

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**  
**Федеральное государственное автономное образовательное учреждение  
высшего образования**

**«СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций  
Институт цифрового развития**

**ОТЧЁТ**

**по лабораторной работе №2.4**

Дисциплина: «Основы кроссплатформенного программирования»

Тема: «Работа со списками в языке Python»

Выполнил: студент 1 курса

группы ИВТ-б-о-21-1

Назаров Никита Юрьевич

Ставрополь 2022

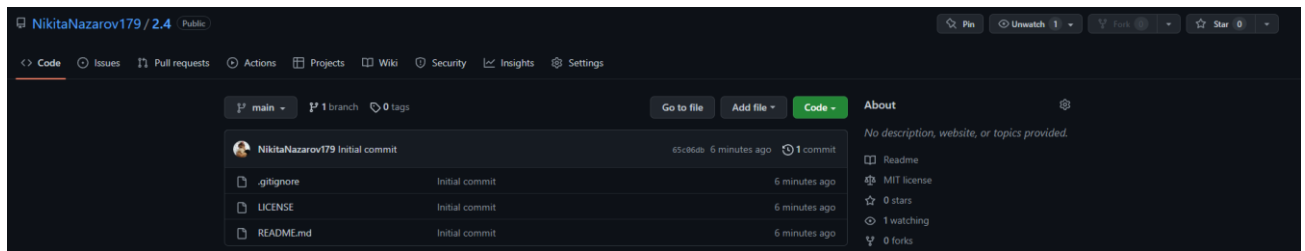


Рисунок 1 – новый репозиторий

```
H:\cross\git\nazarov>git clone https://github.com/NikitaNazarov179/2.4.git
Cloning into '2.4'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.
```

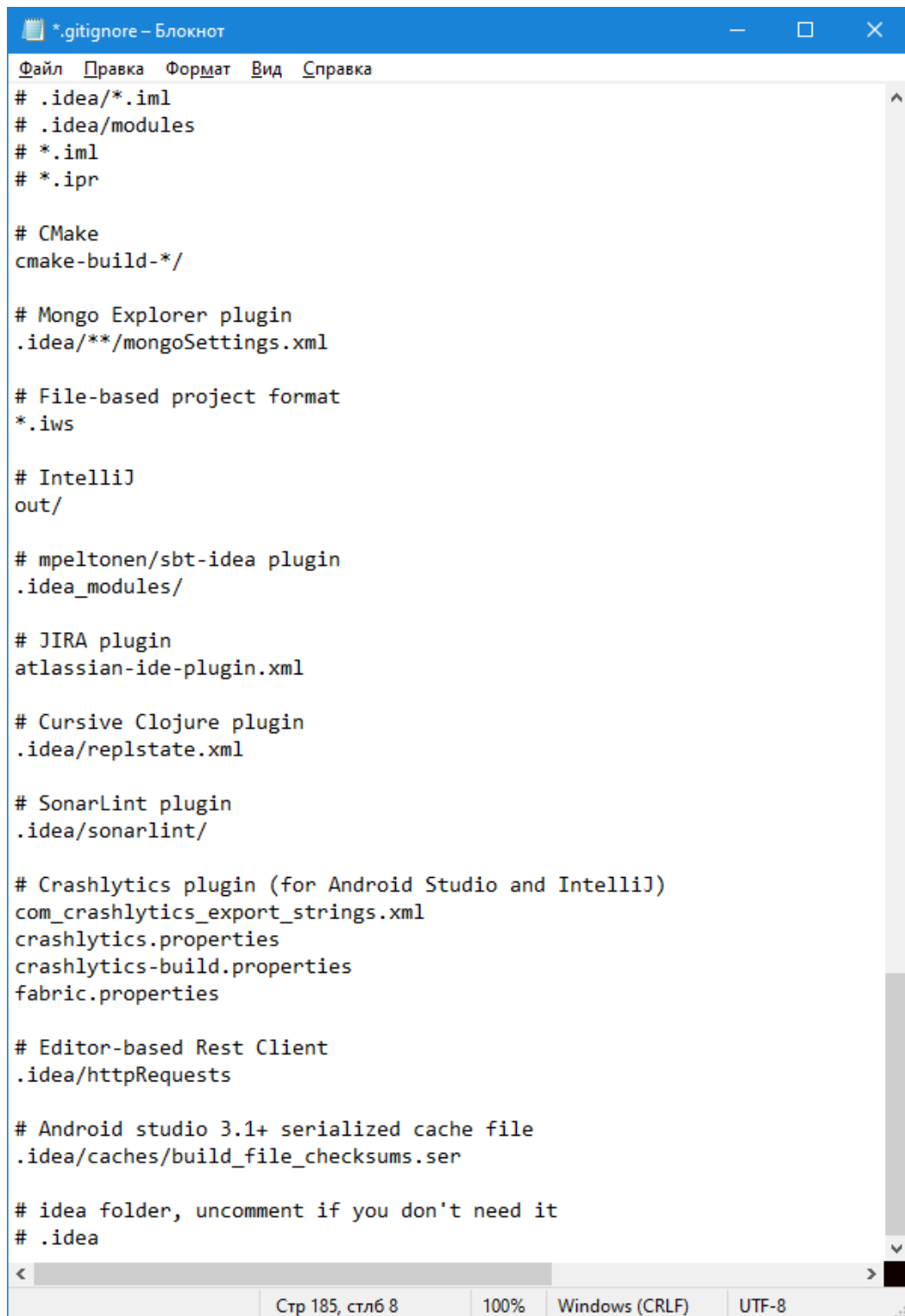
Рисунок 2 – клонирование репозитория

```
H:\cross\git\nazarov\2.4>git flow init

Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [H:/cross/git/nazarov/2.4/.git/hooks]
```

