

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»

Выполнила: студентка 1 курса,
группы ИВТ-б-о-21-1
Диченко Дина Алексеевна

Ставрополь 2022

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

Практическая часть:

1. Создала и клонировала репозиторий.

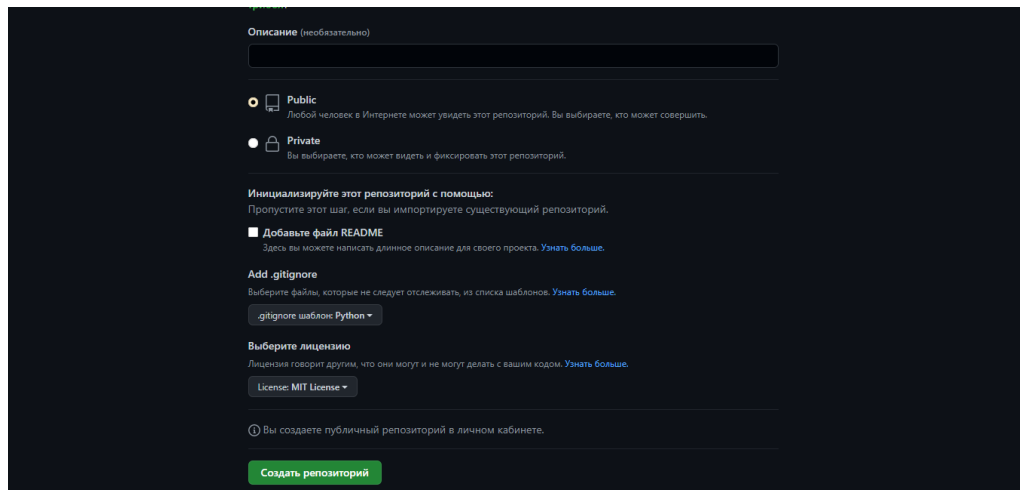


Рисунок 1. Создание репозитория

```
Microsoft Windows [Version 10.0.19043.1706]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\Дом\Desktop\Дина\Документы\СКФУ\Программирование>git clone https://github.com/DinaDichenko/Lab2.4.git
Cloning into 'Lab2.4'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.

C:\Users\Дом\Desktop\Дина\Документы\СКФУ\Программирование>
```

Рисунок 2. Клонирование репозитория

2. Изменила файл .gitignore.

```
# Создано https://www.toptal.com/developers/gitignore/api/python , pycharm
# Редактировать по адресу https://www.toptal.com/developers/gitignore?templates=python , pycharm

### PyCharm ###
# Охватывает IDE JetBrains: IntelliJ, RubyMine, PhpStorm, AppCode, PyCharm, CLion, Android Studio, WebStorm и Rider
# Ссылка: https://intellij-support.jetbrains.com/hc/en-us/articles/206544839

# Пользовательские вещи
.идея /**/workspace.xml
.идея /**/tasks.xml
.идея /**/usage.statistics.xml
.идея /**/словари
.идея /**/полка
.идея/

# AWS User-specific
.идея /**/aws.xml

# Сгенерированные файлы
.идея /**/contentModel.xml

# Чувствительные или файлы с высоким оттоком
```

