Федеральное государственное автономное образовательное учреждение
высшего образования «Северо-Кавказский федеральный университет»
Институт цифрового развития

ОТЧЕТ ПО ДИСЦИПЛИНЕ
ОСНОВЫ ПРОГРАММНОЙ ИНЖЕНЕРИИ

Лабораторная работа №8

Работа с кортежами в языке Python

Выполнила:
Мирзаева Камилла Мирзаевна
2 курс, группа ПИЖ-б-о-20-1

Принял:
Воронкин Роман Александрович

Ставрополь, 2021 г.

Ход работы:

```
>>> print(a)
[1, 2, 3]
>>> a[1] = 15
>>> b = (1, 2, 30
... )
>>> b = (1, 2, 3)
>>> print(b)
(1, 2, 3)
>>> b[1] = 15
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'tuple' object does not support item assignment
>>>
```

Рисунок 1 - Пример кортежа

```
>>> lst = [10, 20, 30]
>>> tpl = (10, 20, 30)
>>> print(lst.__sizeof__())
104
>>> print(tpl.__sizeof__())
48
>>>
```

Рисунок 2 - Размер кортежа

```
>>> a = ()
>>> print(type(a))
<class 'tuple'>
>>>  b = tuple()
  File "<stdin>", line 1
    b = tuple()
IndentationError: unexpected indent
>>> print(type(b))
<class 'tuple'>
>>>
```

Рисунок 3 - Создание кортежа

```
>>> a = (1, 2, 3, 4, 5)
>>> print(a[0])
1
>>> print(a[1:3])
(2, 3)
>>> a[1] = 3
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'tuple' object does not support item assignment
>>>
```

Рисунок 4 - Доступ к элементам кортежа

```
>>> del a[0]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'tuple' object doesn't support item deletion
```

Рисунок 5 -Удаление кортежа

```
>>> a = (1, 2, 3, 4, 5)
>>> del a
>>> print(a)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'a' is not defined
>>>
```

Рисунок 6 - Удаление кортежа

```
>>> lst = [1, 2, 3, 4, 5]
>>> print(type(lst))
<class 'list'>
>>> print(lst)
[1, 2, 3, 4, 5]
>>> tpl = tuple(lst)
>>> print(type(tpl))
<class 'tuple'>
>>> print(tpl)
(1, 2, 3, 4, 5)
>>>
```

Рисунок 7 - Преобразование кортежа в список и обратно

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

(name, age) = name_and_age
name # 'Bob'
age  # 42
```

Рисунок 8 - Деструктуризация

```
(quotient, modulo) = div_mod(13, 4)
```

Рисунок 9 – Деструктуризация

```
(a,) = (42,)
a # 42
```

Рисунок 10 - Деструктуризация

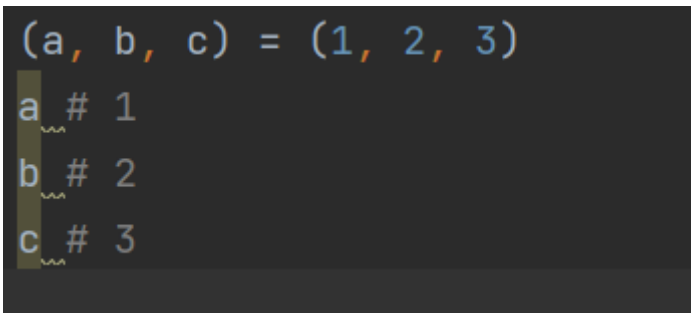


Рисунок 11 - Пример присваивания

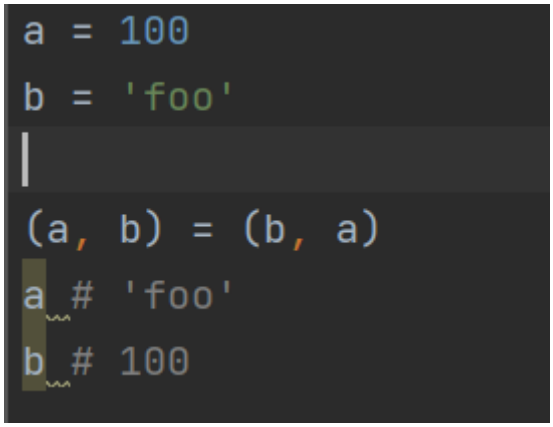


Рисунок 12 - Пример присваивания