Рисунок 3 – организация репозитория согласно модели gitflow



```
*.gitignore – Блокнот
Файл  Правка  Формат  Вид  Справка
# .idea/*.iml
# .idea/modules
# *.iml
# *.ipr

# CMake
cmake-build-*/

# Mongo Explorer plugin
.idea/**/mongoSettings.xml

# File-based project format
*.iws

# IntelliJ
out/

# mpeltonen/sbt-idea plugin
.idea_modules/

# JIRA plugin
atlassian-ide-plugin.xml

# Cursive Clojure plugin
.idea/replstate.xml

# SonarLint plugin
.idea/sonarlint/

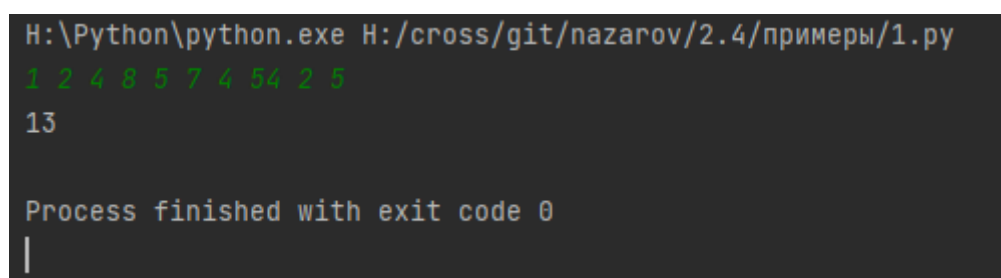
# Crashlytics plugin (for Android Studio and IntelliJ)
com_crashlytics_export_strings.xml
crashlytics.properties
crashlytics-build.properties
fabric.properties

# Editor-based Rest Client
.idea/httpRequests

# Android studio 3.1+ serialized cache file
.idea/caches/build_file_checksums.ser

# idea folder, uncomment if you don't need it
# .idea
```

Рисунок 4 – редактирование gitignore



```
H:\Python\python.exe H:/cross/git/nazarov/2.4/примеры/1.py
1 2 4 8 5 7 4 54 2 5
13

Process finished with exit code 0
|
```

Рисунок 5 – пример 1

```
H:\Python\python.exe H:/cross/git/nazarov/2.4/примеры/2.py
345 -43 567 456576 65 -6 4563 234 54 -66 -6784356
4
```

Рисунок 6 – пример 2

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

#Ввести список A из 10 элементов. Определить количество элементов, кратных 3 и индекс
#последнего такого элемента.

import sys

if __name__ == '__main__':
    # Ввести список одной строкой.
    A = list(map(int, input().split()))
    # Проверить количество элементов списка.
    if len(A) != 10:
        print("Неверный размер списка", file=sys.stderr)
        exit(1)

    counter = 0
    last = 0
    for i, x in enumerate(A):
        if x % 3 == 0:
            counter += 1
            last = i
    print("количество элементов, кратных 3:", counter, ", последний элемент:", last)
```

Рисунок 6 – программа для решения индивидуального задания 1

