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 №7
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

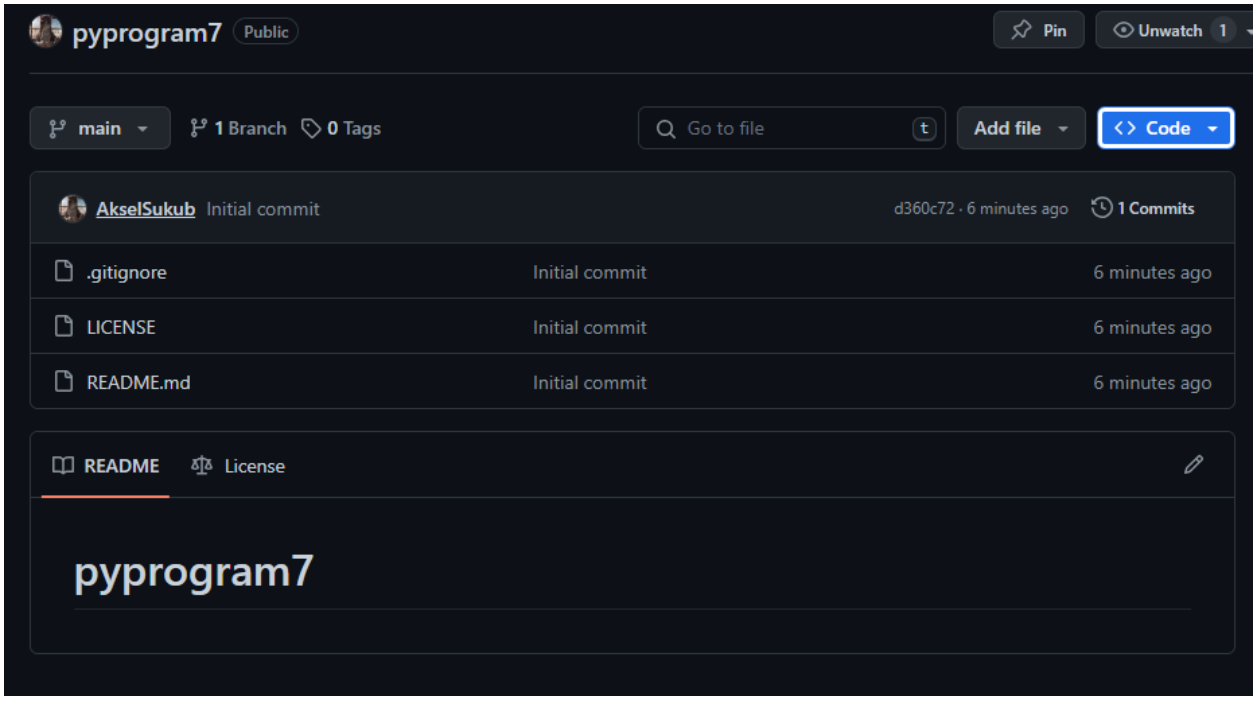

Рисунок 1. Созданный репозиторий

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

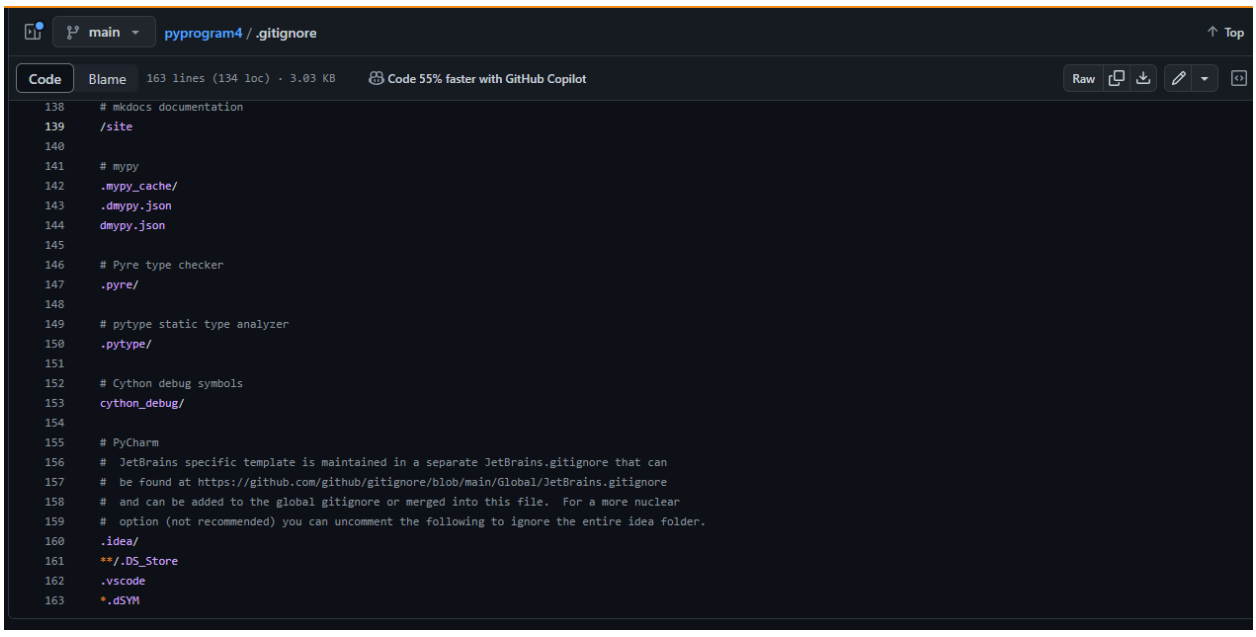

Рисунок 2. Измененный файл .gitignore

3. Выполнил задания

В списках *A*, *G*, *F* содержатся оценки учащихся по алгебре, геометрии и физике соответственно. Определить среднюю оценку по алгебре и количество учащихся, не имеющих ни одной «двойки».

zadanie1.py > ...

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

# Оценки по алгебре, геометрии и физике соответственно
grades_algebra = [5, 4, 3, 4, 5, 2, 4, 5, 4, 3]
grades_geometry = [4, 5, 3, 4, 2, 5, 4, 3, 4, 5]
grades_physics = [3, 4, 5, 3, 4, 5, 2, 4, 5, 4]

# Функция для определения средней оценки по предмету
def average_grade(grades):
    return sum(grades) / len(grades) if len(grades) > 0 else 0

# Функция для определения количества учащихся без двоек
def students_without_twos(grades):
    return len([grade for grade in grades if grade != 2])

# Определение средней оценки по алгебре с использованием цикла
average_algebra_loop = average_grade(grades_algebra)
print(f'Средняя оценка по алгебре (цикл): {average_algebra_loop:.2f}')

# Определение количества учащихся без двоек по алгебре с использованием цикла
students_without_twos_algebra_loop = students_without_twos(grades_algebra)
print(f'Количество учащихся без двоек по алгебре (цикл): {students_without_twos_algebra_loop}')

