

**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 РАБОТЕ №2.4
дисциплины «Основы кроссплатформенного программирования»**

Выполнил:
Сластёнов Андрей Сергеевич
1 курс, группа ИТС-б-о-21-1,
11.03.02 «Инфокоммуникационные
технологии и системы связи»,
направленность (профиль) Сети связи и
системы коммутации,
очная форма обучения

(подпись)

Руководитель практики от университета:
Воронкин Р.А., канд. тех. наук,
доцент, доцент кафедры
инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г

Тема: работа со строками в языке Python.

Цель работы: приобретение навыков по работе со строками при написании программ с помощью языка программирования Python версии 3.x.

Порядок выполнения работы:

1. Создал общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT.

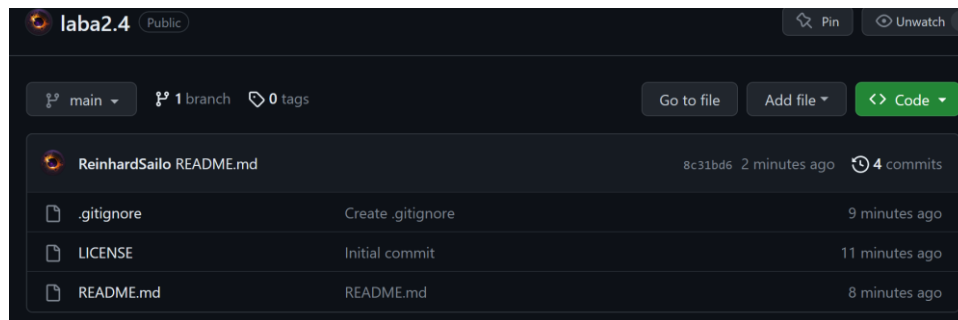


Рисунок 1. Новый репозиторий

2. В ходе данной лабораторной работы работал с моделью ветвления git flow

```
PS C:\Users\Andre> git clone https://github.com/ReinhardSailo/laba2.4.git
Cloning into 'laba2.4'...
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 11 (delta 1), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (11/11), done.
Resolving deltas: 100% (1/1), done.
PS C:\Users\Andre> cd laba2.4
PS C:\Users\Andre\laba2.4> 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? [C:/Users/Andre/laba2.4/.git/hooks]
PS C:\Users\Andre\laba2.4> git branch
* develop
main
PS C:\Users\Andre\laba2.4> |
```

Рисунок 2. Клонирование и модель ветвления git-flow

3. Пример №1. Условие примера: Ввести список A из 10 элементов, найти сумму элементов, меньших по модулю 5, и вывести ее на экран.

