

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:

Сотников Андрей Александрович
2 курс, группа ПИЖ-б-о-21-1,
09.03.04 «Программная
инженерия», направленность
(профиль) «Разработка и
сопровождение программного
обеспечения», очная форма
обучения

(подпись)

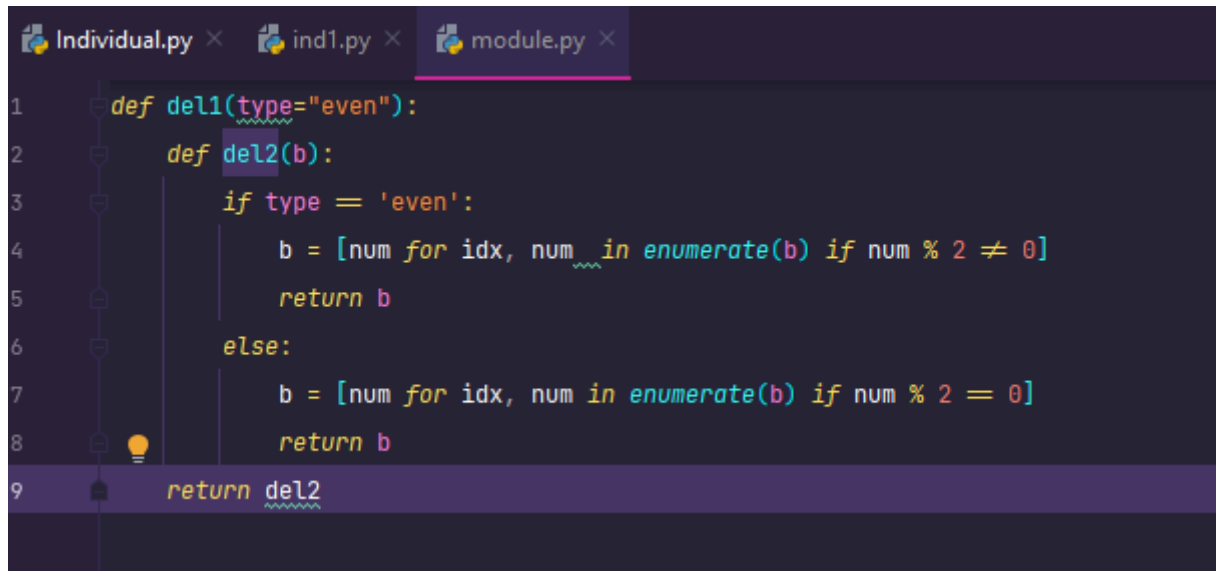
Проверил:

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

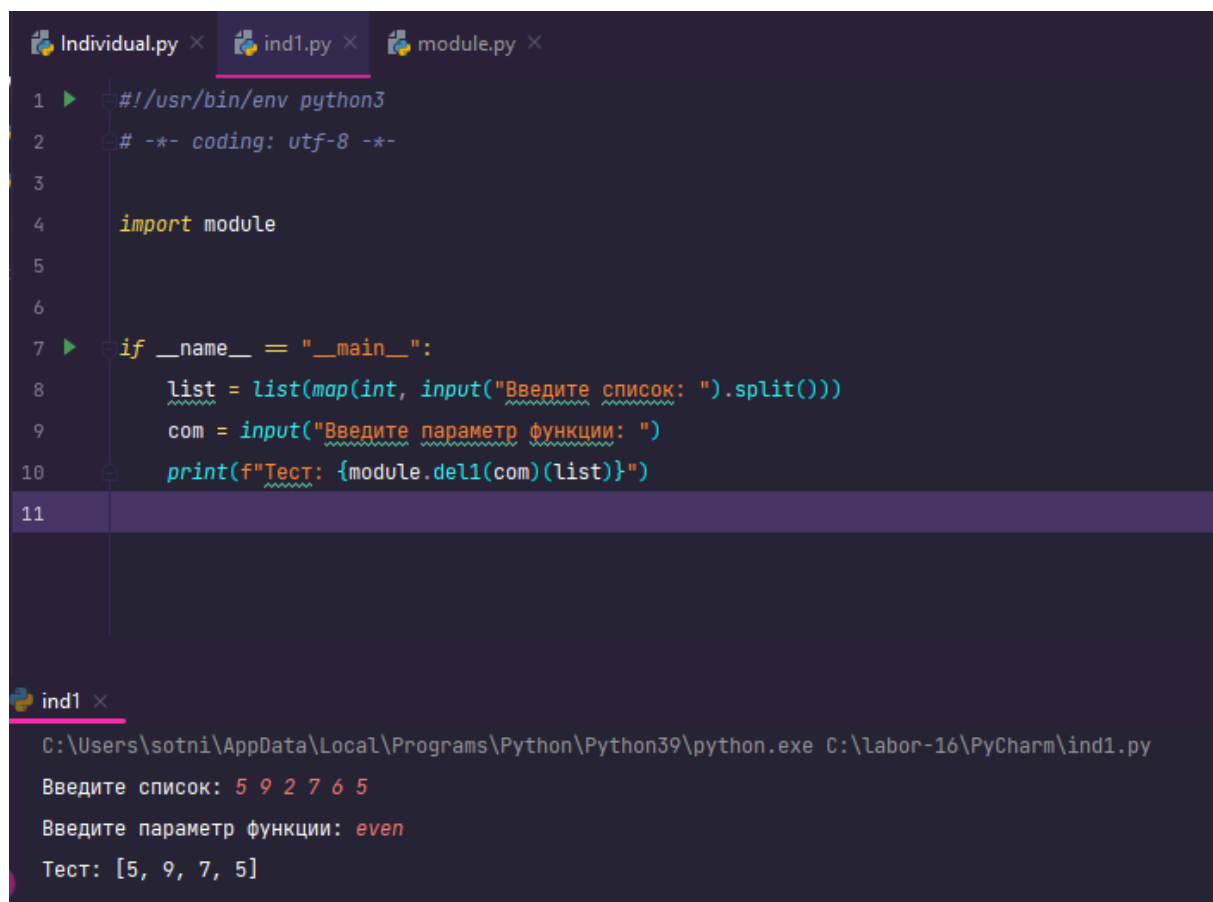
Ставрополь, 2022 г.

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 del1(type="even"):
2     def del2(b):
3         if type == 'even':
4             b = [num for idx, num in enumerate(b) if num % 2 != 0]
5             return b
6         else:
7             b = [num for idx, num in enumerate(b) if num % 2 == 0]
8             return b
9     return del2
```

Рисунок 1 – Код модуля индивидуального задания №1



```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import module
5
6
7 if __name__ == "__main__":
8     list = list(map(int, input("Введите список: ").split()))
9     com = input("Введите параметр функции: ")
10    print(f"Тест: {module.del1(com)(list)}")
11
```

ind1 x

C:\Users\sotni\AppData\Local\Programs\Python\Python39\python.exe C:\labor-16\PyCharm\ind1.py

Введите список: 5 9 2 7 6 5

Введите параметр функции: even

Тест: [5, 9, 7, 5]

Рисунок 2 – Код и результат работы программы индивидуального задания

Индивидуальное задание №2: 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.

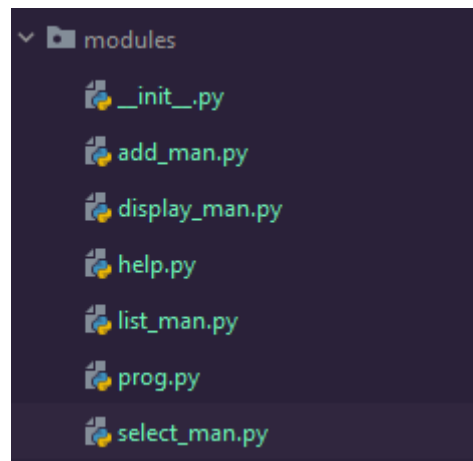


Рисунок 3 – Созданный пакет modules

