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 ЛАБОРАТОРНОЙ РАБОТЕ №2.4**

**дисциплины «Программирование на Python»**

**Вариант\_\_\_**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Выполнил:  Иващенко Олег Андреевич  2 курс, группа ИВТ-б-о-22-1,  09.03.02 «Информационные и вычислительные машины», направленность (профиль) «Программное обеспечение средств вычислительной техники и автоматизированных систем»  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | | Руководитель практики:  Воронкин Роман Александрович, 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 написании программ с помощью языка программирования Python версии 3.x.

Порядок выполнения работы

Таблица 1.1 – Код программы example\_1.py

|  |
| --- |
| #!/usr/bin/env python3  # -\*- coding: utf-8 -\*-  import sys  if \_\_name\_\_ == '\_\_main\_\_':  # 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 = 0  for item in A:  if abs(item) < 5:  s += item    print(s) |



Рисунок 1 – Вывод программы example\_1.py

Таблица 1.2 – Код программы example\_1.py

|  |
| --- |
| #!/usr/bin/env python3  # -\*- coding: utf-8 -\*-  import sys  if \_\_name\_\_ == '\_\_main\_\_':  # 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 1.2 – Вывод программы example\_2.py

Таблица 2 – Код программы example\_2.py

|  |
| --- |
| #!/usr/bin/env python3  # -\*- coding: utf-8 -\*-  import sys  if \_\_name\_\_ == '\_\_main\_\_':  # Ввести список одной строкой.  a = list(map(int, input().split()))  # Если список пуст, завершить программу.  if not a:  print("Заданный список пуст", file=sys.stderr)  exit(1)  # Определить индексы минимального и максимального элементов.  a\_min = a\_max = a[0]  i\_min = i\_max = 0  for i, item in enumerate(a):  if item < a\_min:  i\_min, a\_min = i, item  if item >= a\_max:  i\_max, a\_max = i, item    # Проверить индексы и обменять их местами.  if i\_min > i\_max:  i\_min, i\_max = i\_max, i\_min  # Посчитать количество положительных элементов.  count = 0  for item in a[i\_min+1:i\_max]:  if item > 0:  count += 1    print(count) |



Рисунок 2 – Вывод программы example\_2.py

Индивидуальное задание 1. Ввести список A из 10 элементов, найти сумму элементов, меньших по модулю 3 и кратных 9, их количество и вывести результаты на экран.

Таблица 3 – Код программы individual\_1.py

|  |
| --- |
| #!/usr/bin/python3  # -\*- coding: utf-8 -\*-  if \_\_name\_\_ == "\_\_main\_\_":  A = list(map(int, input("Введите 10 целочисленных"  "элементов массива: ").split()))  if (len(A) != 10):  print("Количество элементов не равно 10. Завершение программы...")  exit(1)    result = []  summ = 0  result = [i for i in A if i < abs(3) and i % 9 == 0]  summ = sum(i for i in A if i < abs(3) and i % 9 == 0)  print(f"Элементы списка: {result}")  print(f"Количество элементов: {len(result)}")  print(f"Сумма элементов: {summ}") |

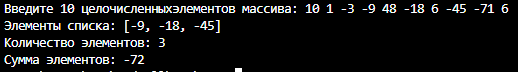


Рисунок 3 – Вывод программы individual\_1.py

Индивидуальное задание 2. В списке, состоящем из вещественных элементов, вычислить:

1. Количество элементов списка, равных 0;
2. Сумму элементов списка, расположенных после минимального элемента.

Таблица 4 – Код программы individual\_2.py

|  |
| --- |
| #!/usr/bin/python3  # -\*- coding: utf-8 -\*-  if \_\_name\_\_ == "\_\_main\_\_":  A = list(map(float, input("Введите несколько"  "вещественных чисел: ").split()))  i\_min = 0  min\_sum = 0  null\_count = 0  #Поиск минимального значения  for i in A:  if i < i\_min:  i\_min = i  if i == float(0):  null\_count += 1  for i in range(A.index(i\_min) + 1, len(A), 1):  min\_sum += A[i]  print(f"Наименьший элемент - [{A.index(i\_min)}] {i\_min}")  print(f"Количество элементов, равных нулю - {null\_count}")  print(f"Сумма элементов, находящихся после минимального - {min\_sum}") |

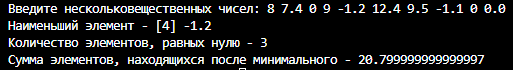


Рисунок 4 – Вывод программы individual\_2.py

Контрольные вопросы

1. Что такое списки в языке Python?

Списки в Python – это упорядоченные изменяемые коллекции объектов. Они могут содержать элементы различных типов данных и позволяют обращаться к элементам по индексу.

1. Как осуществляется создание списка в Python?

Пример создания списка:

my\_list = [1, 2, 3, “apple”, “orange”]

1. Как организовано хранение списков в оперативной памяти?

Элементы списка хранятся в памяти последовательно, и каждый элемент имеет свой индекс.

1. Каким образом можно перебрать все элементы списка?

Перебрать все элементы списка можно с помощью циклов for и for..in

1. Какие существуют арифметические операции со списками?

Арифметические операции со списками:

* Сложение списков (конкатенация) – result = list1 + list2;
* Умножение списка – result = list1 \* 3

1. Как проверить, есть ли элемент в списке?

Для проверки наличия элемента в списке можно использовать оператор in. Пример:

if element in list:

1. Как определить число вхождений заданного элемента в списке?

Для определения числа вхождений заданного элемента в списке можно использовать метод count(). Пример:

count = list.count(element)

1. Как осуществляется добавление (вставка) элементов в списке?

Добавление (вставка) элементов в список осуществляется методами append() или insert().

1. Как выполнить сортировку списка?

Для выполнения сортировки списка можно использовать метод sort().

1. Как удалить один или несколько элементов из списка?

Удаление элементов из списка можно осуществить с помощью метода remove(). Пример:

list.remove(element)

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

Создание нового списка с использованием выражения и цикла в одной строки можно осуществить следующим образом:

list = [x\*\*2 for x in range(10)]

1. Как осуществить доступ к элементам списков с помощью срезов?

Доступ к элементам списков с помощью срезов можно осуществить следующим образом:

result = list[start\_index:end\_index]

1. Какие существуют функции агрегации для работы со списками?

Функции агрегации для работы со списками:

* len(list) – длина списка;
* sum(list) – сумма элементов списка;
* max(list) – максимальный элемент списка;
* min(list) – минимальный элемент списка.

1. Как создать копию списка?

Создание копии списка осуществляется методом copy(). Пример:

copy\_list = list.copy()

1. Самостоятельно изучите функцию sorted языка Python. В чём отличие метода sort списков?

Функция sorted() создаёт новый отсортированный список. Метод sort() сортирует уже существующий список.

**Выводы**: В процессе выполнения лабораторной работы были приобретены навыки по работе со списками при написании программ с помощью языка программирования Python версии 3.x, были написаны 4 программы, 2 из которых являются примерами из лабораторной работы, и 2 индивидуальные задачи.