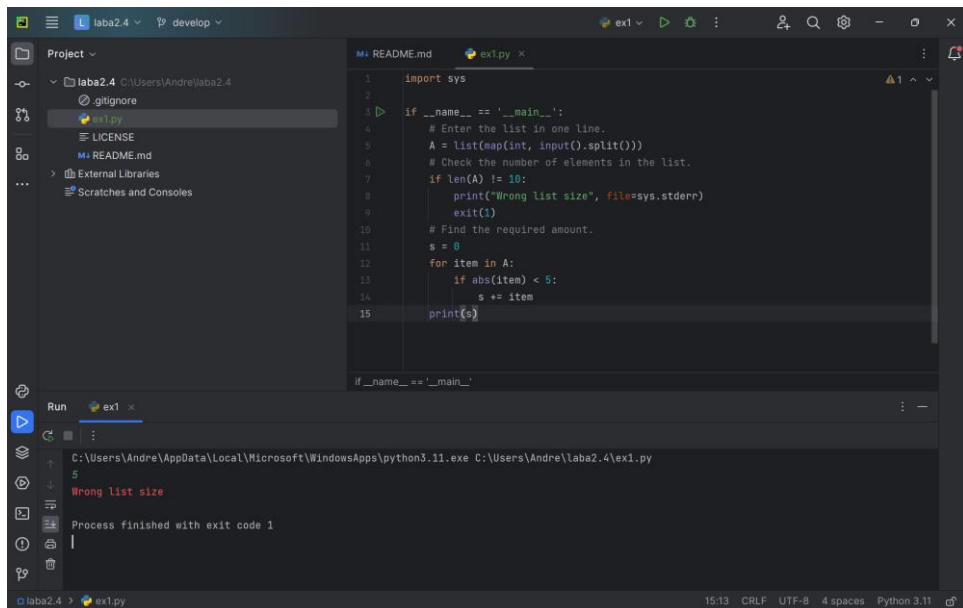


Рисунок 3. Программа и ее результат.

4. Пример №2. Условие примера: написать программу, которая для целочисленного списка определяет, сколько положительных элементов располагается между его максимальным и минимальным элементами.

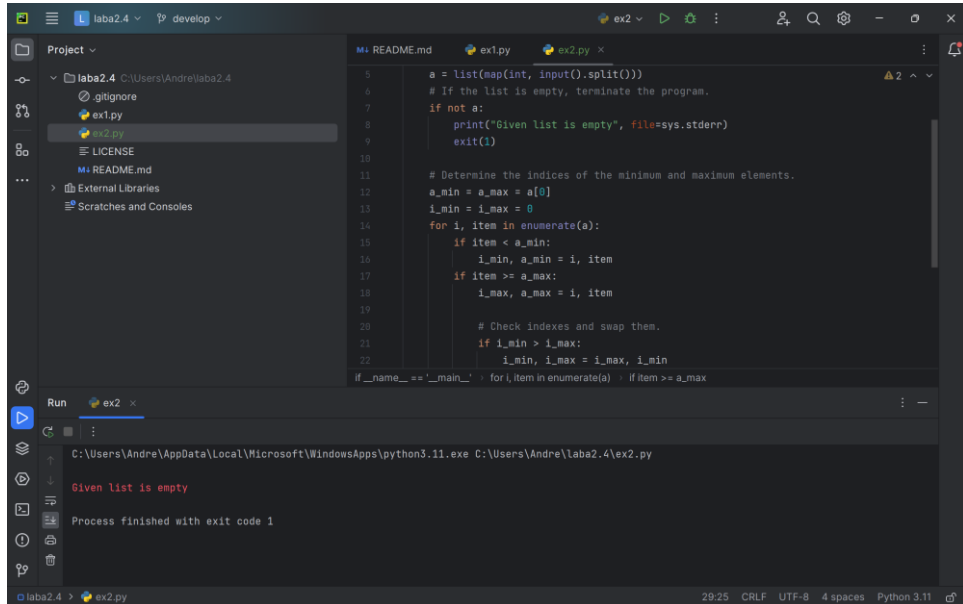


Рисунок 4. Программа и ее результат.

5. Индивидуальное задание №1

Вариант 11. Условие задания: Ввести список A из 10 элементов, найти сумму отрицательных элементов кратных 7, их количество и вывести результаты на экран.

