

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)

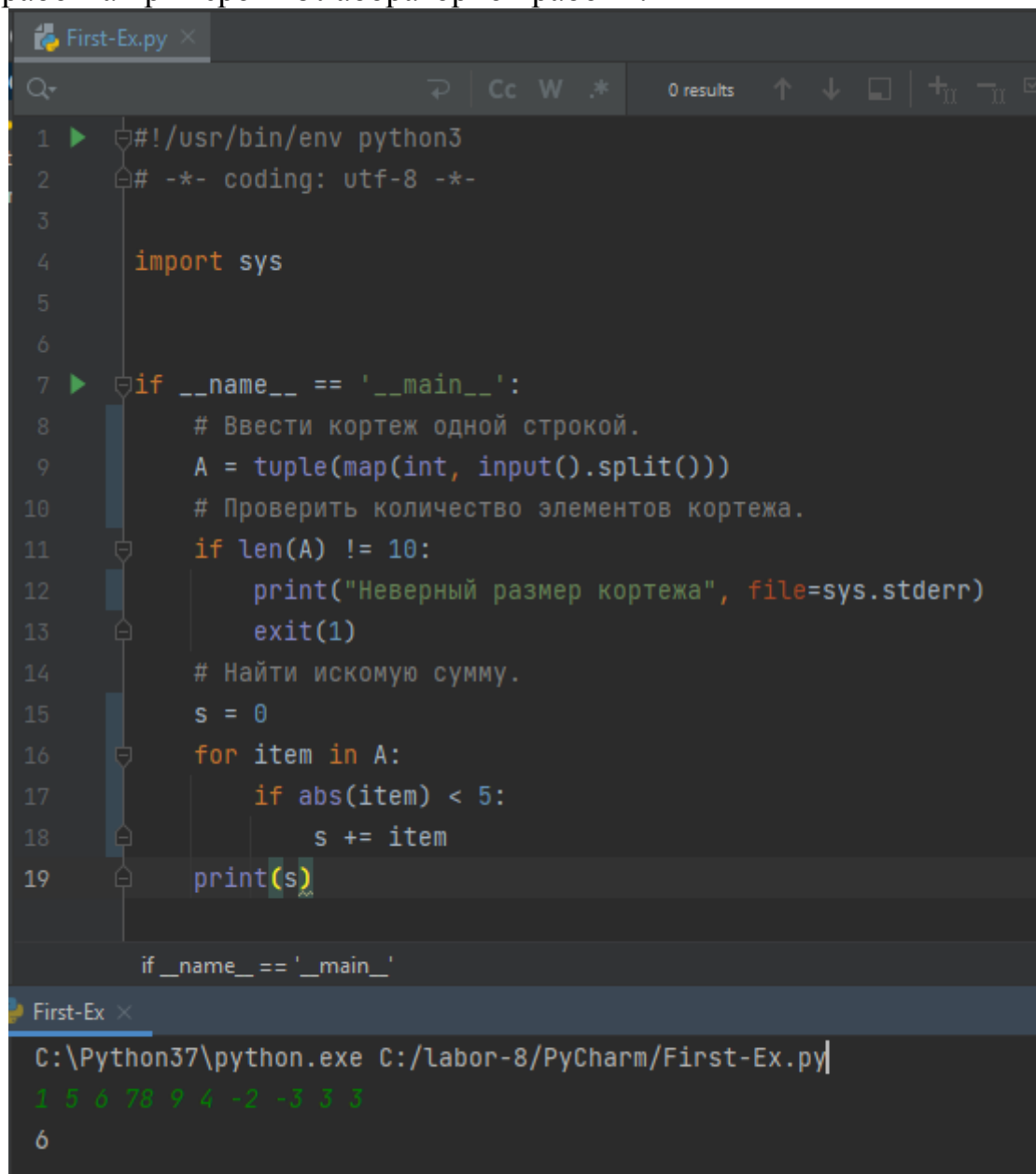
Проверил:

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2022 г.

