**ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ**  
**«ОПТИКО-МЕХАНІЧНИЙ ФАХОВИЙ КОЛЕДЖ КИЇВСЬКОГО**  
**НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ ІМЕНІ ТАРАСА ШЕВЧЕНКА»**  
Циклова комісія програмування та інформаційних технологій

**ЗВІТ**

**З НАВЧАЛЬНОЇ ПРАКТИКИ**

спеціальність *121 Інженерія програмного забезпечення*  
освітньо-кваліфікаційний рівень *«Фаховий молодший бакалавр»*

|  |  |
| --- | --- |
| ОЦІНКА  (оцінка цифрою та прописом)  Керівник практики від коледжу:  (підпис)  « \_» 20 року | Виконав:  студент 3-го курсу групи ІПЗ-32  *Шевченко Ростислав Андрійович*  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (підпис виконавця)  « \_» 20 року |

КИЇВ - 2025

ЗАТВЕРДЖУЮ

Директор ВСП «ОМФК КНУ

ім. Тараса Шевченка»

\_\_\_\_\_\_\_\_\_\_\_Борис ГАПРИНДАШВІЛІ

«\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2025р

**ІНДИВІДУАЛЬНЕ ЗАВДАННЯ**

на період *Навчальної* практики

(навчальної, технологічної, виробничої)

*Глушко Анна Федорівна*

(П І.Б студента )

Спеціальність *121 Інженерія програмного забезпечення* Група *ІПЗ-32*

|  |  |  |
| --- | --- | --- |
|  | **Питання для вивчення** | **Відмітки про виконання** |
| **1** | Вступне заняття. Вивчення питань охорони праці. Ознайомлення з електронною бібліотекою коледжу. Вивчення правил оформлення звітної документації. |  |
| **2** | Знайомство з мовою програмування Python. Історія мови, особливості мови, сфери застосування мови |  |
| 3 | Основи мови Python. Синтаксис мови, середовище виконання, запуск програм |  |
| 4 | Змінні та літерали, типи даних. Вбудовані типи даних, ініціалізація змінних, змінювані і незмінювані типи |
| 5 | Програмування лінійних та розгалужених обчислювальних процесів |  |
| 6 | Програмування циклічних обчислювальних процесів |  |
| 7 | Типи даних визначені користувачем, магічні методи, виключення (Exceptions) та їх обробка |  |
| 8 | Вбудовані структури даних, складні типи |  |
| 9 | Функції |  |
| 10 | Потокове введення, виведення та обробка даних |  |
| 11 | Побудова проекту, модулі, пакети |  |
| 12 | Робота з файловою системою, оператор with, контекстний менеджер |  |
| 13 | Робота з протоколом HTTP, бібліотека requests |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Голова циклової комісії |  |  |  |
|  | (підпис) | (ініціали, прізвище) | (дата) |
| Керівник практики від коледжу |  |  |  |
|  | (підпис) | (ініціали, прізвище) | (дата) |

**Зміст**

[ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ 4](#_Toc565466204)

[Вступ 5](#_Toc1137144064)

[1 Теоретичні відомості 7](#_Toc928310420)

[1.1 Вступне заняття 7](#_Toc1749277208)

[1.2 Знайомство з мовою програмування Python. Історія мови, особливості мови, сфери застосування мови 7](#_Toc1313556716)

[Історія мови Python 7](#_Toc1402805331)

[Особливості мови Python 8](#_Toc1124402173)

[Сфери застосування Python 8](#_Toc141203036)

[1.3 Основи мови Python. Синтаксис мови, середовище виконання, запуск програм 9](#_Toc975059889)

[Синтаксис мови Python 9](#_Toc1551059058)

[Середовище виконання Python 9](#_Toc1045124425)

[Запуск програм на Python 9](#_Toc886069885)

[1.4 Змінні та літерали, типи даних. Вбудовані типи даних, ініціалізація змінних, змінювані і незмінювані типи 10](#_Toc457620007)

[Змінні та літерали 10](#_Toc1001588155)

[Типи даних 10](#_Toc327235305)

