

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 по дисциплине «Основы
программной инженерии»

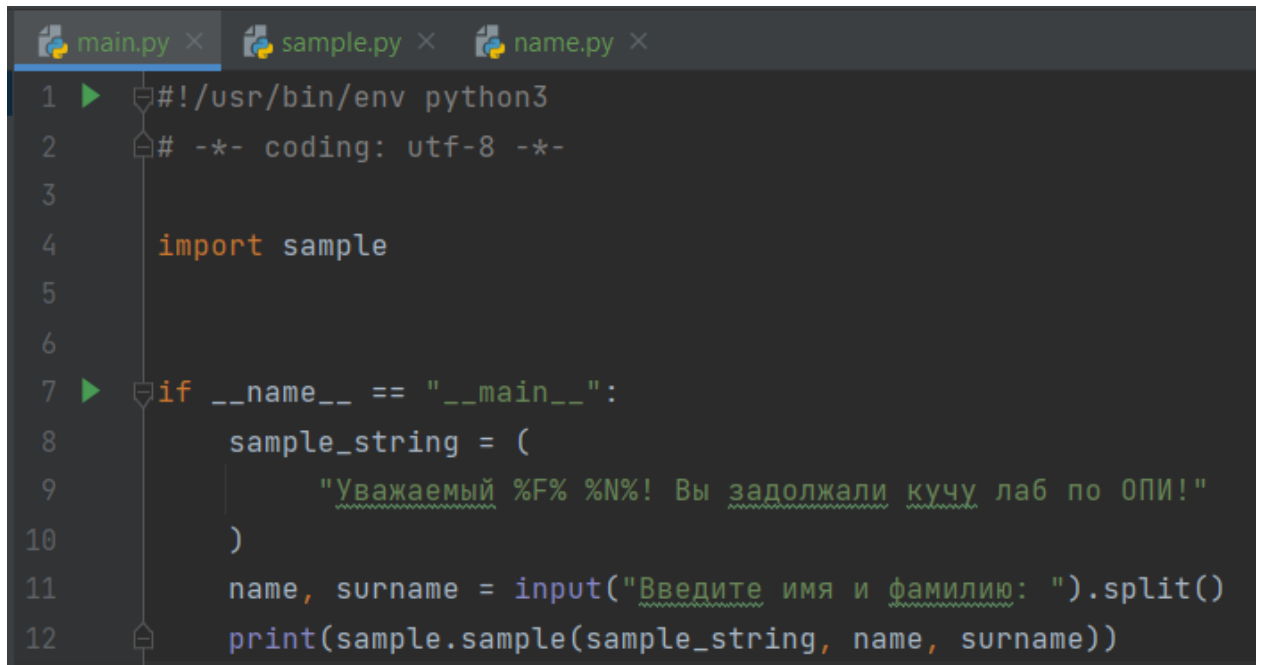
Выполнил:
Чернова Софья Андреевна,
2 курс, группа ПИЖ-б-о-20-1,
Проверил:
Доцент кафедры инфокоммуникаций,
Воронкин Р.А.

Ставрополь, 2021 г

1. Ход работы

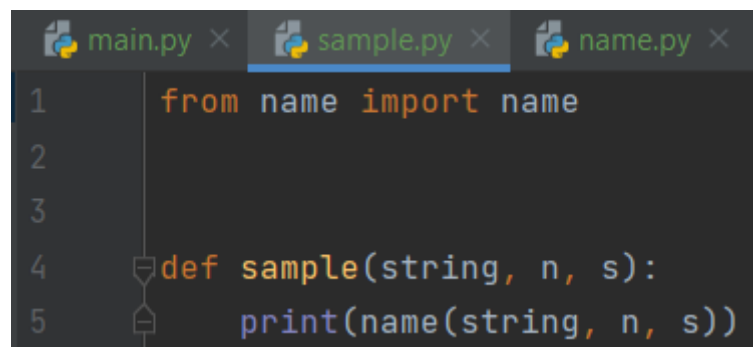
1.1 Индивидуальное задание №1 (рис. 1-4)

Условие: 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`.



