

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 ПРАКТИЧЕСКОЙ РАБОТЕ №12**  
**дисциплины «Программирование на Python»**  
**Вариант 23**

Выполнил:  
Мотовилов Вадим Борисович  
2 курс, группа ИВТ-б-о-22-1,  
09.03.01 «Информатика и  
вычислительная техника»,  
направленность (профиль)  
«Информатика и вычислительная  
техника», очная форма обучения

---

(подпись)

Руководитель практики:  
Воронкин Роман Александрович

---

(подпись)

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

Ставрополь, 2023 г.

Порядок выполнения работы:

1. Создал репозиторий и скопировал его

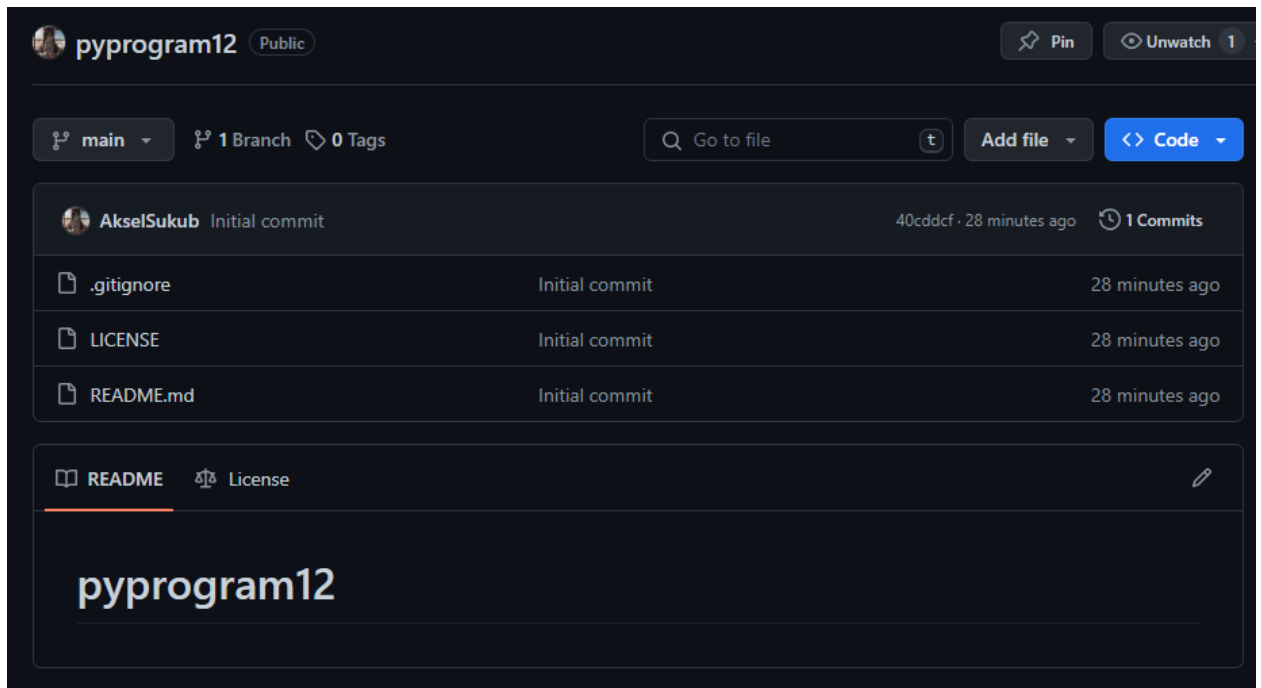


Рисунок 1. Созданный репозиторий

2. Изменил файл .gitignore и README.rm и добавил git flow

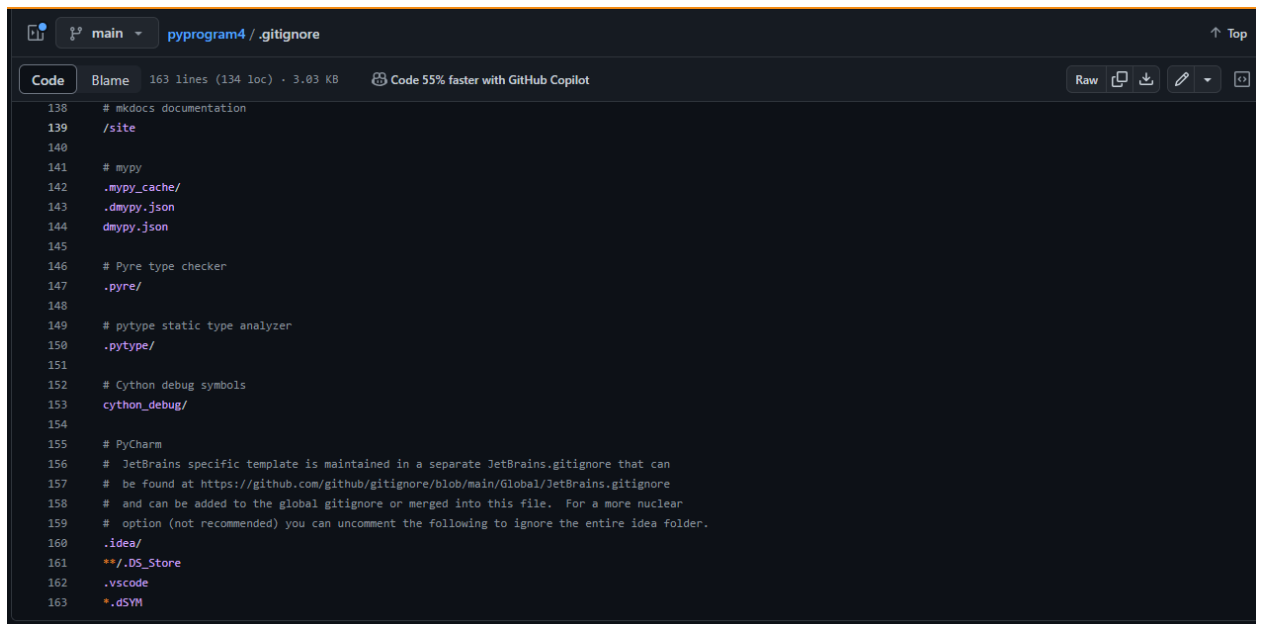


Рисунок 2. Измененный файл .gitignore

3. Выполнил задания

Напечатать в обратном порядке последовательность чисел, признаком конца которой является 0.

```
zadanie1.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  def print_reverse_sequence():
5      num = int(input("Введите число: "))
6      if num != 0:
7          print_reverse_sequence()
8      print(num)
9
10 print_reverse_sequence()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\1\pyprogram12\prog> & "C:/Program Files/Python39/python.exe" zadanie1.py
Введите число: 3
Введите число: 7
Введите число: 9
Введите число: 11
Введите число: 17
Введите число: 0
0
17
11
9
7
3
```

Рисунок 3 . Выполнение 1 задания

Ответы на контрольные вопросы:

1. Рекурсия - это процесс, при котором функция вызывает саму себя. Она используется для решения задач, которые могут быть разбиты на более простые подзадачи. Рекурсия позволяет решать такие задачи более элегантным и понятным способом.

Преимущества использования рекурсии:

Упрощение кода: Рекурсивные решения могут быть более компактными и понятными, поскольку они отображают структуру задачи более наглядно.

Решение сложных задач: Рекурсия может быть полезна для решения задач, которые сложно решить итеративными методами.

Работа с рекурсивными структурами данных: Рекурсия позволяет эффективно работать с рекурсивными структурами данных, такими как деревья или списки.