[Ініціалізація змінних 10](#_Toc306893266)

[Змінювані та незмінювані типи 11](#_Toc2083310759)

[2 Програмування лінійних та розгалужених обчислювальних процесів 11](#_Toc1918252953)

[2.1 Програмування циклічних обчислювальних процесів 12](#_Toc151871907)

[2.2 Типи даних визначені користувачем, магічні методи, виключення (Exceptions) та їх обробка 12](#_Toc1332761692)

[Типи даних, визначені користувачем 12](#_Toc60247023)

[2.3 Вбудовані структури даних, складні типи 12](#_Toc141563928)

[Списки (Lists) 13](#_Toc327833781)

[Словники (Dictionaries) 13](#_Toc1190511440)

[Множини (Sets) 13](#_Toc1319207991)

[Складні типи даних 13](#_Toc491495181)

[3 Функції 14](#_Toc952659577)

[3.1 Потокове введення, виведення та обробка даних 14](#_Toc565199148)

[Потокове введення 14](#_Toc436532189)

[Потокове виведення 14](#_Toc1342440961)

[Обробка даних 14](#_Toc1003759454)

[Робота з файлами CSV 14](#_Toc1380895627)

[3.2 Побудова проекту, модулі, пакети 14](#_Toc1104850220)

[Модулі 14](#_Toc2043141505)

[Пакети 14](#_Toc622146731)

[Структура проекту 15](#_Toc1399309708)

[Віртуальне середовище 15](#_Toc945261717)

[Файл requirements.txt 15](#_Toc753743243)

[4 Робота з файловою системою, оператор with, контекстний менеджер 15](#_Toc669335905)

[4.1 Робота з протоколом HTTP, бібліотека requests 15](#_Toc1383604342)

[Основи HTTP 15](#_Toc1473466461)

[Бібліотека requests 15](#_Toc1749593572)

[5 Виконання завдань 16](#_Toc1141070261)

[5.1 Завдання 1 16](#_Toc1730155898)

[5.2 Завдання 2 17](#_Toc952682175)

[5.3 Завдання 3 18](#_Toc1979880078)

[6 Висновки 20](#_Toc937489042)

# **ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ**

* **API** (Application Programming Interface) - інтерфейс програмування додатків
* **HTML** (Hypertext Markup Language) - мова розмітки гіпертексту
* **URL** (Uniform Resource Locator ) - Уніфіко́ваний лока́тор ресу́рсів

# **Вступ**

Метою практики є поглиблення знань та набуття практичних навичок у мові Python. Під час практики ми ознайомлювались з основами мови програмування: типи даних, функції, взаємодія з API і т.д. Виконували практичні заняття.

У звіті описано основні етапи проходження практики, виконані завдання, використані технології та інструменти, а також отримані результати.

# **Теоретичні відомості**

## **1.1 Вступне заняття**

Метою вступного заняття було ознайомлення з програмою та основними напрямками курсу.

Під час вступного заняття було розглянуто такі питання:

* **Організація практики:** обговорено терміни проходження практики, розподіл часу на виконання завдань, порядок консультацій з керівником практики.
* **Мета та завдання практики:** сформульовано основну мету практики - поглиблення знань з Python та набуття практичних навичок розробки програмного забезпечення з використанням цієї мови програмування. Окреслено конкретні завдання, які студенти повинні виконати під час практики, зокрема:
  + Вивчення основних концепцій Python: типи даних, оператори, цикли, функції, класи.
  + Практичне застосування Python для вирішення конкретних завдань з програмування.
  + Розробка невеликого програмного проекту на Python.
  + Оформлення звіту про практику.
* **Вимоги до звіту:** детально роз'яснено структуру звіту, вимоги до оформлення коду, використання літератури та інших джерел. Наголошено на важливості чіткого та структурованого викладу матеріалу, а також на необхідності дотримуватися академічної доброчесності.
* **Рекомендації щодо виконання завдань:** надано практичні поради щодо ефективного виконання завдань, зокрема щодо пошуку необхідної інформації, використання онлайн-ресурсів та інструментів розробки.
* **Відповіді на питання:** студенти мали можливість поставити питання щодо організації практики, завдань та вимог до звітності.