```
H:\Python\python.exe H:/cross/git/nazarov/2.4/задания/1.py
234 54867 4506 4576 45764 596 45 6 345 654
количество элементов, кратных 3: 7 , последний элемент: 9
```

Рисунок 7 - результат работы программы

```
H:\Python\python.exe H:/cross/git/nazarov/2.4/задания/2.py
Введите количество переменных: 5
1
354
2
4357567
-45
Сумма положительных элементов списка = 4357924
Произведение элементов списка, расположенных между максимальным по модулю и минимальным по модулю элементами = 708
Отсортированный список по убыванию: [4357567, 354, 2, 1, -45]

Process finished with exit code 0
```

Рисунок 8 – индивидуальное задание 2.

```

H:\cross\git\nazarov\2.4>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

H:\cross\git\nazarov\2.4>git merge develop
Updating 65c06db..a4aad9e
Fast-forward
 .gitignore | 150 ++++++-----
 .../1.py" | 23 +++++
 .../2.py" | 43 ++++++
 .../1.py" | 20 +++
 .../2.py" | 34 +++++
5 files changed, 223 insertions(+), 47 deletions(-)
create mode 100644 "\320\267\320\260\320\264\320\260\320\275\320\270\321\217\1.py"
create mode 100644 "\320\267\320\260\320\264\320\260\320\275\320\270\321\217\2.py"
create mode 100644 "\320\277\321\200\320\270\320\274\320\265\321\200\321\213\1.py"
create mode 100644 "\320\277\321\200\320\270\320\274\320\265\321\200\321\213\2.py"

```

Рисунок 9 – произвел слияние веток

```

H:\cross\git\nazarov\2.4>git push
Enumerating objects: 26, done.
Counting objects: 100% (26/26), done.
Delta compression using up to 12 threads
Compressing objects: 100% (20/20), done.
Writing objects: 100% (24/24), 5.43 KiB | 1.36 MiB/s, done.
Total 24 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), done.
To https://github.com/NikitaNazarov179/2.4.git
65c06db..a4aad9e main -> main

```

Рисунок 10 – пуш на удаленный репозиторий

### Контр. вопросы и ответы на них:

#### 1. Что такое списки в языке Python?

Список (list) – это структура данных для хранения объектов различных типов.

#### 2. Как осуществляется создание списка в Python?

Для создания списка нужно заключить элементы в квадратные скобки.

#### 3. Как организовано хранение списков в оперативной памяти?

Список является изменяемым типом данных. При его создании в памяти резервируется область, которую можно условно назвать некоторым “контейнером”, в котором хранятся ссылки на другие элементы данных в памяти. В отличие от таких типов данных как число или строка, содержимое “контейнера” списка можно менять.

#### 4. Каким образом можно перебрать все элементы списка?

```
for elem in my_list:
```

**5. Какие существуют арифметические операции со списками?**

+, \*

**6. Как проверить есть ли элемент в списке?**

Для того, чтобы проверить, есть ли заданный элемент в списке Python необходимо использовать оператор in.

**7. Как определить число вхождений заданного элемента в списке?**

list.count('элемент')

**8. Как осуществляется добавление (вставка) элемента в список?**

Метод insert можно использовать, чтобы вставить элемент в список.

**9. Как выполнить сортировку списка?**

list.sort()

**10. Как удалить один или несколько элементов из списка?**

Удалить элемент можно, написав его индекс в методе pop.

**11. Что такое списковое включение и как с его помощью осуществлять обработку списков?**

List Comprehensions чаще всего на русский язык переводят как абстракция списков или списковое включение, является частью синтаксиса языка, которая предоставляет простой способ построения списков.

**12. Как осуществляется доступ к элементам списков с помощью срезов?**

list[<начало среза>:<конец среза>:<шаг>]

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

Для работы со списками Python предоставляет следующие функции:

- len(L) - получить число элементов в списке L .
- min(L) - получить минимальный элемент списка L .
- max(L) - получить максимальный элемент списка L .
- sum(L) - получить сумму элементов списка L , если список L

содержит только числовые значения

**14. Как создать копию списка?**

Для создания копии списка необходимо использовать либо метод copy, либо использовать оператор среза

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

Отличие заключается в том, что метод `list.sort()` определён только для списков, в то время как `sorted()` работает со всеми итерируемыми объектами.