином себе та іншої множини.
isdisjoint()	Повертає True, якщо дві множини не мають перетину.
issubset()	Повертає True, якщо інша множина містить поточну множину.
issuperset()	Повертає True, якщо поточна множина містить іншу множину.
len()	Повертає довжину (кількість елементів) у множині.

max()	Повертає найбільший елемент у множині.
min()	Повертає найменший елемент у множині.
pop()	Видаляє та повертає довільний елемент множини. Викликає помилку <code>KeyError</code> , якщо множина порожня.
remove()	Видаляє елемент із множини. Якщо елемент не є членом множини, виникає помилка <code>KeyError</code> .
sorted()	Повертає новий відсортований список з елементів множини (але не виконує сортування).
sum()	Повертає суму всіх елементів множини.
symmetric_difference()	Повертає симетричну різницю двох множин у вигляді нової множини.
symmetric_difference_update()	Оновлює множину з симетричною різницею себе та іншої.
union()	Повертає об'єднання множин у новій множині.
update()	Оновлює множину з об'єднанням себе та інших.

Обробка помилок в Python

Існує три типи помилок, що виникають під час виконання програми:

- синтаксичні – це помилки, що породжуються неправильним використанням синтаксичних конструкцій мови програмування;
- помилки виконання (`runtime error`) – це помилки, що виникають у синтаксично правильних програмах під час їхнього виконання, коли інтерпретатор не може коректно розв'язати проблему, що виникла або подальше виконання програми є неможливим;
- логічні помилки – це помилки пов'язані з логікою програми. Інтерпретатор не зможе виявити і ідентифікувати такі помилки, оскільки з точки зору синтаксису програма написана правильно.

Якщо у програмі виникає помилка першого або другого типу, то будемо казати, що відбулася виключна ситуація. Будь-яка виключна ситуація генерує (також використовується термін породжує) виключення.

Виключення (eng: exception) – спеціальний тип даних, що використовується для ідентифікації та детального опису помилки, що виникла у програмі.

Слід зауважити, що виключення породжуються не лише у разі виникнення виключної ситуації, а і як повідомлення про те, що відбулася певна подія. Наприклад, метод списку `pop()` генерує виключення `IndexError`, якщо список порожній.

Для обробки виключень використовується оператор `try`. Його загальний синтаксис такий

```
try:  
processing_code_with_potential_exception  
except exception_1  
handling_exception_1  
...  
except exception_N  
handling_exception_N  
else:  
state_else  
finally:  
obligatory_code
```

Щоб перевірити стан програми (наприклад для того, щоб переконатися, що подальше її виконання має сенс) використовують інструкцію `assert`. Вона має такий синтаксис

```
assert condition, message
```

де `condition` – умова (логічний вираз), `message` – необов'язковий параметр, що є даними, що передаються на обробку при виникненні виключної ситуації. Інструкція **`assert`** генерує виключення типу `AssertionError`, якщо логічний вираз `condition` є хибним. Наприклад,

```
x = 0  
try:
```

```
assert x != 0
except AssertionError:
print("Ділити на 0 не можна")
else:
print(1 / x)
```

Завдання для індивідуальної роботи

I. Використовуючи множини, скласти програму на мові Python для розв'язання наступних завдань (номер варіанту відповідає порядковому номеру студента у списку підгрупи).

- 1) Троє сусідів садівників Антон, Тарас, Олег посадили нові куці та дерева. У Антона ростуть вишні, яблуні, груші, виноград, калина, ліщина волоський горіх. У Тараса росте алича, слива, ліщина, яблуня, груша, черешня, малина, виноград. У Олега – яблуня, груша, слива, черешня, малина, порічка, смородина чорна, абрикос, персик, виноград. Визначити, які однакові рослини ростуть у всіх трьох садівників? Які рослини ростуть у Олега, але немає у інших? Які рослини ростуть у Тараса і Олега, і не ростуть у Антона? Який перелік рослин у всіх трьох?
- 2) Три каси на автовокзалі продавали квитки на автобусні рейси. Використовуючи множини, визначити, на які спільні рейси купляли квитки в кожній касі, на які рейси купляли в 1-й касі, а в 2-й ні, які спільні рейси продавали у 2 і 3-й? Скільки всіх різних рейсів продавали каси?
- 3) Студенти 108 групи відвідують спортивні секції: футбол, волейбол, плавання. Багато студентів відвідує не одну, а кілька секцій. Створити 3 колекції, які будуть містити прізвища студентів, які ходять в кожну секцію. Визначити, які студенти ходять тільки на футбол; на футбол і плавання; які ходять на всі три секції; які ходять тільки на плавання; які ходять на волейбол і не ходять на футбол?
- 4) Алла, Інна ті Іванна склали свої квіткові композиції на конкурс квітів. Алла використала тюльпани, нарциси, гербери, гвоздики, фіалки, орхідеї, Інна – троянди, гербери, орхідеї, піони, лілії, ромашки, фіалки, Іванна – айстри, піони, ромашки, орхідеї, амариліс, лаванду, гербери, троянди. Визначити список всіх квітів, які використали дівчата. Які квіти використовувалися всіма трьома дівчатами? Які квіти