```python
# Операция tuple()
# 1. Создание кортежа из слова 'Hello'
d = tuple('Hello'); # d = ('H', 'e', 'l', 'l', 'o')

# 2. Создание кортежа из списка
# Заданный список
lst = [2, "abc", 3.88]

# Создать кортеж
e = tuple(lst) # e = (2, 'abc', 3.88)

# 3. Создание кортежа из другого кортежа
f = tuple((3, 2, 0, -5)) # f = (3, 2, 0, -5)
```

Рисунок 13 - Создание кортежа из итерированного объекта

```python
# Операция [i:j] - взятие среза
# 1. Кортеж, содержащий целые числа
A = (0, 1, 2, 3)
item = A[0:2] # item = (0, 1)

# 2. Кортеж, содержащий список
A = (2.5, ['abcd', True, 3.1415], 8, False, 'z')
item = A[1:3] # item = (['abcd', True, 3.1415], 8)

# 3. Кортеж, содержащий вложенный кортеж
A = (3, 8, -11, "program")
B = ("Python", A, True)
item = B[:3] # item = ('Python', (3, 8, -11, 'program'), True)
item = B[1:] # item = ((3, 8, -11, 'program'), True)
```

Рисунок 14 - Взятие среза в кортеже

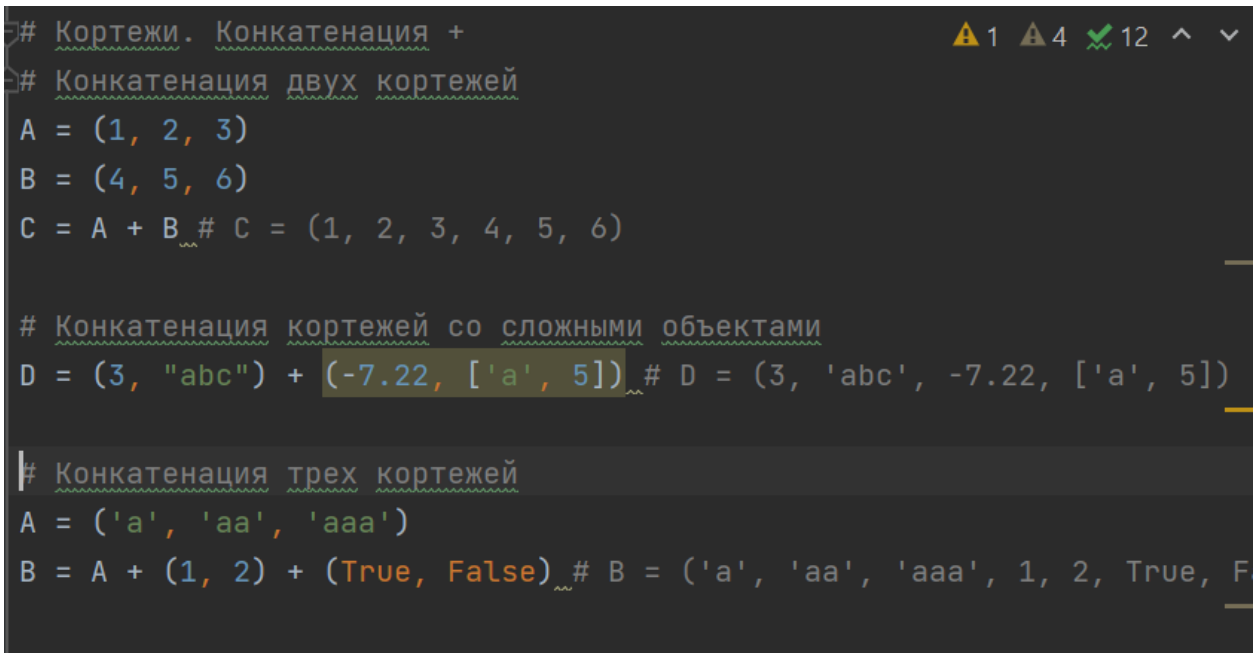


Рисунок 15 – Конкатенация

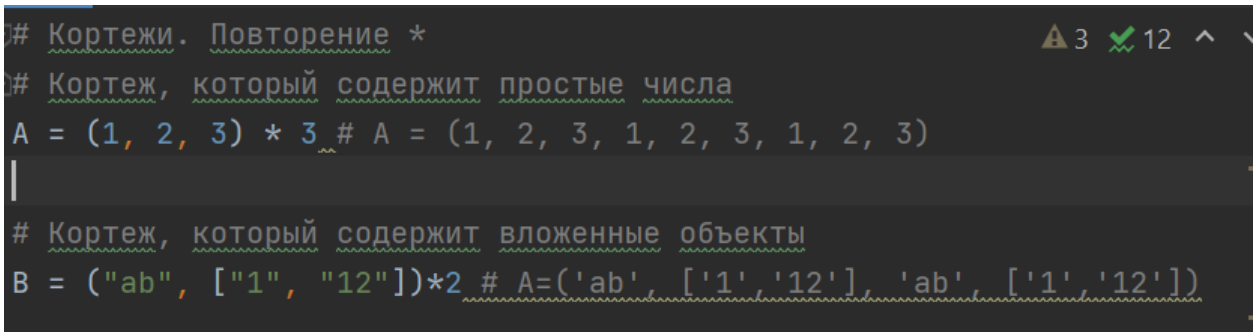


Рисунок 16 - Повторение

