

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ  
ФЕДЕРАЦИИ**

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

**Кафедра инфокоммуникаций**

**Отчет по лабораторной работе № 2.11  
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. Проработать примеры лабораторной работы.

```
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 def fun1(a):
5     x = a * 3
6
7     def fun2(b):
8         nonlocal x
9         return b + x
10
11     return fun2
12
13
14 ▶ if __name__ == '__main__':
15     test_fun = fun1(4)
16     print(test_fun(7))
17
```

1\_Task ×

D:\fgit\14\_LR\PyCharm\Scripts\python.exe D:\f  
19

Process finished with exit code 0

Рисунок 1 – Результат работы программы

```
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 ▶ if __name__ == '__main__':
5     tpl = lambda a, b: (a, b)
6     a = tpl(1, 2)
7     print(a)
8     b = tpl(3, a)
9     print(b)
10    c = tpl(a, b)
11    print(c)
12
```

2\_Task\_(primer) ×

D:\fgit\14\_LR\PyCharm\Scripts\python.exe "  
(1, 2)  
(3, (1, 2))  
((1, 2), (3, (1, 2)))

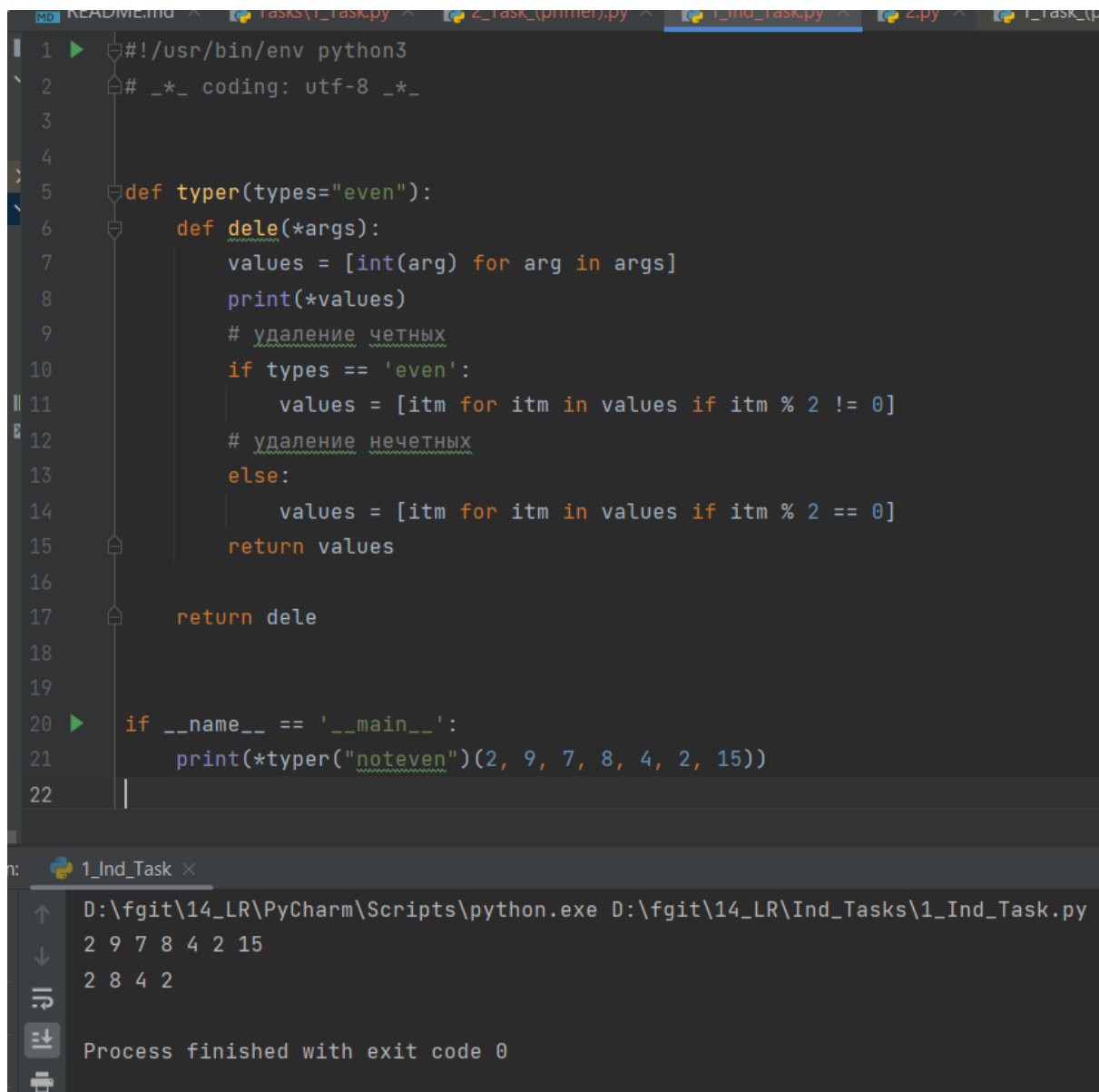
Process finished with exit code 0

Рисунок 2 – Результат работы программы

8. Выполнить индивидуальное задание.

Вариант – 10.

Условие. Используя замыкания функций, объявите внутреннюю функцию, которая принимает в качестве аргумента список целых чисел и удаляет из него все четные или нечетные значения в зависимости от значения параметра `type`. Если `type` равен «even», то удаляются четные значения, иначе – нечетные. По умолчанию `type` должно принимать значение «even». Вызовите внутреннюю функцию замыкания и отобразите на экране результат ее работы.



```
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          print(*values)
9          # удаление четных
10         if types == 'even':
11             values = [itm for itm in values if itm % 2 != 0]
12         # удаление нечетных
13         else:
14             values = [itm for itm in values if itm % 2 == 0]
15         return values
16
17     return dele
18
19
20 if __name__ == '__main__':
21     print(*typer("noteven")(2, 9, 7, 8, 4, 2, 15))
22
