

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

Институт цифрового развития
Кафедра инфокоммуникаций

«Модули и пакеты»

ОТЧЕТ
по лабораторной работе №16
дисциплины
«Основы программной инженерии»

Выполнил:

Мизин Глеб Егорович

2 курс, группа ПИЖ-б-о-21-1,

09.03.04 «Программная

инженерия», направленность

(профиль) «Разработка и

сопровождение программного

обеспечения», очная форма

обучения

(подпись)

Проверил:

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2022 г.

Индивидуальное задание №1: 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.

```
1  def func_1(type_1='max'):  
2      def func_2(b):  
3          if type_1 == 'max':  
4              return max(b)  
5          else:  
6              return min(b)  
7      return func_2  
8
```

Рисунок 1 – Код модуля индивидуального задания №1

```
1  #!/usr/bin/env python3  
2  # -*- coding: utf-8 -*-  
3  
4  
5  import module  
6  
7  if __name__ == "__main__":  
8      test_list = [1, 2, 3, 5, 0, -13, 100]  
9      test_tuple = [1.5, 2, 123, 5, 0, -167, 100]  
10  
11     print(f'Max of entered list: {module.func_1()(test_list)}')  
12     print(f'Min of entered tuple: {module.func_1("min")(test_tuple)}')  
13
```

Run: ind_1 × ind ×

```
F:\GitLab\LR-2.13\venv\Scripts\python.exe F:\GitLab\LR-2.13\PyCharm\ind_1.py  
Max of entered list: 100  
Min of entered tuple: -167  
Process finished with exit code 0
```

Рисунок 2 – Код и результат работы программы индивидуального задания

Индивидуальное задание №1: 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.

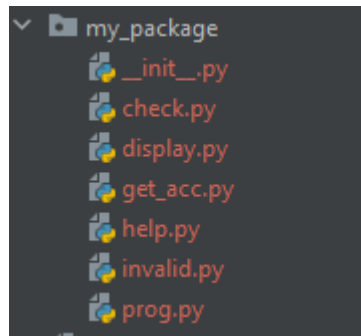


Рисунок 3 – Созданный пакет `my_package`

```
1  __all__ = ['help', 'invalid', 'check', 'display', 'get_acc', 'prog']
2  |
```

Рисунок 4 – Код модуля `__init__.py`

```
1  def sum_check(requisites, account):
2      full_summa = 0
3      for sender_req in requisites:
4          if int(sender_req.get("s_b_a")) == int(account):
5              full_summa += float(sender_req.get("t_a"))
6
7      if full_summa == 0:
8          print("This bank account does not exist")
9      else:
10         print(full_summa)
11
12
```

Рисунок 5 – Код модуля `check.py`

```

1  def display_acc(accounts):
2      if accounts:
3          line = '+--{}---{}---{}---{}---+'.format(
4              '-' * 2,
5              '-' * 25,
6              '-' * 25,
7              '-' * 10
8          )
9          print(line)
10         print(
11             '| {:^2} | {:^25} | {:^25} | {:^10} |'.format(
12                 "№",
13                 "Sender bank account",
14                 "beneficiary account",
15                 "Amount",
16             )
17         )
18         print(line)
19
20         for ind, requisite in enumerate(accounts, 1):
21             print(
22                 '| {:^2} | {:^25} | {:^25} | {:^10} |'.format(
23                     ind,
24                     requisite.get('s_b_a'),
25                     requisite.get('b_a'),
26                     requisite.get('t_a'),
27                 )
28             )
29             print(line)
30     else:
31         print("You have no bank accounts for now!")
32

```

Рисунок 6 – Код модуля display.py

```

1  def get_bank_acc():
2      while True:
3          s_b_a = input("Enter the sender's bank account: ")
4          if len(s_b_a) != 20 or s_b_a.isdigit() is False:
5              print("Incorrect bank account!")
6          else:
7              break
8
9      while True:
10         b_a = input("Enter the beneficiary's account: ")
11         if len(b_a) != 20 or b_a.isdigit() is False:
12             print("Incorrect bank account!")
13         else:
14             break
15
16         t_a = input("Enter transfer amount in ₺: ")
17
18         return {
19             "s_b_a": s_b_a,
20             "b_a": b_a,
21             "t_a": t_a,
22         }
23

