

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ**

**Отчет о лабораторной работе № 2.13 Модули и пакеты**

Выполнил:

Шальнев Владимир Сергеевич,  
2 курс, группа ПИЖ-б-о-20-1,

Проверил:

Доцент кафедры  
прикладной математики и  
компьютерной безопасности,  
Воронкин Р.А.

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

Ставрополь, 2022 г.

## ВЫПОЛНЕНИЕ:

### Задача 1

```
1 def individual_func(string):
2     def replace_string(name, surname):
3         nonlocal string
4         answer = string.replace("%N%", name)
5         answer = answer.replace("%F%", surname)
6         return answer
7
8     return replace_string
9
```

Файл module.py

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 from module import individual_func as indiv
5
6 if __name__ == "__main__":
7     replacer = indiv(
8         "Уважаемый %F%, %N%! Вы делаете работу по замыканиям функций."
9     )
10    print(replacer('Vladimir', 'Shalnev'))
11    print(replacer('Dinis', 'Proverkovich'))
12    replacer = indiv("Уважаемый %F%, %N%!")
13    print(replacer('Nestikus', 'Dimov'))
14
```

Файл task.py

```
Уважаемый Shalnev, Vladimir! Вы делаете работу по замыканиям функций.
Уважаемый Proverkovich, Dinis! Вы делаете работу по замыканиям функций.
Уважаемый Dimov, Nestikus!
```

Значение №1

### Задача 2

```
1 __all__ = ["data_commands", "interaction"]
2
```

Файл \_\_init\_\_.py

```

from datetime import datetime

def add():
    destination = input("Название пункта назначения? ")
    number = int(input("Номер поезда? "))
    time = input("Время отправления ЧЧ:ММ? ")
    time = datetime.strptime(time, '%H:%M')
    train = {
        'destination': destination,
        'number': number,
        'time': time,
    }
    return train

def select(command, trains):
    count = 0
    parts = command.split(' ', maxsplit=1)
    time = datetime.strptime(parts[1], '%H:%M')
    for train in trains:
        if train.get("time") > time:
            count += 1
            print(
                '{:>4}: {} {}'.format(
                    count,
                    train.get('destination', ''),
                    train.get("number")
                )
            )
    if count == 0:
        print("Отправлений позже этого времени нет.")

```

Файл data\_commands.py

```

def print_list(trains):
    line = '+-{}-+-{}-+-{}-+-{}-+'.format(
        '-' * 4,
        '-' * 28,
        '-' * 14,
        '-' * 19
    )
    print(line)
    print(
        '| {:^4} | {:^28} | {:^14} | {:^19} |'.format(
            "No",
            "Название пункта назначения",
            "Номер поезда",
            "Время отправления"
        )
    )
    print(line)
    for idx, train in enumerate(trains, 1):
        print(
            '| {:>4} | {:<28} | {:<14} | {:>19} |'.format(
                idx,
                train.get('destination', ''),
                train.get('number', ''),
                train.get('time', 0).strftime("%H:%M")
            )
        )
    print(line)

```

```

def print_help():
    print("Список команд:\n")
    print("add - добавить отправление;")
    print("list - вывести список отправлений;")
    print("select <ЧЧ:ММ> - вывод на экран информации о "
          "поездах, отправляющихся после этого времени;")
    print("help - отобразить справку;")
    print("exit - завершить работу с программой.")

def get_command():
    return input(">>> ").lower()

```

Файл interaction.py

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-
# variant 6

import sys
from solver import interaction as inter
from solver import data_commands as d_com

if __name__ == '__main__':
    trains = []
    while True:
        command = inter.get_command()
        if command == 'exit':
            break

        elif command == 'add':
            trains.append(d_com.add())
            if len(trains) > 1:
                trains.sort(key=lambda item: item.get('destination', ''))

        elif command == 'list':
            inter.print_list(trains)

        elif command.startswith('select '):
            d_com.select(command, trains)

        elif command == 'help':
            inter.print_help()
        else:
            print(f"Неизвестная команда {command}", file=sys.stderr)

```

Файл task.py

```
>>> add
Название пункта назначения? Butovo
Номер поезда? 127
Время отправления ЧЧ:ММ? 12:30
>>> list
```

No	Название пункта назначения	Номер поезда	Время отправления
1	Butovo	127	12:30

Значение №1

```
add
Название пункта назначения? Moscow
Номер поезда? 126
Время отправления ЧЧ:ММ? 14:20
>>> help
Список команд:

add - добавить отправление;
list - вывести список отправок;
select <ЧЧ:ММ> - вывод на экран информации о поездах, отправляющихся после этого времени;
help - отобразить справку;
exit - завершить работу с программой.
>>> select 13:20
1: Moscow 126
>>> select 12:00
1: Butovo 127
2: Moscow 126
>>> |
```

Значение №2

## ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ:

### 1. Что является модулем языка Python?

Под модулем в Python понимается файл с расширением .py.

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 as псевдоним_объекта
```

```
from имя_модуля import *
```

```
from имя_пакета 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`.

**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` ?**

Она указывает список атрибутов, которые могут быть подключены.