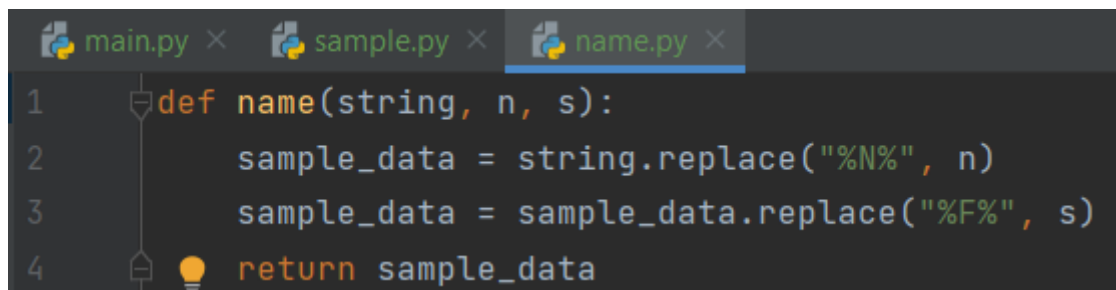
```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import sample
5
6
7  if __name__ == "__main__":
8      sample_string = (
9          "Уважаемый %F% %N%! Вы задолжали кучу лаб по ОПИ!"
10     )
11     name, surname = input("Введите имя и фамилию: ").split()
12     print(sample.sample(sample_string, name, surname))
```

Рисунок 1 – код модуля main



```
1  from name import name
2
3
4  def sample(string, n, s):
5      print(name(string, n, s))
```

Рисунок 2 – код модуля sample



```
1  def name(string, n, s):
2      sample_data = string.replace("%N%", n)
3      sample_data = sample_data.replace("%F%", s)
4      return sample_data
```

Рисунок 3 – код модуля name

```
Введите имя и фамилию: Илья Ваньянц  
Уважаемый Ваньянц Илья! Вы задолжали кучу лаб по ОПИ!  
None  
  
Process finished with exit code 0
```

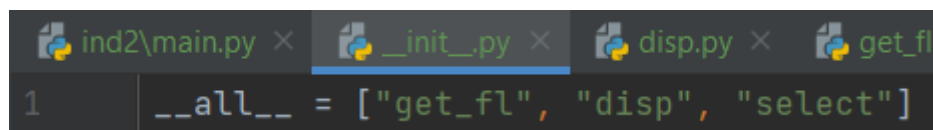
Рисунок 4 – результат выполнения программы

1.2 Индивидуальное задание №2 (рис. 5-12)