## **1.2 Знайомство з мовою програмування Python. Історія мови, особливості мови, сфери застосування мови**

### **Історія мови Python**

Мову Python було створено Гвідо ван Россумом у 90-х роках. Назва мови походить від назви комедійного серіалу "Monty Python's Flying Circus". Ван Россум працював над Python в Інституті CWI в Нідерландах, де він займався розробкою розподіленої операційної системи Amoeba.

Перша версія Python була опублікована у 1991 році. З того часу Python постійно розвивався та вдосконалювався. У 2000 році вийшла версія Python 2.0, а у 2008 році - Python 3.0. Python 3.0 був значним кроком вперед, оскільки він містив багато змін, які покращили мову та її можливості.

### **Особливості мови Python**

Python має кілька особливостей, які роблять його привабливим для розробників:

* **Простота та читабельність:** Python має простий та зрозумілий синтаксис, що робить його легким для вивчення та використання. Код на Python легко читається та розуміється, що зменшує час на розробку та підтримку програмного забезпечення.
* **Великий вибір бібліотек:** Python має велику кількість бібліотек, які охоплюють різні сфери застосування, від веб-розробки до аналізу даних та машинного навчання. Це дозволяє розробникам швидко та ефективно створювати складні програми.
* **Кросплатформність:** Python є кросплатформною мовою, що означає, що програми, написані на Python, можуть працювати на різних операційних системах, таких як Windows, macOS та Linux.
* **Об'єктно-орієнтоване програмування:** Python підтримує об'єктно-орієнтоване програмування, що дозволяє розробникам створювати модульні та масштабовані програми.
* **Велика спільнота:** Python має велику та активну спільноту розробників, які завжди готові допомогти та поділитися своїм досвідом.

### **Сфери застосування Python**

Python використовується в різних сферах, зокрема:

* **Веб-розробка:** Python є популярним вибором для веб-розробки завдяки таким фреймворкам, як Django та Flask.
* **Аналіз даних та машинне навчання:** Python має потужні бібліотеки для аналізу даних та машинного навчання, такі як NumPy, Pandas та Scikit-learn.
* **Автоматизація:** Python використовується для автоматизації різних завдань, таких як обробка файлів, веб-скрепінг та тестування програмного забезпечення.
* **Розробка ігор:** Python може використовуватися для розробки ігор за допомогою бібліотеки Pygame.
* **Освіта:** Python є популярною мовою для навчання програмуванню завдяки своїй простоті та зрозумілості.

## **1.3 Основи мови Python. Синтаксис мови, середовище виконання, запуск програм**

### **Синтаксис мови Python**

Синтаксис Python відрізняється своєю лаконічністю та інтуїтивністю. Ось деякі ключові особливості:

* **Відсутність крапок з комою:** Python не використовує крапки з комою для позначення кінця оператора. Замість цього використовується перехід на новий рядок.
* **Відступи:** Python використовує відступи для позначення блоків коду. Це робить код більш читабельним та структурованим.
* **Динамічна типізація:** Python автоматично визначає тип змінної під час виконання програми.
* **Великий вибір вбудованих функцій та типів даних:** Python має велику кількість вбудованих функцій та типів даних, що полегшує розробку програм.

### **Середовище виконання Python**

Для виконання програм на Python необхідно встановити інтерпретатор Python. Існує кілька способів встановлення Python:

* **Завантаження з офіційного сайту:** Ви можете завантажити останню версію Python з офіційного сайту python.org.
* **Використання дистрибутивів:** Існують різні дистрибутиви Python, такі як Anaconda та ActivePython, які включають в себе інтерпретатор Python та інші корисні інструменти.

Після встановлення Python ви можете використовувати різні середовища розробки для написання та виконання програм:

* **IDLE:** IDLE - це інтегроване середовище розробки, яке постачається разом з Python.
* **Visual Studio Code:** Visual Studio Code - це популярний редактор коду з підтримкою Python.
* **PyCharm:** PyCharm - це професійне середовище розробки для Python.

