

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт перспективной инженерии
Департамент цифровых, робототехнических систем и электроники

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №7
дисциплины
«Основы кроссплатформенного программирования»
Вариант 1

Выполнил:
Бакулин Вадим Романович
2 курс, группа ИТС-б-о-23-1,
11.03.02 «Инфокоммуникационные
технологии и системы связи», очная
форма обучения

(подпись)

Проверил:
Доцент департамента цифровых,
робототехнических систем и
электроники Воронкин Р.А.

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2024 г

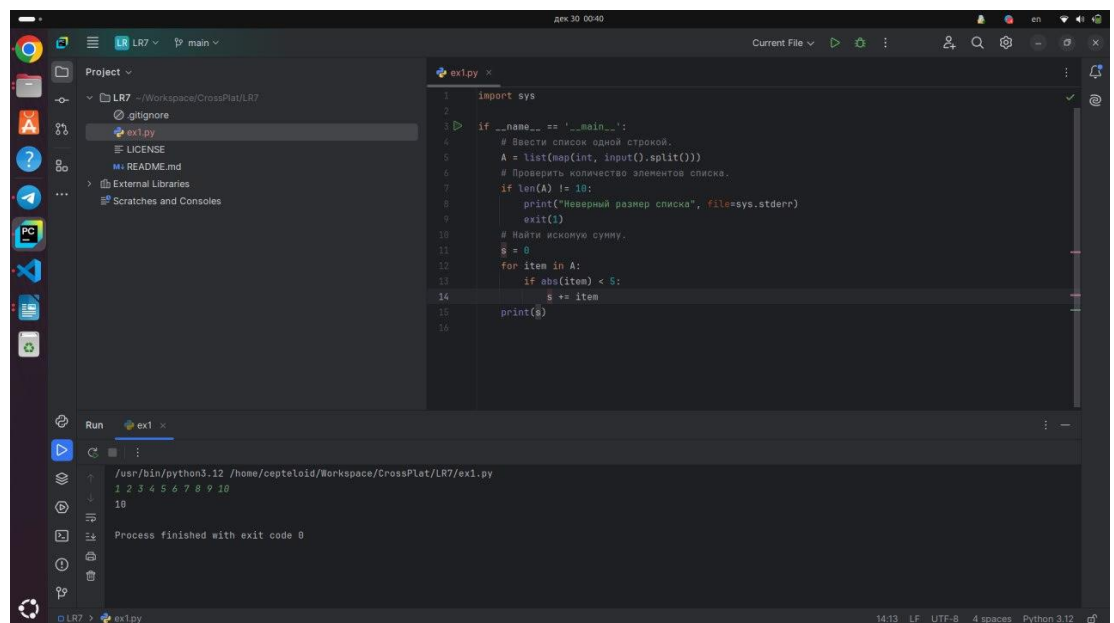
Тема: Работа со списками в языке Python

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

Ссылка на GitHub: <https://github.com/zepteloid/LR7>

Порядок выполнения работы:

1. Был изучен теоретический материал.
2. В ходе выполнения лабораторной работы были выполнены задания.
3. Был создан и клонирован репозиторий
4. Были разобраны предоставленные примеры:

