

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 №2.5 (8)**  
**дисциплины**  
**«Программирование на Python»**

Выполнил:  
Ибрагимов Муса Айнудинович  
2 курс, группа ИВТ-6-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:  
Воронкин Роман Александрович

---

(подпись)

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

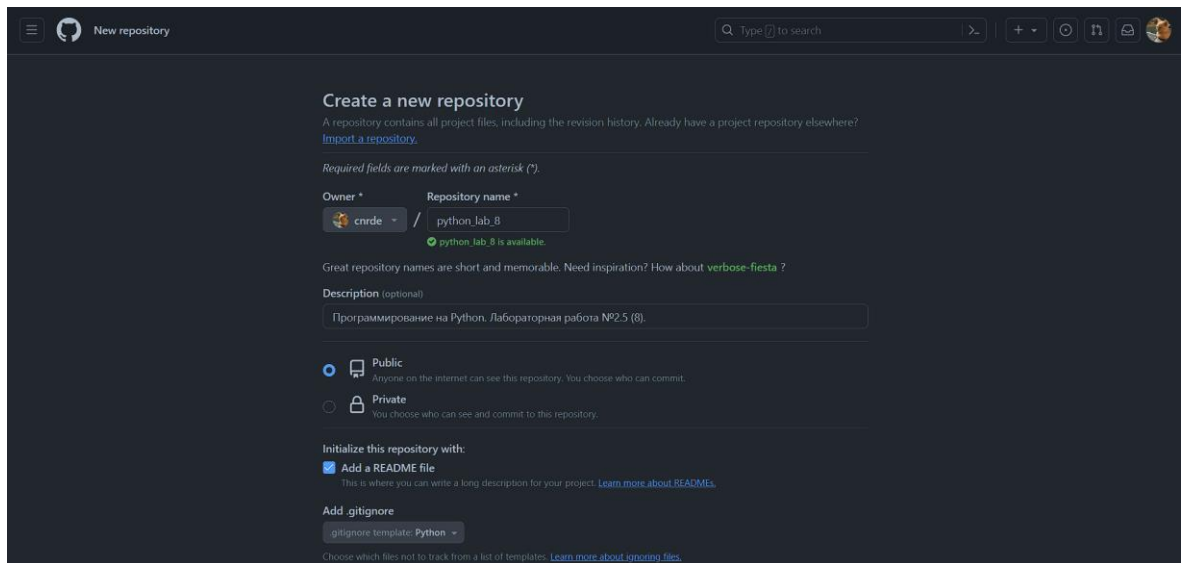
Ставрополь, 2024 г.

Тема: Работа с кортежами в языке Python.

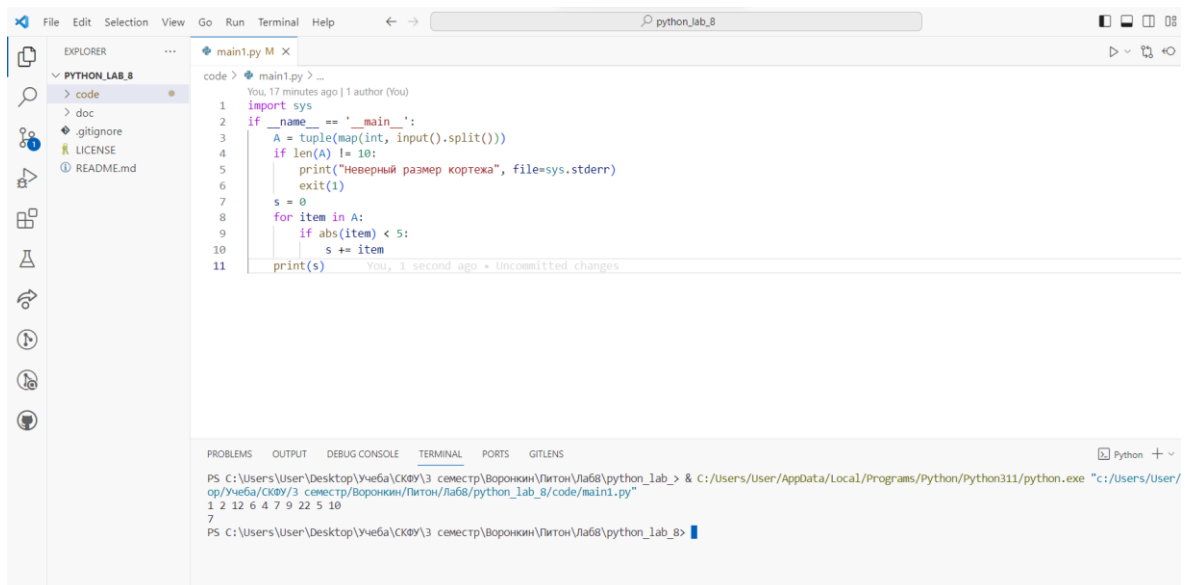
Цель: Приобретение навыков по работе с кортежами при написании программ с помощью языка программирования Python версии 3.x.