```
k = 0 # количество положительных чисел

while i < len(A):
    if (A[i]<0):
        k = k + 1
    i = i + 1

# Вывести результат
print("k = ", k)

# 3. Обход в цикле for
# Заданный кортеж, содержащий строки
A = ("abc", "ad", "bcd")
```

```
main
C:\Users\milymirr\PycharmProjects\laba2_5\venv\Scripts\python.ex
abc
abcd
bcd
cde
k =  3
A =  ('abc', 'ad', 'bcd')
B =  ['abcabc', 'adad', 'bcdbcd']
```

Рисунок 17 - Обход кортежа в цикле

```
# Проверка вхождения элемента в кортеж
# Оператор in
# Заданный кортеж, который содержит строки
A = ("abc", "abcd", "bcd", "cde")

# Ввести элемент
item = str(input("s = "))

if (item in A):
    print(item, " in ", A, " = True")
else:
    print(item, " in ", A, " = False")
```

```
if (item in A)
main
C:\Users\milymirr\PycharmProjects\laba2_5\venv\Scripts\pyt
s = abc
abc  in  ('abc', 'abcd', 'bcd', 'cde')  = True

Process finished with exit code 0
```

Рисунок 18 - Проверка вхождения элемента в кортеж

```
# Заданный кортеж
A = ("Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat")

# Запрос к вводу названия дня недели
day = str(input("Enter day: "))

# Корректно вычислить индекс
if day in A: # проверка, есть ли строка day в кортеже A
    num = A.index(day)
    print("Number of day = ", num + 1)
else:
    num = -1
    print("Wrong day.")
```

```
example1
C:\Users\milymirr\PycharmProjects\laba2_5\venv\Scripts\pyt
Enter day: Sun
Number of day =  1
```

Рисунок 19 - Поиск позиции элемента в кортеже

```
# Метод count - подсчет количества вхождений элемента в
# Заданный кортеж
A = ("ab", "ac", "ab", "ab", "ca", "ad", "jklmn")

d1 = A.count("ab") # d1 = 3
d2 = A.count("jprst") # d2 = 0
d3 = A.count("ca") # d3 = 1

