

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ**

**Отчет по лабораторной работе №8 по дисциплине:
основы программной инженерии**

Выполнила:

студент группы ПИЖ-б-о-20-1

Лазарева Дарья Олеговна

Проверил:

доцент кафедры инфокоммуникаций

Романкин Р.А.

Ставрополь, 2021 г.

Ход работы:

1. Изменение элемента в списке

```
>>> a = [1, 2, 3]
>>> print(a)
[1, 2, 3]
>>> a[1] = 15
>>> print(a)
[1, 15, 3]
```

2. Попытка внести изменения в кортеже

```
>>> 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
```

3. Создание кортежей

```
>>> a = ()
>>> print(type(a))
<class 'tuple'>
>>> b = tuple()
>>> print(type(b))
<class 'tuple'>
```

```
>>> a = (1, 2, 3, 4, 5)
>>> print(type(a))
<class 'tuple'>
>>> print(a)
(1, 2, 3, 4, 5)
```

4. Использование функции tuple в качестве создания кортежей

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

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

6. Удаление кортежей. Отдельные элементы удалить невозможно

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

7. Преобразование кортежа в список и обратно

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

```
>>> tpl = (2, 4, 6, 8, 10)
>>> print(type(tpl))
<class 'tuple'>
>>> print(tpl)
(2, 4, 6, 8, 10)
>>> lst = list(tpl)
>>> print(type(lst))
<class 'list'>
>>> print(lst)
[2, 4, 6, 8, 10]
```

8. Операции над кортежами. Обход кортежа в цикле

```
1 A = ("abc", "abcd", "bcd", "cde")
2 for item in A:
3     print(item)
4
```

modul1 x

C:\Users\79616\anaconda3\python.exe D:/YЧЕБ

abc
abcd
bcd
cde

```
A = (-1, 3, -8, 12, -20)
i = 0
k = 0
while i < len(A):
    if (A[i]<0):
        k = k + 1
    i = i + 1

print("k = ", k)
```

modul1 x

C:\Users\79616\anaconda3\python.exe D:/YЧЕБ

k = 3

9. Обход кортежа в цикле. Кортеж, содержащий строки

```
A = ("abc", "ad", "bcd")
B = [item * 2 for item in A]
print("A = ", A)
print("B = ", B)
```

modul1 x

C:\Users\79616\anaconda3\python.exe D:/УЧЕБА/ОПИ/8/modul1.py

A = ('abc', 'ad', 'bcd')

B = ['abcabc', 'adad', 'bcdbcd']

10. Операция in. Проверка вхождения элемента в кортеж

```
A = ("abc", "abcd", "bcd", "cde")
item = str(input("s = "))
if (item in A):
    print(item, " in ", A, " = True")
else:
    print(item, " in ", A, " = False")
```

modul1 x

C:\Users\79616\anaconda3\python.exe D:/УЧЕБА/ОПИ/8/modul1.py

s = abc

abc in ('abc', 'abcd', 'bcd', 'cde') = True

11. Методы работы с кортежами. Метод index()

```
A = ("Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat")
day = str(input("Enter day: "))
if day in A:
    num = A.index(day)
    print("Number of day = ", num + 1)
else:
    num = -1
    print("Wrong day.")
```

modul1 x

C:\Users\79616\anaconda3\python.exe D:/УЧЕБА/ОПИ/8/modul1.py

Enter day: Tue

Number of day = 3

12. Метод count(). Количество вхождений элемента в 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)
```

modul1 x

C:\Users\79616\anaconda3\python.exe D:/УЧЕБА/ОПИ/8/modul1.py

d1 = 3

d2 = 0

d3 = 1

Пример 1: решить задачу, используя кортежи

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import sys

if __name__ == '__main__':
    A = tuple(map(int, input().split()))
    if len(A) != 10:
        print("неверный размер кортежа", file=sys.stderr)
        exit(1)

    s = 0
    for item in A:
        if abs(item) < 5:
            s += item

    print(s)
```

modul1 x

C:\Users\79616\anaconda3\python.exe D:/УЧЕБА/ОПИ/8/modul1.py

1 2 3 4 5 6 7 8 9 5

10

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import sys

if __name__ == '__main__':
    A = list(map(int, input().split()))
    if len(A) != 10:
        print("неверный размер списка", file=sys.stderr)
        exit(1)

    s = sum(a for a in A if abs(a) < 5)
    print(s)
```

modul1.1 x

C:\Users\79616\anaconda3\python.exe D:/УЧЕБА/ОПИ/8/modul1.1.py

1 2 3 4 5 6 7 8 9 5

10

Пример 2: Решить задачу, с помощью списковых включений

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import sys

if __name__ == '__main__':
    A = tuple(map(int, input().split()))
    if len(A) != 10:
        print("неверный размер кортежа", file=sys.stderr)
        exit(1)

    s = 0
    for item in A:
        if abs(item) < 5:
            s += item

    print(s)
```

if __name__ == '__main__'

modul1 x

C:\Users\79616\anaconda3\python.exe D:/УЧЕБА/ОПИ/8/modul1.py

1 2 1 3 4 5 6 7 8 9 9

неверный размер кортежа

Выполнение индивидуального задания (вариант 13): 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.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import sys

if __name__ == '__main__':
    A = tuple(map(int, input("Введите оценки: ").split()))
    if len(A) == 0:
        print("Неверный размер кортежа", file=sys.stderr)
        exit(1)

    s = 0
    for item in A:
        if item == 5:
            s += 1
    print(f"Количество пятерок :", s)
```

C:\Users\79616\anaconda3\python.exe D:/	C:\Users\79616\anaconda3\python.exe
Введите оценки: 5 5 5 5 4 3 4 2 3	Введите оценки: 5 5 5 5 5 5 5
Количество пятерок : 4	Количество пятерок : 7

Вопросы для защиты работы:

1. Что такое кортеж в языке Python? Кортеж – неизменяемая структура данных, которая по своему подобию очень похожа на список
2. Каково назначение кортежей в языке Python? В целях экономии места, безопасности данных от случайного изменения
3. Как осуществляется создание кортежей?

```
>>> a = ()
>>> print(type(a))
<class 'tuple'>
>>> b = tuple()
>>> print(type(b))
<class 'tuple'>
```

4. Как осуществляется доступ к элементам кортежа? Доступ к элементам кортежа осуществляется как к элементам списка – через указание индекса
5. Зачем нужна распаковка (деструктуризация) кортежа? Обращение по индексу не самый удобный способ работы с кортежами. Кортежи часто содержат значения разных типов, и помнить, по какому индексу что лежит – непросто. Можно разобрать кортеж

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

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

Именно таким способом принято получать и сразу разбирать значения, которые возвращает функция (если таковая возвращает несколько значений, конечно):

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

Соответственно кортеж из одного элемента нужно разбирать так:

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

6. Какую роль играют кортежи в множественном присваивании? Используя множественное присваивание можно совершить обмен между двумя переменными

```

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.

Например

```

# 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)

```

8. Как выполняется конкатенация и повторение кортежей?

```
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.

9. Как выполняется обход элементов кортежа? Элементы кортежа можно последовательно просмотреть с помощью операторов цикла `while` или `for`