

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 ПРАКТИЧЕСКОЙ РАБОТЕ №2.4
дисциплины «Программирование на Python»

Выполнил:
Степанов Леонид Викторович
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение
средств вычислительной техники
и автоматизирование систем»,
очная форма обучения

(подпись)

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

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

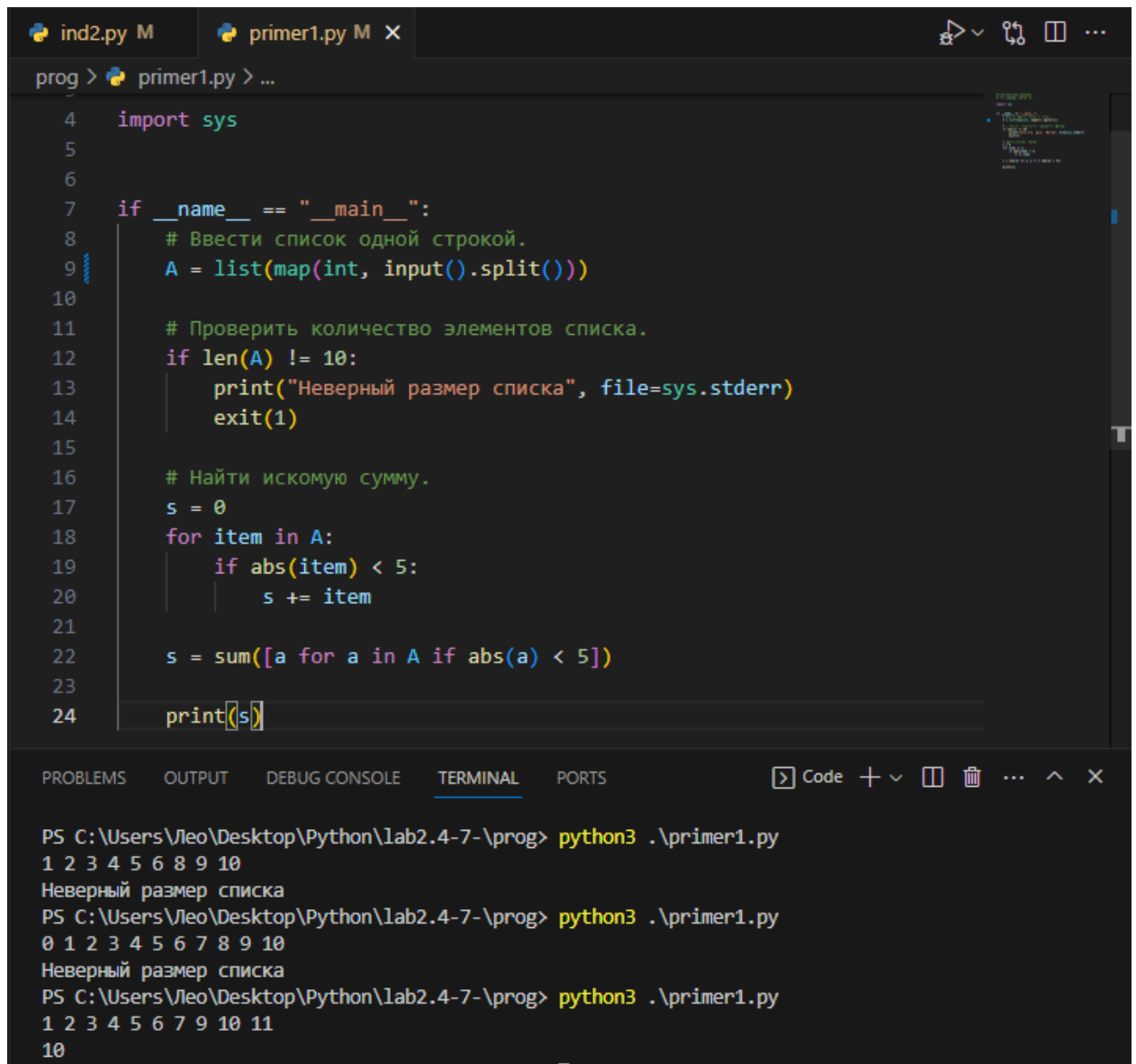
Ставрополь, 2023 г.

