

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 №10
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

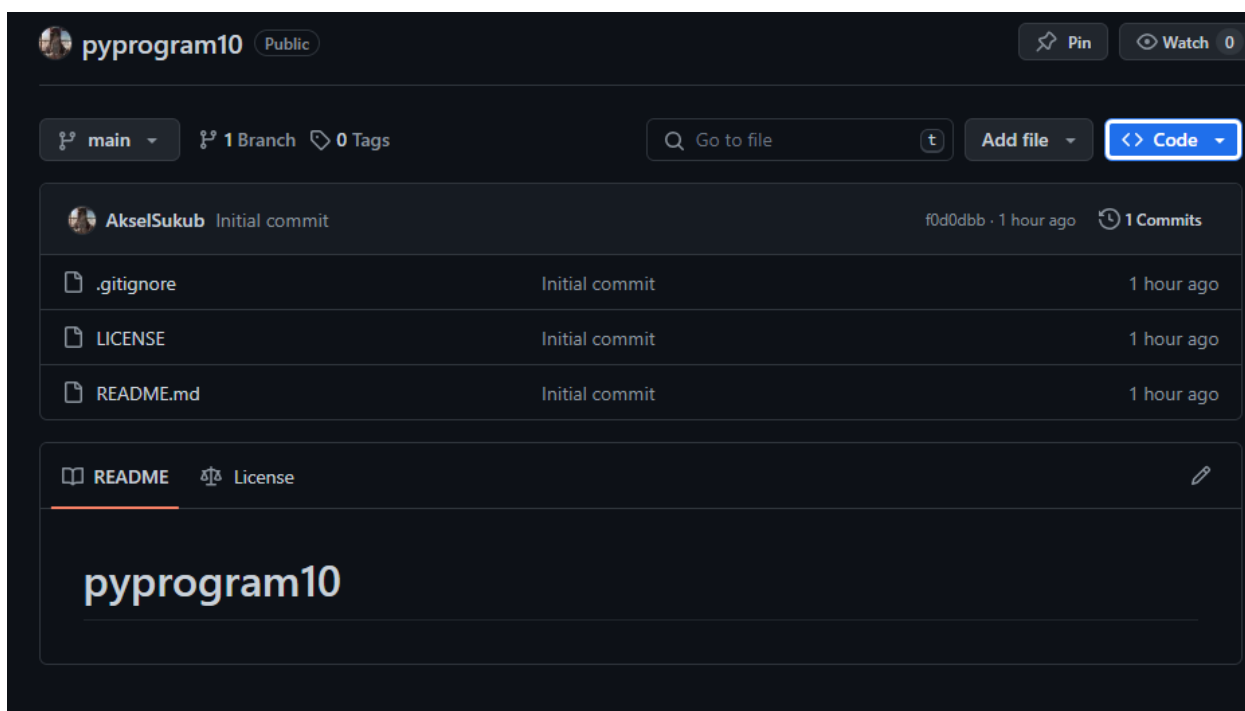


Рисунок 1. Созданный репозиторий

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

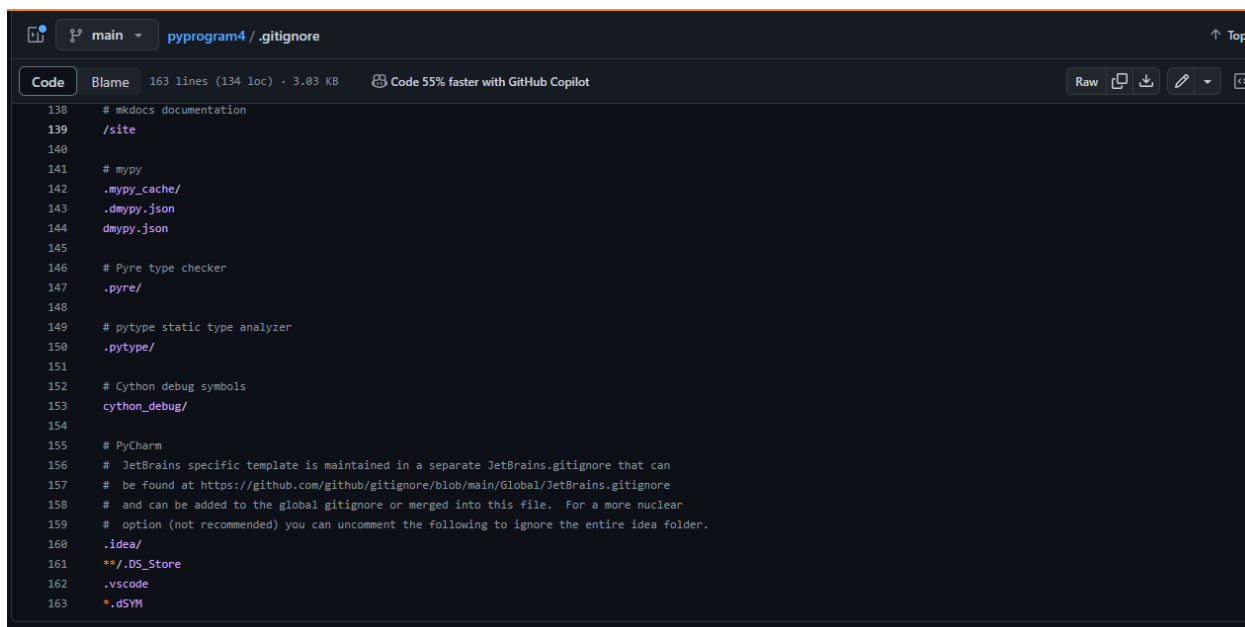


Рисунок 2. Измененный файл .gitignore

3. Выполнил задания

$$X = (A/B) \cap (C \cup D); \quad Y = (A/D) \cup (\bar{C}/\bar{B}).$$
$$A = \{a, b, d, I, x\}; \quad B = \{d, e, h, i, n, u\}; \quad C = \{e, f, m, n\}; \quad D = \{a, c, h, k, r, s, w, x\};$$

```
zadanie1.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  def set_operations(set1, set2):
5      print("Множество 1:", set1)
6      print("Множество 2:", set2)
7
8      # Объединение
9      union_set = set1.union(set2)
10     print("\nОбъединение множеств 1 и 2:", union_set)
11
12     # Пересечение
13     intersection_set = set1.intersection(set2)
14     print("Пересечение множеств 1 и 2:", intersection_set)
15
16     # Разность
17     difference_set = set1.difference(set2)
18     print("Разность множеств 1 и 2:", difference_set)
19
20     # Симметрическая разность
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\1\pyprogram10\prog> & "C:/Program Files/Python311/python.exe" c:/Users/1/pyprogram10/prog/zadanie1.py
Введите элементы множества 1 (через запятую): 1, 3, 5, 7
Введите элементы множества 2 (через запятую): 2, 4, 8, 3
Множество 1: {'5', '1', '3', '7'}
Множество 2: {'2', '4', '8', '3'}

Объединение множеств 1 и 2: {'2', '8', '3', '4', '5', '7', '1'}
Пересечение множеств 1 и 2: {'3'}
Разность множеств 1 и 2: {'5', '1', '7'}
Симметрическая разность множеств 1 и 2: {'2', '8', '4', '5', '7', '1'}
```

Рисунок 3 . Выполнение 1 задания

Ответы на контрольные вопросы:

1. В языке Python множество (set) - это неупорядоченная коллекция уникальных элементов. Множество в Python поддерживает операции, характерные для математических множеств, такие как объединение, пересечение, разность и другие.

2. В Python множество можно создать с помощью фигурных скобок {} или с помощью функции set(). Например:

```
set1 = {1, 2, 3} # Создание множества с помощью фигурных скобок
```

```
set2 = set([4, 5, 6]) # Создание множества с помощью функции set()
```

3. Для проверки присутствия или отсутствия элемента в множестве в Python используется оператор in. Например:

```
my_set = {1, 2, 3}
```

```
if 1 in my_set:
```

```
    print("Элемент 1 присутствует в множестве")
```

```
if 4 not in my_set:
```

```
    print("Элемент 4 отсутствует в множестве")
```

