

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития  
Кафедра инфокоммуникаций

**ОТЧЕТ**  
**ПО ЛАБОРАТОРНОЙ РАБОТЕ №2.13**  
**дисциплины «Программирование на Python»**

Выполнил:  
Дзуев Альберт Мухаметович  
2 курс, группа ИВТ-б-о-22-1,  
09.03.01 «Информатика и  
вычислительная техника»,  
направленность (профиль)  
«Программное обеспечение средств  
вычислительной техники и  
автоматизированных систем», очная  
форма обучения

---

(подпись)

Руководитель практики:  
Воронкин Р.А., доцент кафедры  
инфокоммуникаций

---

(подпись)

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

Ставрополь, 2023 г.

**Тема:** Модули и пакеты

**Цель:** приобретение навыков по работе с модулями и пакетами языка программирования Python версии 3.x.

**Порядок выполнения работы:**

1. Создал новый репозиторий, клонировал его, в нем создал ветку developer и перешел на нее.
2. Выполнить индивидуальное задание лабораторной работы 2.11, оформив все функции программы в виде отдельного модуля. Разработанный модуль должен быть подключен в основную программу с помощью одного из вариантов команды import. Номер варианта уточнить у преподавателя.

Используя замыкания функций, объявите внутреннюю функцию, которая принимает в качестве аргумента коллекцию (список или кортеж) и возвращает или минимальное значение, или максимальное, в зависимости от значения параметра type внешней функции. Если type равен «max», то возвращается максимальное значение, иначе - минимальное. По умолчанию type должно принимать значение «max». Вызовите внутреннюю функцию замыкания и отобразите на экране результат ее работы.

**Код модуля ind1\_m.py:**

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
def max_min(type='max'):
    def helper(collection):
        if type == 'max':
            return max(collection)
        else:
            return min(collection)
    return helper
```

**Код программы ind1.py:**

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
from ind1_m import max_min
if __name__ == '__main__':
    print(max_min(input("Введите max или min: ")))
    (map(int, input("Введите список: ").split(' ')))
```

```
Введите max или min: max
Введите список: 1 -3 23 5 0 -11 -9
23
```

Рисунок 1. Вывод программы ind1.py

3. Выполнить индивидуальное задание лабораторной работы 2.8, оформив все классы программы в виде отдельного пакета. Разработанный пакет должен быть подключен в основную программу с помощью одного из вариантов команды `import`. Настроить соответствующим образом переменную `_all` в файле `__init__.py` пакета.

Использовать словарь, содержащий следующие ключи: название начального пункта маршрута; название конечного пункта маршрута; номер маршрута. Написать программу, выполняющую следующие действия: ввод с клавиатуры данных в список, состоящий из словарей заданной структуры; записи должны быть упорядочены по номерам маршрутов, вывод на экран информации о маршруте, номер которого введен с клавиатуры; если таких маршрутов нет, выдать на дисплей соответствующее сообщение.

#### Содержание пакета:

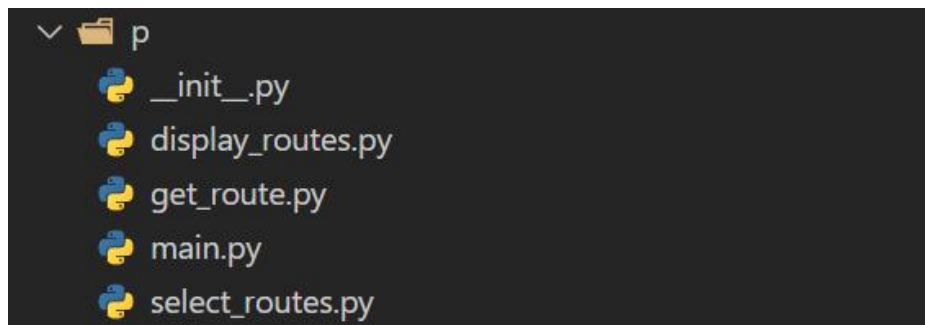


Рисунок 2. Структура пакета

#### Код модуля get route.py:

```
def get_route():
    """
    Запросить данные о маршруте.
    """
    start = input("Введите начальный пункт: ")
    end = input("Введите конечный пункт: ")
    count = int(input("Введите номер маршрута: "))
```

```

return {
'начальный пункт': start,
'конечный пункт': end,
'номер маршрута': count
}

```

#### **Код модуля display\_routes.py:**

```

def display_routes(routes):
    """
    Отобразить список маршрутов.
    """
    if routes:
        line = '+-{}-+-{}-+-{}-+'.format(
            '-' * 30,
            '-' * 20,
            '-' * 8)
        print(line)
        print(
            '| {:^30} | {:^20} | {:^8} |'.format(
                "Начало",
                "Конец",
                "Номер"))
        print(line)
        for route in routes:
            print(
                '| {:<30} | {:<20} | {:>8} |'.format(
                    route.get('начальный пункт', ''),
                    route.get('конечный пункт', ''),
                    route.get('номер маршрута', '')))
            print(line)
        else:
            print("Список маршрутов пуст.")