Тема: Работа со списками в языке Python

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

Порядок выполнения работы:

1. Создал файл (primer1.py) в котором проработал пример 1, в нем ищется сумма элементов двумя способами: с помощью перебора и при помощи спискового включения.



```
4 import sys
5
6
7 if __name__ == "__main__":
8     # Ввести список одной строкой.
9     A = list(map(int, input().split()))
10
11     # Проверить количество элементов списка.
12     if len(A) != 10:
13         print("Неверный размер списка", file=sys.stderr)
14         exit(1)
15
16     # Найти искомую сумму.
17     s = 0
18     for item in A:
19         if abs(item) < 5:
20             s += item
21
22     s = sum([a for a in A if abs(a) < 5])
23
24     print(s)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code + - - X

```
PS C:\Users\Leo\Desktop\Python\lab2.4-7-\prog> python3 .\primer1.py
1 2 3 4 5 6 8 9 10
Неверный размер списка
PS C:\Users\Leo\Desktop\Python\lab2.4-7-\prog> python3 .\primer1.py
0 1 2 3 4 5 6 7 8 9 10
Неверный размер списка
PS C:\Users\Leo\Desktop\Python\lab2.4-7-\prog> python3 .\primer1.py
1 2 3 4 5 6 7 9 10 11
10
```

Рисунок 1 – Результат выполнения программы primer1.py

2. Создал файл (primer2.py) в котором проработал пример 2, в нём определяется число положительных элементов между максимальным и между минимальным значением.

```
ind2.py M primer1.py M primer2.py X
prog > primer2.py > ...

4 import sys
5
6
7 if __name__ == "__main__":
8     # Ввести список одной строкой.
9     a = list(map(int, input().split()))
10    # Если список пуст, завершить программу.
11    if not a:
12        print("Заданный список пуст", file=sys.stderr)
13        exit(1)
14
15    # Определить индексы минимального и максимального элементов.
16    a_min = a_max = a[0]
17    i_min = i_max = 0
18    for i, item in enumerate(a):
19        if item < a_min:
20            i_min, a_min = i, item
21        if item >= a_max:
22            i_max, a_max = i, item
23
24    # Проверить индексы и обменять их местами.
25    if i_min > i_max:
26        i_min, i_max = i_max, i_min
27
28    # Посчитать количество положительных элементов.
29    count = 0
30    for item in a[i_min+1:i_max]:
31        if item > 0:
32            count += 1
33
34    print(count)

