

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО”
Факультет прикладної математики
Кафедра прикладної математики

Звіт
із лабораторної роботи №1
із дисципліни “Програмування”
на тему
ВИКОРИСТАННЯ ФУНКЦІЙ

Виконав:
Студент групи КМ-83
Касіч Б.В.

Керівник:
ст.вик.
Дрозденко О.М.

Київ – 2018

1. ПОСТАНОВКА ЗАДАЧІ

1.1 Мета роботи

Вивчення написання і застосування функцій на мові Python

1.2 Теми для попереднього опрацювання:

- призначення і правила створення функцій
- аргументи і змінні
- типи аргументів при виклику функцій
- розміщення локальних і глобальних змінних
- повернення з функції (який може бути результат роботи)

1.3 Узагальнене формулювання 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.
У звіті обов'язково повинно бути описано, які типи аргументів передаються при виклику функції.
- 3) Вхідні дані і результат роботи супроводжувати відповідною інформацією на екрані.
- 4) Показати розроблену програму викладачеві.
- 5) Письмово відповісти на Питання для самоперевірки.
- 6) Оформити звіт відповідно до вимог.

Завдання на виконання лабораторної роботи складається з 2-х частин:

- функції на обчислення і логіку;
- функції для роботи з рядками.

2 ОПИС ПРОГРАМ

2.1 Перша програма

За допомогою команди `print` ми вітаємо користувача та надаємо йому інформацію про дану програму. Імпортуємо бібліотеку `re`, задля подальшої валідації даних. Створюємо нашу основну функцію `arithmetic(x,y,plus,minus,multiply,divide)`, в якій за допомогою конструкцій `if,elif,else`, циклу `while(x/y або y/x)` та методу `try-except` (потрібно при діленні на нуль) реалізуємо умову задачі. Параметри `x` та `y` – змінні які вводить користувач (два дійсних числа). `Plus`, `minus`, `multiply`, `divide` – операції додавання, віднімання, множення, ділення відповідно (1 2 3 4). Далі вводимо змінні `pattern_int` та `pattern_float` для подальшої валідації даних. Створюємо три функції для валідації першого числа, другого та значення номеру бажаної операції (`valid_x()`, `valid_y()`, `valid_d()` відповідно). Це реалізовується за допомогою циклу `while` та функції `match` із бібліотеки `re`. Далі безпосередньо присвоюємо значення `x`, `y` та `d` певним змінним і вставляємо їх у нашу функцію відповідно. Вводимо функцію через команду `print`, і користувачу виведеться результат певної операції.

Далі за допомогою команди `input` ми запитуємо у користувача, хоче він завершити програму чи ні. Якщо хоче, то він має написати `stop` і програма завершиться не почавши цикл `while`. В циклі ми так само запитуємо у користувача, чи хоче він змінити числа чи ні. Якщо хоче, то має ввести слово `switch`, якщо не хоче, то користувач буде мати змогу вибрати нову операцію поки не захоче припинити програму.

2.2 Друга програма

За допомогою команди `print` ми вітаємо користувача та надаємо йому інформацію про дану програму. Імпортуємо бібліотеку `re`, задля подальшої валідації даних. Створюємо основну функцію `Delete(s,n,l)`, яка написана в один рядок, а саме `return (s[:n] + s[l+n:])`. Параметр `s` – це саме речення користувача, `n` – початкова позиція підрядка, який потрібно видалити, `l` – довжина цього підрядка. Потім ми вводимо функцію `valid_position()`, щоб валідувати значення позиції `n`, функцію `valid_length()`, щоб валідувати значення довжини підрядка. Це реалізовується за допомогою циклу `while` та функції `match` із бібліотеки `re`, попередньо ввівши значення змінної `pattern_int`. Далі користувач безпосередньо вводить усі змінні, які нам потрібні і викликається сама функція з відповідними параметрами за допомогою команди `print`. Користувачу виводиться речення без обраного ним підрядка.

Далі за допомогою команди `input` ми запитуємо у користувача, хоче він завершити програму чи ні. Якщо хоче, то він має написати `stop` і програма завершиться не почавши цикл `while`. В циклі ми запитаємо у користувача чи хоче він змінити всі змінні, якщо ні, то він зможе сам обрати ті змінні, які захоче змінити ввівши певні слова які йому будуть представлені для введення. Це реалізовано за допомогою великої кількості конструкцій `if-elif-else`, також при зміні речення та не зміні позиції ми потребуємо від користувача, щоб довжина речення була не менше абсолютного значення позиції (як раніше ми потребували, щоб абсолютне значення позиції не було більше за довжину речення).

