

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития  
Кафедра инфокоммуникаций

**ОТЧЕТ**  
**ПО ЛАБОРАТОРНОЙ РАБОТЕ №8**  
**дисциплины «Программирование на python»**

Выполнил:  
Кожуховский Виктор Андреевич  
2 курс, группа ИВТ-б-о-22-1,  
09.03.01 «Информатика и  
вычислительная техника»,  
направленность (профиль)  
«Программное обеспечение средств  
вычислительной  
техники и автоматизированных систем  
», очная форма обучения

---

(подпись)

Руководитель практики:  
Воронкин Роман Александрович

---

(подпись)

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

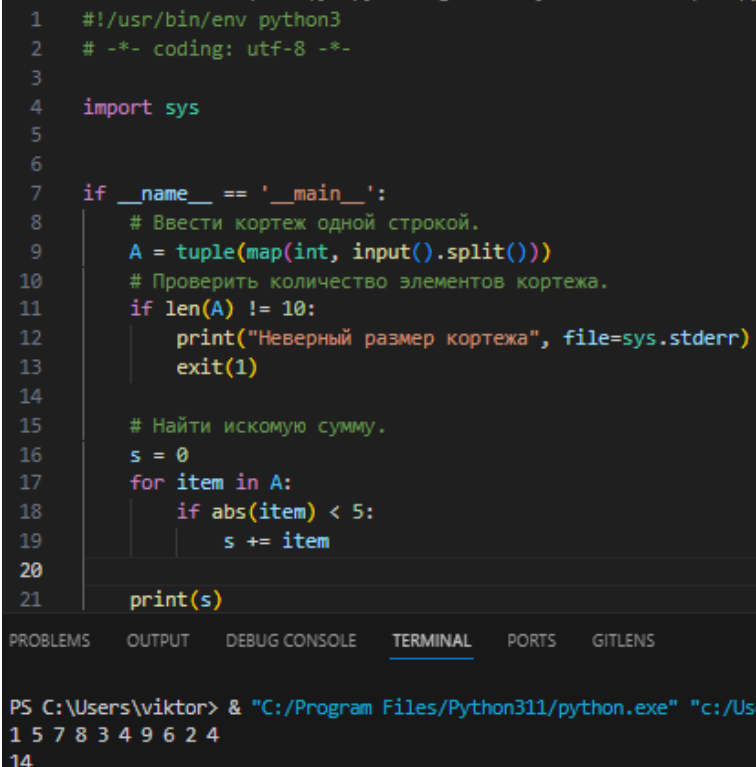
Ставрополь, 2023 г.

**Тема:** Работа с кортежами в языке Python

**Цель работы:** приобретение навыков по работе с кортежами при написании программ с помощью языка программирования Python версии 3.x.