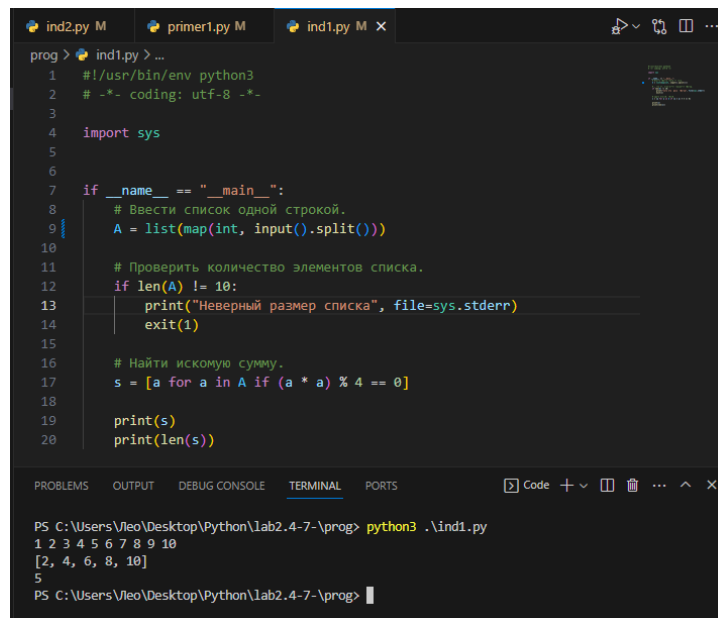
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\leo\Desktop\Python\lab2.4-7-\prog> python3 .\primer2.py
1 2 3 4 5 6
4
PS C:\Users\leo\Desktop\Python\lab2.4-7-\prog> python3 .\primer2.py
10 222 22 33 44 55 -40 -3 22
4
PS C:\Users\leo\Desktop\Python\lab2.4-7-\prog> 
```

Рисунок 2 – Результат выполнения программы primer1.py

Индивидуальное задание (17 Вариант):

1. Ввести список А из 10 элементов, найти квадраты элементов кратных 4 и их количество. Преобразованный массив вывести.

Создал файл (ind1.py), в котором нашёл сумму и количество элементов, квадраты которых кратны 4.



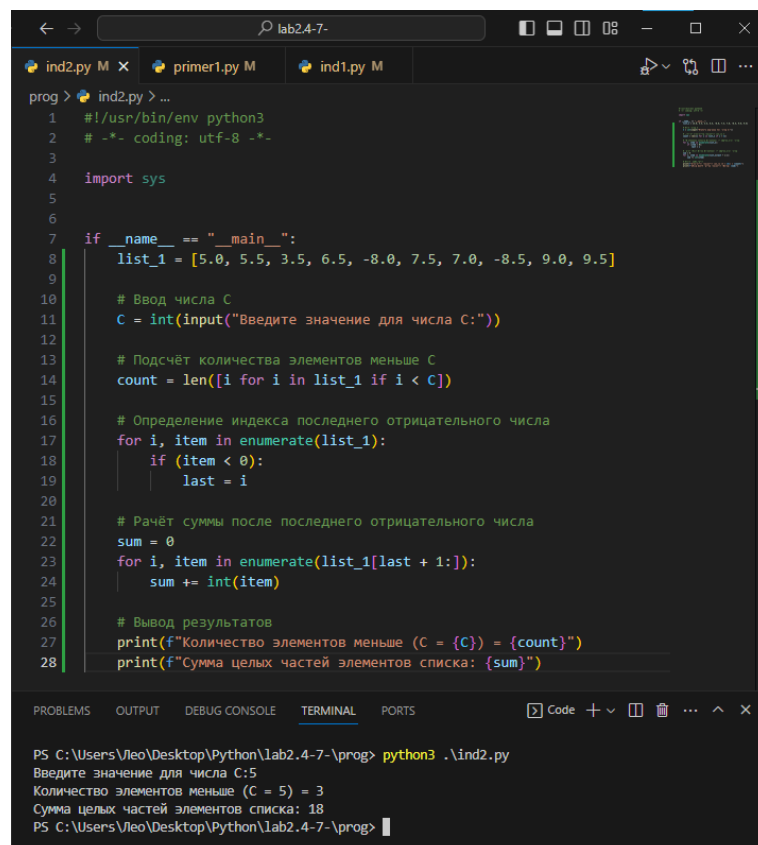
```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import sys
5
6
7 if __name__ == "__main__":
8     # Ввести список одной строкой.
9     A = list(map(int, input().split()))
10
11     # Проверить количество элементов списка.
12     if len(A) != 10:
13         print("Неверный размер списка", file=sys.stderr)
14         exit(1)
15
16     # Найти искомую сумму.
17     s = [a for a in A if (a * a) % 4 == 0]
18
19     print(s)
20     print(len(s))
```

PS C:\Users\Leo\Desktop\Python\lab2.4-7-\prog> python3 .\ind1.py
1 2 3 4 5 6 7 8 9 10
[2, 4, 6, 8, 10]
5
PS C:\Users\Leo\Desktop\Python\lab2.4-7-\prog>

Рисунок 3 – Результат выполнения программы ind1.py

2. В списке, состоящем из вещественных элементов, вычислить:
- 1) количество элементов списка, меньших C;
 - 2) сумму целых частей элементов списка, расположенных после последнего отрицательного элемента.

Создал файл (ind2.py), в котором выполнил индивидуальное задание 2:



```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import sys
5
6
7 if __name__ == "__main__":
8     list_1 = [5.0, 5.5, 3.5, 6.5, -8.0, 7.5, 7.0, -8.5, 9.0, 9.5]
9
10     # Ввод числа C
11     C = int(input("Введите значение для числа C:"))
12
13     # Подсчёт количества элементов меньше C
14     count = len([i for i in list_1 if i < C])
15
16     # Определение индекса последнего отрицательного числа
17     for i, item in enumerate(list_1):
18         if (item < 0):
19             last = i
20
21     # Рачёт суммы после последнего отрицательного числа
22     sum = 0
23     for i, item in enumerate(list_1[last + 1:]):
24         sum += int(item)
25
26     # Вывод результатов
27     print(f"Количество элементов меньше (C = {C}) = {count}")
28     print(f"Сумма целых частей элементов списка: {sum}")
```

PS C:\Users\Leo\Desktop\Python\lab2.4-7-\prog> python3 .\ind2.py
Введите значение для числа C: 5
Количество элементов меньше (C = 5) = 3
Сумма целых частей элементов списка: 18
PS C:\Users\Leo\Desktop\Python\lab2.4-7-\prog>

Рисунок 3 – Результат выполнения программы ind1.py

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

Контрольные вопросы:

1. Что такое списки в языке Python?

Список – это структура данных для хранения объектов 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 ссылки на какие-либо другие объекты или структуры.

2. Как осуществляется создание списка в Python?

Для создания списка нужно заключить элементы в квадратные скобки: `list_1 = [1, 2, 3, 4,]`. Так же при помощи `list()`: `list_2 = list (1, 2, 3, 4)`

3. Как организовано хранение списков в оперативной памяти?

При его создании в памяти резервируется область («контейнер»), в котором хранятся ссылки на другие элементы в памяти. Содержимое контейнера можно менять в отличие от чисел или строк.

4. Каким образом можно перебрать все элементы списка?

Все элементы списка можно перебрать разными способами, при помощи цикла `for` и с помощью `range(len(list))`: `for i in range(len(list)): ...` или при помощи также цикла `for` и с помощью `enumerate(a)`: `for i, item in enumerate(a):`

5. Какие существуют арифметические операции со списками?

Объединение списков при помощи оператора сложения («+») и операция повторения при помощи оператора умножения («*»).

6. Как проверить есть ли элемент в списке?

Проверить есть ли заданный элемент в списке может оператор `in`, если нет заданного элемента `not in`

7. Как определить число вхождений заданного элемента в списке?

Число вхождений заданного элемента в списке может определить метод `count`, который в качестве аргумента принимает искомый элемент.

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

Добавление элемента в список может осуществляться при помощи метода `append(a)`, который добавляет элемент `a` в конец списка, также можно добавить больше одного элемента методом `extend(a)`.

9. Как выполнить сортировку списка?

Отсортировать массив можно при помощи метода `sort()`, чтобы отсортировать по «убыванию», `sort(reverse=True)`.

10. Как удалить один или несколько элементов из списка?

Удалить элемент по его индексу может метод `pop(i)`.

Удалить элемент по его значению может метод `remove()`.

Оператор `del` может удалять также как метод `remove`, но сразу несколько элементов: `del list[1:3]`.

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

Списковое включение – способ построения списков, пример: `a = [i for i in range(7)]`. В этом примере создастся массив из 6 элементов: от 0 до 6.

Также при помощи данного способа можно осуществлять обработку списков: с элементом `i` проводить арифметические операции, и после цикла писать условие вхождения в список.

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

С помощью срезов доступ к элементам списков осуществляется так:

- 1) `list[:]` – копия списка;
- 2) `list[0:n]` – первые `n` элементы;
- 3) `list[n:m]` – получить элементы с `n+1` по `m`;
- 4) `list[::n]` – взять элементы списка с шагом `n`;
- 5) `list[n:m:s]` – взять элементы с `n+1` по `m` с шагом `s`.

13. Какие существуют функции агрегации для работы со списками?

Функции агрегации: получить число элементов: `len()`, получить минимальный элемент списка: `min()`, получить максимальный элемент списка: `max()`, получить сумму элементов списка: `sum()`.

14. Как создать копию списка?

Создать копию списка можно при помощи среза `a = b[:]` и при помощи метода `copy()`.

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

Функция `sorted()` возвращает новый отсортированный список, оставляя исходный список неизменным. Она принимает список (или другую итерируемую последовательность) в качестве аргумента и возвращает новый список, содержащий отсортированные элементы. Основное отличие между `sorted()` и `sort()` заключается в том, что `sorted()` возвращает новый отсортированный список, оставляя исходный список неизменным, в то время как `sort()` изменяет сам список, сортируя его элементы на месте.