

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО
ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего
образования
«Северо-Кавказский федеральный
университет» Кафедра инфокоммуникаций**

Отчет по лабораторной работе №2.5

**по дисциплине «Основы программной
инженерии»**

**Выполнил студент группы ПИЖ-б-о-20-1
Турклиев В Н « » _____ 20_г.
Подпись студента _____
Работа защищена « » _____ 20_г.
Проверил Воронкин Р.А. _____
(подпись)**

ВЫПОЛНЕНИЕ

Пример 1.

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

import sys

if __name__ == '__main__':
    # Ввести кортеж одной строкой.
    A = tuple(map(int, input().split()))
    # Проверить количество элементов кортежа.
    if len(A) != 10:
        print("Неверный размер кортежа", file=sys.stderr)
        exit(1)

    # Найти искомую сумму.
    s = 0
    for item in A:
        if abs(item) < 5:
            s += item

    print(s)

if __name__ == '__main__':
```

1 x

C:\Users\Vova\AppData\Local\Programs\Python\Python39\python.exe

5 6 -4 3 2 1 15 18 9 10

2

Пример 2.

```
#!/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)
```

if __name__ == '__main__'

2 x

C:\Users\Vova\AppData\Local\Programs\Python\Python39\python.exe

1 2 3 4 5 6 7 8 9 5

10

Индивидуальное задание

```
m = (1, 2, 3, 4, 5, 6, 78, 8, 4, 5, 64, 3, 2, 34)
n = []
for i in range(len(m)):
    if (i + 1) % 2 != 0:
        n.append(i*m[i])
    else:
        n.append(m[i]/i)
n = tuple(n)
print(type(n))
print(n)
```

zadanie x

C:\Users\Vova\AppData\Local\Programs\Python\Python39\python.exe "C:/Users/Vova/Desktop/учеба/основы им/lab8/lab8/pyProj/zadanie.py"

<class 'tuple'>

(0, 2.0, 6, 1.3333333333333333, 20, 1.2, 468, 1.1428571428571428, 32, 0.5555555555555556, 640, 0.2727272727272727, 24, 2.6153846153846154)

Ссылка - <https://github.com/vegas007gof/lab8>

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`

10. Как проверить принадлежность элемента кортежу?

Проверить принадлежность элемента можно с помощью операции `in`

11. Как методы работы с кортежами вам известны? Для поиска позиции элемента в кортеже – метод `index()`. Количество вхождений элемента в кортеж – метод `count()`

12. Допустимо ли использование функций агрегации таких как `len()` `sum()` 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())
```