Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное образовательное учреждение высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития Кафедра инфокоммуникаций

**«Работа с кортежами в языке Python»**

**ОТЧЕТ**

**по лабораторной работе №8 дисциплины**

**«Основы программной инженерии»**

|  |
| --- |
| Выполнил:  Сотников Андрей Александрович  2 курс, группа ПИЖ-б-о-21-1,  09.03.04 «Программная инженерия», направленность (профиль) «Разработка и сопровождение программного обеспечения», очная форма обучения  (подпись) |
| Проверил:  (подпись) |

Отчет защищен с оценкой Дата защиты

Ставрополь, 2022 г.

Проработка примеров из лабораторной работы:

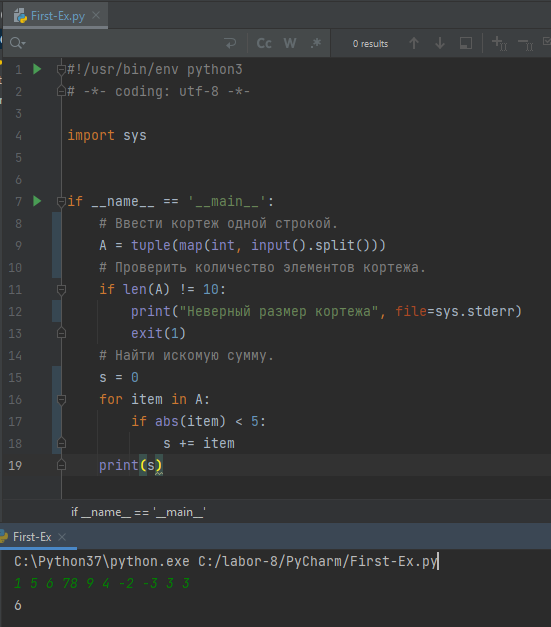


Рисунок №1 – Пример №1

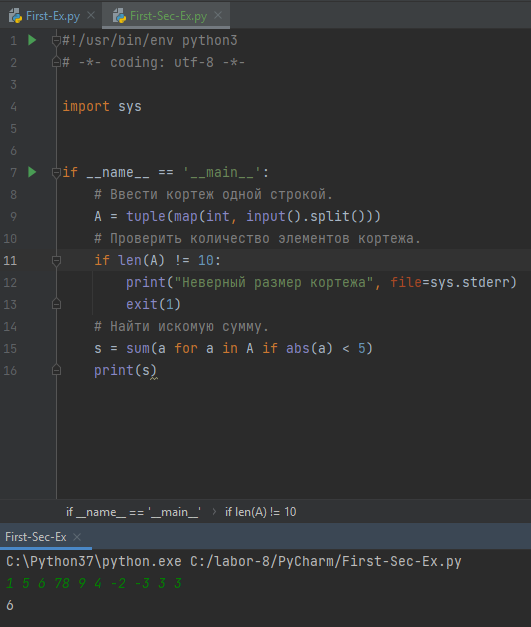


Рисунок №2 – Пример №1 с использованием списковых включений

Индивидуальное задание: известны данные о численности населения (в миллионах жителей) и площади (в тысячах квадратных километров) 28 государств. Определить общую численность населения в «маленьких» государствах (чья площадь не превышает A тысяч квадратных километров).