Рисунок 4. Редактирование файла .gitignore

3. Организовала репозиторий в соответствии с моделью ветвления git flow.

```
C:\Users\Дом\Desktop\Дина\Документы\СКФУ\Программирование\Lab2.4>git flow init

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

Which branch should be used for integration of the "next release"?
  - main
Branch name for "next release" development: [] mail
fatal: Local branch 'mail' does not exist.

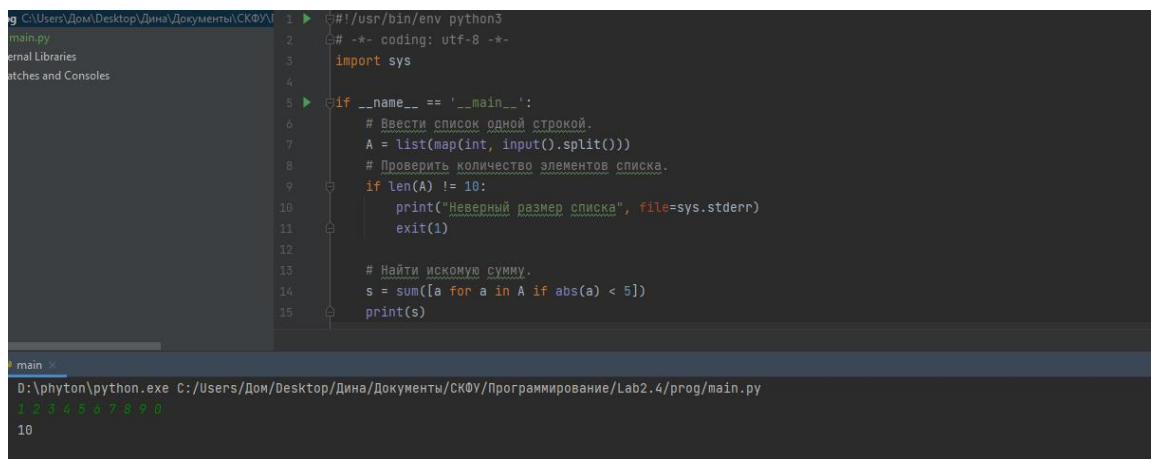
C:\Users\Дом\Desktop\Дина\Документы\СКФУ\Программирование\Lab2.4>git flow init

Which branch should be used for integration of the "next release"?
  - main
Branch name for "next release" development: [] main

How to name your supporting branch prefixes?
Feature branches? [feature/] feat
Bugfix branches? [bugfix/] bug
Release branches? [release/] rel
Hotfix branches? [hotfix/] hot
Support branches? [support/] sup
Version tag prefix? [] pre
Hooks and filters directory? [C:/Users/Дом/Desktop/Дина/Документы/СКФУ/Программирование/Lab2.4/.git/hooks] hook
```

Рисунок 5. Организация репозитория в соответствии с git-flow

4. Проработала примеры из лабораторной работы, сохранила изменения на удаленном репозитории.

