

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:
Пазенко Данила Сергеевич
1 курс, группа ИТС-б-о-22-1,
11.03.02 «Инфокоммуникационные
технологии и системы связи»,
направленность (профиль)
«Инфокоммуникационные системы и
сети», очная форма обучения

(подпись)

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

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Тема: работа со списками в языке Python.

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

Порядок выполнения работы:

1. Создал новый репозиторий и клонировал его на свой компьютер

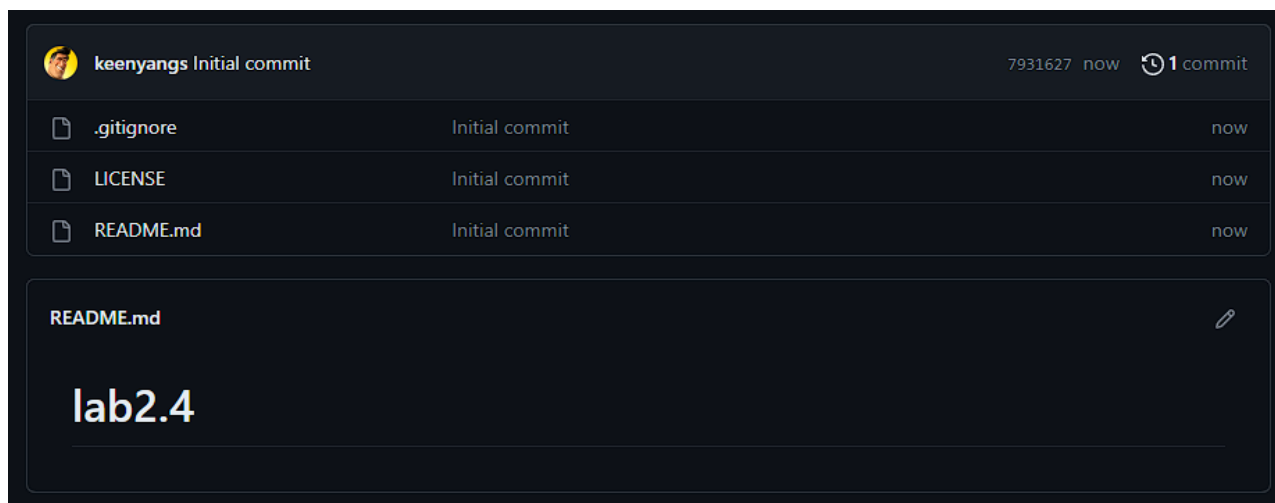


Рисунок 1 – Новый репозиторий

2. Проклонировал репозиторий на свой компьютер. В ходе данной лабораторной работы работал с моделью ветвления git-flow.

```
stron@KPA3 MINGW64 /v/programms/project (master)
$ git clone https://github.com/keenyangs/lab2.4.git
Cloning into 'lab2.4'...
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.

stron@KPA3 MINGW64 /v/programms/project (master)
$ cd /v/programms/project/lab2.4

stron@KPA3 MINGW64 /v/programms/project/lab2.4 (main)
$ 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? [v:/programms/project/lab2.4/.git/hooks]
```

Рисунок 2 – Клонирование репозитория

3. Создал проект PyCharm в папке репозитория.

Пример №1.

Добавил новый файл ex1.py.

Условие примера: Ввести список A из 10 элементов, найти сумму элементов, меньших по модулю 5, и вывести ее на экран.