### **Запуск програм на Python**

Існує кілька способів запуску програм на Python:

* **Запуск з командного рядка:** Ви можете запустити програму Python з командного рядка, вказавши шлях до файлу з розширенням .py.
* **Запуск з інтегрованого середовища розробки:** Ви можете запустити програму Python безпосередньо з інтегрованого середовища розробки, натиснувши кнопку запуску.

## **1.4 Змінні та літерали, типи даних. Вбудовані типи даних, ініціалізація змінних, змінювані і незмінювані типи**

У Python змінні використовуються для зберігання даних. Літерали - це безпосередньо задані значення, наприклад, числа, рядки або логічні значення. Кожна змінна має тип даних, який визначає, які значення вона може зберігати. Python має кілька вбудованих типів даних, які ми розглянемо нижче.

### **Змінні та літерали**

Змінні в Python можуть зберігати значення різних типів. Ім'я змінної може складатися з літер, цифр та символу підкреслення, але воно повинно починатися з літери або символу підкреслення. Python є динамічно типізованою мовою, тому тип змінної визначається під час присвоєння їй значення.

### **Типи даних**

Python має кілька вбудованих типів даних, які використовуються для представлення різних видів інформації.

**Основні типи даних:**

* **int:** цілі числа (наприклад, 10, -5, 0)
* **float:** числа з плаваючою крапкою (наприклад, 3.14, -2.5, 0.0)
* **str:** рядки (наприклад, "Hello", "Python")
* **bool:** логічні значення (True або False)
* **list:** списки (впорядковані колекції елементів)
* **tuple:** кортежі (незмінювані списки)
* **dict:** словники (асоціативні масиви)
* **set:** множини (невпорядковані колекції унікальних елементів)

### **Ініціалізація змінних**

Ініціалізація змінної - це присвоєння їй першого значення. У Python змінні ініціалізуються за допомогою оператора присвоєння =.

### **Змінювані та незмінювані типи**

У Python типи даних поділяються на змінювані (mutable) та незмінювані (immutable).

* **Незмінювані типи:** значення незмінюваних типів не можуть бути змінені після створення. До них відносяться int, float, str, bool та tuple.
* **Змінювані типи:** значення змінюваних типів можуть бути змінені після створення. До них відносяться list, dict та set.

# **2 Програмування лінійних та розгалужених обчислювальних процесів**

## **2.1 Програмування циклічних обчислювальних процесів**

Цикли є однією з фундаментальних концепцій програмування, що дозволяє виконувати блок коду кілька разів. Python надає два основних типи циклів: for та while.

## **2.2 Типи даних визначені користувачем, магічні методи, виключення (Exceptions) та їх обробка**

### **Типи даних, визначені користувачем**

У Python, крім вбудованих типів даних, таких як цілі числа (int), числа з плаваючою крапкою (float), рядки (str) тощо, програмісти можуть створювати власні типи даних, які називаються класами. Класи дозволяють структурувати дані та поведінку в об'єкти, що робить код більш організованим та модульним.

спеціальні методи - це методи, які мають спеціальні імена, що починаються та закінчуються двома підкресленнями (\_\_). Ці методи автоматично викликаються Python в певних ситуаціях, що дозволяє змінювати поведінку об'єктів.

**Деякі з найпоширеніших магічних методів:**

* \_\_init\_\_(self, ...): конструктор класу.
* \_\_str\_\_(self): повертає рядкове представлення об'єкта.
* \_\_repr\_\_(self): повертає представлення об'єкта для розробника.
* \_\_len\_\_(self): повертає довжину об'єкта.
* \_\_getitem\_\_(self, key): дозволяє отримувати доступ до елементів об'єкта за ключем.
* \_\_setitem\_\_(self, key, value): дозволяє встановлювати значення елементів об'єкта за ключем.

## **2.3 Вбудовані структури даних, складні типи**

### **Списки (Lists)**

Списки є впорядкованими, змінюваними колекціями елементів. Елементи списку можуть бути різних типів.