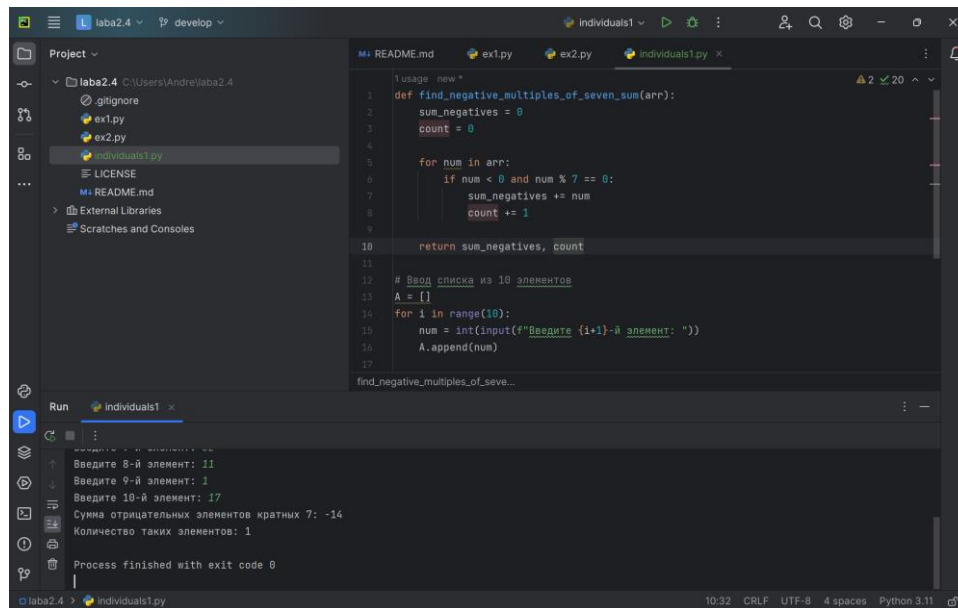


Рисунок 5. Программа и ее результат.

6. Индивидуальное задание №2

Вариант 11. Условие задания: В списке, состоящем из вещественных элементов, вычислить:

1. номер максимального по модулю элемента списка;
2. сумму элементов списка, расположенных после первого положительного элемента.

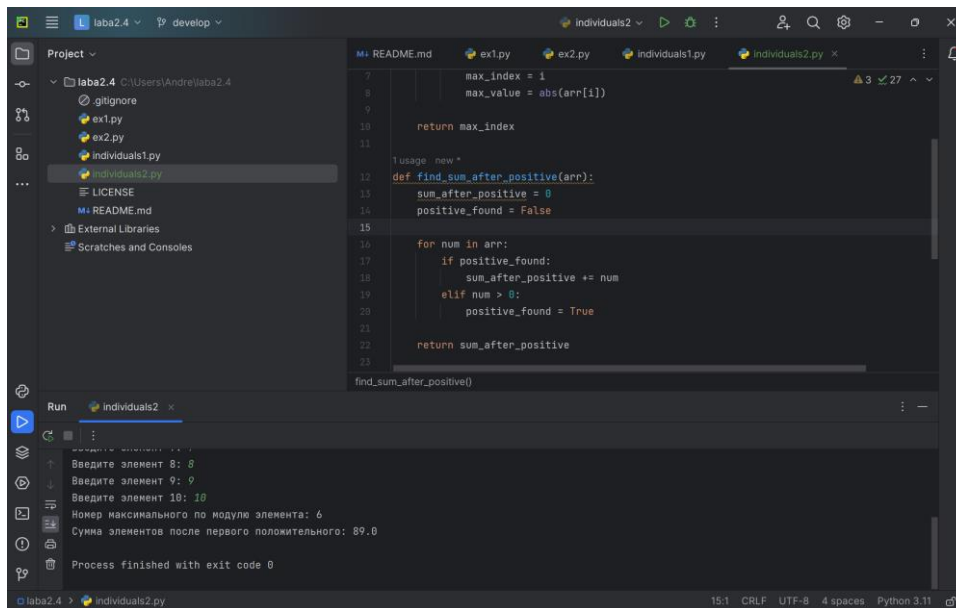


Рисунок 6. Программа и ее результат.

7. Слил ветку develop с веткой main и отправил на удаленный сервер – Github.

```

PS C:\Users\Andre> cd laba2.4
PS C:\Users\Andre\laba2.4> git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
PS C:\Users\Andre\laba2.4> git merge develop
Updating 8c31bd6..31c8b73
Fast-forward
 ex1.py      | 15 ++++++
 ex2.py      | 29 ++++++
 individuals1.py | 23 ++++++
 individuals2.py | 33 ++++++
 4 files changed, 100 insertions(+)
 create mode 100644 ex1.py
 create mode 100644 ex2.py
 create mode 100644 individuals1.py
 create mode 100644 individuals2.py
PS C:\Users\Andre\laba2.4> git push

```

Рисунок 7. Слил ветки.