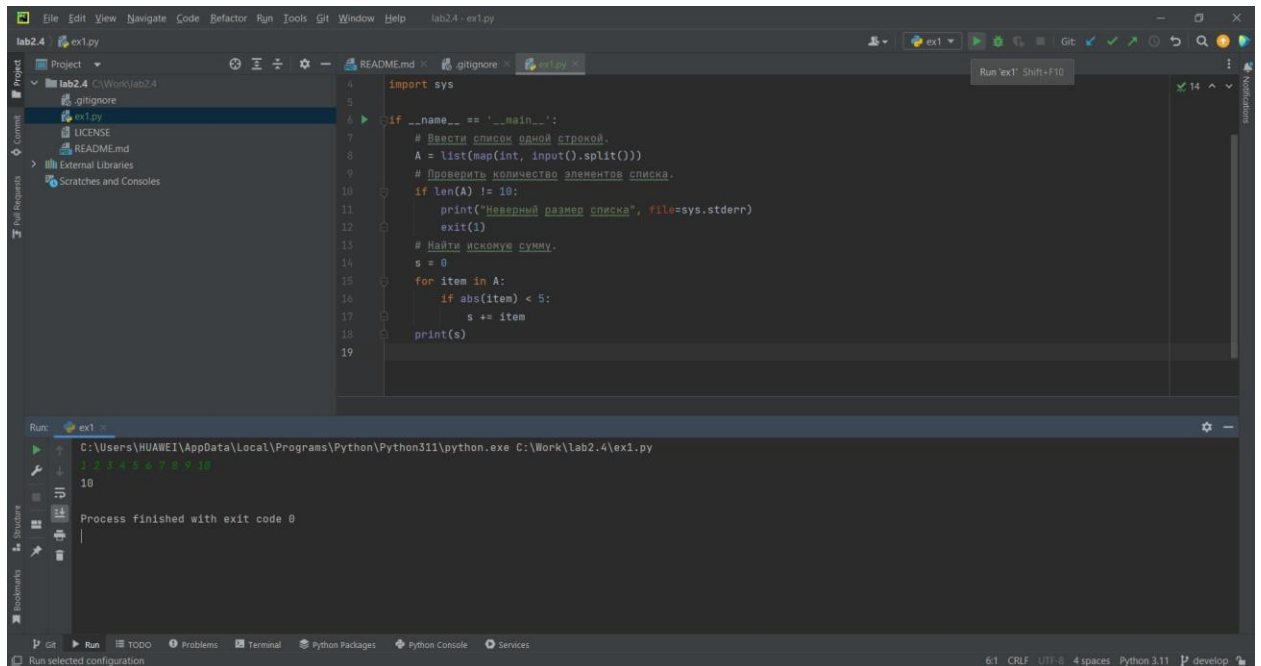


Рисунок 3 – Программа и ее результат

4. Создал новый файл под названием *ex2.py*

Пример №2.

Условие примера: написать программу, которая для целочисленного списка определяет, сколько положительных элементов располагается между его максимальным и минимальным элементами.