3. РЕЗУЛЬТАТИ ВИПРОБУВАНЬ

3.1 Усі можливі результати першої програми продемонстровано на Рисунок 3.1

```

=====
                        ЛАБОРАТОРНА РОБОТА №4
                ПРОГРАМУВАННЯ З ВИКОРИСТАННЯМ ФУНКЦІЙ
                        (ОБЧИСЛЕННЯ І ЛОГІКА)
                АВТОР: КАСІЧ БОГДАН, КМ-83 (6 ВАРІАНТ)
=====

Добрий день! Дана програма дозволить вам ввести два дійсних числа , а потім
ви зможете дізнатися їх суму, різницю, добуток і частку ввівши 1,2,3 або 4 відповідно.
Програма написана за допомогою функції ariphmetic(x,y,plus,minus,multiply,divide)

Design your float x: fwe
Please print correct data: 12
Design your float y: fewr
Please print correct data: 23
print '1' if u want to add numbers
print '2' if u want to subtract numbers
print '3' if u want to multiply numbers
print '4' if u want to divide numbers: 1
('Sum', 35.0)
print 'stop' if u want to stop the prog:
print 'switch' if u want to swtich the number:
print '1' if u want to add numbers
print '2' if u want to subtract numbers
print '3' if u want to multiply numbers
print '4' if u want to divide numbers: 2
('Difference', -11.0)
print 'stop' if u want to stop the prog:
print 'switch' if u want to swtich the number:
print '1' if u want to add numbers
print '2' if u want to subtract numbers
print '3' if u want to multiply numbers
print '4' if u want to divide numbers: 3
('Product', 276.0)
print 'stop' if u want to stop the prog:
print 'switch' if u want to swtich the number: switch
Design your float x: 4
Design your float y: 0
print '1' if u want to add numbers
print '2' if u want to subtract numbers
print '3' if u want to multiply numbers
print '4' if u want to divide numbers: 4
print '1' if u want to x/y
print '2' if u want to y/x: 1
Zero cannot be divided. Please don't do that
print 'stop' if u want to stop the prog: stop
>>> |

```

Рисунок 3.1

3.2 Усі можливі результати другої програми продемонстровано на Рисунок 3.2

```

=====
                        ЛАБОРАТОРНА РОБОТА №4
ПРОГРАМУВАННЯ З ВИКОРИСТАННЯМ ФУНКЦІЙ
(РОБОТА З РЯДКАМИ)
АВТОР: КАСІЧ ВОГДАН, КМ-83 (6 ВАРІАНТ)
=====

Добрий день! Дана програма дозволить вам ввести своє речення , а потім ви зможете
видалити звідти певний підрядок, який забажаєте. Програма зроблена за допомогою
функції Delete(s,n,l), де s - ваше речення , n - стартова позиція , l - довжина підрядка

Print your sentence: Hello World
Type integer starting position of your substring: 8321
value of position must not exceed the length of your sentence: -3128
value of position must not exceed the length of your sentence: 3
Type the positive integer length of your substring: -2
Print integer POSITIVE number please: jgewfrg
Print INTEGER POSITIVE number please: fwe
Print INTEGER POSITIVE number please: -3
Print integer POSITIVE number please: 5
Helrld
print 'stop' if u want to stop the prog:
print 'everything' if u want to change everything: everything
Print your sentence: hello
Type integer starting position of your substring: 2
Type the positive integer length of your substring: 2
heo
print 'stop' if u want to stop the prog:
print 'everything' if u want to change everything:
print 'length' if u want to change length:
print 'position' if u want to change position:
print 'sentence' if u want to change sentence: sentence
Print your sentence: ok
the length of your sentence must not be less than value of position: oko
ok
print 'stop' if u want to stop the prog: stop
>>> |