Ответы на контрольные вопросы:

Что такое списки в языке Python? Список (list) – это структура данных для хранения объектов различных типов. В нем можно хранить объекты

различных типов. Размер списка не статичен, его можно изменять.

Список по

своей природе является изменяемым типом данных. Переменная, определяемая как список, содержит ссылку на структуру в памяти, которая в

свою очередь хранит на какие-либо другие объекты или структуры.

2. Как осуществляется создание списка в Python? Для создания списка нужно заключить элементы в квадратные скобки.

3. Как организовано хранение списков в оперативной памяти? При создании списка в памяти резервируется область, которую можно условно назвать некоторым “контейнером”, в котором хранятся ссылки другие элементы данных в памяти. В отличие от таких типов данных число или строка, содержимое “контейнера” списка можно менять.

4. Каким образом можно перебрать все элементы списка? Читать элементы списка можно с помощью следующего цикла:
`my_list = ['один', 'два', 'три', 'четыре', 'пять']`
`for elem in my_list: print(elem)`

5. Какие существуют арифметические операции со списками? Для объединения списков можно использовать оператор сложения (+).

Список

можно повторить с помощью оператора умножения (*).

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? Метод `count` можно использовать для определения числа сколько раз данный элемент встречается в списке.

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

`append` можно использовать для добавления элемента в список. Метод `insert`

можно использовать, чтобы вставить элемент в список.

9. Как выполнить сортировку списка? Для сортировки списка нужно использовать метод `sort`. Для сортировки списка в порядке убывания необходимо вызвать метод `sort` с аргументом `reverse=True`.

10. Как удалить один или несколько элементов из списка? Удалить элемент можно, написав его индекс в методе `pop`. Если не указывать индекс,

то функция удалит последний элемент. Элемент можно удалить с помощью

метода `remove`. Оператор `del` можно использовать для тех же целей. Можно удалить несколько элементов с помощью оператора среза. Можно удалить все

элементы из списка с помощью метода `clear`.

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.

В языке Python есть две очень мощные функции для работы с коллекциями:

`map` и `filter`. Они позволяют использовать функциональный стиль программирования, не прибегая к помощи циклов, для работы с такими типами как `list`, `tuple`, `set`, `dict` и т.п. Списковое включение позволяет обойтись

без этих функций.

12. Какие существуют функции агрегации для работы со списками? Для работы со списками Python предоставляет следующие функции: 1. `len(L)` -

получить число элементов в списке L 2. `min(L)` - получить минимальный элемент списка L 3. `max(L)` - получить максимальный элемент списка L 4. `sum(L)` - получить сумму элементов списка L, если список L содержит только числовые значения.

13. Как создать копию списка? `copy.copy(x)`

14. Самостоятельно изучите функцию `sorted` языка Python. В чем ее отличие от метода `sort` списков? Функция `sorted()` в Python возвращает отсортированный список из элементов в итерируемом объекте. `list.sort()` на 13% быстрее, чем `sorted()`.

15. Самостоятельно изучите функцию `sorted` языка Python. В чем ее отличие от метода `sort` списков? Функция `sort()` очень похожа на `sorted()`, но в

отличие от `sorted` она ничего не возвращает и не вносит изменений в исходную

последовательность. Более того, `sort()` является методом класса `list` и может

использоваться только со списками. Синтаксис: `List_name.sort(key, reverse=False)` Параметры: `ключ`: Функция, которая служит ключом для сравнения сортировки. `реверс`: Если `true`, то список сортируется в порядке убывания.

Вывод: в ходе выполнения лабораторной работы приобретены навыки по работе со списками при написании программ с помощью языка программирования Python версии 3.x.