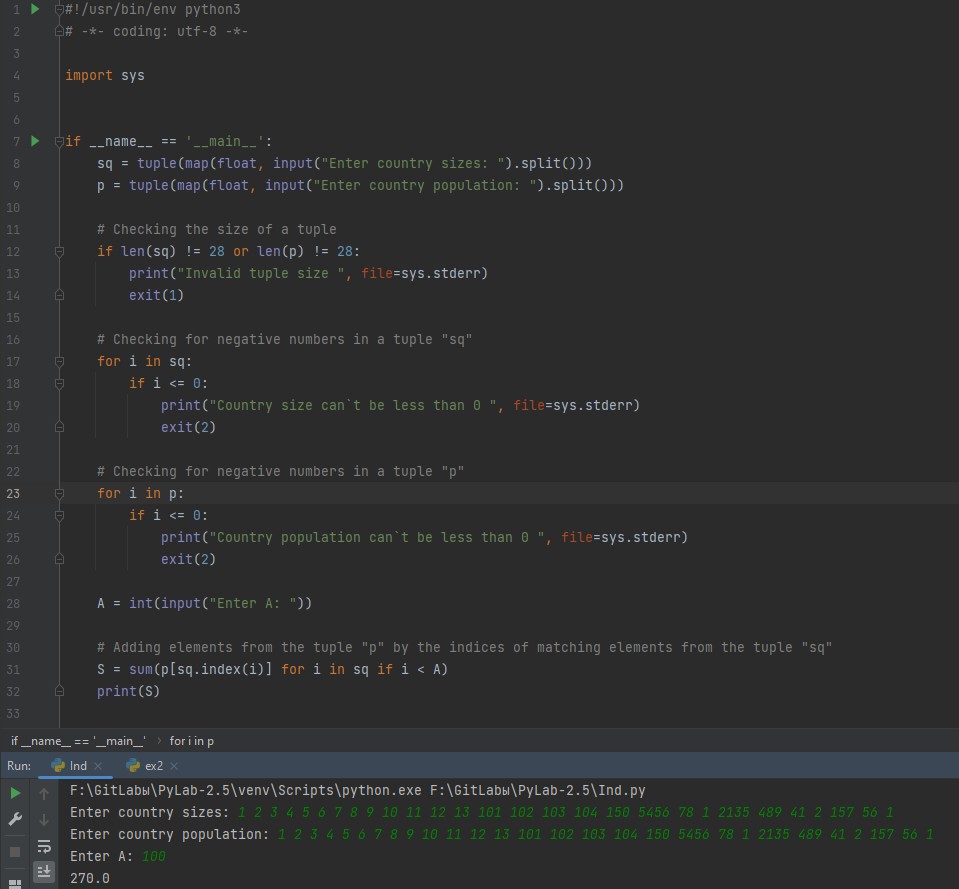


Рисунок №3 – Индивидуальное задание №1

**Контрольные вопросы**

**Вопросы для защиты работы**

1. Что такое списки в языке Python?

Список (list) – это структура данных для хранения объектов различных типов.

1. Каково назначение кортежей в языке Python?

Кортеж (tuple) – это неизменяемая структура данных, которая по своему подобию очень похожа на список. 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. В – третьих - 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). Важно также отметить, что кортежи можно использовать в качестве ключа у словаря.

1. Как осуществляется создание кортежей?

Для создания пустого кортежа можно воспользоваться одной из следующих команд.

>>> a = ()

>>> print(type(a))

<class 'tuple'>

>>> b = tuple()

>>> print(type(b))

<class 'tuple'>

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

>>> a = (1, 2, 3, 4, 5)

>>> print(type(a))

<class 'tuple'>

>>> print(a) (1, 2, 3, 4, 5)

При желании можно воспользоваться функцией tuple().

>>> a = tuple([1, 2, 3, 4])

>>> print(a) (1, 2, 3, 4)

1. Как осуществляется доступ к элементам кортежа?

Доступ к элементам кортежа осуществляется также как к элементам списка – через указание индекса. Но, как уже было сказано – изменять элементы кортежа нельзя!

>>> a = (1, 2, 3, 4, 5)

>>> print(a[0])

>>> print(a[1:3]) (2, 3)

>>> a[1] = 3

Traceback (most recent call last):

File "<pyshell#24>", line 1, in <module> a[1] = 3

TypeError: 'tuple' object does not support item assignment

1. 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

1. Какую роль играют кортежи в множественном присваивании?

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

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

a # 1

b # 2

c # 3

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

a = 100

b = 'foo'

(a, b) = (b, a) a # 'foo'

b # 100

Строку (a, b) = (b, a) нужно понимать как "присвоить в а и b значения из кортежа, состоящего из значений переменных b и a ".

1. Как выбрать элементы кортежа с помощью среза?

С помощью операции взятия среза можно получить другой кортеж.

Общая форма операции взятия среза для кортежа следующая T2 = T1[i:j]

здесь 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.

Операция взятия среза для кортежа может иметь модификации такие же как и для списков.

1. 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

где

T1, T2 – кортежи, для которых нужно выполнить операцию конкатенации. Операнды T1, T2 обязательно должны быть кортежами. При выполнении операции конкатенации для кортежей, использовать в качестве операндов любые другие типы (строки, списки) запрещено;

T3 – кортеж, который есть результатом.

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

T2 = T1 \* n

здесь

T2 – результирующий кортеж;

T1 – исходный кортеж, который нужно повторить n раз; n – количество повторений кортежа T1.

1. Как выполняется обход элементов кортежа?

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

1. Как проверить принадлежность элемента кортежу? Операция in.
2. Какие методы работы с кортежами Вам известны? Метод index(). Поиск позиции элемента в кортеже Метод count(). Количество вхождений элемента в кортеж
3. Допустимо ли использование функций агрегации таких как len() , sum() и т. д. при работе с кортежами?

Да, допустимо.

1. Как создать кортеж с помощью спискового включения.

В этом примере показано использование списковых включений для расчета суммы, однако 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() .