```

Рисунок 3.2

Текст першої програми:

```

print("""
=====
                ЛАБОРАТОРНА РОБОТА №4
    ПРОГРАМУВАННЯ З ВИКОРИСТАННЯМ ФУНКЦІЙ
    (ОВЧИСЛЕННЯ І ЛОГІКА)
    АВТОР: КАСІЧ БОГДАН, КМ-83 (6 ВАРІАНТ)
=====
Добрий день! Дана програма дозволить вам ввести два дійсних числа , а потім
ви зможете дізнатися їх суму,різницю,добуток і частку ввівши 1,2,3 або 4 відповідно.
Програма написана за допомогою функції ariphmetic(x,y,plus,minus,multiply,divide)
""")

import re

def ariphmetic(x,y,plus,minus,multiply,divide):
    if plus=="1":
        return "Sum",x+y
    elif minus=="2":
        return "Difference",x-y
    elif multiply=="3":
        return "Product",x*y
    else:
        divide=input("print '1' if u want to x/y \nprint '2' if u want to y/x: ")
        while divide!="1" and divide!="2":
            divide=input("Please print correct data: ")
        if divide=="1":
            try:
                return "Quotient(x/y)",x/y
            except ZeroDivisionError:
                return "Zero cannot be divided. Please don't do that"
        else:
            try:
                return "Quotient(y/x)",y/x
            except ZeroDivisionError:
                return "Zero cannot be divided. Please don't do that"

pattern_int = r"^[-\d]\d*$"
pattern_float = r"^[-\d]\d*\.\d*$"
def valid_x():
    a=input("Design your float x: ")
    while (not re.match(pattern_float,a) and not re.match(pattern_int,a)):
        a = input("Please print correct data: ")
    a=float(a)
    return a

def valid_y():
    b=input("Design your float y: ")
    while (not re.match(pattern_float,b) and not re.match(pattern_int,b)):
        b = input("Please print correct data: ")
    b=float(b)
    return b

def valid_d():
    d=input("print '1' if u want to add numbers \nprint '2' if u want to substract numbers \nprint '3' if u want to multiply numbers \nprint '4' if u want to divide numbers: ")
    while (d!="1" and d!="2" and d!="3" and d!="4"):
        d = input("Please print correct data: ")
    return d

a=valid_x()
b=valid_y()
d=valid_d()
print(arihmetic(a,b,d,d,d,d))

answer=input("print 'stop' if u want to stop the prog: ")
while answer!="stop":
    switch=input("print 'switch' if u want to swtich the number: ")
    if switch=="switch":
        a=valid_x()
        b=valid_y()
        d=valid_d()
        print(arihmetic(a,b,d,d,d,d))
        answer=input("print 'stop' if u want to stop the prog: ")
        continue
    else:
        d=valid_d()
        print(arihmetic(a,b,d,d,d,d))
        answer=input("print 'stop' if u want to stop the prog: ")
        continue

```

Текст другої програми:

```
print("""
=====
                        ЛАБОРАТОРНА РОБОТА №4
                ПРОГРАМУВАННЯ З ВИКОРИСТАННЯМ ФУНКЦІЙ
                        (РОБОТА З РЯДКАМИ)
                АВТОР: КАСІЧ БОГДАН, КМ-83 (6 ВАРІАНТ)
=====
Добрий день! Дана програма дозволить вам ввести своє речення , а потім ви зможете
видалити звідти певний підрядок, який забажаєте. Програма зроблена за допомогою
функції Delete(s,n,l), де s - ваше речення , n - стартова позиція , l - довжина підрядка
""")

import re
def Delete(s,n,l):
    return (s[:n] + s[l+n:])

def valid_position():
    position = input("Type integer starting position of your substring: ")
    while not re.match(pattern_int,position):
        position = input("Type integer only please: ")
    position=int(position)
    while position >= len(sentence) or abs(position) > len(sentence):
        position = input("value of position must not exceed the length of your sentence: ")
        while not re.match(pattern_int,position):
            position = input("Type integer only please: ")
    position=int(position)

    return position

def valid_length():
    length = input("Type the positive integer length of your substring: ")
    while not re.match(pattern_int,length):
        length=input("Print INTEGER POSITIVE number please: ")
    length=int(length)
    while length <= 0:
        length=input("Print integer POSITIVE number please: ")
        while not re.match(pattern_int,length):
            length=input("Print INTEGER POSITIVE number please: ")
        length=int(length)
    return length

|

pattern_int = r"^[-\d]\d*$"
```



```

sentence=input("Print your sentence: ")
position= valid_position()
length= valid_length()
print(Delete(sentence,position,length))

answer=input("print 'stop' if u want to stop the prog: ")
while answer!="stop":
    change_everything=input("print 'everything' if u want to change everything: ")
    if change_everything=="everything":
        sentence=input("Print your sentence: ")
        position= valid_position()
        length= valid_length()
        print(Delete(sentence,position,length))
        answer=input("print 'stop' if u want to stop the prog: ")
    else:
        change_length=input("print 'length' if u want to change length: ")
        change_position=input("print 'position' if u want to change position: ")
        change_sentence=input("print 'sentence' if u want to change sentence: ")
        if change_length=="length" and change_position=="position" and change_sentence=="sentence":
            print("You could achieve the same just by typing 'everything' earlier...")
            sentence=input("Print your sentence: ")
            position= valid_position()
            length= valid_length()
            print(Delete(sentence,position,length))
            answer=input("print 'stop' if u want to stop the prog: ")
        elif change_length=="length" and change_position=="position" and change_sentence!="sentence":
            position= valid_position()
            length= valid_length()
            print(Delete(sentence,position,length))
            answer=input("print 'stop' if u want to stop the prog: ")
        elif change_length!="length" and change_position=="position" and change_sentence=="sentence":
            sentence=input("Print your sentence: ")
            position= valid_position()
            print(Delete(sentence,position,length))
            answer=input("print 'stop' if u want to stop the prog: ")
        elif change_length!="length" and change_position=="position" and change_sentence!="sentence":
            position= valid_position()
            print(Delete(sentence,position,length))
            answer=input("print 'stop' if u want to stop the prog: ")
        elif change_length=="length" and change_position!="position" and change_sentence=="sentence":
            sentence=input("Print your sentence: ")
            while len(sentence) <= position or len(sentence) < abs(position):
                sentence=input("the length of your sentence must not be less than value of position: ")
            length= valid_length()
            print(Delete(sentence,position,length))
            answer=input("print 'stop' if u want to stop the prog: ")
        elif change_length=="length" and change_position!="position" and change_sentence!="sentence":
            position= valid_length()
            print(Delete(sentence,position,length))
            answer=input("print 'stop' if u want to stop the prog: ")
        elif change_length!="length" and change_position!="position" and change_sentence=="sentence":
            sentence=input("Print your sentence: ")
            while len(sentence) <= position or len(sentence) < abs(position):
                sentence=input("the length of your sentence must not be less than value of position: ")
            print(Delete(sentence,position,length))
            answer=input("print 'stop' if u want to stop the prog: ")
    else:
        answer=input("You have not changed anything, which means that the result will be the same. \nBut if you want to stop the prog print 'stop': ")

