

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 № 1 по дисциплине
«Основы программной инженерии»

Выполнил студент
2 курса, группы ПИЖ-б-о-20-1
Тотубалина С.С.
Проверил:
Доцент кафедры инфокоммуникаций,
Воронкин Р.А.

Ставрополь, 2022 г.

1. Ход работы

```
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import math, datetime
5
6 ▶ if __name__ == '__main__':
7     print(math.cos(math.pi / 9))
8     print(datetime.date(2019, 6, 7))
```

Рис. 1 – импорт двух модулей в одну строку

```
0.9396926207859084
2019-06-07
```

```
Process finished with exit code 0
```

Рис. 2 – результат импорта двух модулей в одну строку

```
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import math as m
5
6 ▶ if __name__ == '__main__':
7     print(m.sin(m.pi / 7))
```

Рис. 3 – задание своих имён импортированным модулям

```
0.4338837391175581
```

```
Process finished with exit code 0
```

Рис. 4 – результат задания своих имён импортированным модулям

```
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 from math import cos
5
6 ▶ if __name__ == '__main__':
7     print(cos(8))
8     print(sin(8))
```

Рис. 5 – импорт отдельных объектов из модуля

```
C:\Users\admin\anaconda3\python.exe "C:/Users/admin/PycharmProjects/OPI labs/2.13/example/ex3.py"
Traceback (most recent call last):
  File "C:\Users\admin\PycharmProjects\OPI labs\2.13\example\ex3.py", line 8, in <module>
    print(sin(8))
NameError: name 'sin' is not defined
-0.14550003380861354

Process finished with exit code 1
```

Рис. 6 – результат импорта отдельных объектов из модуля

```
1  ▶  #!/usr/bin/env python3
2      # -*- coding: utf-8 -*-
3
4      from math import cos, sin, pi
5
6  ▶  if __name__ == '__main__':
7      print(cos(pi))
8      print(sin(pi))
```

Рис. 7 – импорт нескольких отдельных функций из модуля

```
-1.0
1.2246467991473532e-16

Process finished with exit code 0
```

Рис. 8 – результат импорта нескольких отдельных функций из модуля

```
1  ▶  #!/usr/bin/env python3
2      # -*- coding: utf-8 -*-
3
4      from math import *
5  ▶  if __name__ == '__main__':
6      print(cos(pi))
7      print(sin(pi))
8      print(factorial(2))
```

Рис. 9 – импорт всех функций из модуля при помощи *

```
-1.0
1.2246467991473532e-16
2

Process finished with exit code 0
```

Рис. 10 – результат импорта всех функций из модуля при помощи *

```

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

```

Рис. 11 – код файла sample.py для индивидуального задания №1

```

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

```

Рис. 12 – код файла name_surname.py для индивидуального задания №1

```

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 делаете работу по замыканиям функции."
10     )
11     name, surname = input("Введите имя и фамилию: ").split()
12     print(sample.sample(sample_string, name, surname))

```

Рис. 13 – код файла main.py для индивидуального задания №1

```

Введите имя и фамилию: Светлана Тотубалина
Уважаемый Тотубалина Светлана! Вы делаете работу по замыканиям функции.
None

Process finished with exit code 0

```

Рис. 14 – результат выполнения программы

```

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_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             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_flights(flights, airplane_type)
31             display_flights(selected)
32
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("select <тип самолета> - запросить рейсы указанного типа "
39                 "самолета;")
40             print("help - отобразить справку;")
41             print("exit - завершить работу с программой.")

```

Рис. 15 - код файла main.py для индивидуального задания №2

```

1  __all__ = ["get_fl.py", "disp.py", "select.py"]

```

Рис. 16 - код файла _init_.py для индивидуального задания №2

```
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     }
```

Рис. 17 – код файла get_fl.py для индивидуального задания №2

```

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

```

Рис. 18 - код файла disp.py для индивидуального задания №2

```
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
```

Рис. 19 - код файла select.py для индивидуального задания №2


```

>>> add
Введите название пункта назначения Москва
Введите номер рейса 8
Введите тип самолета Грузовой
>>> add
Введите название пункта назначения Ставрополь
Введите номер рейса 7
Введите тип самолета Грузовой
>>> list
+-----+-----+-----+
| No | Пункт назначения | Номер рейса | Тип самолета |
+-----+-----+-----+
| 1 | Москва | 8 | Грузовой |
| 2 | Ставрополь | 7 | Грузовой |
+-----+-----+-----+
>>> select Грузовой
Для типа самолета Грузовой:
+-----+-----+-----+
| No | Пункт назначения | Номер рейса | Тип самолета |
+-----+-----+-----+
| 1 | Москва | 8 | Грузовой |
| 2 | Ставрополь | 7 | Грузовой |
+-----+-----+-----+
>>> select 0
Для типа самолета 0:
рейсы не найдены
Список рейсов пуст
>>> ghj
>>> Неизвестная команда ghj
exit

Process finished with exit code 0

```

Рис. 20 – результат выполнения программы

Ответы на вопросы:

1. Что является модулем языка Python?

Под модулем 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?

При помощи следующих конструкций:

- 1) `import имя_модуля`
- 2) `import имя_модуля1, имя_модуля2`
- 3) `import имя_модуля as новое_имя`
- 4) `from имя_модуля import имя_объекта`
- 5) `from имя_модуля import имя_объекта1, имя_объекта2`
- 6) `from имя_модуля import имя_объекта as псевдоним_объекта`
- 7) `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.

```
from имя_пакета import *
```

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

Переменная `__all__` хранит список модулей, который импортируется при загрузке через конструкцию.