

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ
ФЕДЕРАЦИИ**
**Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

Институт цифрового развития

ОТЧЁТ

по лабораторной работе №2.8

Дисциплина: «Программирование на Python»

Тема: «Работа с функциями в языке Python»

Выполнил: студент 2 курса

группы ИВТ-б-о-21-1

Пуценко Иван Алексеевич

Ставрополь 2022

Выполнение работы:

1. Создал репозиторий в GitHub «rep 2.6» в который добавил .gitignore, который дополнил правила для работы с IDE PyCharm с ЯП Python, выбрал лицензию MIT, клонировал его на лок. сервер и организовал в соответствии с моделью ветвления git-flow.

```
2. #!/usr/bin/env python3
# -*- coding: utf-8 -*-

def exit_to_program():
    print('всего доброго')
    return exit(1)

def help_program():
    print("add - добавление рейса\n"
          "help - помощь по командам\n"
          "select \"пункт назначения\" - вывод самолетов летящих в п.н.\n"
          "exit - выход из программы")

def add_program(planes):
    plane = dict()
    plane["destination"] = input("Пункт назначения:\n")
    plane["flight_number"] = int(input("Номер рейса:\n"))
    plane["type_plane"] = input("Тип самолета\n")
    planes.append(plane)
    planes.sort(key=lambda key_plane: key_plane.get("flight_number"))
    return planes

def select_program(planes):
    lst = list(map(lambda x: x.get("destination"), planes))
    point = input('выберите нужное вам место\n')
    print("результаты поиска")
    if point in lst:
        print('рейсы в эту точку')
        for i in planes:
            if point == i["destination"]:
                print(f"{i['flight_number']}.....{i['type_plane']}")
    else:
        print("рейсов не найдено")

def plane_print(planes):
    for i in planes:
        print(f"куда - {i['destination']} номер - {i['flight_number']} самолет - {i['type_plane']}")

def error():
    print('неверная команда')

def menu(lst_plane):
    command = input('введите команду("help" - руководство по командам)\n>>>').lower()
    if command == 'exit':
        exit_to_program()
    elif command == 'help':
```

```
        help_program()
    elif command == 'add':
        lst_plane = add_program(lst_plane)
    elif command == 'select':
        select_program(lst_plane)
    elif command == 'plane_print':
        plane_print(lst_plane)
    else:
        error()

if __name__ == '__main__':
    lst_planes = list()

    while True:
        menu(lst_planes)
```

Рисунок 1.1- листинг программы индивидуального задания

```

Тип самолета
zxcv
введите команду("help" - руководство по командам)
>>>add
Пункт назначения:
fghj
Номер рейса:
1
Тип самолета
dfgh
введите команду("help" - руководство по командам)
>>>help
add - добавление рейса
help - помощь по командам
select "пункт назначения" - вывод самолетов летящих в п.н.
exit - выход из программы
введите команду("help" - руководство по командам)
>>>plane_print
куда - fghj номер - 1 самолет - dfgh
куда - qwer номер - 3 самолет - zxcv
куда - qwer номер - 5 самолет - asdf
введите команду("help" - руководство по командам)
>>>select
выберите нужное вам место
qwer
результаты поиска
рейсы в эту точку
3.....zxcv
5.....asdf
введите команду("help" - руководство по командам)

```

Рисунок 1.2- вывод программы индивидуального задания

Контр. вопросы и ответы на них:

Ответы на вопросы:

1. Каково назначение функций в языке программирования Python?

Главной задачей функций в Python, как и в других языках

программирования, является сокращение объёма кода и его структуризация. В функции, как правило, выносятся те части кода, которые выполняются в программе многократно.

2. Каково назначение операторов `def` и `return`?

Оператор `def` необходим для определения функции. После него идёт название самой функции, передаваемые в функцию параметры и само тело функции. Оператор `return` служит для возвращения результата выполнения функции в основную программу, где эта функция была вызвана.

3. Каково назначение локальных и глобальных переменных при написании функций Python?

Локальные переменные существуют только внутри функции. В другой части программы как-либо вызывать или изменить их невозможно.

Глобальные напротив – существуют во всей программе.

4. Как вернуть несколько значений из функции Python?

После оператора `return` необходимо записать все возвращаемые переменные через запятую, а при вызове функции нужно задать необходимое количество переменных. Куда будут возвращены параметры.

5. Какие существуют способы передачи значений в функцию?

По ссылке и по значению.

6. Как задать значение аргументов функции по умолчанию?

Нужно в скобках передаваемых параметров присвоить им значение.

7. Каково назначение `lambda`-выражений в языке Python?

`Lambda`-выражения – это небольшие функции, которые вызываются в программе один раз.

8. Как осуществляется документирование кода согласно PEP257?

Если пояснение функции содержит одну строку, то достаточно двух кавычек с каждой стороны строки. Пример: `"""Пояснение"""`. Если это многострочное пояснение, то необходимо три кавычки с каждой стороны. Пояснение находится в теле функции, сразу после её объявления.