

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

**Основы кроссплатформенного программирования
Отчет по лабораторной работе №7**

Работа с кортежами в языке Python

Выполнила студентка группы
ИТС-б-о-20-1 (2)

Швецова К.С. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил к.т.н., доцент

Кафедры инфокоммуникаций

Воронкин Р.А.

(подпись)

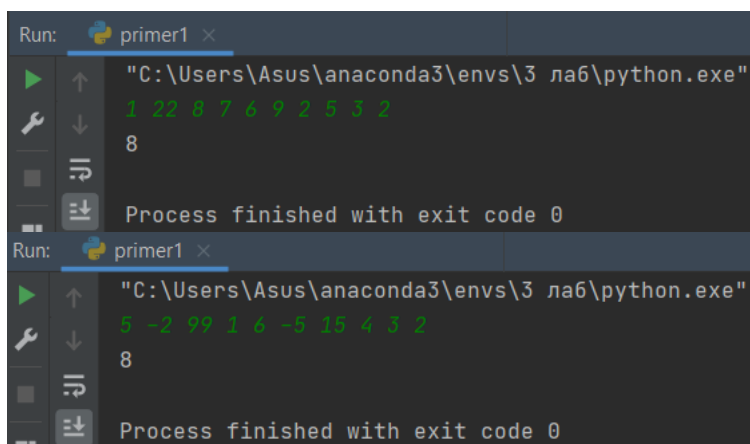
Ставрополь 2021

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

Ссылка на репозиторий - <https://github.com/ShveczovaKS/3lab2k>

Ход работы:

Пример 1. Ввести кортеж A из 10 элементов, найти сумму элементов, меньших по модулю 5, и вывести ее на экран. Использовать в программе вместо списков кортежи.

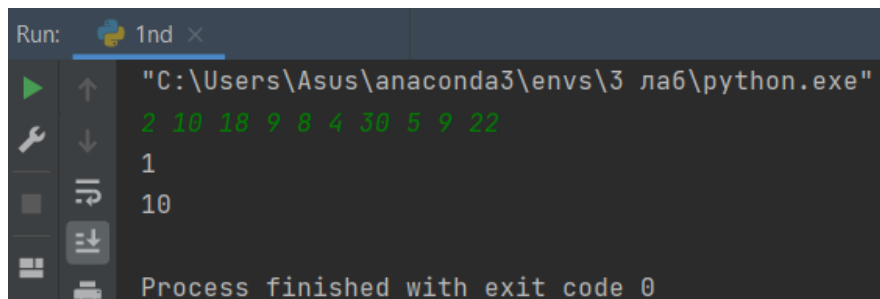


```
Run: primer1 x
"C:\Users\Asus\anaconda3\envs\3 лаб\python.exe"
1 22 8 7 6 9 2 5 3 2
8
Process finished with exit code 0

Run: primer1 x
"C:\Users\Asus\anaconda3\envs\3 лаб\python.exe"
5 -2 99 1 6 -5 15 4 3 2
8
Process finished with exit code 0
```

Рисунок 1 и 2. Результат выполнения программы с различными исходными данными

Индивидуальное задание 1. 15 вариант. Ввести список A из 10 элементов, найти произведение элементов, больших 8 и меньших 18 и кратных 10, их количество и вывести результаты на экран.



```
Run: 1nd x
"C:\Users\Asus\anaconda3\envs\3 лаб\python.exe"
2 10 18 9 8 4 30 5 9 22
1
10
Process finished with exit code 0
```

Рисунок 3. Результат выполнения программы

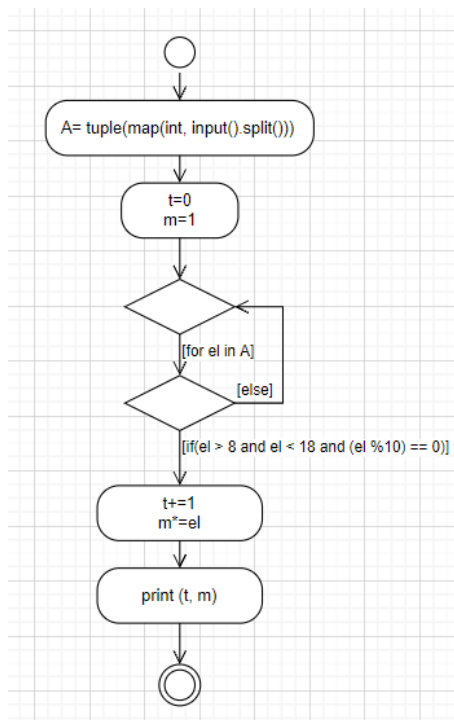


Рисунок 4. UML-диаграммы деятельности решения

Индивидуальное задание 2. 14 вариант. В списке, состоящем из вещественных элементов, вычислить:

- 1) количество элементов списка, больших C;
- 2) произведение элементов списка, расположенных после максимального элемента.