print("d1 = ", d1)
print("d2 = ", d2)
print("d3 = ", d3)
```

```
example1
C:\Users\milymirr\PycharmProjects\laba2_5\venv\Scripts\pyt
d1 =  3
d2 =  0
d3 =  1
```

Рисунок 20 - Количество вхождений элемента в кортеж

Пример №1.

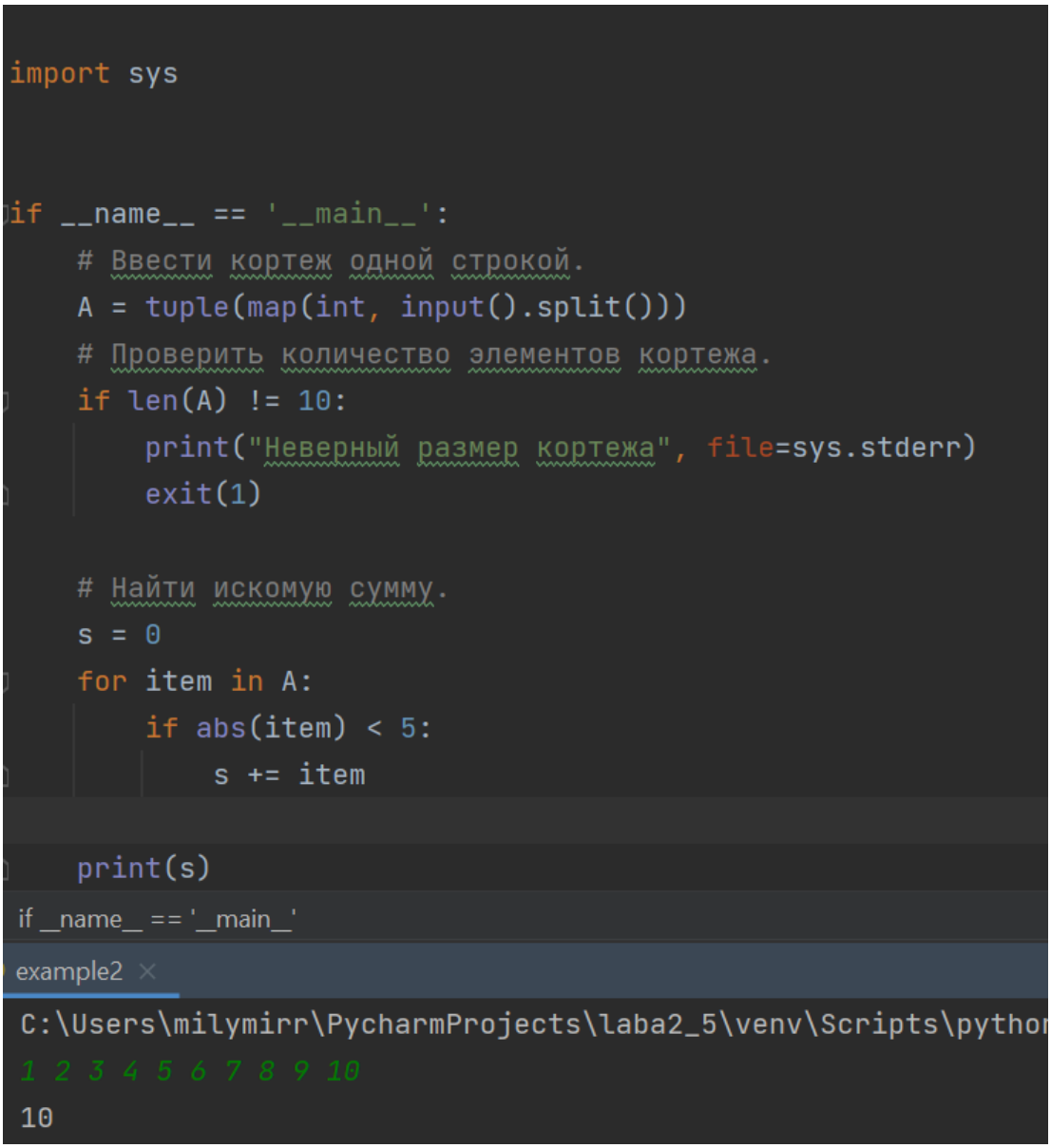
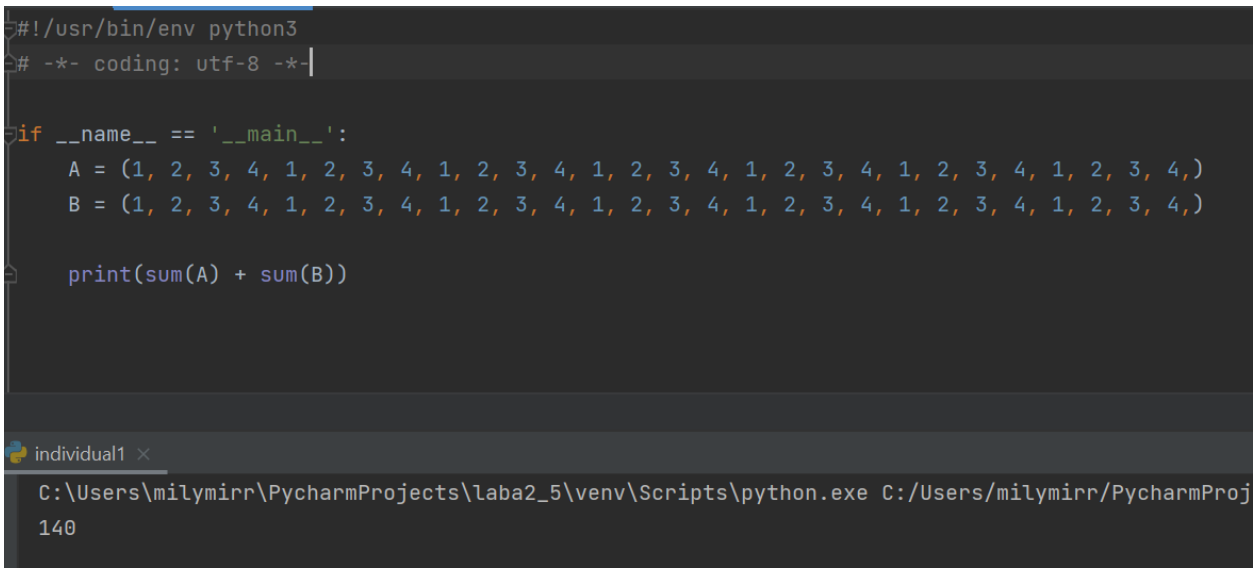


Рисунок 21 - Пример №1

Индивидуально задание.

Вариант №15.

Известно количество мячей, забитых футбольной командой за каждую игру в двух чемпионатах, которое хранится в двух кортежах. В каждом из чемпионатов команда сыграла 26 игр. Найти общее количество мячей, забитых командой в двух чемпионатах.



Контрольные вопросы:

1. Что такое списки в языке Python?

Список (list) - структура данных для хранения объектов различных типов. Так вот, список очень похож на массив, только, как было уже сказано выше, 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?

Кортеж (tuple) – это неизменяемая структура данных, которая по своему подобию очень похоже на список.

Назначение кортежей:

1) Обезопасить данные от случайного изменения.

2) Экономия места.

3) Прирост производительности, который связан с тем, что кортежи работают быстрее, чем списки.

3. Как осуществляется создание кортежей?

1) A = ()

2) A = tuple([1, 2 , 3])

4. Как осуществляется доступ к элементам кортежа?

Доступ к элементам кортежа осуществляется через указание индекса. print(a[0]) или print(a[1:3])