# Определение средней оценки по алгебре с использованием List Comprehensions
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Средняя оценка по алгебре (цикл): 3.90
Количество учащихся без двоек по алгебре (цикл): 9
Средняя оценка по алгебре (List Comprehensions): 3.90
Количество учащихся без двоек по алгебре (List Comprehensions): 9
```

Рисунок 3 . Выполнение 1 задания

В списке, состоящем из вещественных элементов, вычислить:

1. сумму элементов списка с нечетными номерами;
2. сумму элементов списка, расположенных между первым и последним отрицательными элементами.

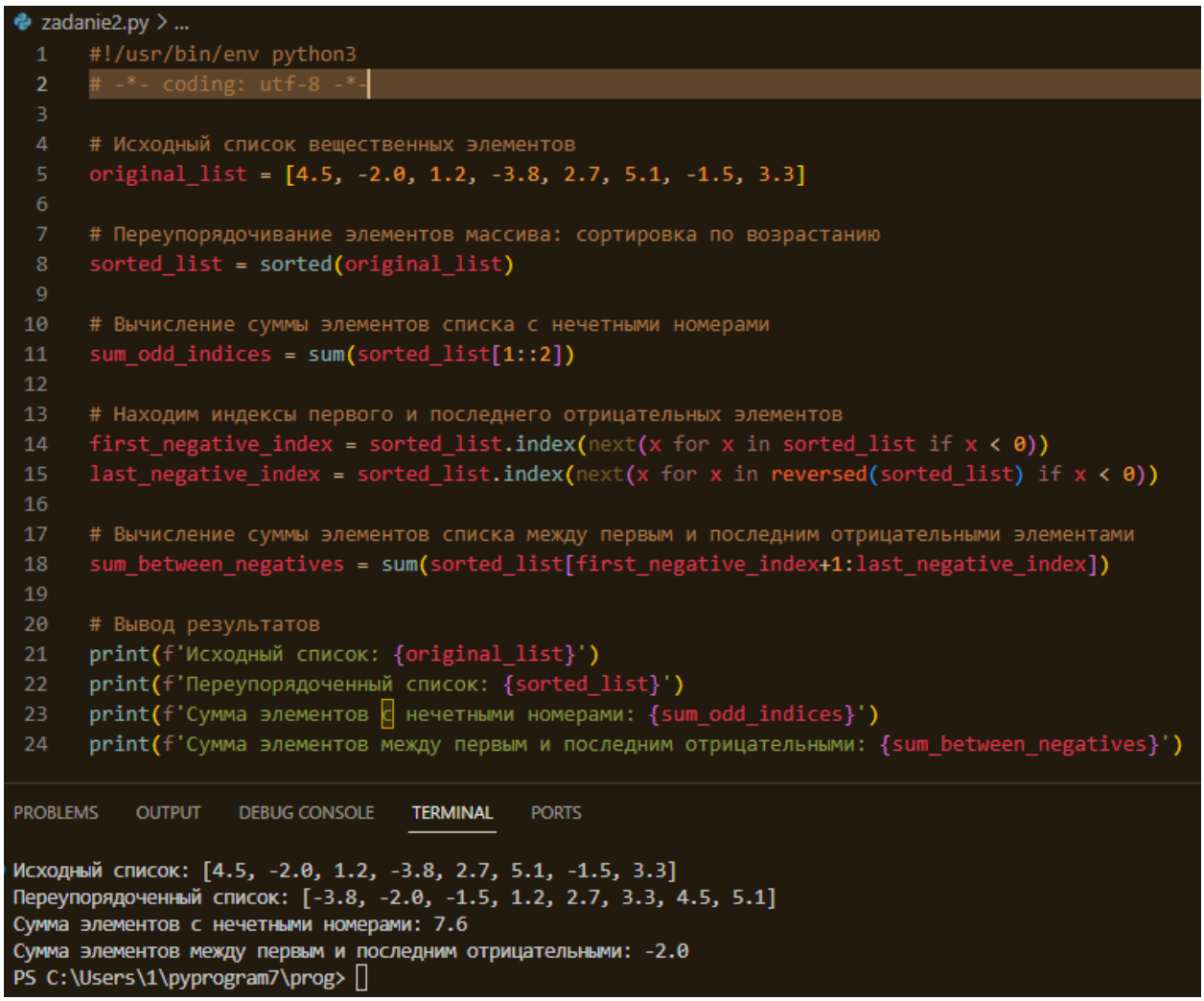

Рисунок 4 . Выполнение 2 задания

Ответы на контрольные вопросы:

1. **Что такое списки в языке Python?** Список в Python - это упорядоченная изменяемая коллекция объектов. Элементы списка могут быть любого типа данных, включая другие списки.

2. **Как осуществляется создание списка в Python?** Список создается с использованием квадратных скобок [] и разделяется запятыми. Пример: my_list = [1, 2, 3].

3. **Как организовано хранение списков в оперативной памяти?** Списки в Python хранятся в памяти как массивы, где каждый элемент занимает место в памяти, а доступ осуществляется по индексу.

4. **Каким образом можно перебрать все элементы списка?** С использованием цикла for. Например:

for element in my_list:

```
print(element)
```

5. **Какие существуют арифметические операции со списками?**
Арифметические операции со списками включают сложение списков и умножение списка на число.

6. **Как проверить есть ли элемент в списке?**
Используйте оператор in. Пример: element in my_list.

7. **Как определить число вхождений заданного элемента в списке?**
Метод count() может использоваться для определения числа вхождений. Например: my_list.count(element).

8. **Как осуществляется добавление (вставка) элемента в список?**
Методы append() для добавления в конец и insert(index, element) для вставки на определенную позицию.

9. **Как выполнить сортировку списка?**
Используйте метод sort() для сортировки в порядке возрастания. Для сортировки без изменения исходного списка используйте функцию sorted().

10. **Как удалить один или несколько элементов из списка?**
Методы remove(element) для удаления по значению и pop(index) для удаления по индексу.

11. **Что такое списковое включение и как с его помощью осуществлять обработку списков?**
Списковое включение (list comprehension) - это синтаксическое средство для создания списков с более кратким и выразительным кодом.

12. **Как осуществляется доступ к элементам списков с помощью срезов?**
С использованием срезов, например, my_list[start:stop], чтобы получить подсписок с элементами с индексами от start до stop-1.

13. **Какие существуют функции агрегации для работы со списками?**

Функции, такие как sum(), max(), и min(), используются для агрегации данных в списках.

14. **Как создать копию списка?**

Используйте метод copy() или просто выполните срез my_list[:].

15. **Самостоятельно изучите функцию sorted языка Python. В чем ее отличие от метода sort списков?**

sorted() возвращает новый отсортированный список, не изменяя оригинальный, в то время как метод sort() сортирует сам список, изменяя его.