

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 «Основы программной инженерии»**

Выполнил:  
Плугатырев Владислав Алексеевич  
2 курс, группа ПИЖ-б-о-22-1,  
09.03.04 «Программная инженерия»,  
направленность (профиль) «Разработка и  
сопровождение программного  
обеспечения», очная форма обучения

---

(подпись)

Доцент кафедры инфокоммуникаций  
Воронкин Роман Александрович

---

(подпись)

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

Ставрополь, 2023 г.

**Тема:** работа со кортежами в языке Python.

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

Порядок выполнения работы

1. Создание репозитория GitHub.

**Create a new repository**

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

---

*Required fields are marked with an asterisk (\*).*

**Owner \*** **Repository name \***

BigLofanbl4 / lab8

✔ lab8 is available.

Great repository names are short and memorable. Need inspiration? How about **crispy-octo-potato** ?

**Description** (optional)

---

☐ **Public**  
Anyone on the internet can see this repository. You choose who can commit.

☒ **Private**  
You choose who can see and commit to this repository.

---

**Initialize this repository with:**

☐ **Add a README file**  
This is where you can write a long description for your project. [Learn more about READMEs.](#)

**Add .gitignore**

.gitignore template: **Python**

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

**Choose a license**

License: **MIT License**

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

---

You are creating a private repository in your personal account.

---

**Create repository**

Рисунок 1 – Создания репозитория

2. Проработал примеры из лабораторной работы.

```
#!/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)
```

Рисунок 1 – Код из примера 1

```
1 3 -4 3 5 6 -34 -45 -2 10
1
```

Рисунок 2 – Вывод программы из примера 1

3. Фирме принадлежат два магазина. Известна стоимость товаров, проданных в каждом магазине за каждый день в июле и августе, которая хранится в двух массивах. Получить общую стоимость проданных фирмой товаров за два месяца.

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

if _ (variable) julySales: tuple[Literal[15000], Literal[7300]]
    julySales = (15000, 7300)
    augustSales = (3456, 8900)
    print(f"Sum of sales in July and August = {sum(julySales) + sum(augustSales)}")
```

Рисунок 3.1 – Код программы

**Sum of sales in July and August = 34656**

### Рисунок 3.2 – Вывод программы

#### Ответы на контрольные вопросы

1. Список – это структура данных, которая хранит различные объекты.
2. Кортежи используются для хранения неизменяемых последовательностей объектов разных типов.
3. Кортежи создаются с использованием круглых скобок и элементы разделяются запятыми. Например: `my_tuple = (1, 2, 3)`. Или с помощью функции `tuple`. Например: `my_tuple = tuple()`.
4. Доступ к элементам кортежа осуществляется по индексу.
5. Распаковка (деструктуризация) кортежа позволяет присвоить его элементы отдельным переменным.
6. Используя множественное присваивание, можно провернуть интересный трюк: обмен значениями между двумя переменными.
7. Для выбора элементов кортежа с помощью среза используется синтаксис `[start:end]`.
8. Конкатенация кортежей выполняется с помощью оператора `+`. Повторение кортежа выполняется с помощью оператора `*`.
9. Обход элементов кортежа может осуществляться с помощью циклов.
10. Принадлежность элемента кортежу можно проверить с помощью оператора `in`.
11. `count()`, `index()`.
12. Да, функции агрегации такие как `len()`, `sum()` и т.д. могут быть использованы при работе с кортежами.
13. Кортежи могут быть созданы с помощью спискового включения. Например: `my_tuple = tuple(i for i in range(10))`.