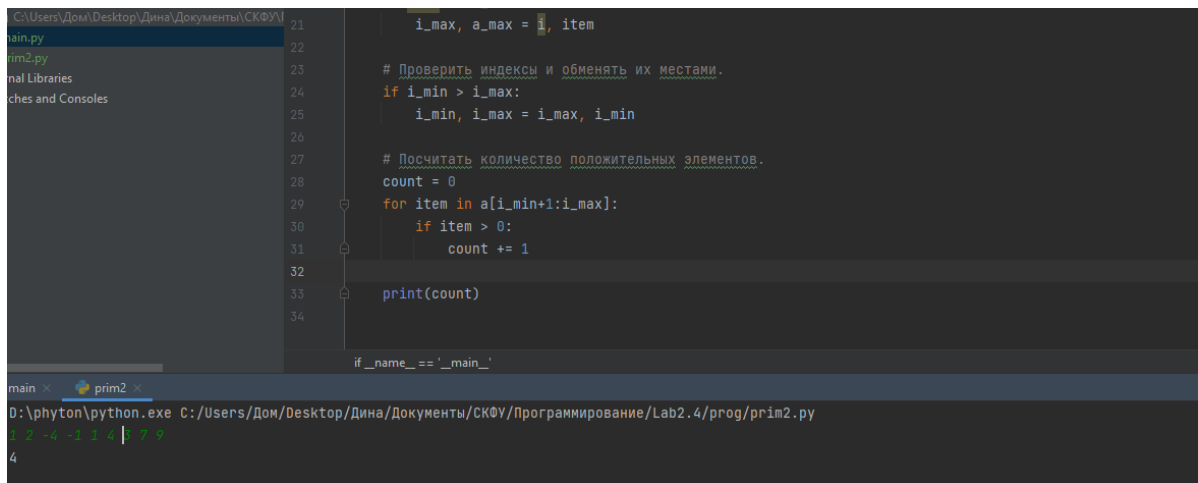


```
C:\Users\Дом\Desktop\Дина\Документы\СКФУ\Программирование\Lab2.4>python main.py
1
2
3
4
5
6
7
8
9
0
10

1
2
3
4
5
6
7
8
9
0
10

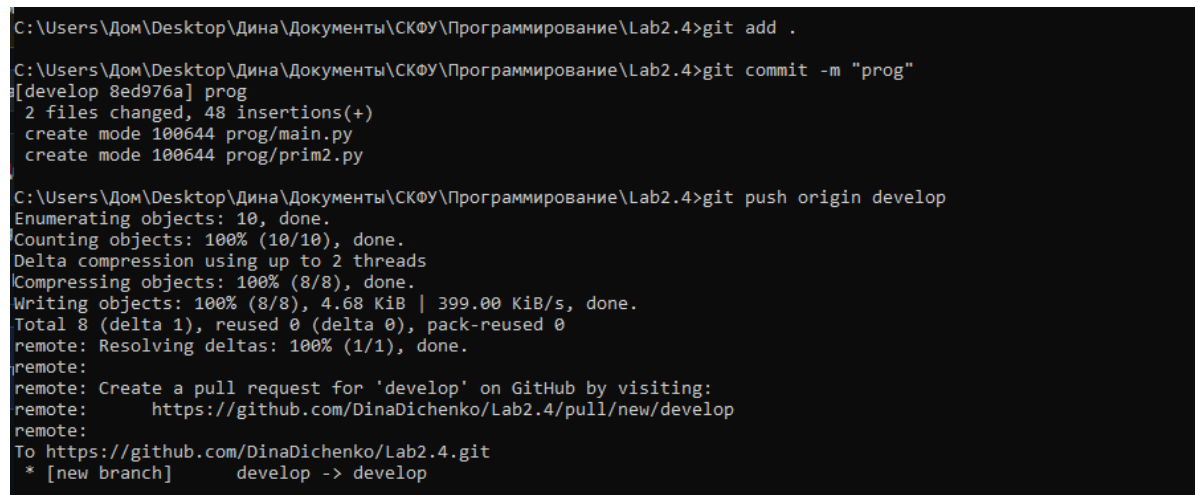
# Ввести список одной строкой.
A = list(map(int, input().split()))
# Проверить количество элементов списка.
if len(A) != 10:
    print("Неверный размер списка", file=sys.stderr)
    exit(1)
# Найти искомую сумму.
s = sum([a for a in A if abs(a) < 5])
print(s)
```

Рисунок 6. Результат работы программы



```
21     i_max, a_max = i, item
22
23     # Проверить индексы и обменять их местами.
24     if i_min > i_max:
25         i_min, i_max = i_max, i_min
26
27     # Посчитать количество положительных элементов.
28     count = 0
29     for item in a[i_min+1:i_max]:
30         if item > 0:
31             count += 1
32
33     print(count)
34
35 if __name__ == '__main__':
36     main()
37
38 main x prim2 x
D:\phyton\python.exe C:/Users/Дом/Desktop/Дина/Документы/СКФУ/Программирование/Lab2.4/prog/prim2.py
1 2 -3 -1 4 | 7
4
```