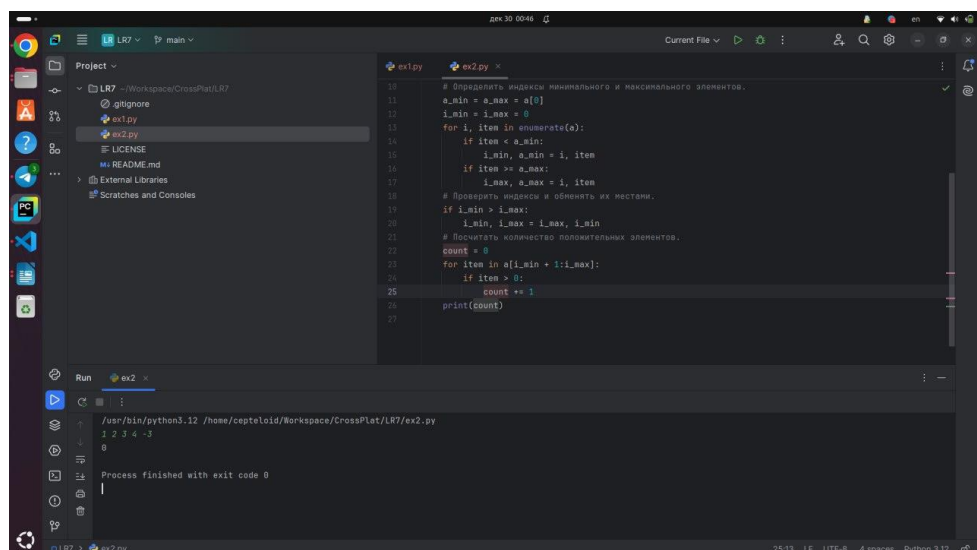


```
1 import sys
2
3 if __name__ == '__main__':
4     # Ввести список одной строкой.
5     A = list(map(int, input().split()))
6     # Проверить количество элементов списка.
7     if len(A) != 10:
8         print("Неверный размер списка", file=sys.stderr)
9         exit(1)
10    # Найти искомую сумму.
11    s = 0
12    for item in A:
13        if abs(item) < 5:
14            s += item
15    print(s)
```

Run ex1 x

```
/usr/bin/python3.12 /home/zepteloid/Workspace/CrossPlatform/LR7/ex1.py
1 2 3 4 5 6 7 8 9 10
10
Process finished with exit code 0
```

Рисунок 1. Код и вывод программы первого примера



```
10 # Определить индексы минимального и максимального элементов.
11 a_min = a_max = a[0]
12 i_min = i_max = 0
13 for i, item in enumerate(a):
14     if item < a_min:
15         i_min, a_min = i, item
16     if item > a_max:
17         i_max, a_max = i, item
18 # Проверить индексы и поменять их местами.
19 if i_min > i_max:
20     i_min, i_max = i_max, i_min
21 # Посчитать количество положительных элементов.
22 count = 0
23 for item in a[i_min + 1:i_max]:
24     if item > 0:
25         count += 1
26 print(count)
```

Run ex2 x

```
/usr/bin/python3.12 /home/zepteloid/Workspace/CrossPlatform/LR7/ex2.py
1 2 3 4 -3
0
Process finished with exit code 0
```

Рисунок 2. Код и вывод программы второго примера

5. В ходе выполнения лабораторной работы были выполнены индивидуальные задания:

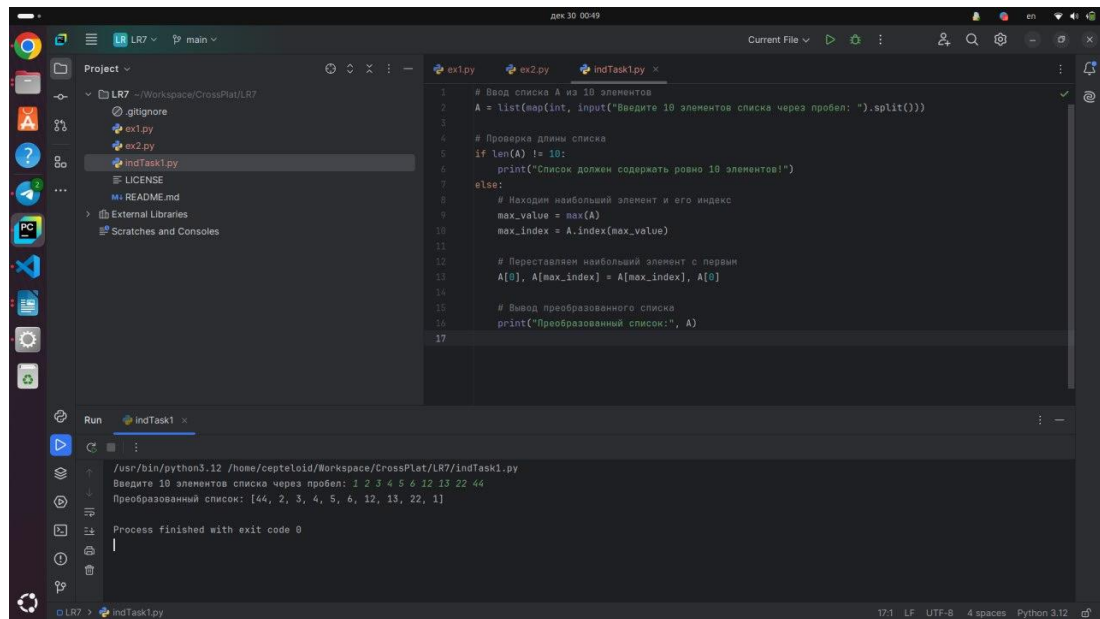


Рисунок 3. Код и вывод программы первого индивидуального задания

from functools import reduce

Ввод списка вещественных элементов

```
A = list(map(float, input("Введите вещественные элементы списка через пробел: ").split()))
```

1. Сумма отрицательных элементов

```
negative_sum = sum(x for x in A if x < 0)
print(f"Сумма отрицательных элементов: {negative_sum}")
```

2. Произведение элементов между максимальным и минимальным элементами

```
max_index = A.index(max(A))
min_index = A.index(min(A))
```

Определяем границы между максимальным и минимальным

```
start, end = sorted([max_index, min_index])
```

```

# Вычисляем произведение элементов между ними
if end - start > 1: # Проверяем, есть ли элементы между max и min
    product = reduce(lambda x, y: x * y, A[start + 1:end])
else:
    product = 0 # Если между max и min нет элементов
print(f'Произведение элементов между максимальным и минимальным:
{product}')

# 3. Упорядочивание элементов списка по возрастанию
A.sort()

print(f'Упорядоченный список: {A}')

```

```

Введите вещественные элементы списка через пробел: -1 -2 3 4 -5 6 -7
Сумма отрицательных элементов: -15.0
Произведение элементов между максимальным и минимальным: 0
Упорядоченный список: [-7.0, -5.0, -2.0, -1.0, 3.0, 4.0, 6.0]

```

Рисунок 4. Вывод второго индивидуального задания

Ответы на контрольные вопросы:

1. Что такое списки в языке Python?

Списки в Python — это изменяемые последовательности, которые могут содержать элементы различных типов, включая числа, строки и другие объекты. Списки позволяют хранить и управлять коллекциями данных.

2. Как осуществляется создание списка в Python?

Список создается с помощью квадратных скобок [], также можно использовать функцию list.

3. Как организовано хранение списков в оперативной памяти?

Списки хранятся в виде массивов указателей на объекты. Каждый элемент списка хранит ссылку на объект, а не сам объект. Это позволяет спискам содержать элементы разных типов.

4. Каким образом можно перебрать все элементы списка?

Перебор элементов списка можно осуществить с помощью цикла `for`.

5. Какие существуют арифметические операции со списками?

Основные арифметические операции со списками: конкатенация— объединяет два списка. Повторение — повторяет список `n` раз.

6. Как проверить есть ли элемент в списке?

Через оператора `in` `if element in my list`

7. Как определить число вхождений заданного элемента в списке?

Через метод `count`

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

Через метод `append` для добавления в конец списка или `insert` для вставки по индексу.

9. Как выполнить сортировку списка?

Через `sort` для сортировки списка на месте или функцию `sorted` для создания нового отсортированного списка

10. Как удалить один или несколько элементов из списка?

Через `remove` для удаления первого вхождения элемента, `pop(index)` для удаления элемента по индексу, или `del` для удаления по индексу

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

Списковое включение — это способ создания нового списка на основе существующего.

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

Срезы позволяют извлекать подпоследовательности. Синтаксис: `my list[start:end]`, где `start` — начальный индекс, а `end` — конечный индекс.

13. Какие существуют функции агрегации для работы со списками?

Основные функции: `sum` — сумма элементов. `min` — минимальный элемент. `max` — максимальный элемент. `len` — количество элементов.

14. Как создать копию списка?

Через метод `copy`

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

Функция `sorted` создает новый отсортированный список и не изменяет оригинальный, а метод `sort` сортирует список на месте и возвращает `None`

Вывод: в ходе выполнения лабораторной работы было изучено то, как можно работать со списками, а именно, как создавать списки и основные методы для работы с ними.