```

Відповіді до контрольних запитань

1. Що являє собою об'єкт функція?

Функції - це такі ділянки коду, які ізольовані від решти програми і виконуються тільки тоді, коли викликаються.

2. Які правила створення функції?

```
def func(x,y,...):
```

```
    body
```

```
    return something --- може повернути саму функцію(рекурсія)
```

3. Різниця між аргументами і змінними в функції.

Термін «аргумент» має на увазі, що конкретно і до якої конкретної функції було передано. А змінна є сталою по відношенню до функції.

4. Що таке обов'язкові аргументи?

Якщо при створенні функції ми вказали кількість переданих їй аргументів і їх порядок, то і викликати її ми повинні з тією ж кількістю аргументів, заданих в потрібному порядку.

5. Що таке аргументи за умовчанням?

Аргумент за замовчуванням, це аргумент, значення для якого задано спочатку, при створенні функції.

6. Що таке аргументи-ключові слова?

Аргументи - ключові слова використовуються при виконанні функції. Завдяки ключовим аргументів, ви можете задавати довільний (тобто не такий яким він описаний, при створенні функції) порядок аргументів.

7. Що таке аргументи довільної довжини?

Іноді виникає ситуація, коли ви заздалегідь не знаєте, яка кількість аргументів буде необхідно прийняти функції. В цьому випадку слід використовувати аргументи довільної довжини. Вони задаються довільним ім'ям змінної, перед якою ставиться зірочка (*).

8. Області видимості змінних.

-Глобальні змінні

-Локальні змінні

9. Як працює оператор `return`?

Він повертає певне значення (`return something`) , якщо `return` не буде , то функція поверне `None`.

10. Для чого використовується оператор `pass`?

Найчастіше `pass` використовується там, де код поки ще не з'явився, але планується. Крім цього, іноді, його використовують при налагодженні, розмістивши на сходінці з ним точку зупинки.

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

Анонімні функції можуть містити лише один вислів, а й виконуються вони швидше. Анонімні функції створюються за допомогою інструкції `lambda`. Крім цього, їх не обов'язково привласнювати змінній. `lambda` функції, на відміну від звичайної, не потрібно інструкція `return`.

ЗМІСТ

1 ПОСТАНОВКА ЗАДАЧІ.....	2
2 ОПИС ПРОГРАМ.....	3-4
Результати випробувань.....	5-6
Тексти програм.....	7-9
Відповіді до контрольних запитань.....	10-11