Проработка примеров из лабораторной работы:



The image shows a PyCharm IDE window with a Python script named 'First-Ex.py' and its execution output. The script is a Python program that takes a tuple of integers as input and calculates the sum of elements whose absolute value is less than 5. The output shows the input tuple and the resulting sum.

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import sys
5
6
7 if __name__ == '__main__':
8     # Ввести кортеж одной строкой.
9     A = tuple(map(int, input().split()))
10    # Проверить количество элементов кортежа.
11    if len(A) != 10:
12        print("Неверный размер кортежа", file=sys.stderr)
13        exit(1)
14    # Найти искомую сумму.
15    s = 0
16    for item in A:
17        if abs(item) < 5:
18            s += item
19    print(s)
```

if __name__ == '__main__'

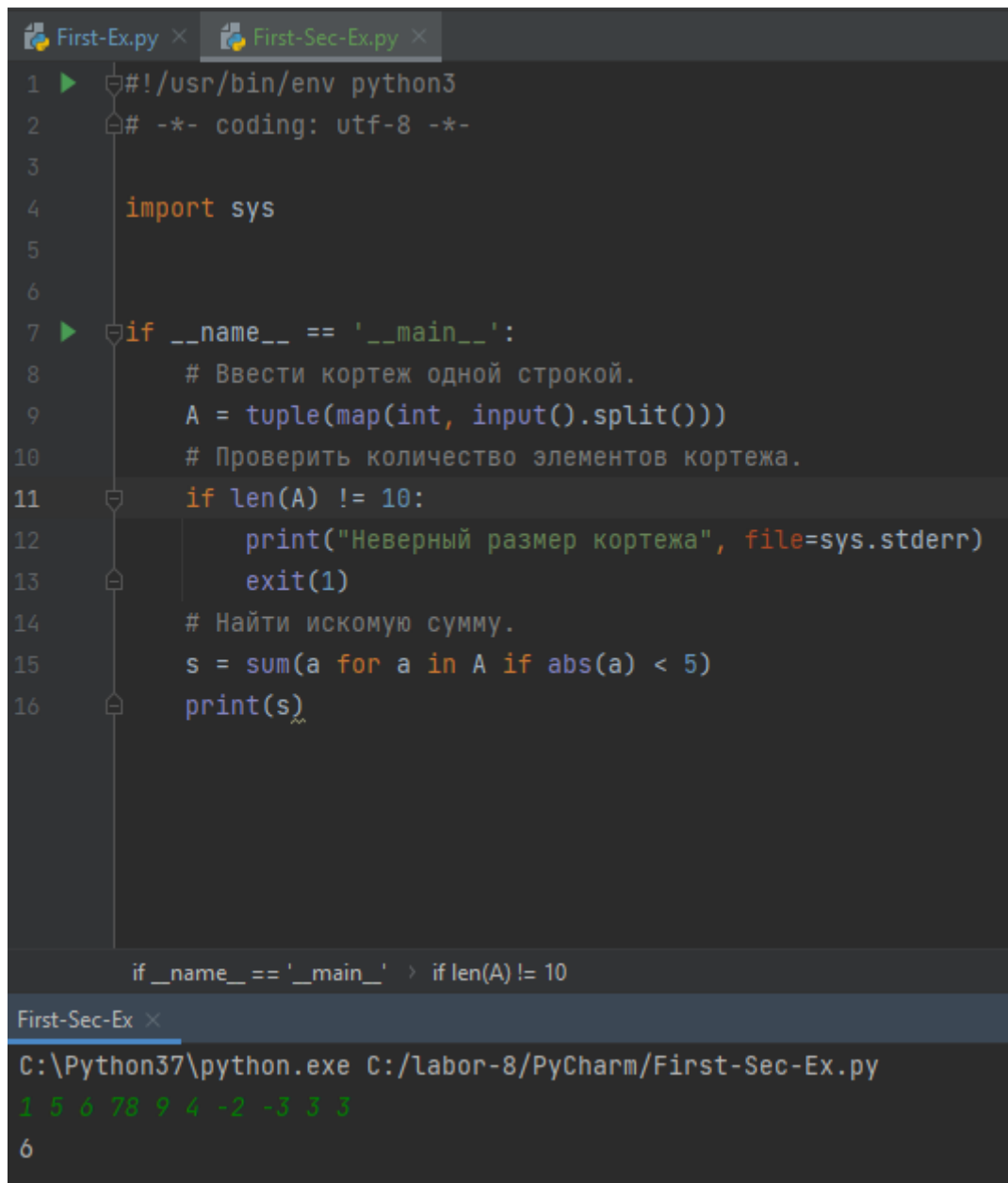
First-Ex ×

C:\Python37\python.exe C:/labor-8/PyCharm/First-Ex.py|

1 5 6 78 9 4 -2 -3 3 3

6

Рисунок №1 – Пример №1



The screenshot shows a PyCharm IDE with two tabs: 'First-Ex.py' and 'First-Sec-Ex.py'. The 'First-Sec-Ex.py' tab is active, displaying a Python script. The script starts with a shebang line and a UTF-8 encoding declaration. It imports the 'sys' module. A main block is defined by an 'if __name__ == '__main__':' condition. Inside this block, a tuple 'A' is created from the input string '1 5 6 78 9 4 -2 -3 3 3'. The script then checks if the length of 'A' is not equal to 10. If true, it prints an error message to stderr and exits with code 1. Otherwise, it calculates the sum of elements in 'A' whose absolute value is less than 5 and prints the result. The output window at the bottom shows the command used to run the script and the input string, followed by the output '6'.

