

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 №9**  
**дисциплины «Основы программной инженерии»**

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

---

(подпись)

Руководитель практики:  
Богданов С.С., ассистент кафедры  
инфокоммуникаций

---

(подпись)

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

Ставрополь, 2023 г.

**Тема:** Работа со словарями в языке Python.

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

## Порядок выполнения работы

1. Создал репозиторий GitHub.

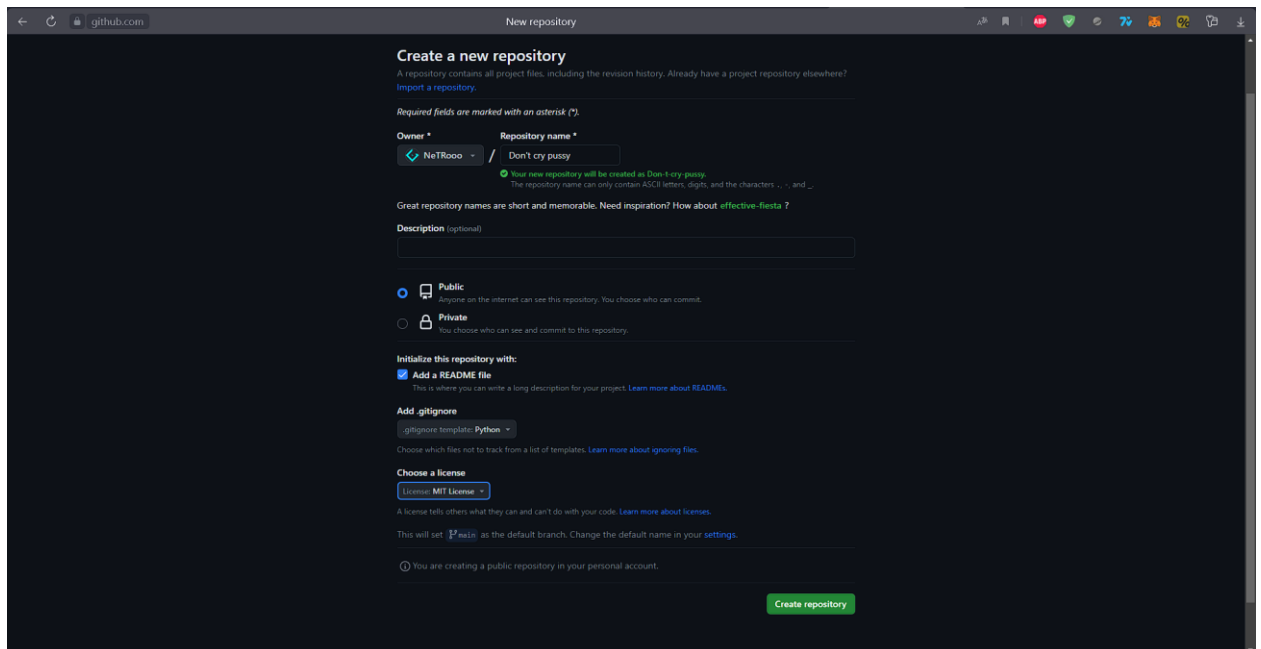


Рисунок 1- Создание репозитория

2. Проработал примеры из лабораторной работы.