```
1  __all__ = ['help', 'display_man', 'select_man', 'list_man', 'add_man', 'prog']
```

Рисунок 4 – Код модуля `__init__.py`

```
2
3  def add_man():
4      """
5      Добавление людей
6      """
7
8      name = input("Фамилия и инициалы? ")
9      post = input("Телефон? ")
10     year = input("Год рождения? ")
11     year = year.split(".")
12     year = date(int(year[0]), int(year[1]), int(year[2]))
13
14     # Создать словарь.
15     man = {
16         'name': name,
17         'tel': post,
18         'date': year,
19     }
20
21     # Добавить словарь в список.
22     return man
```

Рисунок 5 – Код модуля
`add_man.py`

```

def display_man(staff):
    if staff:
        line = '+-{}-+-{}-+-{}-+-{}-+'.format(
            '-' * 4,
            '-' * 30,
            '-' * 20,
            '-' * 12
        )
        print(line)
        print(
            '| {:^4} | {:^30} | {:^20} | {:^12} |'.format(
                "№",
                "Ф.И.О.",
                "Телефон",
                "Год рождения"
            )
        )
        print(line)
        for idx, worker in enumerate(staff, 1):
            print(
                '| {:>4} | {:<30} | {:<20} | {:>12} |'.format(
                    idx,
                    worker.get('name', ''),
                    worker.get('tel', ''),
                    str(worker.get('date', 0))
                )
            )
        print(line)
    else:
        print("Список пуст")

```

Рисунок 6 – Код модуля display_man.py

```

def list_man(people):
    """
    Вывод людей
    """

    line = '+-{}--{}-+-{}--{}-+'.format(
        '-' * 4,
        '-' * 30,
        '-' * 20,
        '-' * 12
    )
    print(line)
    print(
        '| {:^4} | {:^30} | {:^20} | {:^12} |'.format(
            "№",
            "Ф.И.О.",
            "Телефон",
            "Год рождения"
        )
    )
    print(line)

    for idx, man in enumerate(people, 1):
        print(
            '| {:>4} | {:<30} | {:<20} | {:>12} |'.format(
                idx,
                man.get('name', ''),
                man.get('tel', ''),
                str(man.get('date', 0))
            )
        )
    print(line)

```

Рисунок 7 – Код модуля list_man.py

```
def help_man():  
    print("Список команд:\n")  
    print("add - добавить человека;")  
    print("list - вывести список людей;")  
    print("select <имя> - запросить людей с этим именем;")  
    print("help - отобразить справку;")  
    print("exit - завершить работу с программой.")
```

Рисунок 8 – Код модуля help.py

