**МИНИCTEPCTBO НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ**

**РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное**

**образовательное учреждение высшего образования**

**«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций**

**Лабораторная работа 7**

Работа с кортежами в языке Python

|  |
| --- |
| Выполнил студент группы ИВТ-б-о-20-1 |
| Симанский М.Ю « »\_\_\_\_\_\_\_\_\_20\_\_г. |
| Подпись студента\_\_\_\_\_\_\_\_\_\_\_ |
| Работа защищена « »\_\_\_\_\_\_\_\_\_\_\_\_20\_\_г. |
| Проверил Воронкин Р.А. \_\_\_\_\_\_\_\_  (подпись) |

**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/MaksimSimanskiy/lab4.git

**Задание 1**

12. Ввести список А из 10 элементов, найти сумму элементов, больших 2 и меньших 20 и кратных 8, их количество и вывести результаты на экран.

**Код**

#!/usr/bin/env python3

# -\*- coding: utf-8 -\*-

import sys

b = ()

print('Введите элементы кортежа в одну строку ')

a = tuple(map(int, input().split()))

if len(a) != 10:

print("Неверный размер кортежа", file=sys.stderr)

exit(1)

b = tuple(filter(lambda x: 20 > x > 2 and x % 8 == 0, a))

print('Список А = {}\nВыбраные элементы = {} \nКоличество элементов = {} \nСумма = {}'.format(a, b, len(b), sum(b)))

