

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

Институт цифрового развития  
Кафедра инфокоммуникаций

**ОТЧЕТ**  
**ПО ЛАБОРАТОРНОЙ РАБОТЕ №11**  
**дисциплины «Анализ данных»**  
**Вариант 9**

Выполнил:  
Дудкин Константин Александрович  
2 курс, группа ИВТ-б-о-22-2,  
09.03.01 «Информатика и  
вычислительная техника»,  
направленность (профиль)  
«Программное обеспечение средств  
вычислительной техники и  
автоматизированных систем»  
очная форма обучения

---

(подпись)

Проверил:  
Кандидат технических наук, доцент  
кафедры инфокоммуникаций, доцент  
Воронкин роман Александрович

---

(подпись)

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

Ставрополь, 2024 г.

Тема: Управление процессами в Python

Цель: Приобретение навыков написания многозадачных приложений на языке программирования Python версии 3.x.

### Порядок выполнения работы

Для индивидуального задания из прошлой лабораторной работы была реализована работа функций в двух отдельных процессах:

```
def var_1(x, queue): 1 usage
    mnum = x
    S, k = mnum, 2

    while fabs(mnum) > E:
        kf = (2 * k) + 1
        mnum = (pow(-1, k) * pow(x, kf)) / kf
        S += mnum
        k += 1

    queue.put(S)

# Вариант 10
def var_2(x, queue): 1 usage
    mnum = x
    S, k = mnum, 2

    while fabs(mnum) > E:
        kf = 2 * k
        mnum = pow(x, kf) / factorial(kf)
        S += mnum
        k += 1

    queue.put(S)
```

Рисунок 1. Изменения в функциях расчета рядов

```

def main(): 1 usage
    queue = Queue()

    x1 = 1.0
    x2 = 1.0

    proc1 = Process(target=var_1, args=(x1, queue))
    proc2 = Process(target=var_2, args=(x2, queue))

    proc1.start()
    proc2.start()

    proc1.join()
    proc2.join()

    result1 = queue.get()
    result2 = queue.get()

    control1 = sin(x1)
    control2 = pow(e, x2)

    print(f"x1 = {x1}")
    print(f"Сумма ряда 1: {round(result1, 4)}")
    print(f"Контрольное значение ряда 1: {round(control1, 4)}")
    print(f"Проверка 1: {isclose(result1, control1, rel_tol=1e-4)}")

    print(f"x2 = {x2}")
    print(f"Сумма ряда 2: {round(result2, 4)}")
    print(f"Контрольное значение ряда 2: {round(control2, 4)}")
    print(f"Проверка 2: {isclose(result2, control2, rel_tol=1e-4)}")

if __name__ == "__main__":
    main()

```

Рисунок 2. Изменения в функции main()

Вывод: В ходе выполнения данной работы были получены навыки написания многозадачных приложений на Python версии 3.x.

## Ответы на вопросы

1. Процессы в Python создаются с использованием модуля `multiprocessing`. Для создания процесса необходимо создать экземпляр класса `Process` и вызвать метод `start()` для запуска процесса. Завершение процесса происходит автоматически после выполнения всей работы внутри процесса.

2. Особенность создания классов-наследников от `Process` заключается в том, что они могут переопределять методы, такие как `run()`, для определения поведения процесса.

3. Принудительное завершение процесса в Python может быть осуществлено путем вызова метода `terminate()` для объекта процесса.

4. Процессы-демоны в Python – это процессы, которые работают в фоновом режиме и завершаются, когда основной процесс завершает свою работу. Для запуска процесса-демона, необходимо установить атрибут `daemon` объекта процесса в значение `True` перед его запуском.