## Завдання до лабораторної роботи №5

**ТЕМА.**Функції. Генератори. Обробка помилок. Виняткові ситуації. Блок try-except.

**МЕТА**: навчитися визначати і викликати функції (def, lambda, return) мовою Python, використовувати їх для реалізації алгоритмів розв’язку задач; навчитися визначати помилки та передбачати появу виняткових ситуації (блок try-except).

***Підготовка до виконання завдань лабораторної роботи***

1. Перед виконанням завдань лабораторної роботи опрацюйте [урок 13](https://sites.google.com/site/pythonukr/urok-13-funkciie) та [урок 14](https://sites.google.com/site/pythonukr/urok-14-parametri-i-argumenti-funkcij-lokalni-ta-globalni-zminni) ресурсу [https://sites.google.com/site/pythonukr/,](https://sites.google.com/site/pythonukr/vstup) теми [п.6.4 "Включення (скорочення синтаксису)"](http://pythonguide.rozh2sch.org.ua/#_%D0%B2%D0%BA%D0%BB%D1%8E%D1%87%D0%B5%D0%BD%D0%BD%D1%8F_%D1%81%D0%BA%D0%BE%D1%80%D0%BE%D1%87%D0%B5%D0%BD%D0%BD%D1%8F_%D1%81%D0%B8%D0%BD%D1%82%D0%B0%D0%BA%D1%81%D0%B8%D1%81%D1%83), [п.6.5."Генератори"](http://pythonguide.rozh2sch.org.ua/#_%D0%B3%D0%B5%D0%BD%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%B8), [п.6.6. "Функції"](http://pythonguide.rozh2sch.org.ua/#_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D1%96%D1%97), [п.6.7. "Обробка помилок"](http://pythonguide.rozh2sch.org.ua/#_%D0%BE%D0%B1%D1%80%D0%BE%D0%B1%D0%BA%D0%B0_%D0%BF%D0%BE%D0%BC%D0%B8%D0%BB%D0%BE%D0%BA) із [http://pythonguide.rozh2sch.org.ua](http://pythonguide.rozh2sch.org.ua/)або  інші електронні джерела.

2. Перед виконанням практичних завдань лабораторної роботи дайте відповіді на контрольні запитання та виконайте вправи (джерело "[Путівник мовою програмування Python](http://pythonguide.rozh2sch.org.ua/#_%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D1%8C%D0%BD%D1%96_%D0%B7%D0%B0%D0%BF%D0%B8%D1%82%D0%B0%D0%BD%D0%BD%D1%8F_3)") .

3.  [Вбудовані винятки в Python](https://docs.python.org/3.6/library/exceptions.html)

#### Контрольні запитання для самоперевірки

1. Навіщо використовувати у програмах функції?
2. Коли виконується код функції: коли вона визначається, чи коли вона викликається?
3. З допомогою якої конструкції створюються функції?
4. Чим відрізняється визначення функції від її виклику?
5. Скільки глобальних областей видимості може мати програма на Python? Скільки локальних?
6. Що таке **значення, яке повертає функція**? Чи може бути це значення у вигляді виразу?
7. Що відбувається із змінними, які знаходяться у локальній області видимості, при поверненні них із функції?
8. Яке значення повертає функція, якщо у ній відсутній оператор **return**?
9. Як змусити змінну у функції посилатися на глобальну змінну?
10. Що таке тип даних **None**?
11. Коли і як використовується анонімна функція **lambda**?
12. Як можна запобігти аварійному завершенні програми?
13. Які блоки коду записують у блок **try**? А які у **except**?

#### **Вправи для самоперевірки**

Виконайте в **інтерактивному інтерпретаторі** на вибір три вправи, що відповідають відповідно темам "Включення (скорочення синтаксису).", "Функції" та "Обробка винятків":

1. Використайте функцію **zip()**, щоб створити словник **movies**, який об’єднує у пари списки: **seasons = ['summer', 'autumn']** і **months = [ 'july', 'november']**. Виведіть вміст словника.
2. Використайте відомі вам структури коду для виведення ключів і значень словника **activity = {'business': 'manager', 'it': 'software developer', 'science': 'scientist'}** у вигляді, на зразок «категорія: професія».
3. Використайте включення списку, щоб створити список, який містить парні числа у діапазоні **range(12)**.
4. Використайте включення словника, щоб створити словник **squares** з ключами у вигляді цілих чисел з діапазону **range(1, 11)**. Значення словника формуються піднесенням ключів до другого степеня.
5. Створіть список чисел від 1 до 1 000 000. Скористайтеся функціями **min()** і **max()** та переконайтеся у тому, що список дійсно починається 1 і закінчується 1 000 000. Викличте функцію **sum()** і подивіться, наскільки швидко Python зможе підсумувати мільйон чисел.
6. Cкористайтеся третім аргументом функції **range()** для створення списку непарних чисел від 1 до 25і виведіть усі числа в окремих рядках у циклі **for**.
7. Створіть список перших 10 кубів (тобто кубів усіх цілих чисел від 1 до 10) і виведіть значення усіх кубів у циклі **for** в один рядок з пропусками.
8. Cтворіть список чисел у діапазоні від 3 до 60 і виведіть усі числа списку у циклі **while** в окремих рядках.
9. Визначте функцію **trees**, яка повертає список **['poplar', 'willow', 'lime']**. Викличте функцію.
10. Напишіть функцію **favorite\_book()**, яка отримує один параметр **title**. Функція повинна виводити повідомлення, на зразок **One of my favorite books is "The Lord of the Rings".**. Викличте функцію і переконайтеся у тому, що назва книги правильно передається як аргумент при виконанні функції.
11. Напишіть функцію **make\_shirt()**, яка отримує розмір футболки і текст, який повинен бути надрукований на ній. Функція повинна виводити повідомлення з розміром і текстом. Викличте функцію з використанням позиційних аргументів. Викличте функцію вдруге з використанням іменованих аргументів. Змініть функцію **make\_shirt()**, щоб футболки за замовчуванням мали розмір **XL**, і на них виводився текст **I love Python!**. Створіть футболку з розміром **XL** і текстом за замовчуванням.
12. Напишіть функцію **city\_country()**, яка отримує назву міста і країну. Функція повинна повертати рядок у форматі **Kiev, Ukraine**. Викличте функцію як мінімум для трьох пар «місто-країна».
13. Визначте виняток, який називається **MyException**. Згенеруйте його, щоб побачити, що станеться. Потім напишіть код, що дозволяє перехопити цей виняток і вивести рядок **This is my exception.**.
14. При введенні числових даних часто зустрічається типова проблема: користувач вводить текст замість чисел. При спробі перетворити дані в **int** генерується виняток **TypeError**. Напишіть функцію, яка приймає два числа, шукає їх суму і виводить результат. Перехопіть виняток **TypeError**, якщо будь-яке із вхідних значень не є числом, і виведіть зручне повідомлення про помилку. Протестуйте функцію: спочатку введіть два числа, а потім введіть текст замість одного з чисел.