Условие: Выполнить индивидуальное задание лабораторной работы

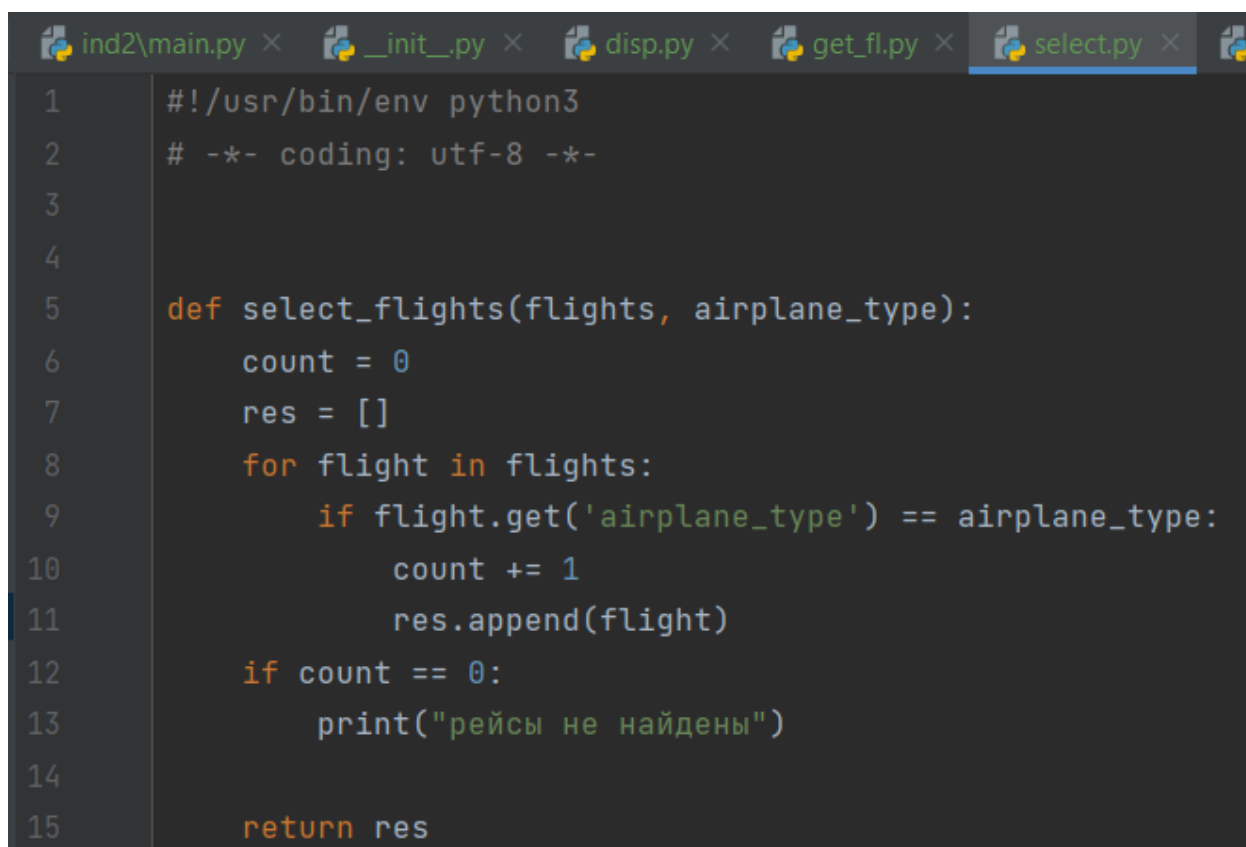
2.8, оформив все классы программы в виде отдельного пакета.

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. Номер варианта уточнить у преподавателя.



```
ind2\main.py ×  __init__.py ×  disp.py ×  get_fl.py ×  
1  __all__ = ["get_fl", "disp", "select"]
```

Рисунок 5 – код модуля `__init__`



```
ind2\main.py ×  __init__.py ×  disp.py ×  get_fl.py ×  select.py ×  
1  #!/usr/bin/env python3  
2  # -*- coding: utf-8 -*-  
3  
4  
5  def select_flights(flights, airplane_type):  
6      count = 0  
7      res = []  
8      for flight in flights:  
9          if flight.get('airplane_type') == airplane_type:  
10             count += 1  
11             res.append(flight)  
12      if count == 0:  
13          print("рейсы не найдены")  
14  
15      return res
```

Рисунок 6 – код модуля `select`

```
ind2\main.py × _init_.py × disp.py × get_fl.py × select.py × ind1\main.py ×
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import sys
5 from flights import get_fl, disp, select
6
7
8 ▶ if __name__ == '__main__':
9     flights = []
10     while True:
11         command = input(">>> ").lower()
12         if command == 'exit':
13             break
14
15         elif command == 'add':
16             flight = get_fl.get_flight()
17             flights.append(flight)
18             if len(flights) > 1:
19                 flights.sort(
20                     key=lambda item:
21                         item.get('flight_destination', ''))
22
23         elif command == 'list':
24             disp.display_flights(flights)
25
26         elif command.startswith('select '):
27             parts = command.split(' ', maxsplit=1)
28             airplane_type = (parts[1].capitalize())
29             print(f"Для типа самолета {airplane_type}:")
30             selected = select.select_flights(flights, airplane_type)
31             disp.display_flights(selected)
```

Рисунок 7 – код модуля main

```

33     elif command == 'help':
34         # Вывести справку о работе с программой.
35         print("Список команд:\n")
36         print("add - добавить рейс;")
37         print("list - вывести список всех рейсов;")
38         print(
39             "select <тип самолета> - запросить рейсы указанного типа "
40             "самолета;")
41         print("help - отобразить справку;")
42         print("exit - завершить работу с программой.")
43     else:
44         print(f"Неизвестная команда {command}", file=sys.stderr)