5. Зачем нужна распаковка (деструктуризация) кортежа?

Деструктуризация кортежа нужно для быстро разбиения кортежа на отдельные элементы, для более удобного доступа и работы.

6. Какую роль играют кортежи в множественном присваивании?

Элементам кортежа можно сразу последовательно присваивать значения. А также используя множественное присваивание, можно провернуть интересный трюк: обмен значениями между двумя переменными.

7. Как выбрать элементы кортежа с помощью среза?

Необходимо ввести: item = A[0:2], item = B[:3], item = B[1:] Общая форма операции взятия среза для кортежа:

```
T2 = T1[i:j]
```

8. Как выполняется конкатенация и повторение кортежей?

Для кортежей можно выполнять операцию конкатенации, которая обозначается символом +. Общая форма операции:

```
T3 = T1 + T2
```

Кортеж может быть образован путём операции повторения, обозначаемой символом *.

Общая форма операции:

```
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?

Проверить принадлежность элемента кортежу можно с помощью операции in.

```
if (item in A):
    print(item, " in ", A, " = True")
else:
    print(item, " in ", A, " = False")
```

11. Какие методы работы с кортежами Вам известны?

Метод index(): pos = A.index(item) Метод count(): k = A.count(item) где item – элемент кортежа.

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.

```
x = 10
a = tuple([i for i in range(x)])
a = tuple(int(i) for i in input().split())
```