### **Словники (Dictionaries)**

Словники є невпорядкованими колекціями пар ключ-значення. Ключі повинні бути унікальними та незмінюваними.

### **Множини (Sets)**

Множини є невпорядкованими колекціями унікальних елементів.

### **Складні типи даних**

Python дозволяє створювати власні складні типи даних, використовуючи класи. Класи дозволяють об'єднувати дані та методи для їх обробки в одному об'єкті.

# **3 Функції**

## **3.1 Потокове введення, виведення та обробка даних**

### **Потокове введення**

Для отримання даних від користувача або з файлу використовуються функції input() та open().

* input(): зчитує рядок з консолі, введений користувачем.
* open(): відкриває файл для читання або запису.

### **Потокове виведення**

Для виведення даних на консоль або у файл використовуються функції print() та write().

* print(): виводить дані на консоль.
* write(): записує дані у файл.

### **Обробка даних**

Python надає різноманітні засоби для обробки даних, зокрема:

* **Оператори:** арифметичні, логічні, порівняння тощо.
* **Функції:** вбудовані та визначені користувачем.
* **Модулі:** бібліотеки з додатковими функціями та інструментами.
* **Структури даних:** списки, кортежі, словники, множини.

### **Робота з файлами CSV**

CSV (Comma Separated Values) - популярний формат для зберігання табличних даних. Python має вбудований модуль csv для роботи з такими файлами.

## **3.2 Побудова проекту, модулі, пакети**

### **Модулі**

Модуль - це окремий файл з розширенням .py, який містить функції, класи, змінні та інші об'єкти. Модулі дозволяють розділити код на логічні блоки та повторно використовувати їх в різних частинах проекту.

### **Пакети**

Пакети - це спосіб організації модулів у ієрархічну структуру. Пакет являє собою каталог, який містить файл \_\_init\_\_.py (може бути порожнім) та інші модулі або підпакети.

### **Структура проекту**

Рекомендована структура Python-проекту включає:

* Кореневий каталог: містить файли налаштувань, документацію та інші файли проекту.
* Каталог з вихідним кодом: зазвичай називається src або згідно з назвою проекту.
* Каталог з тестами: зазвичай називається tests.
* Файл setup.py: для збирання та встановлення проекту.

### **Віртуальне середовище**

Для ізоляції залежностей проекту від інших проектів рекомендується використовувати віртуальне середовище. Для цього можна скористатися інструментом venv

### **Файл requirements.txt**

У файлі requirements.txt перелічуються всі залежності проекту. Його можна створити за допомогою команди: pip freeze > requirements.txt

# **4 Робота з файловою системою, оператор with, контекстний менеджер**

## **4.1 Робота з протоколом HTTP, бібліотека requests**

HTTP (Hypertext Transfer Protocol) є основним протоколом, який використовується для передачі даних між клієнтом (наприклад, браузером) та сервером у World Wide Web. Розуміння принципів роботи з HTTP та вміння використовувати його для отримання та відправлення даних є важливим для розробки веб-додатків та інтеграції з різними сервісами.

### **Основи HTTP**

HTTP базується на запитах та відповідях. Клієнт надсилає запит на сервер, а сервер обробляє цей запит та повертає відповідь. Запит містить метод (наприклад, GET, POST, PUT, DELETE), URL та заголовки. Відповідь містить код стану (наприклад, 200 OK, 404 Not Found) та тіло.

### **Бібліотека requests**

Python має потужну бібліотеку requests, яка спрощує роботу з HTTP. requests дозволяє легко відправляти HTTP-запити різних типів та обробляти відповіді.