```

Рисунок 7 – Код модуля get_acc.py

```

1  def help_me():
2      print("Command List:\n")
3      print("add - Add bank account;")
4      print("list - Display a list of bank accounts;")
5      print("select <bank account> -", end=" ")
6      print("The withdrawn amount from account;")
7      print("help - Display Help;")
8      print("exit - End the program.")
9      print("\n")
10

```

Рисунок 8 – Код модуля help.py

```

1  def invalid_com():
2      print('\n')
3      print(f"Invalid command use help")
4

```

Рисунок 9 – Код модуля invalid.py

```

1  from my_package import help
2  from my_package import check
3  from my_package import display
4  from my_package import get_acc
5  from my_package import invalid
6
7
8  def main():
9      """
10     Main function
11     """
12     requisites = []
13     while True:
14
15         command = input("Enter Command: ").lower()
16         if command == "exit":
17             break
18
19         elif command == "add":
20             requisite = get_acc.get_bank_acc()
21             requisites.append(requisite)
22
23             if len(requisites) > 1:
24                 requisites.sort(key=lambda item: item.get("s_b_a", ""))
25
26         elif command == "list":
27             display.display_acc(requisites)
28
29         elif command.startswith("select "):
30             parts = command.split(" ", maxsplit=1)
31             bank_acc = parts[1]
32             check.sum_check(requisites, bank_acc)
33
34         elif command == 'help':
35             help.help_me()
36
37         else:
38             invalid.invalid_com()
39

```

Рисунок 10 – Код модуля prog.py

```
1 ▶ 1 #!/usr/bin/env python3
2 2 # -*- coding: utf-8 -*-
3
4
5 3 from my_package import *
6
7 ▶ 4 if __name__ == '__main__':
8     prog.main()
9
```

Run: ind_2 (1) × ind ×

F:\GitLab\LR-2.13\venv\Scripts\python.exe F:\GitLab\LR-2.13\PyCharm\ind_2.py

Enter Command: **help**

Command List:

- add - Add bank account;
- list - Display a list of bank accounts;
- select <bank account> - The withdrawn amount from account;
- help - Display Help;
- exit - End the program.

Enter Command: **add**

Enter the sender's bank account: **12345678900987654321**

Enter the beneficiary's account: **12345678980000009878**

Enter transfer amount in ₺: **321321**

Enter Command: **add**

Enter the sender's bank account: **12345678900987654321**

Enter the beneficiary's account: **11111111111111111111**

Enter transfer amount in ₺: **123312**

Enter Command: **list**

№	Sender bank account	beneficiary account	Amount
1	12345678900987654321	12345678980000009878	321321
2	12345678900987654321	11111111111111111111	123312

Enter Command: **select 12345678900987654321**

444633.0

Enter Command:

Рисунок 11 – Код и результат работы программы индивидуального задания

Контрольные вопросы

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?

Самый простой способ импортировать модуль в Python это воспользоваться конструкцией:

```
import имя_модуля
```

За один раз можно импортировать сразу несколько модулей, для этого их нужно перечислить через запятую после слова import:

```
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
```

Для импортирования нескольких функций из модуля, можно перечислить их имена через запятую

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

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` ?

В `__init__.py` файл заставляет Python рассматривать каталоги, содержащие его, как модули. Кроме того, это первый файл, загружаемый в модуль, поэтому вы можете использовать его для выполнения кода, который хотите запускать каждый раз при загрузке модуля, или для указания экспортируемых подмодулей.

5. Каково назначение переменной `__all__` 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 *
```

Например для нашего случая содержимое `__init__.py` может быть вот таким

```
__all__ = ["simper", "compper", "annuity"]
```

```
import fincalc.simper
fv = fincalc.simper.fv(pv, i, n)
import fincalc.simper as sp
fv = sp.fv(pv, i, n)
from fincalc import simper
fv = simper.fv(pv, i, n)
```