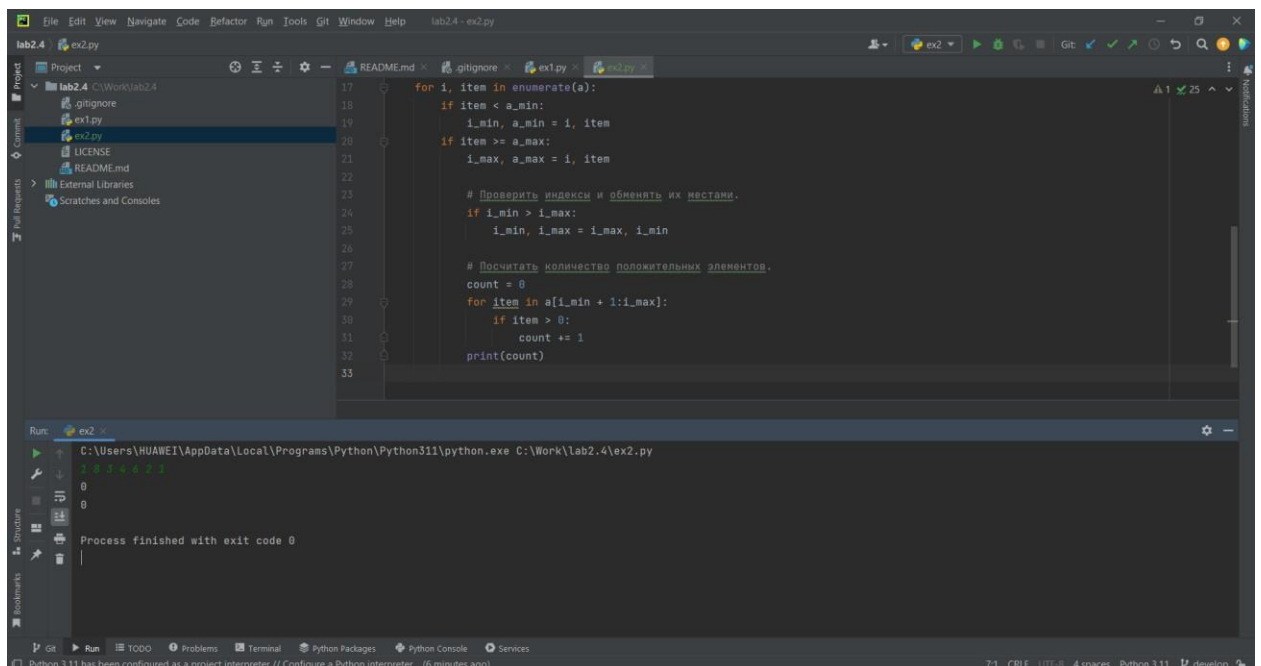


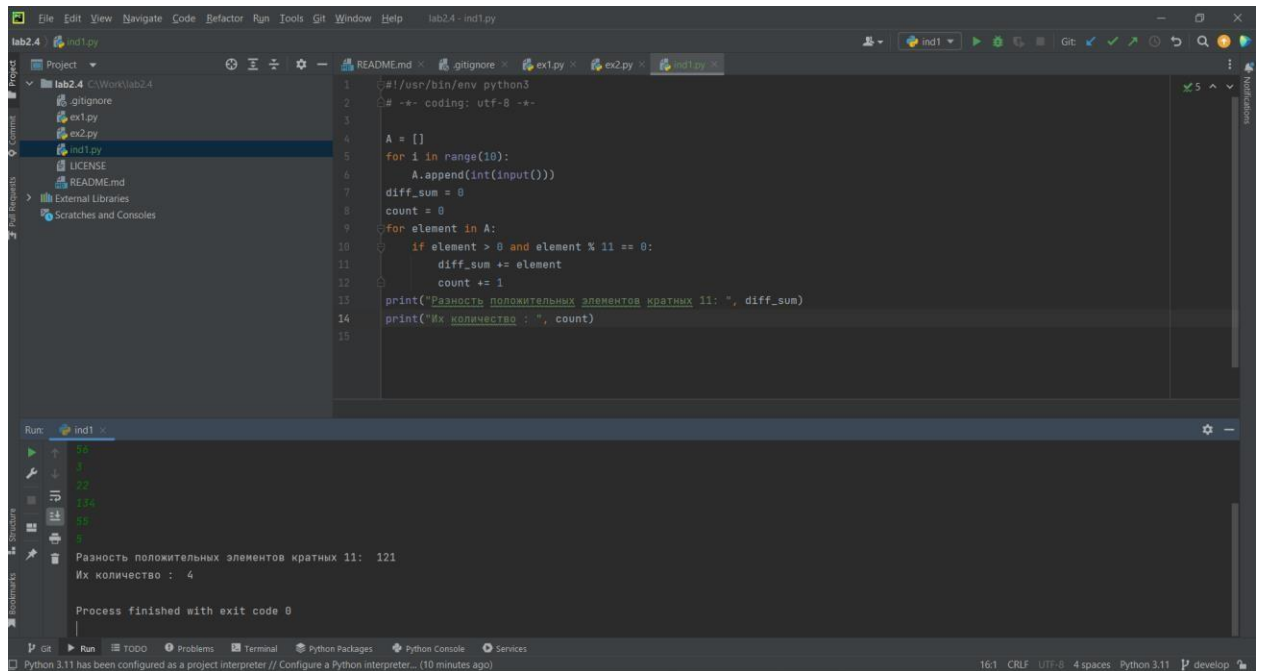
Рисунок 4 – Программа и ее результат

5. Индивидуальное задание №1.

Создал новый файл под названием *ind1.py*

Вариант 14.

Условие задания: ввести список A из 10 элементов, найти разность положительных элементов кратных 11, их количество и вывести результаты на экран.



