**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ ИМЕНИ П.О.СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

Дисциплина «скриптованные языки программирования»

ОТЧЕТ

Лабораторная работа № 3

Основы определения и использования функций

Выполнил студент

Группы ИТД-21

Чайдаков И.М.

Принял преподаватель

Малиновский И. Л.

Гомель 2024

Основы определения и использования функций

**Цель работы:** научиться основам работы с функциями, в том числе с рекурсивными и анонимными; познакомиться с механизмом замыканий функций и декораторами функций

**Задание 1:** Объявить функцию, которая проверяет четность введенного числа, после объявления функции в цикле необходимо считывать целое числовое значение пока не поступит число 1. Четные числа вывести на экран.

**Ход выполнения задания:**

**1.1** Описываем функцию командой def...return и создаем цикл в котором эта функция будет проверять все введенные числа (рисунок 1).



Рисунок 1 — Код копирует лишь четные значения.

**Задание 2:** Вводится список целых чисел в одну строку через пробел. Необходимо вычислить сумму веденных значений используя рекурсивную функцию. Результат вывести на экране.

**Ход выполнения задания:**

**2.1** Описываем функцию для вычисления суммы которая будет вызывать себя вновь до тех пор пока не дойдет до конца списка. После чего преобразуем введенную строку в список чисел, передаем его функции и выводим результат на экран (рисунок 2).



Рисунок 2 — Суммирование списка значений.

**Задание 3:** Используя замыкания функций, объявите внутреннюю функцию которая заключает строку в произвольный тег, содержащийся в параметре внешней функции *tag*.

**Ход выполнения задания:**

**3.1** Создаем функцию которая будет возвращать другую функцию создав замыкание функций, таким образом мы можем сначала определить тег а потом заключить в него любой текст (рисунок 3).

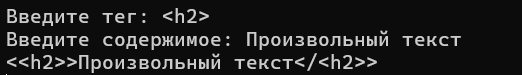


Рисунок 3 — Python-код

**Задание 4:** На вход программы поступает строка из целых чисел, записанных через пробел. Описать функцию *get*\_*list*, которая преобразует строку в список из целых чисел и вернет его. Определить декоратор который дополнительно отсортирует значения возвращаемого списка.

**Ход выполнения задания:**

**1.** Описываем функцию для трансформации строки в список и декоратор который будет изменять порядок чисел в списке после чего выводить его на экран (рисунок 4).



Рисунок 4 — Вывод сортированного списка.

**Вывод:** Функции открывают доступ к многоразовому вызова кода что значительно упрощает читаемость кода и позволяет решать более сложные задачи на языке *Python*.

**ПРИЛОЖЕНИЕ А**

**Текст программ**

**Задание 1:**

'''

Объявить функцию, которая проверяет четность введенного числа,

после объявления функции в цикле необходимо считывать целое числовое значение

пока не поступит число 1. Четные числа вывести на экран.

'''

def numberEven(number):

    return number % 2 == 0

number = 0

while (number != 1):

    number = int(input())

    if numberEven(number):

        print(number)

**Задание 2:**

'''

Вводится список целых чисел в одну строку через пробел.

Необходимо вычислить сумму веденных значений используя рекурсивную функцию.

Результат вывести на экране.

'''

def listSum(numbers):

    if not numbers:

        return 0

    return numbers[0] + listSum(numbers[1:])

numbers = list(map(int, input("Введите список чисел через пробел: ").split()))

result = listSum(numbers)

print("Сумма чисел списка:", result)

input()

**Задание 3:**

'''

Используя замыкания функций, объявите внутреннюю функцию которая

заключает строку в произвольный тег,

содержащийся в параметре внешней функции tag.

'''

def tagSave(tag):

    def textTag(s):

        return f"<{tag}>{s}</{tag}>"

    return textTag

tag = input("Введите тег: ")

text = input("Введите содержимое: ")

textTag = tagSave(tag)

print(textTag(text))

input()

**Задание 4:**

'''

На вход программы поступает строка из целых чисел, записанных через пробел.

Описать функцию get\_list, которая преобразует строку в список из целых чисел

и вернет его. Определить декоратор который

дополнительно отсортирует значения возвращаемого списка.

'''

def sort(func):

    def newList(numbers\_str):

        numbersList = func(numberString)

        return sorted(numbersList)

    return newList

@sort

def get\_list(numberString):

    # Преобразуем строку в список целых чисел

    return list(map(int, numberString.split()))

numberString = input("Введите числа через пробел: ")

list = get\_list(numberString)

print("Отсортированный список:", list)

input()