

**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:
Выходцев Егор Дмитриевич,
2 курс, группа ПИЖ-б-о-20-1,

Проверил:
Доцент кафедры инфокоммуникаций,
Воронкин Р.А.

Ставрополь, 2022 г

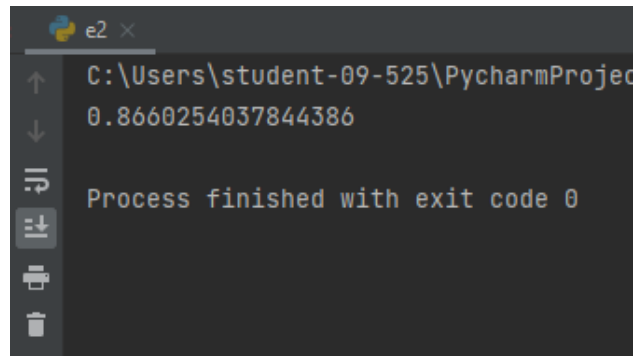
1. Примеры из методических указаний

```
e1.py x
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 / 4))
8     print(datetime.date(2017, 3, 21))
9
```

```
e1 x
C:\Users\student-09-525\PycharmProje
0.7071067811865476
2017-03-21
Process finished with exit code 0
```

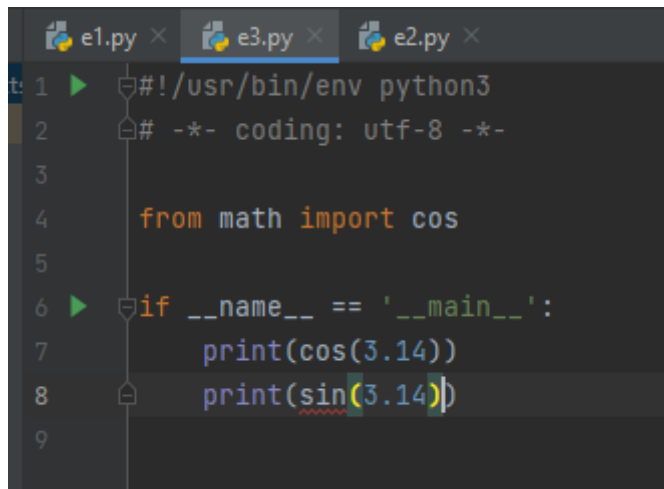
Импорт двух модулей в одну строку

```
e1.py x e2.py x
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 / 3))
8
```

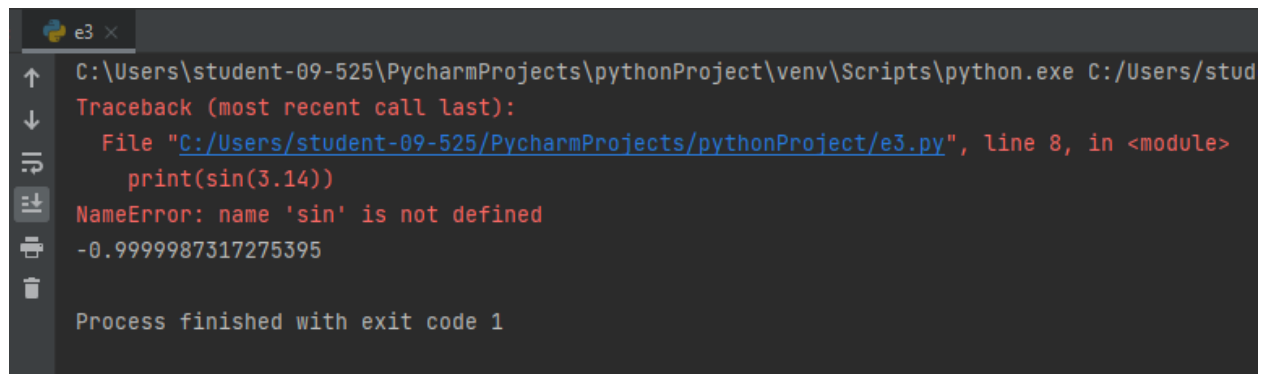


```
e2 x
C:\Users\student-09-525\PycharmProject
0.8660254037844386
Process finished with exit code 0
```

Задание своих имён импортированным модулям



```
e1.py x e3.py x e2.py x
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 from math import cos
5
6 if __name__ == '__main__':
7     print(cos(3.14))
8     print(sin(3.14))
9
```