```

1\_Ind\_Task ×

D:\fgit\14\_LR\PyCharm\Scripts\python.exe D:\fgit\14\_LR\Ind\_Tasks\1\_Ind\_Task.py

2 9 7 8 4 2 15

2 8 4 2

Process finished with exit code 0

Рисунок 3 – Результат работы программы

9. Зафиксируйте изменения в репозитории.

10. Добавьте отчет по лабораторной работе в формате PDF в папку doc репозитория. Зафиксируйте изменения.

11. Выполните слияние ветки для разработки с веткой master/main.

12. Отправьте сделанные изменения на сервер GitHub.

13. Отправьте адрес репозитория GitHub на электронный адрес преподавателя.

Вопросы для защиты работы

1. Что такое замыкание?

Замыкание – это функция, в теле которой присутствуют ссылки на переменные, объявленные вне тела этой функции в окружающем коде и не являющиеся её параметрами.

2. Как реализованы замыкания в языке программирования Python?

```
def mul(a):  
    def helper(b):  
        return a * b  
    return helper
```

3. Что подразумевает под собой область видимости Local?

Эту область видимости имеют переменные, которые создаются и используются внутри функций.

4. Что подразумевает под собой область видимости Enclosing?

Суть данной области видимости в том, что внутри функции могут быть вложенные функции и локальные переменные, так вот локальная переменная функции для её вложенной функции находится в enclosing области видимости.

#### 5. Что подразумевает под собой область видимости Global?

Переменные области видимости global – это глобальные переменные уровня модуля (модуль – это файл с расширением .py)

#### 6. Что подразумевает под собой область видимости Build-in?

Эти сущности доступны в любом модуле Python и не требуют предварительного импорта. Built-in – это максимально широкая область видимости.

#### 7. Как использовать замыкания в языке программирования Python?

В случае с реализацией выше:

```
test_fun = fun1(4)
print("ex1")
print(test_fun(7))
```

19

#### 8. Как замыкания могут быть использованы для построения иерархических данных?

```
tpl = lambda d, e: (d, e)
s = tpl(1, 2)
print(s)
f = tpl(3, s)
print(f)
c = tpl(s, f)
print(c)
```

```
(1, 2)
(3, (1, 2))
((1, 2), (3, (1, 2)))
```