The screenshot shows a Python IDE with a file named *ind1.py* open. The code in the file is as follows:

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 A = []
5 for i in range(10):
6     A.append(int(input()))
7 diff_sum = 0
8 count = 0
9 for element in A:
10     if element > 0 and element % 11 == 0:
11         diff_sum += element
12         count += 1
13 print("Разность положительных элементов кратных 11: ", diff_sum)
14 print("Их количество : ", count)
```

Below the code editor, the Run window shows the execution results:

```
Run: ind1
20
3
22
134
88
5
Разность положительных элементов кратных 11: 121
Их количество : 4
Process finished with exit code 0
```

Рисунок 5 – Программа и ее результат

6. Индивидуальное задание №2.

Создал новый файл под названием *ind2.py*

Вариант 14.

Условие задания:

14. В списке, состоящем из вещественных элементов, вычислить:

1. количество элементов списка, больших C;
2. произведение элементов списка, расположенных после максимального по модулю элемента.

Преобразовать список таким образом, чтобы сначала располагались все отрицательные элементы, а потом - все положительные (элементы, равные 0, считать положительными).

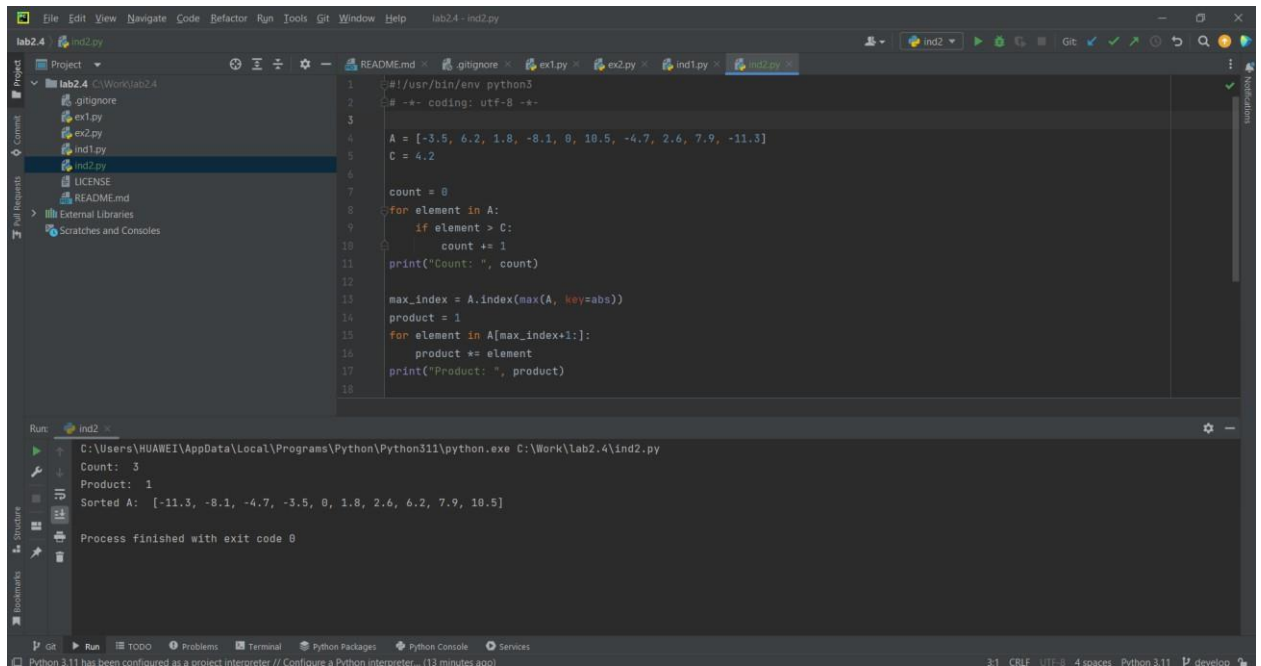


Рисунок 6 – Результат программы

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

```

stron@KPA3 MINGW64 /v/programms/project/lab2.4 (develop)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

stron@KPA3 MINGW64 /v/programms/project/lab2.4 (main)
$ git merge develop
Updating 7931627..bcf31a0
Fast-forward
 .gitignore | 4 ++++
 README.md | 3 ++-
 ex1.py     | 18 ++++++
 ex2.py     | 32 ++++++
 indi1.py   | 14 ++++++
 indi2.py   | 23 ++++++
 6 files changed, 93 insertions(+), 1 deletion(-)
 create mode 100644 ex1.py
 create mode 100644 ex2.py
 create mode 100644 indi1.py
 create mode 100644 indi2.py

```

Рисунок 7 – Слияние ветки develop с основной веткой main

Ссылка: <https://github.com/keenyangs/lab2.4>

Контрольные вопросы:

1. 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 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? 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.