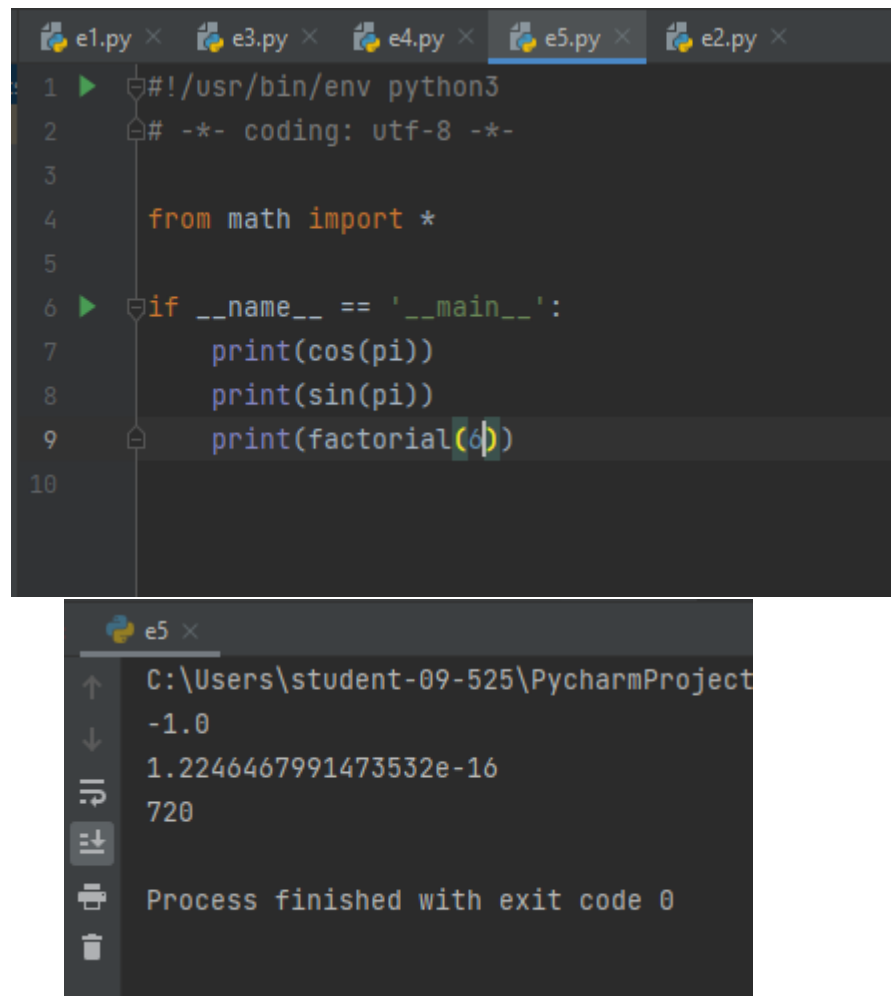
```
e3 x
C:\Users\student-09-525\PycharmProjects\pythonProject\venv\Scripts\python.exe C:/Users/stud
Traceback (most recent call last):
  File "C:/Users/student-09-525/PycharmProjects/pythonProject/e3.py", line 8, in <module>
    print(sin(3.14))
NameError: name 'sin' is not defined
-0.9999987317275395
Process finished with exit code 1
```

Импорт отдельных объектов из модуля

```
e1.py × e3.py × e4.py × e2.py ×
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))
9
```

```
e4 ×
↑ C:\Users\student-09-525\PycharmProje
↓ -1.0
: 1.2246467991473532e-16
⌵ Process finished with exit code 0
⌵
⌵
```

Импорт нескольких отдельных функций из модуля



The image shows a PyCharm IDE with two windows. The top window displays a Python script in a file named e5.py. The script contains the following code:

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 from math import *
5
6 if __name__ == '__main__':
7     print(cos(pi))
8     print(sin(pi))
9     print(factorial(6))
10
```

The bottom window shows the output of the script. It displays the results of the calculations: -1.0, 1.2246467991473532e-16, and 720. The output also indicates that the process finished with exit code 0.

Импорт всех функций из модуля разом

2. Индивидуальные задания

2.1. 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`.