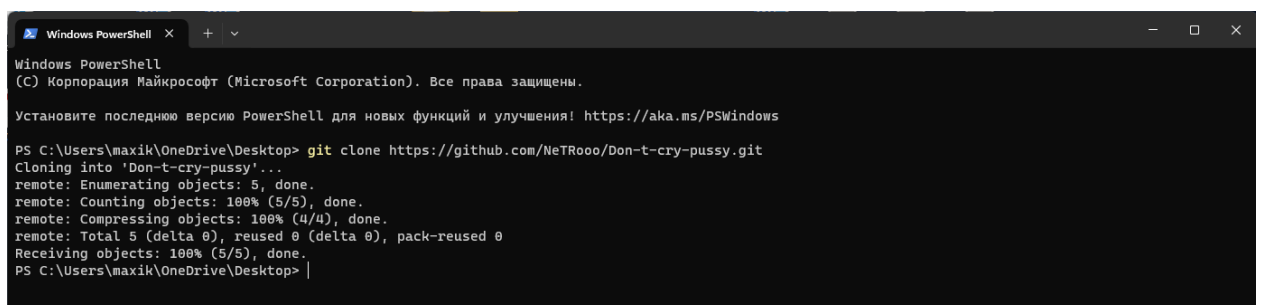


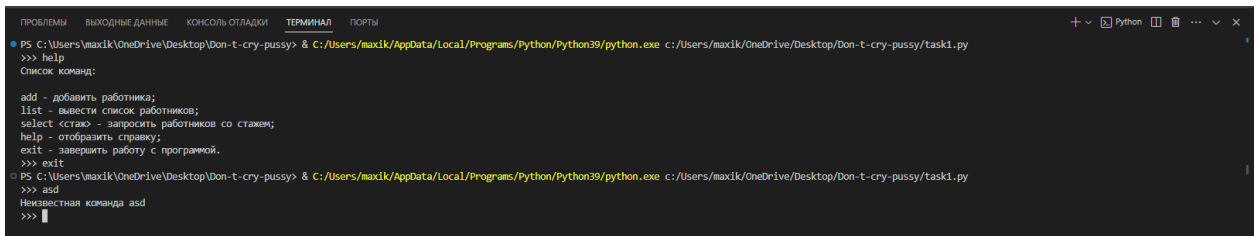
Рисунок 2 – Выполнил клонирование созданного репозитория

```
task1.py U X
task1.py
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3  import sys
4  from datetime import date
5  if __name__ == '__main__':
6      # Список работников.
7      workers = []
8      # Организовать бесконечный цикл запроса команд.
9      while True:
10         # Запросить команду из терминала.
11         command = input(">>> ").lower()
12         # Выполнить действие в соответствие с командой.
13         if command == 'exit':
14             break
15         elif command == 'add':
16             # Запросить данные о работнике.
17             name = input("Фамилия и инициалы? ")
18             post = input("Должность? ")
19             year = int(input("Год поступления? "))
20             # Создать словарь.
21             worker = {
22                 'name': name,
23                 'post': post,
24                 'year': year,
25             }
26             # Добавить словарь в список.
27             workers.append(worker)
28             # Отсортировать список в случае необходимости.
29             if len(workers) > 1:
30                 workers.sort(key=lambda item: item.get('name', ''))
31         elif command == 'list':
32             # Заголовок таблицы.
33             line = '+-{}-+-{}-+-{}-+-{}-+'.format(
34                 '-' * 4,
35                 '-' * 30,
36                 '-' * 20,
37                 '-' * 8
38             )
39             print(line)
40             print(
41                 '| {:^4} | {:^30} | {:^20} | {:^8} |'.format(
42                     "№",
43                     "Ф.И.О.",
44                     "Должность",
45                     "Год"
46                 )
47             )
```

Рисунок 3 – Код примера из лабораторной работы (задание №1) (1)

```
task1.py U X
task1.py
46         )
47     ]
48     print(line)
49     # Вывести данные о всех сотрудниках.
50     for idx, worker in enumerate(workers, 1):
51         print(
52             '| {:>4} | {:<30} | {:<20} | {:>8} |'.format(
53                 idx,
54                 worker.get('name', ''),
55                 worker.get('post', ''),
56                 worker.get('year', 0)
57             )
58         )
59     print(line)
60     elif command.startswith('select '):
61         # Получить текущую дату.
62         today = date.today()
63         # Разбить команду на части для выделения номера года.
64         parts = command.split(' ', maxsplit=1)
65         # Получить требуемый стаж.
66         period = int(parts[1])
67         # Инициализировать счетчик.
68         count = 0
69         # Проверить сведения работников из списка.
70         for worker in workers:
71             if today.year - worker.get('year', today.year) >= period:
72                 count += 1
73                 print(
74                     '{:>4}: {}'.format(count, worker.get('name', ''))
75                 )
76         # Если счетчик равен 0, то работники не найдены.
77         if count == 0:
78             print("Работники с заданным стажем не найдены.")
79     elif command == 'help':
80         # Вывести справку о работе с программой.
81         print("Список команд:\n")
82         print("add - добавить работника;")
83         print("list - вывести список работников;")
84         print("select <стаж> - запросить работников со стажем;")
85         print("help - отобразить справку;")
86         print("exit - завершить работу с программой.")
87     else:
88         print(f"Неизвестная команда {command}", file=sys.stderr)
```