### Практическая часть:

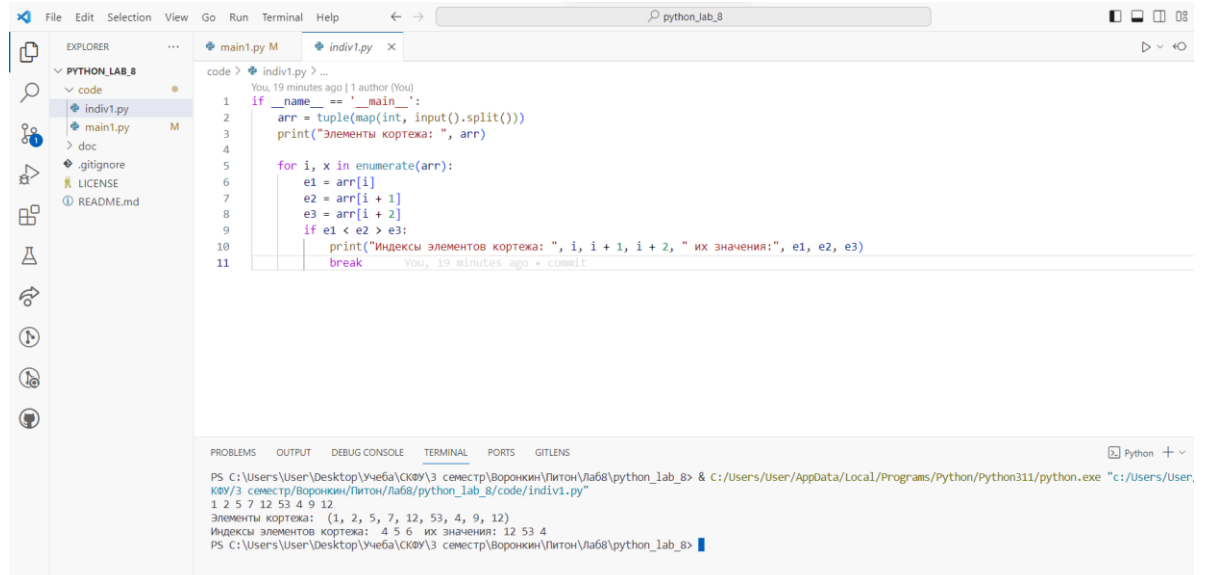
1. Изучил теоретический материал работы.
2. Создал общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и язык программирования Python.



3. Выполнил задание из первого примера: Ввести A из 10 элементов, найти сумму элементов, меньших по модулю 5, и вывести её на экран. Использовать в программе вместо списков кортежи.



4. Выполнил индивидуальное задание: Определить, есть ли в кортеже хотя бы одна тройка соседних чисел, в которой средний элемент больше своих «соседей», т. е. предшествующего и последующего. В случае положительного ответа определить номера элементов первой из таких троек.



The screenshot shows a VS Code editor with a file explorer on the left containing a project named 'PYTHON\_LAB\_8' with files 'main1.py' and 'indiv1.py'. The editor is open to 'indiv1.py', which contains the following Python code:

```
1 if __name__ == '__main__':
2     arr = tuple(map(int, input().split()))
3     print("Элементы кортежа: ", arr)
4
5     for i, x in enumerate(arr):
6         e1 = arr[i]
7         e2 = arr[i + 1]
8         e3 = arr[i + 2]
9         if e1 < e2 > e3:
10            print("Индексы элементов кортежа: ", i, i + 1, i + 2, " их значения: ", e1, e2, e3)
11            break
```

Below the code editor, the terminal output shows the execution of the script:

```
PS C:\Users\User\Desktop\Учеба\СкВУ\3 семестр\Воронкин\Питон\лаб8\python_lab_8> & C:\Users\User\AppData\Local\Programs\Python\Python311\python.exe "c:/Users/User/Desktop/Учеба/СкВУ/3 семестр/Воронкин/Питон/лаб8/python_lab_8/code/indiv1.py"
1 2 5 7 12 53 4 9 12
Элементы кортежа: (1, 2, 5, 7, 12, 53, 4, 9, 12)
Индексы элементов кортежа: 4 5 6 их значения: 12 53 4
PS C:\Users\User\Desktop\Учеба\СкВУ\3 семестр\Воронкин\Питон\лаб8\python_lab_8>
```

## Ответы на контрольные вопросы

### 1. Что такое кортежи 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. Если мы получили откуда-то массив данных, и у нас есть желание поработать с ним, но при этом непосредственно менять данные мы не собираемся, тогда, это как раз тот случай, когда кортежи придутся как нельзя кстати. Кортежи в памяти занимают меньший объем по сравнению со списками. Кортежи работают быстрее, чем списки.

### 3. Как осуществляется создание кортежей?

a = ()

b = tuple()

### 4. Как осуществляется доступ к элементам кортежа?

Доступ к элементам кортежа осуществляется также как к элементам списка – через указание индекса.

### **5. Зачем нужна распаковка (деструктуризация) кортежа?**

Обращение по индексу, это не самый удобный способ работы с кортежами. Дело в том, что кортежи часто содержат значения разных типов, и помнить, по какому индексу что лежит — очень непросто.

### **6. Какую роль играют кортежи в множественном присваивании?**

Используя множественное присваивание, можно проверить интересный трюк: обмен значениями между двумя переменными.

### **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.

### **8. Как выполняется конкатенация и повторение кортежей?**

Для кортежей можно выполнять операцию конкатенации, которая обозначается символом +.

$$T3 = T1 + T2$$

### **9. Как выполняется обход элементов кортежа?**

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

### **10. Как проверить принадлежность элемента кортежу?**

Проверка вхождения элемента в кортеж - оператор in.

### **11. Какие методы работы с кортежами Вам известны?**

index(), count().

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

Доступно.

**13. Как создать кортеж с помощью спискового включения.**

Также как и список.

**Вывод:** Приобрел навыки по работе с кортежами при написании программ с помощью языка программирования Python версии 3.x.