```

Рисунок 8 – код модуля main (продолжение)

```

ind2\main.py ×  _init_.py ×  disp.py ×  get_fl.py ×  select.py ×  ind1\main.py ×
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4
5  def get_flight():
6      flight_destination = input("Введите название пункта назначения ")
7      flight_number = input("Введите номер рейса ")
8      airplane_type = input("Введите тип самолета ")
9      return {
10         'flight_destination': flight_destination,
11         'flight_number': flight_number,
12         'airplane_type': airplane_type,
13     }

```

Рисунок 9 – код модуля get_fl

```
ind2\main.py × _init_.py × disp.py × get_fl.py × select.py × ind1\main
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4
5 def display_flights(flights):
6     if flights:
7         line = '+-{}--{}--{}--{}--+'.format(
8             '-' * 4,
9             '-' * 30,
10            '-' * 20,
11            '-' * 15
12        )
13        print(line)
14        print(
15            '| {:^4} | {:^30} | {:^20} | {:^15} |'.format(
16                "No",
17                "Пункт назначения",
18                "Номер рейса",
19                "Тип самолета"
20            )
21        )
22        print(line)
23
24        for idx, flight in enumerate(flights, 1):
25            print(
26                '| {:>4} | {:<30} | {:<20} | {:<15} |'.format(
27                    idx,
28                    flight.get('flight_destination', ''),
29                    flight.get('flight_number', ''),
30                    flight.get('airplane_type', 0)
31                )
32            )
```

Рисунок 10 – код модуля disp

```
32         )
33         print(line)
34
35     else:
36         print("Список рейсов пуст")
```

Рисунок 11 – код модуля disp (продолжение)

```

>>> add
Введите название пункта назначения London
Введите номер рейса 7685
Введите тип самолета Passenger
>>> add
Введите название пункта назначения Berlin
Введите номер рейса 9087
Введите тип самолета Passenger
>>> add
Введите название пункта назначения Moscow
Введите номер рейса 1627
Введите тип самолета Military
>>> list
+-----+-----+-----+-----+
| No | Пункт назначения | Номер рейса | Тип самолета |
+-----+-----+-----+-----+
| 1 | Berlin | 9087 | Passenger |
| 2 | London | 7685 | Passenger |
| 3 | Moscow | 1627 | Military |
+-----+-----+-----+-----+
>>> select Passenger
Для типа самолета Passenger:
+-----+-----+-----+-----+
| No | Пункт назначения | Номер рейса | Тип самолета |
+-----+-----+-----+-----+
| 1 | Berlin | 9087 | Passenger |
| 2 | London | 7685 | Passenger |
+-----+-----+-----+-----+
>>> |

```

Рисунок 12 – результат работы программы

2. 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. Стоит заметить, что модули могут быть написаны не только на языке Python, но и на других языках (например C).

2) Какие существуют способы подключения модулей в языке Python?

- import имя_модуля
- import имя_модуля1, имя_модуля2

Если вы хотите задать псевдоним для модуля в вашей программе, можно воспользоваться вот таким синтаксисом:

- `import имя_модуля as новое_имя`

Используя любой из вышеперечисленных подходов, при вызове функции из импортированного модуля, вам всегда придется указывать имя модуля (или псевдоним). Для того, чтобы этого избежать делайте импорт через конструкцию `from ... import...`

- `from имя_модуля import имя_объекта`

При этом импортируется только конкретный объект, остальные функции недоступны, даже если при их вызове указать имя модуля.

- `from имя_модуля import имя_объекта1, имя_объекта2`

Импортируемому объекту можно задать псевдоним.

- `from имя_модуля import имя_объекта as псевдоним_объекта`

Если необходимо импортировать все функции, классы и т. п. из модуля, то воспользуйтесь следующей формой оператора `from ... import ...*`

- `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). Для импортирования пакетов используется тот же синтаксис, что и для работы с модулями.

4) Каково назначение файла `__init__.py` ?

Данный файл необходим для того, чтобы Python мог расценивать папки как пакеты. Файл `__init__.py` может быть пустым или может содержать переменную `__all__`.

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

Она хранит список модулей, который импортируется при загрузке через конструкцию `from имя_пакета import *`. К примеру, данная переменная может выглядеть следующим образом: `__all__ = ["simper", "compper", "annuity"]`