```

1  from modules import help
2      from modules import display_man
3      from modules import select_man
4      from modules import list_man
5  from modules import add_man
6
7
8
9  def main():
10     people = []
11
12     while True:
13         command = input(">>> ").lower()
14
15         if command == 'exit':
16             break
17
18         elif command == "add":
19             people.append(add_man.add_man())
20             if len(people) > 1:
21                 people.sort(key=lambda item: item.get('tel', ''))
22
23         elif command == 'list':
24             display_man.display_man(people)
25
26         elif command.startswith('select'):
27             parts = command.split(' ', maxsplit=1)
28             period = parts[1]
29             select = select_man(people, period)
30             display_man.display_man(select)
31
32         elif command == 'help':
33             display_man.help_man()
34
35         else:
36             print(f"Неизвестная команда {command}", file=sys.stderr)

```

Рисунок 9 – Код модуля prog.py

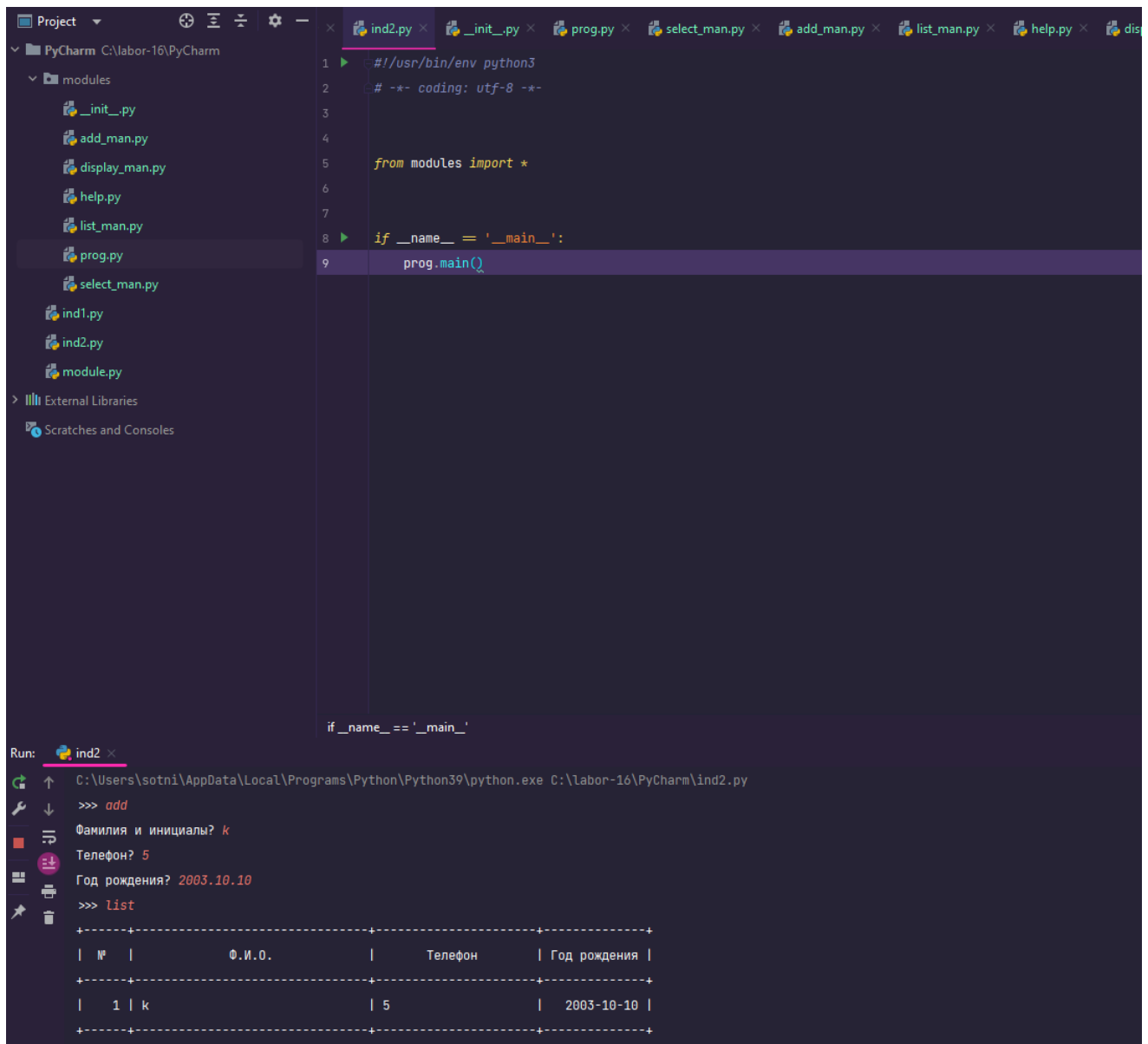


Рисунок 10 – Код и результат работы программы индивидуального задания

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)
```