```
1 > #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import sys
5
6
7 > if __name__ == '__main__':
8     # Ввести кортеж одной строкой.
9     A = tuple(map(int, input().split()))
10    # Проверить количество элементов кортежа.
11    if len(A) != 10:
12        print("Неверный размер кортежа", file=sys.stderr)
13        exit(1)
14    # Найти искомую сумму.
15    s = sum(a for a in A if abs(a) < 5)
16    print(s)
```

if __name__ == '__main__' > if len(A) != 10

First-Sec-Ex ×

C:\Python37\python.exe C:/labor-8/PyCharm/First-Sec-Ex.py

1 5 6 78 9 4 -2 -3 3 3

6

Рисунок №2 – Пример №1 с использованием списковых включений

Индивидуальное задание: Известны оценки по информатике каждого ученика класса. В начале кортежа перечислены все пятерки, затем все остальные оценки. Сколько учеников имеет по информатике оценку «5»? Рассмотреть возможность случая, что такую оценку имеют все ученики. Условный оператор не использовать.

```
First-Ex.py × First-Sec-Ex.py × Individual.py ×
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import sys
5
6
7 ▶ if __name__ == '__main__':
8     k = list(map(int, input().split()))
9     a = sorted(k, reverse=True)
10    a = tuple(a)
11    del(k)
12    count = a.count(5)
13    print(f"Кол-во уч. с 5: {count}\n")
14    print(f"Отсортированный кортеж: {a}")
    if __name__ == '__main__'
```

Individual ×

C:\Python37\python.exe C:/labor-8/PyCharm/Individual.py

2 5 3 3 4 5 4 2 5 5 4

Кол-во уч. с 5: 4

Отсортированный кортеж: (5, 5, 5, 5, 4, 4, 4, 3, 3, 2, 2)

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.

2. 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.

3. 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)
```

4. 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

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 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 `a` и `b` значения из кортежа, состоящего из значений переменных `b` и `a`".

7. 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.

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

$$T3 = T1 + T2$$

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 $T1$.

9. Как выполняется обход элементов кортежа?

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

10. Как проверить принадлежность элемента кортежу?

Операция `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?

Да, допустимо.

13. 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()` .