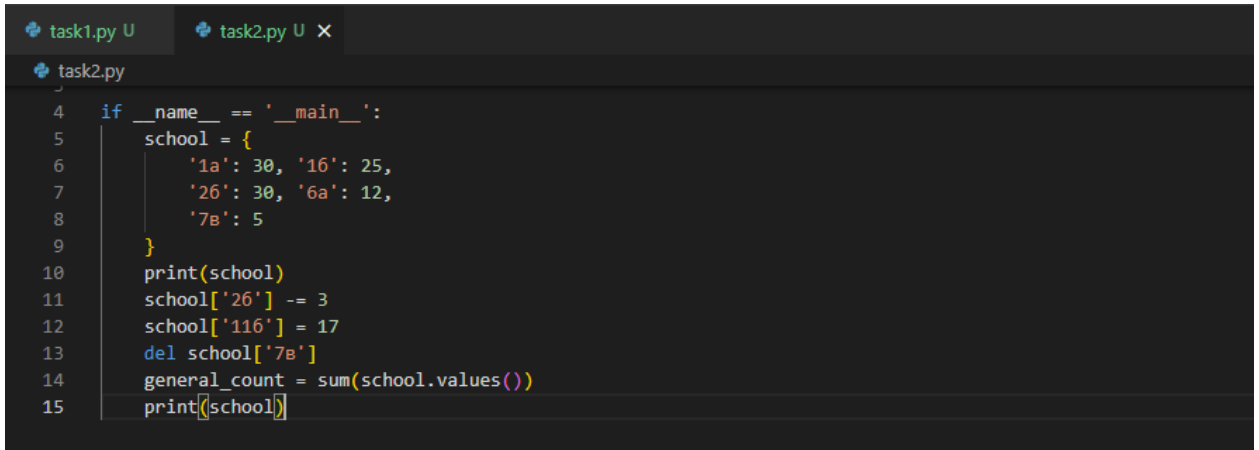
Рисунок 4 – Код примера из лабораторной работы (задание №1) (2)



```
ПРОБЛЕМЫ Выходные данные КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ
PS C:\Users\maxik\OneDrive\Desktop\Don-t-cry-pussy> & C:/Users/maxik/AppData/Local/Programs/Python/Python39/python.exe c:/Users/maxik/OneDrive/Desktop/Don-t-cry-pussy/task1.py
>>> help
Список команд:

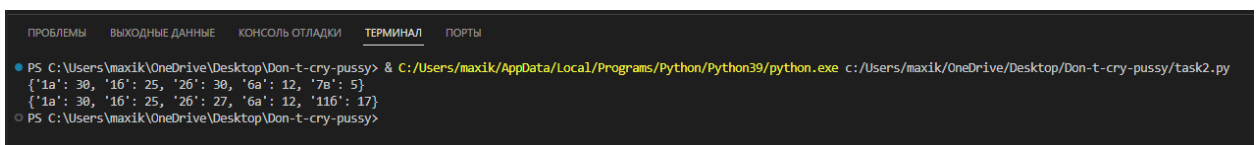
add - добавить работника;
list - вывести список работников;
select <стаж> - запросить работников со стажем;
help - отобразить справку;
exit - завершить работу с программой.
>>> exit
PS C:\Users\maxik\OneDrive\Desktop\Don-t-cry-pussy> & C:/Users/maxik/AppData/Local/Programs/Python/Python39/python.exe c:/Users/maxik/OneDrive/Desktop/Don-t-cry-pussy/task1.py
>>> add
Неизвестная команда add
>>>
```

Рисунок 5 – Результаты выполнения приложения.