4. Для перебора элементов множества в Python можно использовать цикл for. Например:

```
my_set = {1, 2, 3}
```

```
for element in my_set:
```

```
    print(element)
```

5. Set comprehension (генератор множества) - это способ создания множества на основе итерации и условий. Он аналогичен list comprehension, но использует фигурные скобки вместо квадратных скобок. Например:

```
my_set = {x for x in range(1, 10) if x % 2 == 0}
```

```
print(my_set) # Вывод: {2, 4, 6, 8}
```

6. Для добавления элемента во множество в Python используется метод add(). Например:

```
my_set = {1, 2, 3}
```

```
my_set.add(4)
```

```
print(my_set) # Вывод: {1, 2, 3, 4}
```

7. Для удаления одного элемента из множества в Python используется метод remove() или discard(). Для удаления всех элементов множества используется метод clear(). Например:

```
my_set = {1, 2, 3}
```

```
my_set.remove(2)
```

```
print(my_set) # Вывод: {1, 3}
```

```
my_set.discard(4) # Если элемент отсутствует, не будет вызвано исключение
```

```
my_set.clear()
```

```
print(my_set) # Вывод: set()
```

8. Основные операции над множествами в Python:

- Объединение: Используется оператор | или метод union().

- Пересечение: Используется оператор `&` или метод `intersection()`.
- Разность: Используется оператор `-` или метод `difference()`.
- Симметрическая разность: Используется оператор `^` или метод `symmetric_difference()`.

Например:

```
set1 = {1, 2, 3}
```

```
set2 = {3, 4, 5}
```

```
union_set = set1 | set2
```

```
intersection_set = set1 & set2
```

```
difference_set = set1 - set2
```

```
symmetric_difference_set = set1 ^ set2
```

9. Для определения, является ли некоторое множество надмножеством или подмножеством другого множества в Python используется метод `issuperset()` и `issubset()`. Например:

```
set1 = {1, 2, 3}
```

```
set2 = {1, 2}
```

```
if set2.issubset(set1):
```

```
    print("Множество 2 является подмножеством множества 1")
```

```
if set1.issuperset(set2):
```

```
    print("Множество 1 является надмножеством множества 2")
```

10. Множество `frozenset` в Python - это неизменяемое множество. Оно может быть использовано в качестве ключа в словаре, так как является хешируемым типом данных. Например:

```
my_set = frozenset([1, 2, 3])
```

```
my_dict = {my_set: "Значение"}
```

11. Для преобразования множества в строку, список или словарь в Python используются соответствующие функции `str()`, `list()` и `dict()`.
Например:

```
my_set = {1, 2, 3}
set_as_string = str(my_set)
set_as_list = list(my_set)
set_as_dict = dict.fromkeys(my_set)
```