```
main.py x sample.py x name_surname.py x
1  ▶ 1 #!/usr/bin/env python3
2    2 # -*- coding: utf-8 -*-
3
4    4 import sample
5
6
7  ▶ 7 if __name__ == "__main__":
8      8     sample_string = (
9          9         "Greetings %F% %N%! You are doing the function closure."
10         )
11     11     name, surname = input("Enter your name and surname: ").split()
12     12     print(sample.sample(sample_string, name, surname))
13
```

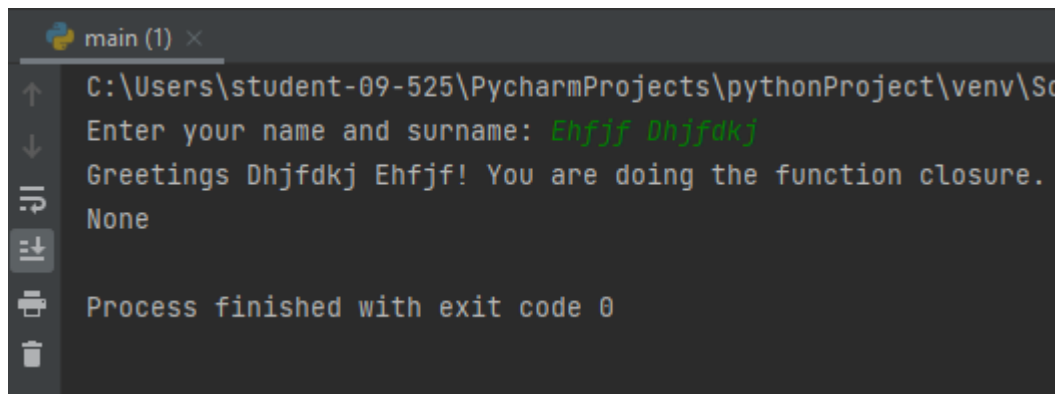
Рисунок 1 – Содержание модуля main

```
main.py x sample.py x name_surname.py x
1  1 from name_surname import name_surname
2
3
4  4 def sample(string, n, s):
5      5     print(name_surname(string, n, s))
6
7
```

Рисунок 2 – Содержание модуля sample

```
main.py x sample.py x name_surname.py x
1  1 def name_surname(string, n, s):
2      2     sample_data = string.replace("%N%", n)
3      3     sample_data = sample_data.replace("%F%", s)
4      4     return sample_data
5
```

Рисунок 3 – Содержание модуля name_surname