**Інші можливості requests:**

* Передача параметрів в URL: requests.get(url, params={"param1": "value1"})
* Встановлення заголовків: requests.get(url, headers={"User-Agent": "My-User-Agent"})
* Робота з файлами: requests.post(url, files={"file": open("my\_file.txt", "rb")})
* Обробка cookies: response.cookies
* Таймаути: requests.get(url, timeout=5)

# **5 Виконання завдань**

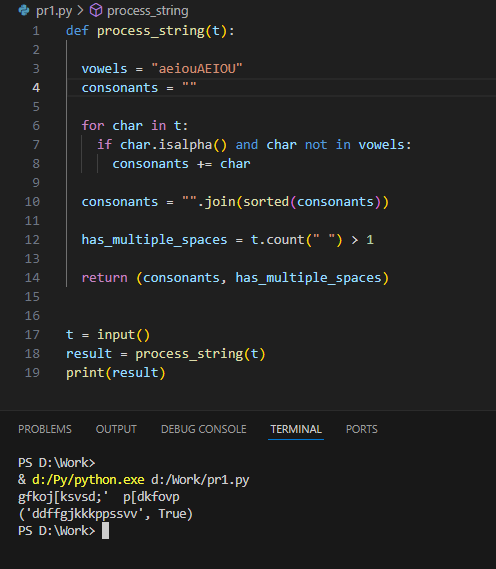
## **5.1 Завдання 1**

**Посилання на гіт хаб:**

<https://github.com/Rost589/Prc/blob/main/Практика/IPZ-32/13_Rostyslav/pr1.py>

**Пояснення:**

Код приймає на вхід строку і повертає кортеж з двох елементів, в яких перший елемент має тип str і включає всі приголосні літери з вхідної строки (алфавітний порядок). Другий елемент - bool - вказує True, якщо у вхідній строці є більше 1го пробілу:



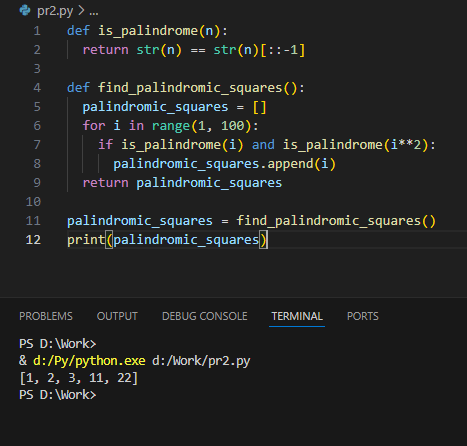
## **5.2 Завдання 2**

**Посилання на гіт хаб:**

<https://github.com/Rost589/Prc/blob/main/Практика/IPZ-32/13_Rostyslav/pr2.py>

**Пояснення:**

Код виводить усі поліндроми першої сотні, якi при піднесенні до квадрата дають паліндром:



## **5.3 Завдання 3**

**Посилання на гіт хаб:**

<https://github.com/Rost589/Prc/blob/main/Практика/IPZ-32/13_Rostyslav/pr3.py>

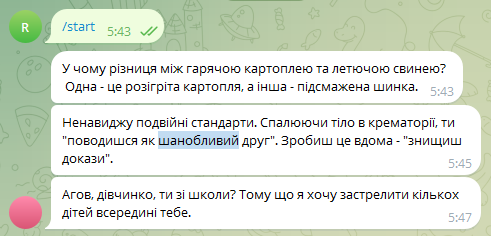
**Пояснення:**

Запускаємо код, водимо токен телеграм бота та ключ DeeplApi:  


Бот активований:



Натискаємо старт. Бот почне надсилати жарти кожні 2 хв.:



# **6 Висновки**

Протягом цього періоду ми ознайомились з основними аспектами Python, від базового синтаксису та типів даних до більш складних тем, таких як об'єктно-орієнтоване програмування, робота з файлами, мережеві запити та розробка веб-додатків.

Поглиблення знань та набуття практичних навичок надає можливість створювати складні проекти та працювати з великими обсягами даних.

# **ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ ІНФОРМАЦІЇ**

1. Посібник з мови Python<https://docs.python.org/3/reference/>
2. Офіційний посібник для початківців <https://docs.python.org/3/tutorial>
3. Початковий посібник для новачків <https://www.python.org/about/gettingstarted>
4. Документація стандартної бібліотеки Python <https://docs.python.org/3/library/>
5. Платформа для практики програмування <https://www.codewars.com/>