МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ

Отчет по лабораторной работе № 13
«Управление процессами в Python»

по дисциплине «Основы программной инженерии»

Выполнила:
Первых Дарья Александровна,
2 курс, группа ПИЖ-б-о-20-1

Проверил:
Доцент кафедры
инфокоммуникаций, Воронкин Р.А.

Ставрополь, 2022 г.

ВЫПОЛНЕНИЕ

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-


from multiprocessing import Process


def func():
    print("Hello from child Process")


if __name__ == "__main__":
    print("Hello from main Process")
    proc = Process(target=func)
    proc.start()
```

```
if __name__ == "__main__"
main
C:\Users\podar\study\anaconda\envs\13LR\python
Hello from main Process
Hello from child Process
```

Рисунок 1 – Создание и ожидание завершения работы процессов

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-


from multiprocessing import Process


def func():
    print("Hello from child Process")


if __name__ == "__main__":
    print("Hello from main Process")
    proc = Process(target=func)
    proc.start()
    print(f"Proc is_alive status: {proc.is_alive()}")
```

```
if __name__ == "__main__"
main
C:\Users\podar\study\anaconda\envs\13LR\python.exe "C:/Use
Hello from main Process
Proc is_alive status: True
Hello from child Process
Goodbye
Proc is_alive status: False
```

