Лабораторная работа № 11
«Организация функций»

1 Цель работы: Приобретение навыка создания функций на Python.

2 Литература:

2.1 Максимов, Н. В. Архитектура ЭВМ и вычислительных систем : учебник / Н. В. Максимов, Т. Л. Партыка, И. И. Попов. – 5-е изд., перераб. и доп. – Москва: ФОРУМ : ИНФРА-М, 2018. – 511 с. – URL: https://znanium.com/catalog/product/944312. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.

3 Подготовка к работе:

3.1 Изучить теоретический материал (см. п. 2),

3.2 Подготовить бланк отчета.

4 Оборудование:

4.1 Персональный компьютер,

4.2 Лазерный принтер.

5 Задание:

5.1 Ознакомиться с теоретическим материалом.

5.2 Выполнить задания из пункта 6.

5.3 Ответить на контрольные вопросы.

6 Порядок выполнения работы:

6.1 Написать и протестировать функцию, возводящую переданное в параметрах число a в степень x. По умолчанию a должно быть равно 2. При реализации можно использовать **. Функцию снабдить комментариями согласно следующему шаблону (описание параметров дано в сигнатуре функции): def названиеФункции(параметр1: ‘описание параметра 1’ = значение1, параметр2: ‘описание параметра 2’ = значение2): """Комментарий к функции """ тело функции

6.2 Разработать алгоритм для предложенной задачи: написать функцию is_year_leap, принимающую 1 аргумент — год, и возвращающую True, если год високосный, и False иначе. Реализовать построенный алгоритм в виде программы на Python.

6.3 Разработать алгоритм для предложенной задачи: написать функцию season, принимающую 1 аргумент — номер месяца (от 1 до 12), и возвращающую время года, которому этот месяц принадлежит (зима, весна, лето или осень). Реализовать построенный алгоритм в виде программы на Python.

6.4 Написать и протестировать функцию, изменяющую значения переданного в нее списка путем умножения каждого из элементов списка на переданное в параметрах число. Если второй параметр отсутствует, умножать на -1.

7 Контрольные вопросы:

7.1 Какие способы передачи параметров в функцию существуют в Python?
7.2 В какой части программы можно объявлять функции?
7.3 Как задать параметры по умолчанию в функциях на Python?
7.4 Что такое «рекурсия»?
7.5 Какие проблемы могут возникать при реализации рекурсивных алгоритмов на электронных вычислительных машинах?
7.6 В каких случаях оправдано применение рекурсивных функций

8 Приложение:

Синтаксис создания функции:

```
def имя_функции (аргументы — необязательно):
    #тело функции
#остальной код
```

Пример функции:

```
def sum (a, b):
   print("Сумма a и b =", a + b)
#данная функция складывает два числа и вывод результат в консоль
sum(3, 8)#вызов функции
a = 7
b = 5
sum(a, b)#тоже вызов функции
```

Возврат значения:

```
def sum(a, b)
    return a + b
#функция возвращает сумму двух чисел
print(sum( 4, 8)) #вывод результата сложения в консоль
```

Значения по умолчанию (необязательные аргументы):

```
def sum( a = 9, b):
    return a +b
#функция возвращает сумму двух чисел
#если не будет указан a, ему присвоится значение по умолчанию
print(sum(8, 5)# выведет 13
print(sum(2)#выведет 7
```

Необязательные аргументы должны идти впереди обязательных!

```
Enter name: Maloman
Enter age: 89
Name:  Maloman
Age:  89
Name:  Jane Doe
Age:  0
Name:  Julia
Age:  0
>>>
```

```
def person(name = "Jane Doe", age = 0):
    print("Name: ", name)
    print("Age: ", age)

name = input("Enter name: ")
age = int(input("Enter age: "))
person(name, age)
person()
person("Julia")
```