### Методика и порядок выполнения работы

1. Изучил теоретический материал работы.
2. Создал общедоступный репозиторий на GitHub, в котором использована лицензия MIT и язык программирования Python.
3. Выполнил клонирование созданного репозитория.
4. Дополнил файл .gitignore необходимыми правилами для работы с IDE PyCharm.
5. Организовал свой репозиторий в соответствии с моделью ветвления git-flow.
6. Создал проект в папке репозитория.
7. Проработал примеры лабораторной работы. Создал для каждого примера отдельный модуль языка Python. Зафиксировал изменения в репозитории.



```
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
15     # Найти искомую сумму.
16     s = 0
17     for item in A:
18         if abs(item) < 5:
19             s += item
20
21     print(s)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

PS C:\Users\viktor> & "C:/Program Files/Python311/python.exe" "c:/Us  
1 5 7 8 3 4 9 6 2 4  
14

Рисунок 1. Код и выполнение примера 1

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

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

PS C:\Users\viktor> & "C:/Program Files/Python311/python.exe" "c:/Us  
1 5 7 8 3 4 9 6 2 4  
14

Рисунок 2. Код и выполнение примера 2

8. Выполнил индивидуальные задания, согласно своему варианту.

Вариант 16

Известны данные о мощности двигателя (в лошадиных силах — л. с.) и стоимости 30 марок легковых автомобилей. Напечатать стоимость каждого из автомобилей, у которых мощность двигателя не превышает 80 л. с.

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import sys
5
6
7  if __name__ == '__main__':
8      # Ввести список одной строкой.
9      A = tuple(map(int, input("Введите мощности через пробел: ").split()))
10     B = tuple(map(int, input("Введите стоимости через пробел: ").split()))
11     # Проверить количество элементов списка.
12     if len(A) != 30 or len(B) != 30:
13         print("Неверный размер списка", file=sys.stderr)
14         exit(1)
15
16     # Найти искомую сумму.
17     prices = [index for item, index in zip(A, B) if item < 80]
18     print("Цены: ", prices)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

PS C:\Users\viktor> & "C:/Program Files/Python311/python.exe" "c:/Users/viktor/Desktop/ckdy/python/gitstuff/PyLab8/indiv 1.py"  
Введите мощности через пробел: 75 82 69 95 76 55 88 60 99 43 67 77 93 85 58 78 72 81 54 63 96 65 74 98 44 97 70 49 86 52  
Введите стоимости через пробел: 20000 15000 23000 21000 18000 12000 25000 13000 26000 11000 17000 22000 24000 16000 10000 19500 17500 15500 12500 14000 21500 16500 18500 20000 10500 20500 13500 14500 19000 11500  
Цены: [20000, 23000, 18000, 12000, 13000, 11000, 17000, 22000, 10000, 19500, 17500, 12500, 14000, 16500, 18500, 10500, 13500, 14500, 11500]

Рисунок 3. Код и выполнение индивидуального задания

10. Зафиксировал сделанные изменения в репозитории.

11. Добавил отчет по лабораторной работе в формате PDF в папку doc репозитория. Зафиксировал изменения.

12. Выполнил слияние ветки для разработки с веткой main / master.

13. Отправил сделанные изменения на сервер GitHub.

### Вопросы для защиты работы

1. Что такое кортеж в языке Python?

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

2. Каково назначение кортежей в языке Python?

Существует несколько причин, по которым стоит использовать кортежи вместо списков. Одна из них – это обезопасить данные от случайного изменения. Во-вторых – 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).

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)
```

4. Как осуществляется доступ к элементам кортежа?

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

```
>>> 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 "<pyshell#24>", line 1, in <module>
```

```
a[1] = 3
```

```
TypeError: 'tuple' object does not support item assignment
```

5. Зачем нужна распаковка (деструктуризация) кортежа?

Кортежи часто содержат значения разных типов, и помнить, по какому индексу что лежит — очень непросто.

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.

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

```
a # 1
```

```
b # 2
```

```
c # 3
```

Используя множественное присваивание, можно произвести обмен значениями между двумя переменными:

```
a = 100
```

```
b = 'foo'
```

```
(a, b) = (b, a)
```

```
a # 'foo'
```

```
b # 100
```

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]
```

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
```

```
# Кортежи. Конкатенация +
```

```
# Конкатенация двух кортежей
```

```
A = (1, 2, 3)
```

```
B = (4, 5, 6)
```

```
C = A + B # C = (1, 2, 3, 4, 5, 6)
```

```
# Конкатенация кортежей со сложными объектами
```

```
D = (3, "abc") + (-7.22, ['a', 5]) # D = (3, 'abc', -7.22, ['a', 5])
```

```
# Конкатенация трех кортежей
```

```
A = ('a', 'aa', 'aaa')
```

```
B = A + (1, 2) + (True, False) # B = ('a', 'aa', 'aaa', 1, 2, True, False)
```

9. Как выполняется обход элементов кортежа?

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

```
# Обход кортежа в цикле
```

```
# 1. Цикл for
```

```
# Заданный кортеж
```

```
A = ("abc", "abcd", "bcd", "cde")
```

```
# Вывести все элементы кортежа
```

```
for item in A:
```

```
    print(item)
```

```

# 2. Цикл while
# Исходный кортеж - целые числа
A = (-1, 3, -8, 12, -20)
# Вычислить количество положительных чисел
i = 0
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")
# Сформировать новый список из элементов кортежа A,
# в новом списке B, каждый элемент удваивается
B = [item * 2 for item in A]
print("A = ", A)
print("B = ", B)

10. Как проверить принадлежность элемента кортежу?
# 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")
```

11. Какие методы работы с кортежами Вам известны?

Методы `index()` (Поиск позиции элемента в кортеже) и `count()` (Количество вхождений элемента в кортеж).

12. Допустимо ли использование функций агрегации таких как `len()` , `sum()` и т. д. при работе с кортежами?

Допустимо.

13. Как создать кортеж с помощью спискового включения.

```
list_of_numbers = [1, 2, 3, 4, 5]
```

```
tuple_of_squares = tuple(number ** 2 for number in list_of_numbers)
```