Рисунок 7. Результат работы программы

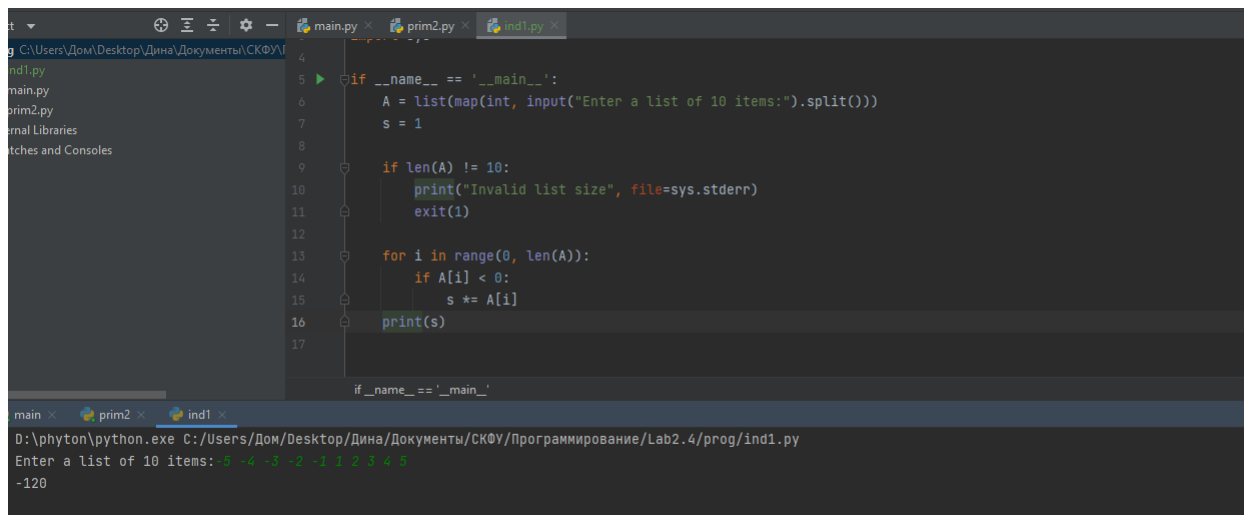


```
C:\Users\Дом\Desktop\Дина\Документы\СКФУ\Программирование\Lab2.4>git add .
C:\Users\Дом\Desktop\Дина\Документы\СКФУ\Программирование\Lab2.4>git commit -m "prog"
[develop 8ed976a] prog
 2 files changed, 48 insertions(+)
 create mode 100644 prog/main.py
 create mode 100644 prog/prim2.py
C:\Users\Дом\Desktop\Дина\Документы\СКФУ\Программирование\Lab2.4>git push origin develop
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 2 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (8/8), 4.68 KiB | 399.00 KiB/s, done.
Total 8 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:   https://github.com/DinaDichenko/Lab2.4/pull/new/develop
remote:
To https://github.com/DinaDichenko/Lab2.4.git
 * [new branch]      develop -> develop
```

Рисунок 8. Сохранение изменений в удаленном репозитории

5. Выполнила индивидуальные задания (вариант 7).

Задание 1. Ввести список А из 10 элементов, найти произведение отрицательных элементов и вывести его на экран.



```
4
5 if __name__ == '__main__':
6     A = list(map(int, input("Enter a list of 10 items:").split()))
7     s = 1
8
9     if len(A) != 10:
10        print("Invalid list size", file=sys.stderr)
11        exit(1)
12
13    for i in range(0, len(A)):
14        if A[i] < 0:
15            s *= A[i]
16    print(s)
17
18 if __name__ == '__main__':
```

main x prim2 x ind1 x

D:\python\python.exe C:/Users/Дом/Desktop/Дина/Документы/СКФУ/Программирование/Lab2.4/prog/ind1.py