використала Інна і не використала Алла? Які квіти використали Інна або Іванна? Які квіти були тільки у Інни, а в інших дівчат – ні?

- 5) Женя, Микола та Олександра вивчають мови програмування. Визначити, які мови вивчають всі троє одночасно? Які вивчає тільки Женя, а інші- ні? Які вивчають Микола та Олександра? Які загалом мови вивчають ці студенти?
- 6) Всі студенти 105 групи записалися на вибіркові курси на наступний навчальний рік, згідно зі своїми навчальними інтересами. Вибіркових курсів було 4, назвемо їх умовно А, В, С, М. Визначити, на який курс записалося найбільше студентів? Які студенти вибрали всі 4 курси? Які вибрали курси А і В одночасно? Які вибрали В або М, і не вибрали А?
- 7) Андрій попросив своїх друзів Ольгу, Катю і Володимира скласти перелік напоїв, які вони люблять. В результаті різні напої вийшли такі: кава без молока, чай чорний, чай зелений, кава з молоком, гарячий шоколад, какао. Визначити, які напої подобаються Андрію і Олі, і не подобаються Каті. Які напої любить Володимир і Ольга? Які любить тільки Андрій і не люблять всі інші?
- 8) Вітя, Аня і Максим мають свої уподобання стосовно мобільних телефонів. Віті подобаються марки Samsung, Meizu, LG, Xiaomi, HTC, Huawei, Ані – Apple, Samsung, Meizu, Nokia, LG, Максиму – Apple, Samsung, Meizu, Nokia, HTC, OnePlus, Google, LG. Визначити спільні інтереси Віті і Ані, Ані і Максима. Які марки подобаються Ані, і не подобаються хлопцям? Які подобаються хлопцям і не подобаються Ані?
- 9) У кожного з трьох хлопчиків-сусідів є своя колекція іграшкових машин. Андрій має машинки BMW, Audi, Volkswagen, Chery, Mitsubishi, Chevrolet, Nissan, Mazda, Citroen, Василь – KIA, Dacia, Daewoo, Volkswagen, Ferrari, Fiat, Chevrolet, Mazda, Ford, Дмитро – Honda, Hummer, Audi, Jeep, BMW, KIA, Mercedes, Mitsubishi, Volkswagen. Визначити, які моделі є тільки у Андрія, і немає у інших хлопців? Які однакові моделі є у Василя та Дмитра? Які є у Василя і нема у Андрія? Які моделі взагалі є у хлопців? Які моделі є у всіх трьох?
- 10) У двох списках є перелік назв продуктів, що є у двох сусідніх магазинах А і В на вул. Головній. Визначити, які продукти є в обох магазинах— і в А і В, які є в магазині А, але нема в В, які продукти є

лише в А або лише в В. Який асортимент продуктів представлений в обох магазинах загалом?

- 11) Андрій, Оксана та Іван мають свої уподобання стосовно ягід і фруктів. Визначити, які фрукти варто замовити на вечірку, щоб задовольнити смаки всіх трьох людей? Які ягоди та фрукти любить тільки Оксана, а хлопці-ні? Які спільні уподобання є у Оксани та Івана? Які ягоди та фрукти любить Андрій, і не люблять Оксана та Іван ?
- 12) Троє дівчат пішли на пленер. Анна взяла з собою червоні, зелені, сині, фіолетові, голубі, жовті фарби. Таня – білі, бежеві, голубі, салаткові, рожеві, сині, зелені, Оля – червоні, чорні, сірі, коричневі, зелені, сині, оранжеві, фіолетові, жовті. Визначити, які кольори у всіх трьох дівчат однакові? Яких кольорів нема у Олі, але є у Тані? Які кольори є у Анни і Тані, але нема у Олі? У кого найбільше фарб?
- 13) Три сусіди-фермери вирощують свою овочеву продукцію. Використовуючи множини, визначити, які спільні овочі вони вирощують? Які овочі є у 1 сусіда і немає у 2 і 3-го? Які загалом овочі у них всіх трьох ростуть? Які овочі ростуть у 2-го і не вирощує 3-й? Які овочі є у 1 і 2-го, але не є їх спільними ?
- 14) Двоє друзів склали список улюблених фільмів. Визначити, скільки разом фільмів є у списках? Які є у одного друга і немає в іншого? Які фільми є спільними? Які фільми є улюбленими в обох друзів але не є спільними?
- 15) Андрій, Оксана і Віктор подорожували Україною. Використовуючи множини, визначити, які міста відвідали всі троє? Які відвідав Віктор і не відвідували Андрій і Оксана? Яке місто відвідав лише один з них?