```
main (1) ×
C:\Users\student-09-525\PycharmProjects\pythonProject\venv\Scripts
Enter your name and surname: Ehjfjf Dhjfdkj
Greetings Dhjfdkj Ehjfjf! You are doing the function closure.
None
Process finished with exit code 0
```

Рисунок 4 – Результат выполнения программы

2.2. Задание 2 (рис 5-

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.

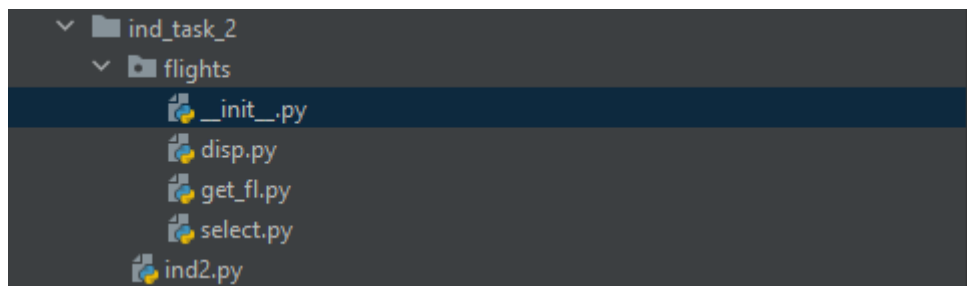


Рисунок 5 – Содержимое пакета «flights»

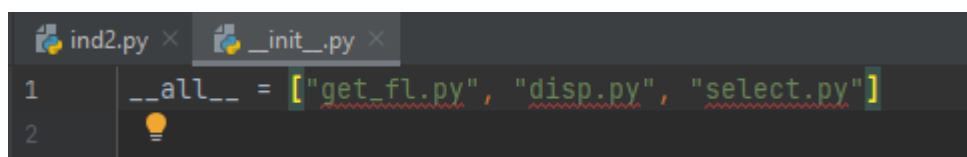


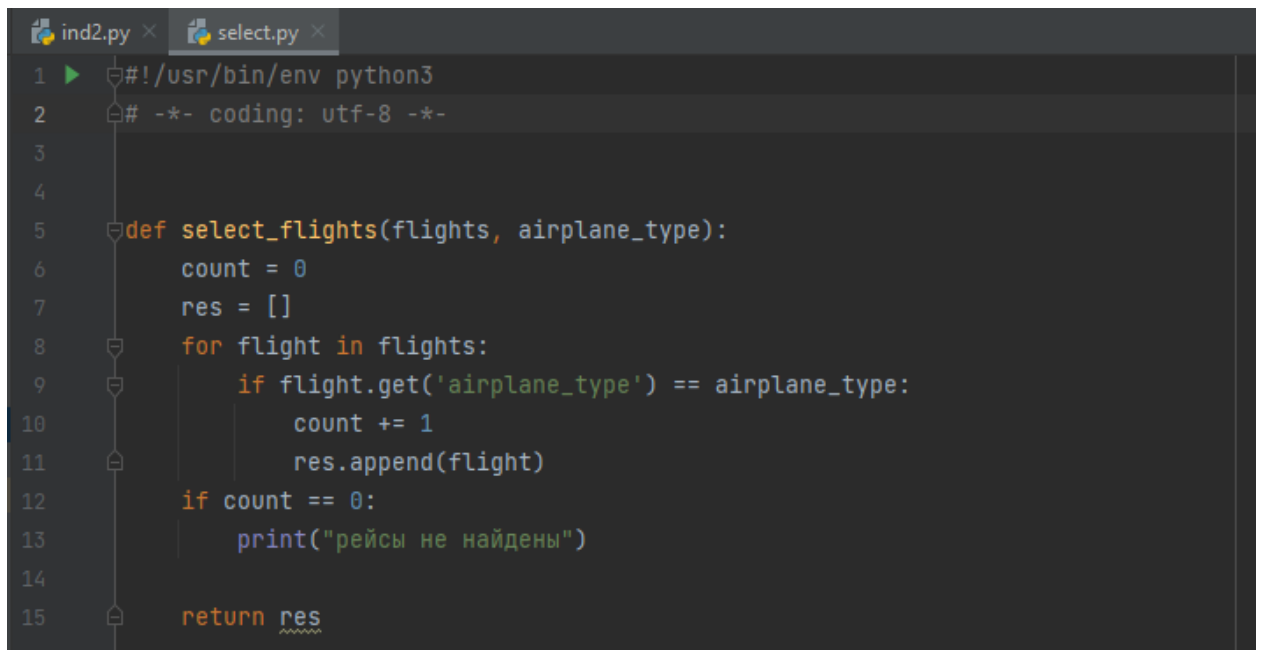
Рисунок 6 – Настройка `__init__.py`

```
ind2.py x get_fl.py x
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 7 – Код модуля get_fl


```
ind2.py x disp.py x
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")
```

Рисунок 8 – Код модуля disp



```
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 9 – Код модуля select

```
ind2.py x select.py x
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)
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(
39                "select <тип самолета> - запросить рейсы указанного типа "
```

Рисунок 10 – Код основного модуля

```
ind2 x
C:\Users\Evil\PycharmProjects\PyCharm\venv\Scripts\python.exe C:/Users/Evil/PycharmPro
>>> add
Введите название пункта назначения Seattle
Введите номер рейса S1342
Введите тип самолета Military
>>> add
Введите название пункта назначения Ukraine
Введите номер рейса U1244
Введите тип самолета Military
>>> add
Введите название пункта назначения Moscow
Введите номер рейса M1241
Введите тип самолета Passenger
>>> list
+-----+-----+-----+-----+
| No | Пункт назначения | Номер рейса | Тип самолета |
+-----+-----+-----+-----+
| 1 | Moscow | M1241 | Passenger |
| 2 | Seattle | S1342 | Military |
| 3 | Ukraine | U1244 | Military |
+-----+-----+-----+-----+
>>> select Military
Для типа самолета Military:
+-----+-----+-----+-----+
| No | Пункт назначения | Номер рейса | Тип самолета |
+-----+-----+-----+-----+
| 1 | Seattle | S1342 | Military |
| 2 | Ukraine | U1244 | Military |
+-----+-----+-----+-----+
>>> select Hdhdj
Для типа самолета Hdhdj:
рейсы не найдены
Список рейсов пуст
>>> djjs
>>> Неизвестная команда djjs
exit
Process finished with exit code 0
```

Рисунок 11 – Результат работы программы

3. 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"]`