Enter a list of 10 items: 1 -2 -1 -2 -1 1 1 1 1 1

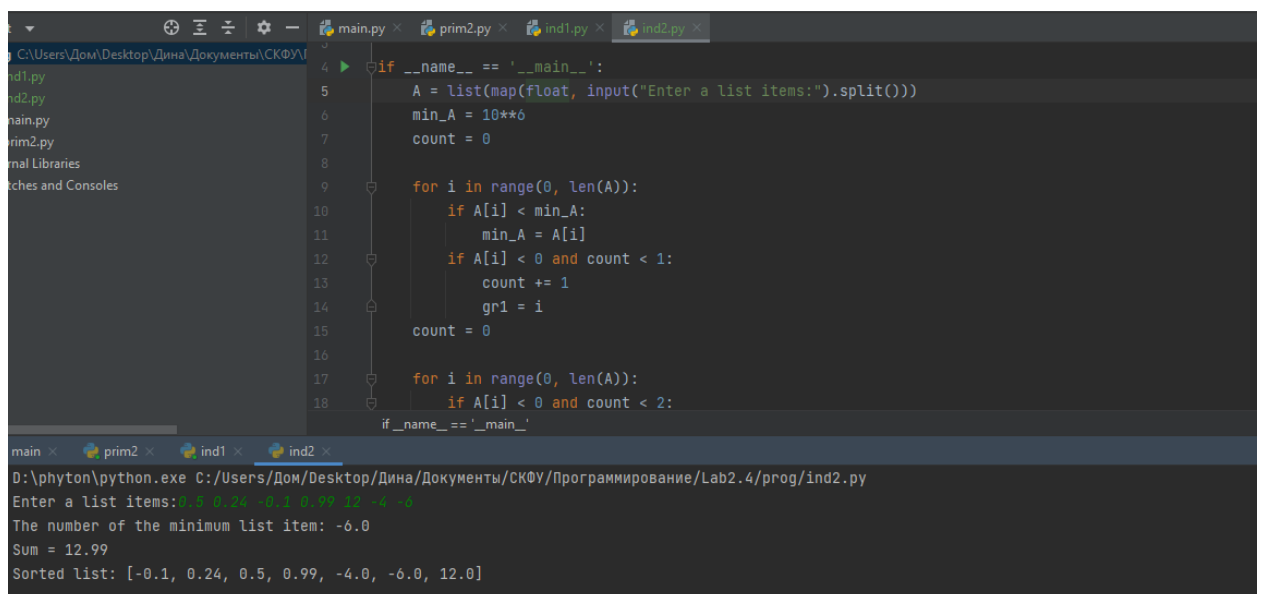
-120

Рисунок 9. Результат работы программы

Задание 2. В списке, состоящем из вещественных элементов, вычислить:

1. номер минимального элемента списка;
2. сумму элементов списка, расположенных между первым и вторым отрицательными элементами.

Преобразовать список таким образом, чтобы сначала располагались все элементы, модуль которых не превышает 1, а потом - все остальные.



```
4 if __name__ == '__main__':
5     A = list(map(float, input("Enter a list items:").split()))
6     min_A = 10**6
7     count = 0
8
9     for i in range(0, len(A)):
10        if A[i] < min_A:
11            min_A = A[i]
12        if A[i] < 0 and count < 1:
13            count += 1
14            gr1 = i
15    count = 0
16
17    for i in range(0, len(A)):
18        if A[i] < 0 and count < 2:
```

main x prim2 x ind1 x ind2 x

D:\python\python.exe C:/Users/Дом/Desktop/Дина/Документы/СКФУ/Программирование/Lab2.4/prog/ind2.py

Enter a list items: 0.1 0.24 -0.1 0.99 12 -4 -0.1

The number of the minimum list item: -6.0

Sum = 12.99

Sorted list: [-0.1, 0.24, 0.5, 0.99, -4.0, -6.0, 12.0]

Рисунок 10. Результат работы программы

Ответы на вопросы:

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?

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?

Можно развернуть порядок элементов в списке с помощью метода reverse.

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.

Если требуется, чтобы элемент отсутствовал в списке, необходимо использовать оператор not 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 `insert` можно использовать, чтобы вставить элемент в список.

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

9. Как выполнить сортировку списка?

Для сортировки списка нужно использовать метод `sort`.

10. Как удалить один или несколько элементов из списка?

Удалить элемент можно, написав его индекс в методе `pop`.

Элемент можно удалить с помощью метода `remove`.

Оператор `del` можно использовать для тех же целей.

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. Как осуществляется доступ к элементам списков с помощью срезов?

Слайсы (срезы) являются очень мощной составляющей Python, которая позволяет быстро и лаконично решать задачи выборки элементов из списка. Выше уже был пример использования слайсов, здесь разберем более подробно работу с ними. Создадим список для экспериментов:

```
a = [i for i in range(10)].
```

Слайс задается тройкой чисел, разделенных запятой: start:stop:step. Start – позиция с которой нужно начать выборку, stop – конечная позиция, step – шаг. При этом необходимо помнить, что выборка не включает элемент определяемый stop.

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.

Для функций min и max элементы списка должны быть сравнимы между собой.

14. Как создать копию списка?

Поэтому 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?

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.

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