**UML-диаграмма**

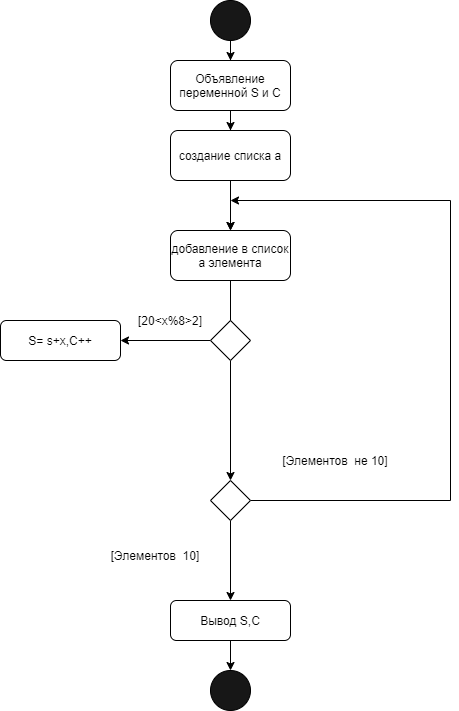
****

Рисунок – UML-диаграмма

**Результат**

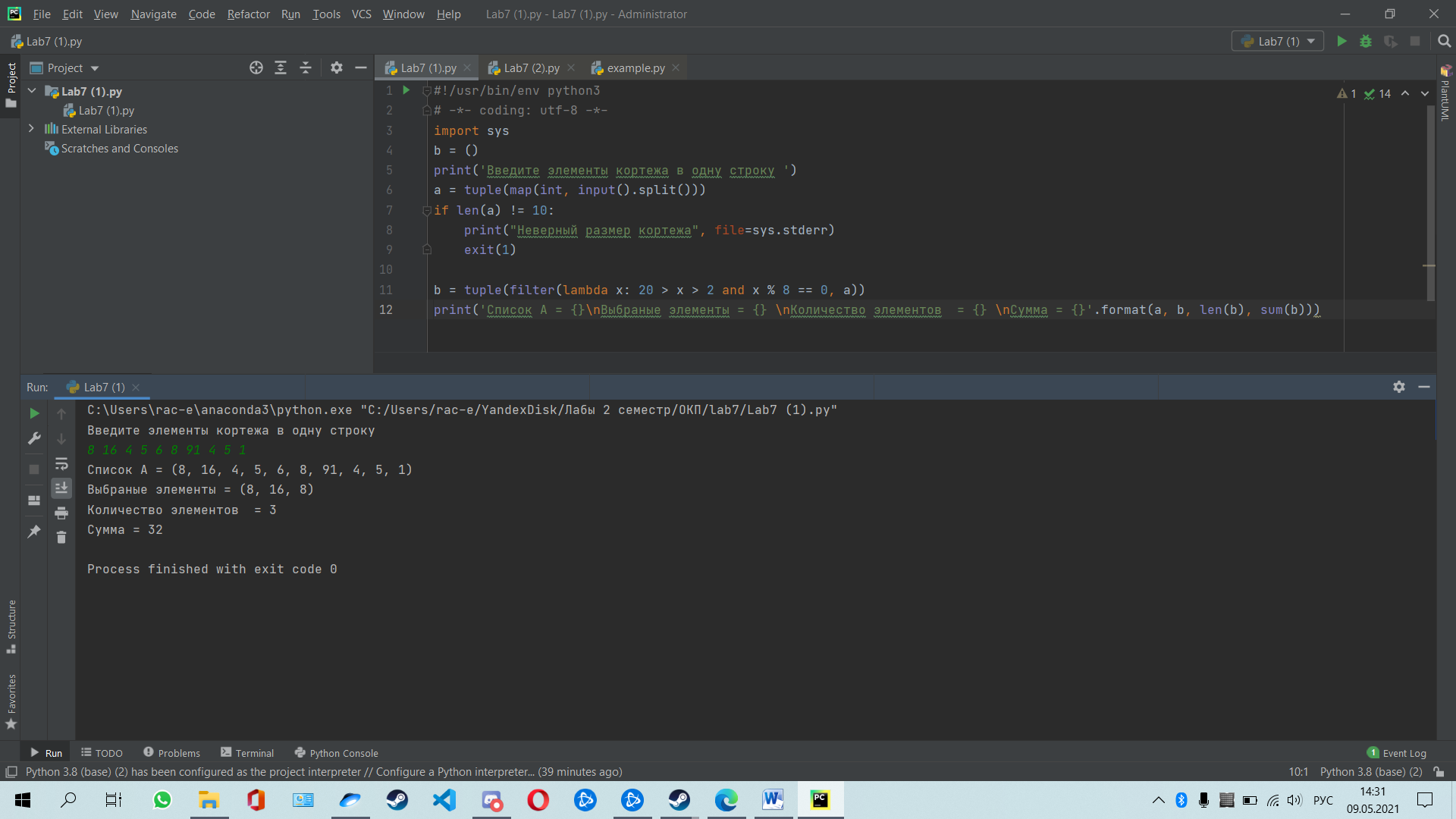


Рисунок – Результат работы

**Задание 2**

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

1) номер максимального по модулю элемента списка;

2) сумму элементов списка, расположенных после первого положительного элемента. Преобразовать список таким образом, чтобы сначала располагались все элементы, целая часть которых лежит в интервале [а, b], а потом - все остальные.

**Код**

#!/usr/bin/env python3

# -\*- coding: utf-8 -\*-

import sys

c = int(input("Введите начало интервала "))

d = int(input("Введите конец интервала "))

print('Введите элементы в одну строку ')

a = tuple(map(int, input().split()))

if len(a) != 5:

print("Неверный размер кортежа", file=sys.stderr)

exit(1)

print('Максимальный элемент ' , max(a, key=abs))

print('Индекс максимального элемента ' , a.index(max(a, key=abs)))

for i in range(len(a)):

if (a[i] > 0.):

k = i

break

print('Сумма чисел после первого положительного числа',sum(a[k:]))

print('Отсортированый кортеж',sorted(a,key = lambda x: d > x > c,reverse = True))

