

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ
ФЕДЕРАЦИИ**

**Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

**Отчет по лабораторной работе № 2.13
по дисциплине «Основы программной инженерии»**

Выполнил студент группы
ПИЖ-б-о-21-1

Трушева В. О. .« » 2022г.

Подпись студента _____

Работа защищена «
» _____ 20__ г.

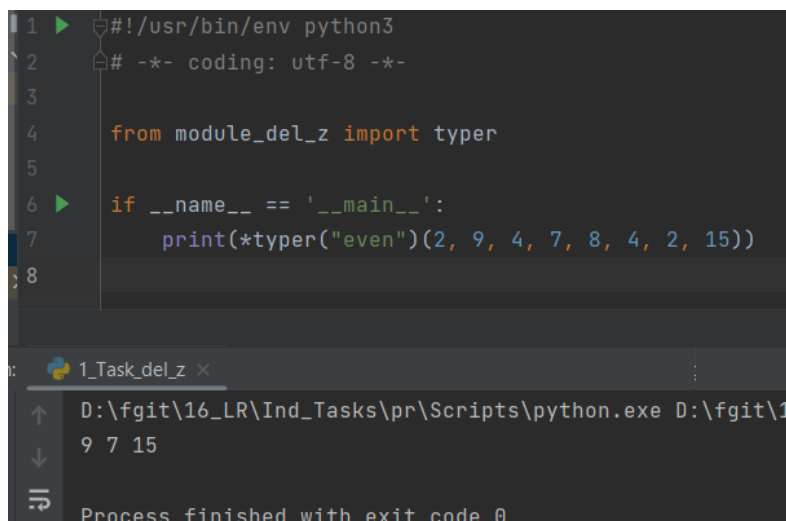
Проверила Воронкин Р.А.

(подпись)

Ставрополь 2022

Методика и порядок выполнения работы

1. Изучить теоретический материал работы.
2. Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и язык программирования Python.
3. Выполните клонирование созданного репозитория.
4. Дополните файл .gitignore необходимыми правилами для работы с IDE PyCharm.
5. Организуйте свой репозиторий в соответствие с моделью ветвления git-flow.
6. Создайте проект PyCharm в папке репозитория.
7. Выполните индивидуальные задания. Приведите в отчете скриншоты работы программ решения индивидуального задания.



```
1  > #!/usr/bin/env python3
2  > # -*- coding: utf-8 -*-
3
4  > from module_del_z import typer
5
6  > if __name__ == '__main__':
7  >     print(*typer("even")(2, 9, 4, 7, 8, 4, 2, 15))
8
9  >
10 >
```

1_Task_del_z x

D:\fgit\16_LR\Ind_Tasks\pr\Scripts\python.exe D:\fgit\16_LR\Ind_Tasks\pr\Scripts\python.exe

9 7 15

Process finished with exit code 0

Рисунок 1 – Результат работы программы

```

1  ▶ #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4
5  def typer(types="even"):
6      def dele(*args):
7          values = [int(arg) for arg in args]
8          i = 0
9
10         # Удаление четных элементов по значению.
11         if types == 'even':
12             for j in values[:]:
13                 if j % 2 == 0:
14                     del values[i]
15                 else:
16                     i += 1
17
18         # Удаление нечетных элементов по значению.
19         else:
20             for j in values[:]:
21                 if j % 2 != 0:
22                     del values[i]
23                 else:
24                     i += 1
25
26         return values
27
28     return dele
29

```

Рисунок 2 – Результат работы программы

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

2_Task x

D:\fgit\16_LR\Ind_Tasks\pr\Scripts\python.exe D:\fgit\16_LR\Ind_Tasks\2_Task.py

Введите команду >>> help

Список команд:

- add - добавить человека;
- list - вывести список людей;
- find - найти человека по фамилии;
- help - отобразить справку;
- exit - завершить работу с программой.

Введите команду >>> add

Фамилия Имя: Трушева Вероника

Номер телефона: 1982301

Дата рождения: 12-12-12

Введите команду >>> list

№	Фамилия Имя	Номер телефона	Дата рождения
1	Трушева Вероника	1982301	0012-12-12

Введите команду >>> find

Введите номер телефона: 21324

Запись не найдена

Рисунок 3 – Результат работы программы

8. Зафиксируйте сделанные изменения в репозитории.
9. Добавьте отчет по лабораторной работе в формате PDF в папку doc репозитория. Зафиксируйте изменения.
10. Выполните слияние ветки для разработки с веткой master/main.
11. Отправьте сделанные изменения на сервер GitHub.
12. Отправьте адрес репозитория GitHub на электронный адрес преподавателя.

Вопросы для защиты работы

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`. 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 *`.