НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

імені Ігоря Сікорського»

Факультет прикладної математики

Кафедра прикладної математики

Звіт

із лабораторної роботи №4

із дисципліни «Основи програмування»

на тему

ВИКОРИСТАННЯ ФУНКЦІЙ

|  |  |
| --- | --- |
| Виконала: | Керівник: |
| студентка групи КМ-93 | ст.вик. Дрозденко О. М. |
| Довгаль Є. О. |  |

Київ — 2019

Зміст

[1 ПОСТАНОВКА ЗАДАЧІ 3](#_Toc22052512)

[1.2 Завдання до лабораторної роботи 3](#_Toc22052514)

[2 ОПИС ПРОГРАМИ 4](#_Toc22052515)

[2.1 Змінні та функції 4](#_Toc22052516)

[Додаток А 5](#_Toc22052517)

[Текст програми 5](#_Toc22052518)

[Додаток Б 6](#_Toc22052519)

[Скріншоти тестувань програми 6](#_Toc22052520)

[Додаток В 7](#_Toc22052521)

[Відповіді на питання для самоперевірки 7](#_Toc22052522)

# ПОСТАНОВКА ЗАДАЧІ

## Мета роботи

Вивчення написання і застосування функцій на мові Python

## 1.2 Завдання до лабораторної роботи

1) Вивчити принципи створення функцій мови Python.

2) Розробити програми з визначенням і викликом функцій, які призначені для користувача, відповідно до варіанта завдання. Функції повинні працювати з параметрами.

3) Вхідні дані і результат роботи супроводжувати відповідною інформацією на екрані.

# 2 ОПИС ПРОГРАМИ

## Змінні та функції

Завдання 1:

*Написати функцію, яка б по введеному номеру пори року в якості параметра (1 - зима, 2 - весна, 3 - літо, 4 - осінь) повертала місяці, кількість днів в кожному з місяців, що відповідають цій порі року.*

Пояснення:

Користувач повинен ввести число від 1 до 4. Якщо вводить щось інше – помилка. Але навіщо користувачу щось вводити, тому перебір автоматичний циклом for.

Маємо функцію choose\_season(season), де season – аргумент, тобто номер пори року (1 - зима, 2 - весна, 3 - літо, 4 - осінь). Назви пор року містяться в списку seasons, місяці в months та кількість днів в days.

Кожній порі року відповідають певні три місяці, а цім місяцям – дні. Як результат маємо print(…), де користувач бачить обраний сезон, його місяці, кількість днів та їх суму.

Завдання 2:

*Функція - Parse(s,t). Призначення - поділ рядка s на дві частини: до першого входження символу t і після нього.*

Пояснення:

Функція parse(s, t) ділить введений рядок на дві частини: до символа t та після нього.

Спочатку користувач задає певний рядок, потім вводіть один символ (якщо їх декілька – помилка). У тілі функції аргумент t замінюється на індекс першого входження цього символа в рядок за допомогою index(). Далі через print(…) та зрізи маємо відповідь.

Завдання повторюється постійно, поки користувач не введе щось, окрім ‘Y’ (можна з малої літери).

# Додаток А

## Текст програми

Зовнішній вигляд збережено для більш зручного читання

def parse(s, t):  
 if len(t) != 1: return -1  
 if s.count(t) == 0: return -1  
 t = s.index(t)  
 print(f'{s[:t]} |||| {s[t:]}')  
  
  
def choose\_season(season):  
 if season not in range(1, 5): return -1  
 print('<<< Виконання першого завдання >>>')  
 seasons = ['зима', 'весна', 'лето', 'осень']  
 months = [['декабрь', 'январь', 'февраль'],  
 ['март', 'апрель', 'май'],  
 ['июнь', 'июль', 'август'],  
 ['сентябрь', 'октябрь', 'ноябрь']]  
 days = [[31, 31, 28], [31, 30, 30], [30, 31, 31], [30, 31, 30]]  
 print(f'Ваш сезон - {seasons[season - 1]}\n'  
 f'Месяца: {", ".join(months[season - 1])}\n'  
 f'Количество дней: {days[season - 1]} (всего: {sum(days[season - 1])})')  
  
  
for i in range(1, 5): choose\_season(i)  
while True:  
 parse(input('\nВведіть текст: '), input('Введіть 1 символ, після якого розбити строку: '))  
 restart = input('Продолжить? Y/N\n')  
 if restart.lower() != 'y': break