#### Практичні завдання лабораторної роботи

**ЗАВДАННЯ 1 (0,5 бал).**Написати простий калькулятор, що зчитує три рядки, які вводить користувач: перше число, друге число і операцію, після чого застосовує операцію до введених числах («перше число» «операція» «друге число») і виводить результат на екран. Підтримувані операції: **+**, **-**, **/**, **\***, **mod**, **pow**, **div**, де **mod** - це взяття залишку від ділення, **pow** - піднесення до степеня, **div** - цілочисельне ділення. У випадку ділення на 0, необхідно використати обробник винятку (*try-except-else-finaly*), який має вивести повідомлення **Division by zero!**.

**ЗАВДАННЯ 2 (0,5 бали)**. У завданнях під номером 2 [**Завдання до лабораторної роботи №4\_1**](https://do.rshu.edu.ua/mod/assign/view.php?id=1446) та [**Завдання до лабораторної роботи №4\_2**](https://do.rshu.edu.ua/mod/assign/view.php?id=1400)оформити частини програми роботи зі списками та відповідно із словиками у вигляді функцій.

!!! АБО, якщо ви виконували ЗАВДАННЯ 4 із [**Завдання до лабораторної роботи №4\_2**](https://do.rshu.edu.ua/mod/assign/view.php?id=1400)**,**то відповідно визначте функцію ***count\_literi***, яка виконує підрахунок  букв у записі числа англійською мовою без врахування пропусків та дефісів, та  функцію ***convetr\_to\_text***, яка виконує переведення числа в текстовий його запис на англійській мові.

**ЗАВДАННЯ 3 (0,5 бали)**Написати програму, що здійснює переведення числа [з десяткової системи числення в двійкову систему числення](https://uk.wikipedia.org/wiki/%D0%94%D0%B2%D1%96%D0%B9%D0%BA%D0%BE%D0%B2%D0%B0_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%87%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%BD%D1%8F). Окремо описати функції, що переводять відповідно цілу та дробову частину десяткового числа у двійковий код. *Примітка*: бажано не використовувати стандартну функцію ***bin().***

*Примітка!*Переведення дробової частини числа цілої частини можна виконати також за наступним алгоритмом:

Нехай result буде порожнім рядком  
Нехай q представляє ціле число для перетворення  
Повторити  
 Встановити r рівним остачі, коли q ділиться на 2  
 Перетворити r у рядок та додати його на початку до result  
 Розділити q на 2, відкидаючи будь-який залишок, і зберегти результат назад у q  
Поки q не дорівнює 0  
Вивести повідомлення про результат переведення

**ЗАВДАННЯ 5 (0,5 бали).**Дано речення. Перетворивши його в список символів (без пропусків), підрахувати скільки разів в ньому зустрічається кожен символ. Наприклад, якщо речення перетворене в список ['h','e','l','l','o'], то в ньому символи 'a',  'e' і 'o' зустрічаються один раз, символ 'l'  - два рази. Для збереження кількості кожного символа, що зустрічається в реченні створити словник {'h':1, 'e':1, 'l':2, 'o':1}. Окремі фрагменти реалізації програми оформити у вигляді функцій.

***Для підготовки звіту потрібно:***

1) зберегти файли проектів та завантажити в папку "Сучасні технології програмування/ЛР5" на свій **Google Диск**або репозиторій на  [https://bitbucket.org](https://bitbucket.org/)**;**

2) слід натиснути кнопку "*Здати роботу*" у поточному завданні;

3) в полі "текст-онлайн" вставити посилання для доступу викладачу до Вашої папки "Сучасні технології програмування/ЛР5",  що містить файли з виконаними завданнями на Google Диску, або на репозиторій, що містить файли виконаних завдань, та натиснути кнопку "*Зберегти*";

3) відправити викладачу для попереднього оцінення натиснувши кнопку "*Відправити на оцінення*";

4) додаткові запитання від викладача та захист в режимі *онлайн*(коментарем або листом) та *офлайн*, тобто в аудиторії.