

**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 программной инженерии»
Тема: «Работа со списками в языке Python»**

**Выполнила:
студентка 2 курса
группы Пиж-б-о-21-1
Логвинов Иван
Васильевич**

Ставрополь 2022

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

Ход работы:

1. Был создан репозиторий в Github в который были добавлены правила gitignore для работы IDE PyCharm, была выбрана лицензия MIT, сам репозиторий был клонирован на локальный сервер и был организован в соответствии с моделью ветвления git-flow.

Владелец * / Имя репозитория *

IvanLogvinov11 / lab2_4 ✓

Отличные названия репозитория короткие и запоминающиеся. Вам нужно вдохновение? Как насчет легендарных очков octo?

Описание (необязательно)

☐ Общедоступный
Любой пользователь Интернета может увидеть этот репозиторий. Вы выбираете, кто может совершать.

☐ Личное
Вы выбираете, кто может видеть и фиксировать в этом репозитории.

Инициализируйте этот репозиторий с помощью:
Пропустите этот шаг, если вы импортируете существующий репозиторий.

☒ Добавьте файл README
Здесь вы можете написать длинное описание для вашего проекта. [Узнать больше.](#)

Добавить .gitignore
Выберите, какие файлы не отслеживать, из списка шаблонов. [Узнать больше.](#)

.шаблон gitignore: Python ▼

Выберите лицензию
Лицензия сообщает другим пользователям, что они могут и чего не могут делать с вашим кодом. [Узнать больше.](#)

Лицензия: MIT License ▼

Это установит `main` в качестве ветки по умолчанию. Измените имя по умолчанию в настройках.

i Вы создаете общедоступный репозиторий в своем личном кабинете.

Создать репозиторий

Рисунок 1.1 – Создание репозитория

```
C:\Users\Иван\Desktop>git clone https://github.com/IvanLogvinov11/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.
C:\Users\Иван\Desktop>
```

Рисунок 1.2 – Клонирование репозитория

2. Был создана папка PyCharm в которой хранятся примеры из лабораторной работы.

```
C:\Users\Иван\AppData\Local\Programs\Python\Python39\pythonw
2 3 4 1 2 6 5 4 3 2
21

Process finished with exit code 0
|
```

Рисунок 2.1 – Результат работы первого примера

```
C:\Users\Иван\AppData\Local\Programs\Python\Python39
5 -3 4 5 8 7 -5
1

Process finished with exit code 0
|
```

Рисунок 2.2 – Результат работы второго примера



 p1	14.12.2022 21:04	JetBrains PyChar...	1 КБ
 p2	14.12.2022 21:09	JetBrains PyChar...	2 КБ

Рисунок 2.3 – Папка с примерами из лабораторной работы

3. Было выполнено два индивидуальных задания согласно 10 варианту

Задание 1

Ввести список А из 10 элементов, найти произведение положительных элементов кратных 3, их количество и вывести результаты на экран.

```
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 import sys
4
5 ▶ if __name__ == '__main__':
6     # Ввести список одной строкой.
7     a = list(map(int, input().split()))
8     # Если список пуст, завершить программу.
9     if not a:
10         print("Заданный список пуст", file=sys.stderr)
11         exit(1)
12
13     s = 1
14     count=0
15     for i in a:
16         if i % 3 == 0:
17             count += 1
18             s *= i
19     print(count, s)
```

```
C:\Users\Иван\AppData\Local\Programs\Python\Pyt
12 3 6 1 2 5 36 11 27 4
5 209952

Process finished with exit code 0
```

Рисунок 3.1 – Результат работы программы

Задание 2

В списке, состоящем из вещественных элементов, вычислить:

1. максимальный элемент списка;
2. сумму элементов списка, расположенных до последнего положительного элемента.

Сжать список, удалив из него все элементы, модуль которых находится в интервале $[a, b]$.

Освободившиеся в конце списка элементы заполнить нулями.

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

if __name__ == '__main__':
    a = list(map(float, input().split()))
    a1 = list(range(0, len(a)))
    in1 = int(input("range 1: "))
    in2 = int(input("range 2: "))
    min_a = a[0]
    pos = 0

    for i in range(0, len(a)):
        if a[i] < min_a:
            min_a = a[i]
            min_i = i

    for i in a:
        if i < 0:
            for j in a[i:len(a)]:
                pos += j
        else:
            break

    for i in range(0, len(a)):
        if (a[i] >= in1) and (a[i] <= in2):
            a.remove(a[i])
            a.append(0)
    for i in range(0, len(a)):
        print(a[i])
    print('\nminimum is: ', min_a, 'sum is', pos)
```

```
C:\Users\Иван\AppData\Local\Programs\Python\Python39\p
1.2 3.5 5.3 -3.2 4.9 1.7
range 1: 2
range 2: 4
1.2
5.3
-3.2
4.9
1.7
0

minimum is: -3.2 sum is 0

Process finished with exit code 0
```

Рисунок 3.2 – Результат работы программы

4. Был осуществлен коммит и слияние веток main и develop, также были запущены изменения на удаленный сервер.

```
C:\Users\Иван\Desktop\lab2_4>git add .
C:\Users\Иван\Desktop\lab2_4>git commit -m "new"
[develop b9999ad] new
10 files changed, 129 insertions(+)
create mode 100644 pycharm/.idea/.gitignore
create mode 100644 pycharm/.idea/inspectionProfiles/profiles_settings.xml
create mode 100644 pycharm/.idea/misc.xml
create mode 100644 pycharm/.idea/modules.xml
create mode 100644 pycharm/.idea/pycharm.iml
create mode 100644 pycharm/.idea/vcs.xml
create mode 100644 pycharm/ind1.py
create mode 100644 pycharm/ind2.py
create mode 100644 pycharm/p1.py
create mode 100644 pycharm/p2.py
C:\Users\Иван\Desktop\lab2_4>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
C:\Users\Иван\Desktop\lab2_4>
```

Рисунок 4.1 – Коммит изменений и переход на ветку main

```

C:\Users\Иван\Desktop\lab2_4>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

C:\Users\Иван\Desktop\lab2_4>git merge develop
Updating 5018865..b9999ad
Fast-forward
 pycharm/.idea/.gitignore          | 3 +++
 .../.idea/inspectionProfiles/profiles_settings.xml | 6 +++++
 pycharm/.idea/misc.xml            | 4 +++
 pycharm/.idea/modules.xml         | 8 ++++++
 pycharm/.idea/pycharm.iml         | 8 ++++++
 pycharm/.idea/vcs.xml             | 6 +++++
 pycharm/ind1.py                   | 18 ++++++++++++++++++
 pycharm/ind2.py                   | 31 ++++++++++++++++++++++
 pycharm/p1.py                     | 17 ++++++++
 pycharm/p2.py                     | 28 ++++++++
10 files changed, 129 insertions(+)
create mode 100644 pycharm/.idea/.gitignore
create mode 100644 pycharm/.idea/inspectionProfiles/profiles_settings.xml
create mode 100644 pycharm/.idea/misc.xml
create mode 100644 pycharm/.idea/modules.xml
create mode 100644 pycharm/.idea/pycharm.iml
create mode 100644 pycharm/.idea/vcs.xml
create mode 100644 pycharm/ind1.py
create mode 100644 pycharm/ind2.py
create mode 100644 pycharm/p1.py
create mode 100644 pycharm/p2.py

```

Рисунок 4.2 – Слияние веток и пуш изменений на удаленный сервер

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

Контрольные 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?

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.