

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 ПРАКТИЧЕСКОЙ РАБОТЕ №2.4**  
**дисциплины «Искусственный интеллект в профессиональной сфере»**

Выполнила:  
Кятов Амаль Алиевич  
3 курс, группа ЭНЭ-б-о-23-1,  
11.03.04 «Электроника и  
наноэлектроника», очная форма  
обучения

---

(подпись)

Проверил:  
Воронкин Роман Александрович  
доцент

---

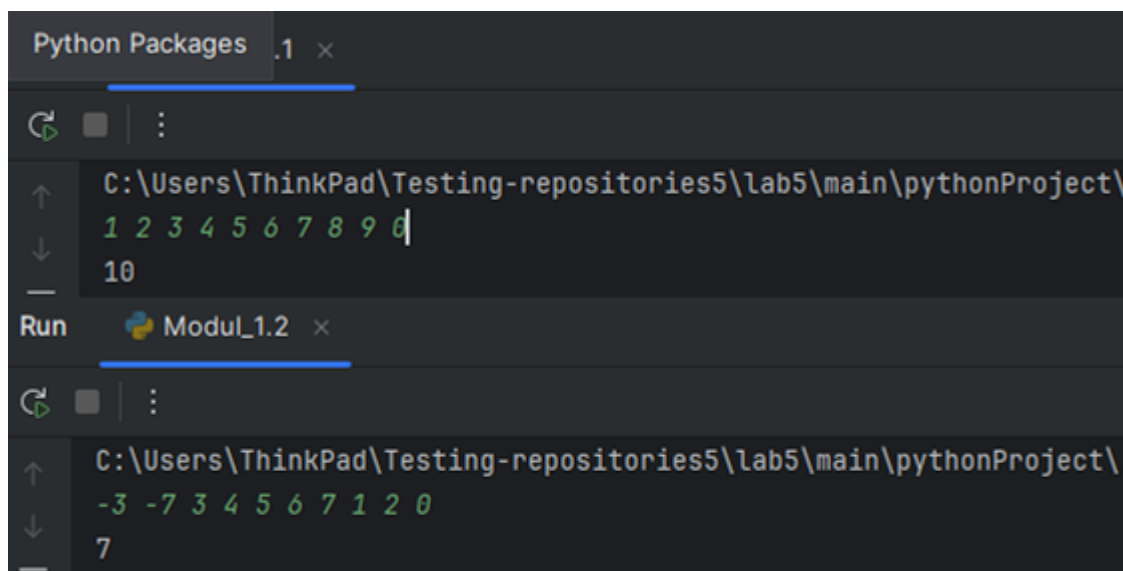
(подпись)

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

Ставрополь, 2024 г.

Тема работы: работа со списками в языке Python.

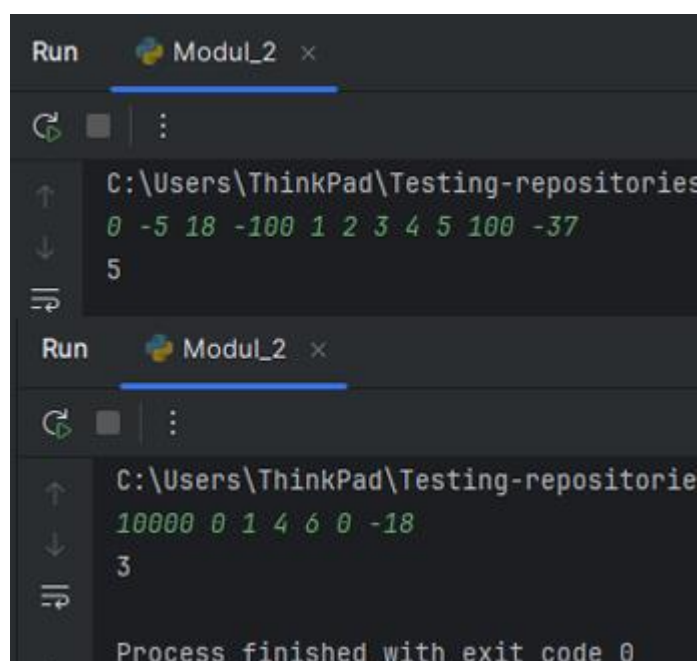
Цель работы: приобретение навыков по работе со списками при написании программ с помощью языка программирования Python версии 3.x.



```
Python Packages .1 x
C:\Users\ThinkPad\Testing-repositories5\lab5\main\pythonProject\
1 2 3 4 5 6 7 8 9 0
10

Run Modul_1.2 x
C:\Users\ThinkPad\Testing-repositories5\lab5\main\pythonProject\
-3 -7 3 4 5 6 7 1 2 0
7
```

Рисунок 1 – Пример модуля 1



```
Run Modul_2 x
C:\Users\ThinkPad\Testing-repositories
0 -5 18 -100 1 2 3 4 5 100 -37
5

Run Modul_2 x
C:\Users\ThinkPad\Testing-repositories
10000 0 1 4 6 0 -18
3
Process finished with exit code 0
```

Рисунок 2 – Пример модуля 2

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

10. В списке, состоящем из вещественных элементов, вычислить:

1. номер минимального по модулю элемента списка;
2. сумму модулей элементов списка, расположенных после первого отрицательного элемента.

```
... product = 1
... count = 0
... for num in A:
...     if num > 0 and num % 3 == 0:
...         product *= num
...         count += 1
... if count > 0:
...     print(f"Произведение положительных элементов, кратных 3: {product}")
...     print(f"Количество таких элементов: {count}")
... else:
...     print("Нет положительных элементов, кратных 3.")
...
Введите элемент 1: >? 4
Введите элемент 2: >? 7
Введите элемент 3: >? 0
Введите элемент 4: >? 9
Введите элемент 5: >? 6
Введите элемент 6: >? 8
Введите элемент 7: >? 3
Введите элемент 8: >? 1
Введите элемент 9: >? 9
Введите элемент 10: >? 11
>>>
```

Рисунок 3 – Решение 1

```
... A = [3, -2, 5, -1.2, 7, -4, 2, -9, 4, 19]
... min_index = 0
... min_value = abs(A[0])
... for i in range(1, len(A)):
...     if abs(A[i]) < min_value:
...         min_value = abs(A[i])
...         min_index = i
... print(f"Номер минимального по модулю элемента: {min_index}")
... sum_after_first_negative = 0
... found_first_negative = False
... for num in A:
...     if num < 0 and not found_first_negative:
...         found_first_negative = True
...     elif found_first_negative:
...         sum_after_first_negative += abs(num)
... print(f"Сумма модулей элементов после первого отрицательного элемента: {sum_after_first_negative}")
Номер минимального по модулю элемента: 3
Сумма модулей элементов после первого отрицательного элемента: 51.2
```

Рисунок 4 – Решение 2

Ответы на вопросы:

1. Что такое списки в языке Python? Списки в языке Python — это упорядоченная коллекция элементов, которая позволяет хранить несколько значений в одной переменной. Если точнее, списки — это массивы ссылок на объекты в памяти компьютера.

2. Как осуществляется создание списка в Python? Создание списка в Python осуществляется через простой синтаксис: элементы заключаются в квадратные скобки и разделяются запятыми. Также для создания списка можно использовать функцию-конструктор `list()`.

3. Как организовано хранение списков в оперативной памяти? Хранение списков в оперативной памяти организовано двумя способами: Последовательное распределение. Элементы списка располагаются в непосредственно следующих друг за другом участках памяти. Связанное (иначе — цепное) распределение. Наряду с другими полями каждый элемент списка содержит дополнительное адресное поле, в котором хранится адрес следующего элемента, либо признак его отсутствия.

4. Каким образом можно перебрать все элементы списка? Для перебора элементов списка можно использовать как цикл `for`, так и цикл `while`.

5. Какие существуют арифметические операции со списками? Существуют следующие арифметические операции со списками в Python: сложение, вычитание, умножение и деление. Сложение списков осуществляется с помощью оператора `+`, который объединяет два списка в один. Если нужно сложить элементы списков поэлементно, можно использовать функцию `zip` и генератор списков.

6. Как проверить есть ли элемент в списке? Чтобы проверить, есть ли элемент в списке в Python, можно использовать оператор `In`. Если элемент присутствует в списке, оператор `in` возвращает `True`, в противном случае — `False`.

7. Как определить число вхождений заданного элемента в списке? Определить число вхождений заданного элемента в списке можно с помощью метода `count()`, который считает, сколько раз элемент повторяется в списке.

8. Как осуществляется добавление (вставка) элемента в список? Для добавления (вставки) элемента в список используется метод `append()` или `insert()`, где `list` — список, `i` — позиция, `x` — нужное значение.

9. Как выполнить сортировку списка? Для сортировки списка можно использовать метод `sort()`, который сортирует элементы списка на месте. Если нужно отсортировать элементы в обратном порядке — от большего к меньшему, в методе есть дополнительный параметр `reverse`.

10. Как удалить один или несколько элементов из списка? Чтобы удалить один или несколько элементов из списка, используют инструкцию `del list[i]`, где `list` — список, `i` — индекс (позиция) элемента в списке.

11. Что такое списковое включение и как с его помощью осуществлять обработку списков? Списковое включение — это способ компактного описания операций обработки списков в языке программирования Python. С его помощью можно генерировать списки, применяя заданное выражение к каждому элементу будущего списка, а также фильтровать элементы согласно заданным условиям.

12. Как осуществляется доступ к элементам списков с помощью срезов? Доступ к элементам списков с помощью срезов осуществляется путём указания начального и конечного индексов. В итоге получается новый список, который содержит часть элементов оригинального. Например, можно сделать срез от начала списка до определённого индекса, от определённого индекса до конца списка и другие.

13. Какие существуют функции агрегации для работы со списками? `len()`. Возвращает количество элементов в списке. `min()` и `max()`. Используются для поиска минимального и максимального значения в списке. `Counter` из модуля `collections`. С её помощью можно подсчитать частоту вхождения элементов в списке. `sum()`. Возвращает сумму элементов списка. Если список пустой, то сумма его элементов равна нулю.

14. Как создать копию списка? Чтобы создать копию списка, можно использовать метод `copy()`, который создаёт поверхностную копию списка.

Для создания глубокой копии нужно использовать метод `deepcopy` из модуля `copy`.

15. Самостоятельно изучите функцию `sorted` языка Python. В чем ее отличие от метода `sort` списков? Функция `sorted()` возвращает отсортированный список, а отличие от метода `sort` в том, что `sorted()` работает с уже отсортированным списком, а метод `sort()` сортирует элементы на месте.

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