Рисунок 2 – Создание и ожидание завершения работы процессов

Рисунок 3 – Создание классов-наследников от Process

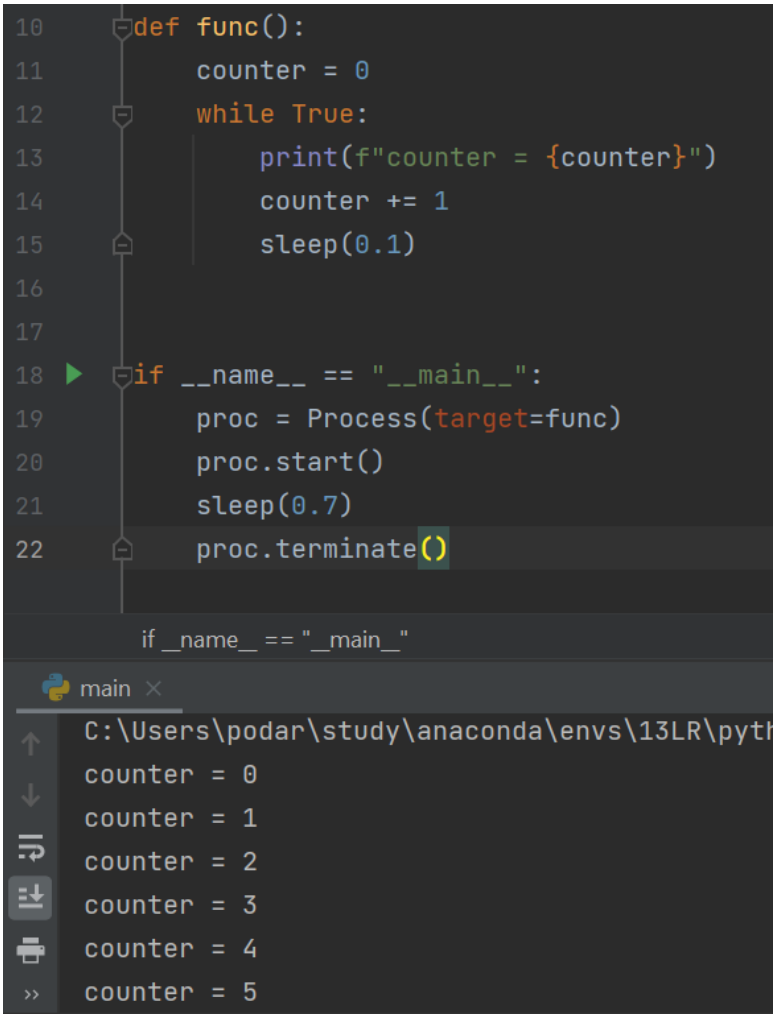

Рисунок 4 – Принудительное завершение работы процессов

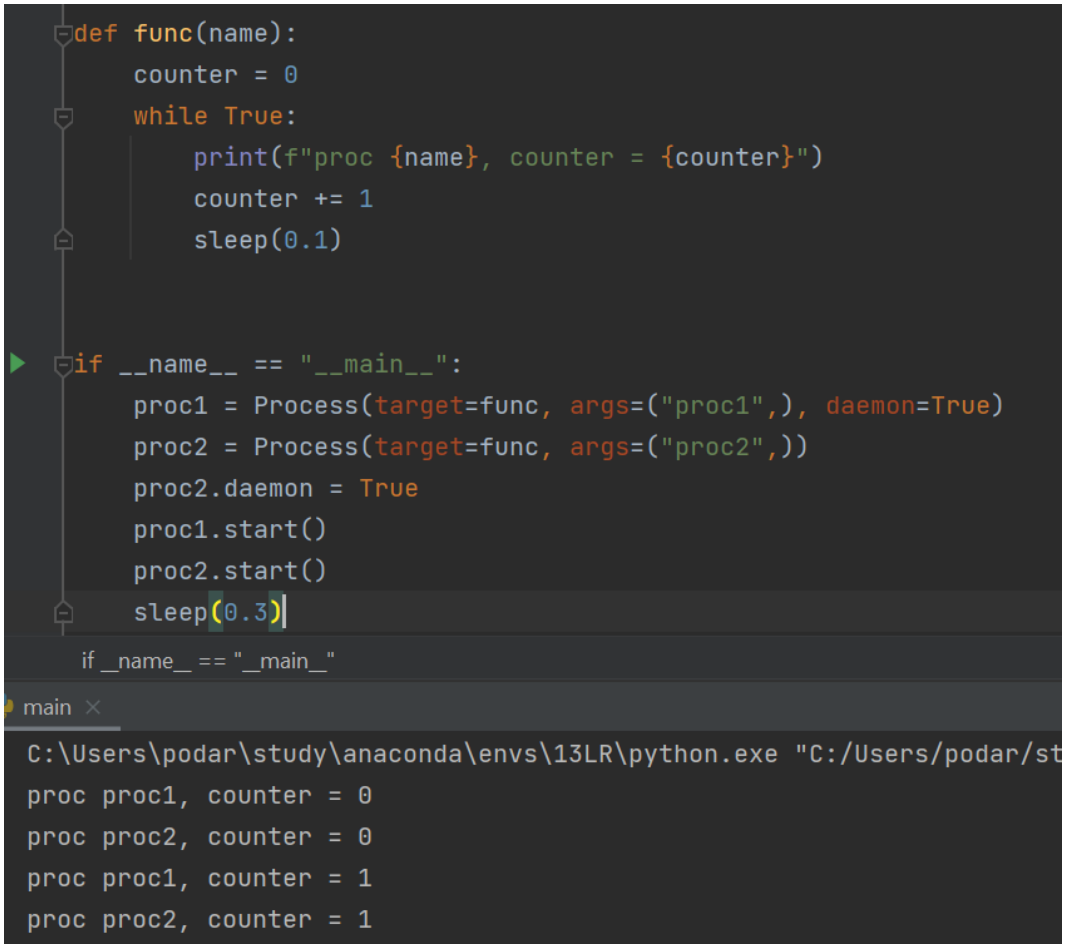

Рисунок 5 – Процессы-демоны

Индивидуальное задание.

Для своего индивидуального задания лабораторной работы 2.23 необходимо реализовать вычисление значений в двух функций в отдельных процессах.

17. $$S = \sum_{n=0}^{\infty}(n+1)x^n = 1 + 2x + 3x^2 + 4x^3 + \ldots;\ x = -0,7;\ y = \frac{1}{(1-x)^2}.$$