```

#### **Код модуля select\_routes.py:**

```

def select_routes(routes, name_punct):
    """
    Выбрать маршруты с заданным пунктом отправления или прибытия.
    """
    selected = []
    for route in routes:
        if str(route['номер маршрута']) == name_punct:
            selected.append(route)
    return selected

```

#### **Код модуля main.py:**

```

import sys
import bisect
import re
from p.select_routes import select_routes
from p.get_route import get_route
from p.display_routes import display_routes
def main():
    """
    Главная функция программы.
    """
    routes = []
    while True:
        command = input(">>> ").lower()
        if command == 'exit':
            break
        elif command == 'add':
            route = get_route()
            if route not in routes:
                bisect.insort(
                    routes, route, key=lambda item: item.get('номер маршрута'))
            else:
                print("Данный маршрут уже добавлен.")
        elif command == 'list':
            display_routes(routes)
        elif (m := re.match(r'select (.+)', command)):
            name_punct = m.group(1)
            selected = select_routes(routes, name_punct)
            display_routes(selected)
        elif command == 'help':
            print("Список команд:")
            print("add - добавить маршрут;")
            print("list - вывести список маршрутов;")
            print("select <номер маршрута> - запросить маршруты, которые имеют  
данный номер")
            print("help - отобразить справку;")
            print("exit - завершить работу с программой.")
        else:
            print(f"Неизвестная команда {command}", file=sys.stderr)

```

### **Код основной программы:**

```

from p.main import main
if __name__ == '__main__':
    main()

```

```

>>> help
Список команд:
add - добавить маршрут;
list - вывести список маршрутов;
select <номер маршрута> - запросить маршруты, которые имеют данный номер
help - отобразить справку;
exit - завершить работу с программой.
>>> add
Введите начальный пункт: Ставрополь
Введите конечный пункт: Моздок
Введите номер маршрута: 10
>>> add
Введите начальный пункт: Моздок
Введите конечный пункт: Грозный
Введите номер маршрута: 12
>>> add
Введите начальный пункт: Моздок
Введите конечный пункт: Ставрополь
Введите номер маршрута: 10
>>> list
+-----+-----+-----+
|          Начало          |          Конец          |          Номер          |
+-----+-----+-----+
|  Ставрополь             |  Моздок                 |             10          |
|  Моздок                 |  Ставрополь             |             10          |
|  Моздок                 |  Грозный                 |             12          |
+-----+-----+-----+
>>> select 10
+-----+-----+-----+
|          Начало          |          Конец          |          Номер          |
+-----+-----+-----+
|  Ставрополь             |  Моздок                 |             10          |
|  Моздок                 |  Ставрополь             |             10          |
+-----+-----+-----+

```

Рисунок 3. Результат выполнения программы

### Ответы на контрольные вопросы:

#### 1. Что является модулем языка Python?

Под модулем в Python понимается файл с расширением .py. Модули предназначены для того, чтобы в них хранить часто используемые функции, классы, константы и т. п. Можно условно разделить модули и программы: программы предназначены для непосредственного запуска, а модули для импортирования их в другие программы.

#### 2. Какие существуют способы подключения модулей в языке Python?

```
import имя_модуля; import имя_модуля1, имя_модуля2;
```

```
import имя_модуля as новое_имя;
```

```
from имя_модуля import имя_объекта;
```

```
from имя_модуля import имя_объекта1, имя_объекта2;
```

```
from имя_модуля import имя_объекта as псевдоним_объекта;
```

`from имя_модуля import*`

3. Что является пакетом языка Python?

Пакет в Python – это каталог, включающий в себя другие каталоги и модули, но при этом дополнительно содержащий файл `__init__.py`. Пакеты используются для формирования пространства имен, что позволяет работать с модулями через указание уровня вложенности (через точку).

4. Каково назначение файла `__init__.py`?

Файл `__init__.py` может быть пустым или может содержать переменную `__all__`, хранящую список модулей, который импортируется при загрузке через конструкцию

5. Каково назначение переменной `__all__` файла `__init__.py`?

В переменную `__all__` вносятся названия модулей, которые импортируются при `from имя_пакета import *`.

**Вывод:** в результате выполнения работы были приобретены навыки по работе с модулями и пакетами языка программирования Python версии 3х.