**UML-диаграмма**

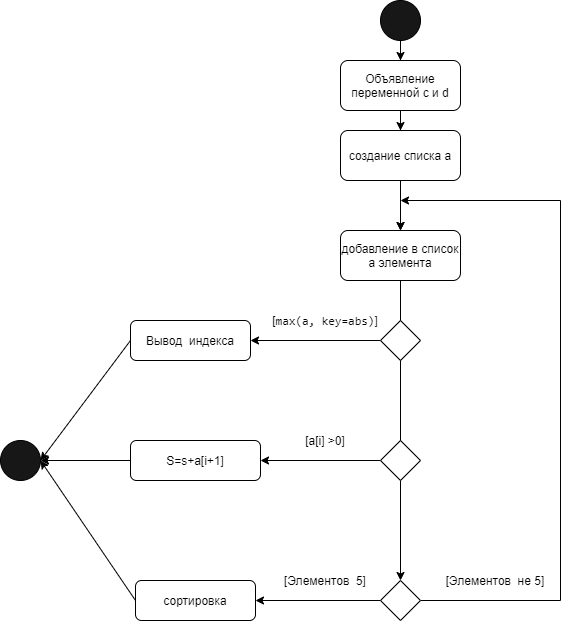
****

Рисунок – UML-диаграмма

**Результат**

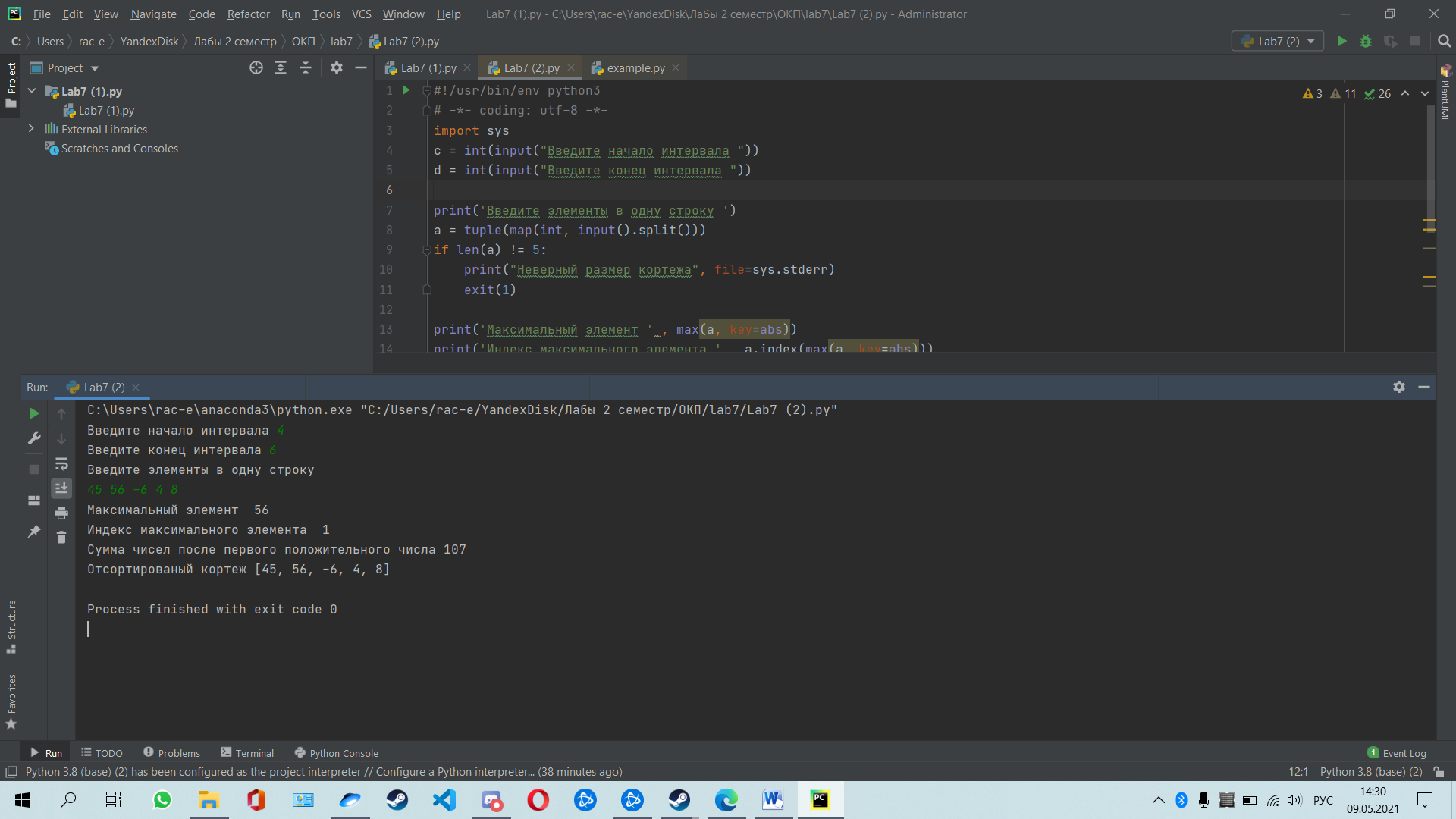


Рисунок – Результат работы

**Ответы на контрольные вопросы**

1. Кортеж (tuple) – это неизменяемая структура данных, которая по своему подобию очень похожа на список.
2. Существует несколько причин, по которым стоит использовать кортежи вместо списков. Одна из них – это обезопасить данные от случайного изменения. Вторая - кортежи в памяти занимают меньший объем по сравнению со списками.
3. Для создания пустого кортежа можно воспользоваться одной из следующих команд: a = (), a = turple().
4. Доступ к элементам кортежа осуществляется также как к элементам списка – через указание индекса.
5. Как мы кортеж собираем, так его можно и разобрать: Именно таким способом принято получать и сразу разбирать значения, которые возвращает функция.
6. Благодаря тому, что кортежи легко собирать и разбирать, в Python удобно делать такие вещи, как множественное присваивание.

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

Используя множественное присваивание, можно провернуть интересный трюк: обмен значениями между двумя переменными.

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

T2 = T1[i:j]

1. Для кортежей можно выполнять операцию конкатенации, которая обозначается символом +.Кортеж может быть образован путем операции повторения, обозначаемой символом \*.
2. Элементы кортежа можно последовательно просмотреть с помощью операторов цикла while или for:

for item in A: print(item)

while i < len(A): if (A[i]

1. if (item in A): print(item, " in ", A, " = True") else: print(item, " in ", A, " = False")
2. Метод index(). Поиск позиции элемента в кортеже

Метод count(). Количество вхождений элемента в кортеж

1. Да ,допустимо
2. Выражение tuple(map(int, input().split())) позволяет ввести целочисленный массив одной строкой при помощи спискового включения.