Однако рекурсия также может потреблять больше памяти и времени выполнения, особенно если не правильно использовать базовый случай или рекурсивные вызовы.

2. База рекурсии - это условие, при котором рекурсивная функция прекращает вызывать саму себя и возвращает результат. База рекурсии определяет точку останова рекурсии и предотвращает бесконечное выполнение функции.

Например, в задаче печати чисел в обратном порядке, базой рекурсии является проверка, равно ли введенное число нулю. Если число равно нулю, функция прекращает вызывать саму себя и возвращает результат.

3. Стек программы - это структура данных, которая хранит информацию о вызовах функций во время выполнения программы. Каждый раз, когда функция вызывается, информация о вызове помещается в стек. Когда функция завершается, информация о вызове удаляется из стека.

Стек программы используется для управления вызовами функций и сохранения контекста выполнения. Когда функция вызывает другую функцию, текущий контекст выполнения сохраняется в стеке, и управление передается вызываемой функции. При завершении вызываемой функции контекст восстанавливается из стека, и выполнение продолжается с точки вызова.

4. В языке Python можно получить текущее значение максимальной глубины рекурсии с помощью функции `sys.getrecursionlimit()`. Она возвращает текущий лимит рекурсии, то есть максимальную глубину рекурсии.

Пример использования:

```
import sys
```

```
recursion_limit = sys.getrecursionlimit()
```

```
print(recursion_limit)
```

5. Если число рекурсивных вызовов превысит максимальную глубину рекурсии в языке Python, будет возбуждено исключение `RecursionError`. Это происходит для предотвращения переполнения стека и перерасхода памяти.

6. Максимальную глубину рекурсии можно изменить с помощью функции `sys.setrecursionlimit()`. Она позволяет установить новое значение для максимальной глубины рекурсии.

Пример использования:

```
import sys
```

```
new_recursion_limit = 5000
```

```
sys.setrecursionlimit(new_recursion_limit)
```

Однако изменение максимальной глубины рекурсии должно быть осторожным, поскольку слишк