# Додаток Б

## Скріншоти тестувань програми

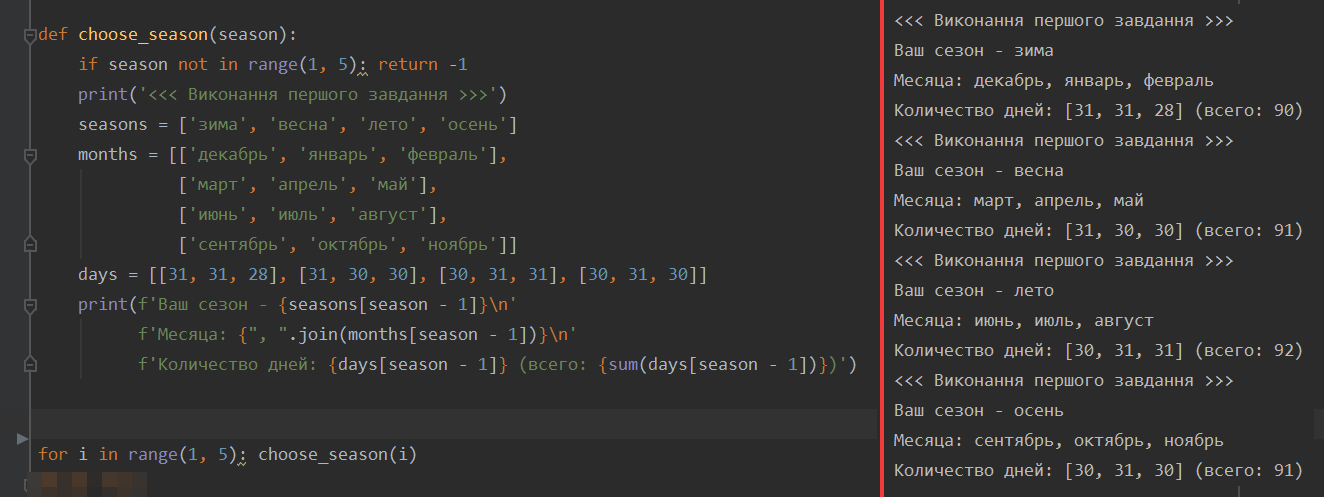


Рисунок 3.1 – Виконання завдання 1

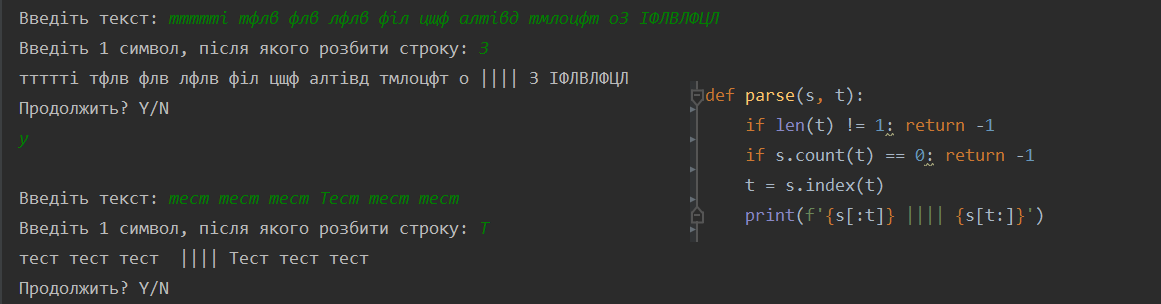


Рисунок 3.2 – Виконання завдання 2

# Додаток В

## Відповіді на питання для самоперевірки

1. Що являє собою об'єкт функція.

Це багаторазово використовувані фрагменти програми, які забезпечують кращу модульність програми та значно підвищують рівень повторного використання коду

2. Які правила створення функції.

Функція починається з def та містить параметри в середині круглих дужок (вони не обов’язкові), після них двокрапка і з нового рядка тіло.

3. Різниця між аргументами і змінними в функції.

Аргументи – це ті ж самі змінні, але передаються безпосередньо при виклику функції та виконують ту ж саму роль, що й інші змінні.

4. Що таке обов'язкові аргументи.

Це ті параметри, які повинні бути заповнені при виклику функції.

5. Що таке аргументи за умовчанням.

Це ті параметри, які не є обов’язковими для заповнення, бо за умовчанням вже мають певні аргументи (константи).

6. Що таке аргументи-ключові слова.

Це аргументи, які вказуються через назву параметра. Наприклад, sort(reverse=True): щоб не перераховувати попередні параметри, ми просто зразу вказали, що це значення для параметра reverse.

7. Що таке аргументи довільної довжини.

Дають змогу вводити безліч параметрів. \*param – всі аргументи будуть кортежем. \*\*param – всі аргументи вводяться як словник.

8. Області видимості змінних.

Область видимості всіх змінних обмежена блоком, в якому вони оголошені, починаючи з точки оголошення імені.

9. Як працює оператор return.

Використовується для повернення з функції, тобто для припинення її роботи і виходу з неї. При цьому можна також повернути деяке значення з функції.

10. Для чого використовується оператор *pass*.

Використовується для позначення порожнього блоку команд.

11. Що являє собою lambda-функція. Яке її застосування.

Безіменна функція з довільним числом аргументів і обчислює один вислів. Тіло такої функції не може містити більше однієї інструкції (або виразу). Дану функцію можна використовувати в рамках будь-яких конвеєрних обчислень.

*lambda-функцію* можна присвоїти будь-якій змінній і надалі використовувати її в якості імені функції.

Доцільно використовувати в місцях, де потрібно інкапсулювати характерний код, який не підлягає повторному використанню, без засмічення програми безліччю маленьких однорядкових функцій.