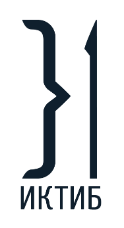
МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное образовательное учреждения высшего образования

«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт компьютерных технологий и информационной безопасности

Кафедра математического обеспечения и применения ЭВМ



**ЛАБОРАТОРНАЯ РАБОТА № 3**

**ОТЧЁТ**

по дисциплине

**«Разработка прикладных решений на языке Python»**

Выполнил:

Студент группы

КТмо2-16

|  |  |  |
| --- | --- | --- |
|  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Руденко К. Д. |
|  | *подпись* |  |

Проверил:

Доцент кафедры САиТ, к.т.н.

|  |  |  |
| --- | --- | --- |
|  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Лапшин В. С. |
|  | *подпись* |  |

Оценка

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2025 г.

**3 лабораторная работа.**

**Руденко Константин Дмитриевич – 5 вариант**

**Вариант 5 :**

**Теоретический вопрос :**

1. Побочные эффекты в программировании, основные правила при создании функций.

Побочный эффект – это когда при вызове функции изменяются любые объекты, которые изменяют нашу программу (или ее состояние). То есть все кроме возвращения значения - return object.

Примеры побочных эффектов:

Изменение глобальных переменных

Изменение переданных mutable-объектов (списков, словарей)

Ввод/вывод данных (print, input, работа с файлами)

Сетевые запросы

Изменение базы данных

Правил при создании функции:

1. Принцип единственной ответственности (Single Responsibility) – то есть функция должна делать только одну вещь
2. Функции должны быть предсказуемыми и не менять внешнее состояние.
3. Имя функции должно четко описывать ее назначение
4. Количество параметров оптимально - 3-4 параметра. Для большего количества используйте \*\*kwargs или объекты
5. Возвращать осмысленные значения
6. Обрабатывайть ошибки корректно
7. Документировать функцию
8. Избегать глобальных переменных

**Задание 1 :**

Изображение выглядит как текст, снимок экрана, Шрифт, белый

Автоматически созданное описание

# Задача 1  
  
def draw\_frame(s, k):  
 *"""  
 Рисует рамку из символа k вокруг строки s  
  
 Args:  
 s (str): строка для обрамления  
 k (str): символ для рамки  
 """* if len(k) != 1:  
 raise ValueError(f"Символ рамки должен быть длиной 1, получено: '{k}' (длина: {len(k)})")  
 # Верхняя и нижняя линии рамки - длина строки + 2 символа по бокам  
 top\_bottom = k \* (len(s) + 2)  
  
 # Средняя линия с текстом  
 middle = f"{k}{s}{k}"  
  
 # Вывод рамки  
 print(top\_bottom)  
 print(middle)  
 print(top\_bottom)  
  
try:  
 s = input("Введите строку которая будет выведена в рамке \"s\": ")  
 k = input("Введите символ для рамки \"k\": ")  
 draw\_frame(s, k)  
except ValueError as e:  
 print(f"Ошибка: {e}")  
except Exception as e:  
 print(f"Произошла непредвиденная ошибка: {e}")  
  
# C:\Users\korudenko\PycharmProjects\botconfectioner\.venv2\Scripts\python.exe C:\Users\korudenko\PycharmProjects\Py3semestr2025RudenkoKTmo2-16\laba3\Task1-1.py  
# Введите строку которая будет выведена в рамке "s": Текст в рамке  
# Введите символ для рамки "k": \*  
# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
# \*Текст в рамке\*  
# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
#  
# Process finished with exit code 0  
  
  
# C:\Users\korudenko\PycharmProjects\botconfectioner\.venv2\Scripts\python.exe C:\Users\korudenko\PycharmProjects\Py3semestr2025RudenkoKTmo2-16\laba3\Task1-1.py  
# Введите строку которая будет выведена в рамке "s": Хочу получить баллов за лабу  
# Введите символ для рамки "k": 6  
# 666666666666666666666666666666  
# 6Хочу получить баллов за лабу6  
# 666666666666666666666666666666  
#  
# Process finished with exit code 0  
  
  
# C:\Users\korudenko\PycharmProjects\botconfectioner\.venv2\Scripts\python.exe C:\Users\korudenko\PycharmProjects\Py3semestr2025RudenkoKTmo2-16\laba3\Task1-1.py   
# Введите строку которая будет выведена в рамке "s": Тест  
# Введите символ для рамки "k": плохой  
# Ошибка: Символ рамки должен быть длиной 1, получено: 'плохой' (длина: 6)  
#   
# Process finished with exit code 0

**Задание 2:**

**Изображение выглядит как текст, Шрифт, алгебра, чек

Автоматически созданное описание**

# Задача 2  
import random  
  
def split\_and\_sort\_numbers(\*args):  
 return (sorted([x for x in args if x < 0], reverse=True),  
 sorted([x for x in args if x >= 0]))  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 print(split\_and\_sort\_numbers(\*[random.randint(-100, 100) for x in range(20)]))  
  
  
# C:\Users\korudenko\PycharmProjects\botconfectioner\.venv2\Scripts\python.exe C:\Users\korudenko\PycharmProjects\Py3semestr2025RudenkoKTmo2-16\laba3\Task3-2.py  
# ([-4, -9, -16, -20, -33, -36, -57, -78, -87, -87, -88], [21, 35, 59, 67, 68, 74, 80, 83, 83])  
#  
# Process finished with exit code 0  
  
# C:\Users\korudenko\PycharmProjects\botconfectioner\.venv2\Scripts\python.exe C:\Users\korudenko\PycharmProjects\Py3semestr2025RudenkoKTmo2-16\laba3\Task3-2.py  
# ([-18, -19, -34, -54, -62, -84], [11, 15, 17, 22, 22, 46, 48, 57, 60, 62, 65, 80, 91, 98])  
#  
# Process finished with exit code 0  
  
# C:\Users\korudenko\PycharmProjects\botconfectioner\.venv2\Scripts\python.exe C:\Users\korudenko\PycharmProjects\Py3semestr2025RudenkoKTmo2-16\laba3\Task3-2.py   
# ([-7, -8, -8, -19, -23, -44, -46, -64, -87, -88], [5, 6, 15, 25, 39, 55, 85, 91, 93, 99])  
#   
# Process finished with exit code 0