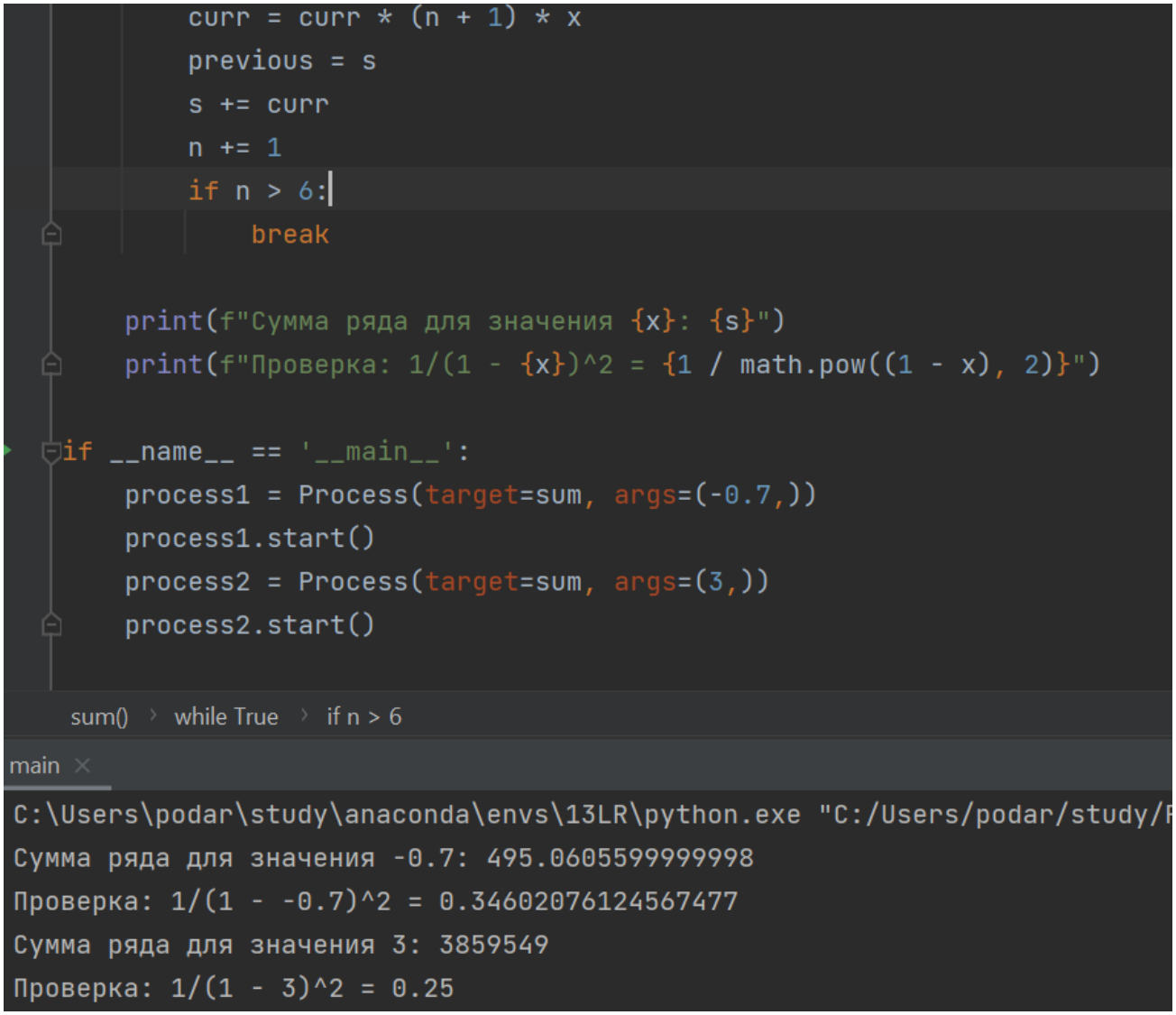

Рисунок 6 – Индивидуальное задание

ВОПРОСЫ

1. Как создаются и завершаются процессы в Python?

proc = Process(target=func)

proc.start()

join() для того, чтобы программа ожидала завершения процесса.

Процессы завершаются при завершении функции, указанной в target, либо принудительно с помощью kill(), terminate()

2. В чем особенность создания классов-наследников от Process?

В классе наследнике от Process необходимо переопределить метод run() для того, чтобы он (класс) соответствовал протоколу работы с процессами.

3. Как выполнить принудительное завершение процесса?

В отличии от потоков, работу процессов можно принудительно завершить, для этого класс Process предоставляет набор методов:

terminate() - принудительно завершает работу процесса. В Unix

отправляется команда SIGTERM, в Windows используется функция TerminateProcess().

kill() - метод аналогичный terminate() по функционалу, только вместо SIGTERM в Unix будет отправлена команда SIGKILL.

4. Что такое процессы-демоны? Как запустить процесс-демон?

Процессы демоны по своим свойствам похожи на потоки-демоны, их суть заключается в том, что они завершают свою работу, если завершился родительский процесс.

Указание на то, что процесс является демоном должно быть сделано до его запуска (до вызова метода start()). Для демонического процесса запрещено самостоятельно создавать дочерние процессы. Эти процессы не являются демонами (сервисами) в понимании Unix, единственное их свойство – это завершение работы вместе с родительским процессом.

```
proc1 = Process(target=func, args=("proc1",), daemon=True)

proc2.daemon = True

proc1.start()

proc2.start()
```