Преобразовать список таким образом, чтобы сначала располагались все отрицательные элементы, а потом - все положительные (элементы, равные 0, считать положительными).

```

Run: 2ind x
"C:\Users\Asus\anaconda3\envs\3 лаб\python.exe" "C:/Users/Asus/OneDrive/Рабочий стол/...
Введите 10 чисел: 1 2 3 4 5 -1 -2 -3 -4 -5
C = -1
Количество элементов больших C: 7
Произведение элементов, расположенных после максимального элемента: -120
Отсортированный список: [-5, -4, -3, -2, -1, 1, 2, 3, 4, 5]
  
```

Рисунок 5. Результат выполнения программы

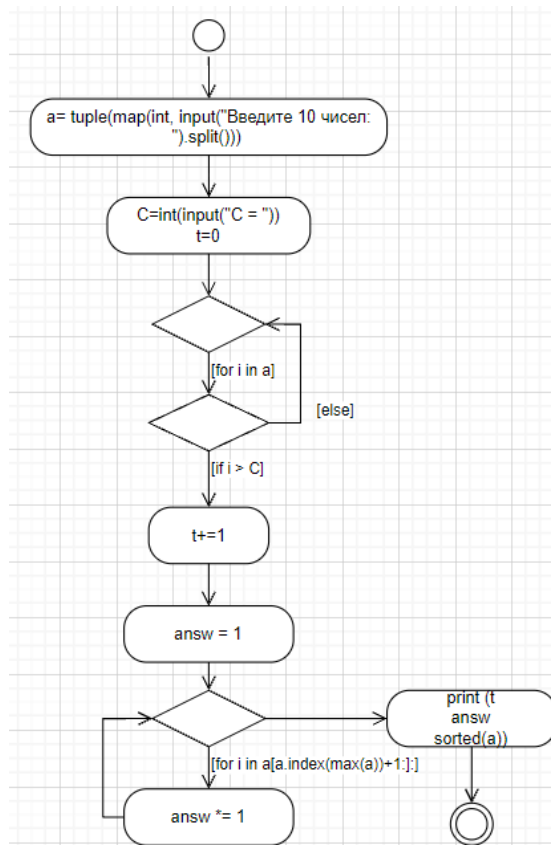


Рисунок 6. UML-диаграммы деятельности решения

Контрольные вопросы:

1. Что такое списки в языке Python?

Список (list) – это структура данных для хранения объектов различных типов. В нем можно хранить объекты 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. Каково назначение кортежей в языке Python?

Во-первых, это экономия места. Дело в том, что кортежи в памяти занимают меньший объем по сравнению со списками. Во-вторых – прирост производительности, который связан с тем, что кортежи работают быстрее, чем списки (т. е. на операции перебора элементов и т. п. будет тратиться

меньше времени). Во-третьих – это безопасность данных от случайного изменения.

3. Как осуществляется создание кортежей?

Для создания пустого кортежа можно воспользоваться одной из следующих команд.

```
>>> a = ()
```

```
>>> b = tuple()
```

Кортеж с заданным содержанием создается также как список, только вместо квадратных скобок используются круглые.

4. Как осуществляется доступ к элементам кортежа?

Доступ к элементам кортежа осуществляется также как к элементам списка – через указание индекса.

5. Зачем нужна распаковка (деструктуризация) кортежа?

Обращение по индексу, это не самый удобный способ работы с кортежами. Дело в том, что кортежи часто содержат значения разных типов, и помнить, по какому индексу что лежит — очень непросто. Но есть способ лучше! Как мы кортеж собираем, так его можно и разобрать:

```
name_and_age = ('Bob', 42)
```

```
(name, age) = name_and_age
```

```
name # 'Bob'
```

```
age # 42
```

6. Какую роль играют кортежи во множественном присваивании?

Благодаря тому, что кортежи легко собирать и разбирать, в Python удобно делать такие вещи, как множественное присваивание:

```
(a, b, c) = (1, 2, 3)
```

a # 1

b # 2

c # 3

7. Как выбрать элементы кортежа с помощью среза?

С помощью операции взятия среза можно получить другой кортеж. Общая форма операции взятия среза для кортежа:

$$T2 = T1[i:j],$$

где

T2 – новый кортеж, который получается из кортежа T1;

T1 – исходный кортеж, для которого происходит срез;

i, j – соответственно нижняя и верхняя границы среза. Фактически берутся ко вниманию элементы, лежащие на позициях i, i+1, ..., j-1. Значение j определяет позицию за последним элементом среза.

8. Как выполняется конкатенация и повторение кортежей?

Для кортежей можно выполнять операцию конкатенации, которая обозначается символом +. В простейшем случае для конкатенации двух кортежей общая форма операции следующая:

$$T3 = T1 + T2$$

Кортеж может быть образован путем операции повторения, обозначаемой символом *. При использовании в выражении общая форма операции следующая:

$$T2 = T1 * n$$

9. Как выполняется обход элементов кортежа?

Элементы кортежа можно последовательно просмотреть с помощью операторов цикла `while` или `for`.

10. Как проверить принадлежность элемента кортежу?

Для того, чтобы проверить, есть ли заданный элемент в кортеже Python необходимо использовать оператор `in`.

11. Какие методы работы с кортежами Вам известны?

Метод `index()` – поиск позиции элемента в кортеже.

Метод `count()` – количество вхождений элемента в кортеж.

12. Допустимо ли использование функций агрегации, таких как `len()` , `sum()` и т. д. при работе с кортежами?

Допустимо.

13. Как создать кортеж с помощью спискового включения.

В отличие от выражения `[a for a in A ...]`, которое на выходе дает нам список, выражение `(a for a in A ...)` дает на выходе специальный объект генератора, а не кортеж. Для преобразования генератора в кортеж необходимо воспользоваться вызовом `tuple()`.

Вывод: в ходе лабораторной работы были приобретены навыки по работе с кортежами при написании программ с помощью языка программирования Python версии 3.x.