II. Скласти програму на мові Python для розв'язання наступних завдань, використовуючи обробку виключень (номер варіанту відповідає порядковому номеру студента у списку підгрупи).

- 1) Задано рядок, серед елементів якого містяться букви, символи пунктуації та цифри. Використовуючи для всіх символів рядка функцію `int()` – перетворення символу у число, обчислити середнє арифметичне значення всіх цифр заданого рядка.

- 2) Знайти півпериметр трикутника за трьома сторонами k, l, h . Оформити перевірку вхідних даних (що трикутник з такими сторонами k, l, h існує) за допомогою оператора **assert**.
- 3) Знайти площу трикутника за трьома сторонами a, b, c . Оформити перевірку вхідних даних (що трикутник з такими сторонами a, b, c існує) за допомогою оператора **assert**.
- 4) Визначити розв'язки рівняння $ax^2 + bx + c = 0$, в залежності від коефіцієнтів a, b, c . Оформити перевірку вхідних даних (що розв'язок існує) за допомогою оператора **assert**.
- 5) Використовуючи операцію **index** (для списків), знайдіть позиції у заданому списку всіх елементів заданої множини.
- 6) Є 5 списків різної довжини. Видалити елемент з 10-ї позиції кожного списку. Оформити перевірку вхідних даних (що така позиція існує в кожному списку) за допомогою оператора **assert**.
- 7) Задано рядок s і символ p . Використовуючи операцію **index()** визначити індекс першого входження символу p у рядок s . Оформити перевірку вхідних даних за допомогою оператора **assert**.
- 8) Задано рядок, серед елементів якого містяться букви, символи, цифри. Використовуючи для всіх елементів рядка функцію **int()** – перетворення символу у число, обчислити кількість цифр у заданому рядку.
- 9) Дано словник s і значення k . Використовуючи операцію $s.pop(k)$, визначити, чи містить словник s ключ k . Оформити перевірку вхідних даних за допомогою оператора **assert**.
- 10) Задано рядок, серед елементів якого містяться цифри. Використовуючи для всіх символів рядка функцію **int()** – перетворення символу у число, обчислити суму цифр заданого рядка.
- 11) Задано рядок. Перевірити, чи є він записом дійсного числа.
- 12) Задано рядок. Перевірити, чи є він записом цілого числа.
- 13) Задано множину M і елемент x . Використовуючи операцію $M.remove(x)$, видалити елемент x з множини M . Оформити перевірку вхідних даних (що такий елемент існує).
- 14) Задані списки s і p однакової довжини n . Знайти суму: $1/(p_1-s_1) + 1/(p_2-s_2) + 1/(p_3-s_3) + \dots + 1/(p_n-s_n)$. Описати виключення «ділення на нуль».
- 15) Дано кілька різних словників, деякі з них порожні. Використовуючи операцію $d.popitem()$, видалити і надрукувати на екран по одному елементу з кожного словника (якщо це можливо). Оформити перевірку вхідних даних за допомогою оператора **assert**.

Запитання для повторення:

1. *Що таке множина у Python?*
2. *Які способи створення множини?*
3. *Які операції над множинами можливі у Python?*
4. *Чим відрізняються множини від інших колекцій?*
5. *Які дії над множинами дозволені?*
6. *Що таке виключення?*
7. *Які типи помилок виникають під час виконання програми?*
8. *Який оператор використовується для обробки виключень? Його загальний синтаксис?*
9. *Для чого і як використовують інструкцію `assert`?*
10. *Який тип виключення використовується для перехоплення помилки «ділення на нуль»?*