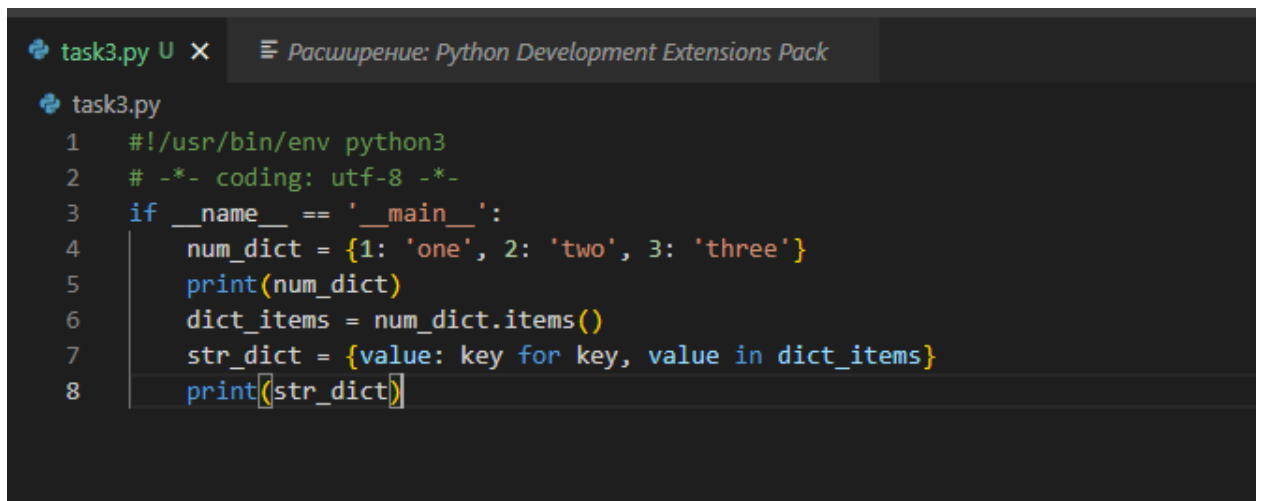
```
task1.py U task2.py U X
task2.py
4 if __name__ == '__main__':
5     school = {
6         '1a': 30, '16': 25,
7         '26': 30, '6a': 12,
8         '7в': 5
9     }
10    print(school)
11    school['26'] -= 3
12    school['116'] = 17
13    del school['7в']
14    general_count = sum(school.values())
15    print(school)
```

Рисунок 6 – Решить задачу: создайте словарь, связав его с переменной school, и наполните данными, которые бы отражали количество учащихся в разных классах (1а, 1б, 2б, 6а, 7в и т.п.). Внесите изменения в словарь согласно следующему: а) в одном из классов изменилось количество учащихся, б) в школе появился новый класс, с) в школе был расформирован (удалён) другой класс. Вычислите общее количество учащихся в школе. Зафиксировать изменения в репозитории. (задание №2)



```
ПРОБЛЕМЫ Выходные данные КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ
PS C:\Users\maxik\OneDrive\Desktop\Don-t-cry-pussy> & C:/Users/maxik/AppData/Local/Programs/Python/Python39/python.exe c:/Users/maxik/OneDrive/Desktop/Don-t-cry-pussy/task2.py
{'1a': 30, '16': 25, '26': 30, '6a': 12, '7в': 5}
{'1a': 30, '16': 25, '26': 27, '6a': 12, '116': 17}
PS C:\Users\maxik\OneDrive\Desktop\Don-t-cry-pussy>
```

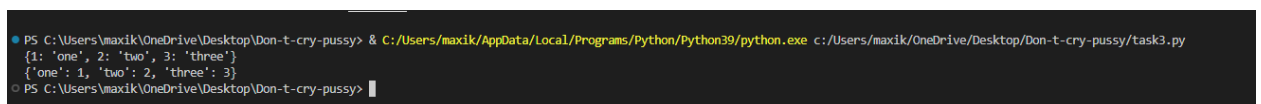
Рисунок 7 – Результат выполнения кода.



```
task3.py U x  Расширение: Python Development Extensions Pack

task3.py
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3  if __name__ == '__main__':
4      num_dict = {1: 'one', 2: 'two', 3: 'three'}
5      print(num_dict)
6      dict_items = num_dict.items()
7      str_dict = {value: key for key, value in dict_items}
8      print(str_dict)
```

Рисунок 8 – Решить задачу: создайте словарь, где ключами являются числа, а значениями – строки. Примените к нему метод `items()`, с помощью полученного объекта `dict_items` создайте новый словарь, «обратный» исходному, т.е. ключами являются строки, а значениями – числа. (задание №3)



```
PS C:\Users\maxik\OneDrive\Desktop\Don-t-cry-pussy> & C:/Users/maxik/AppData/Local/Programs/Python/Python39/python.exe c:/Users/maxik/OneDrive/Desktop/Don-t-cry-pussy/task3.py
{1: 'one', 2: 'two', 3: 'three'}
{'one': 1, 'two': 2, 'three': 3}
PS C:\Users\maxik\OneDrive\Desktop\Don-t-cry-pussy>
```

Рисунок 9 – Результат выполнения кода.

```

task3.py U  ind1.py U X
ind1.py
1  #!/usr/bin/env python3
2  #-*- coding: utf-8 -*-
3  def add_record(phonebook, last_name, first_name, phone_number, birth_date):
4      record = {
5          'фамилия': last_name,
6          'имя': first_name,
7          'номер телефона': phone_number,
8          'дата рождения': birth_date
9      }
10     phonebook.append(record)
11     phonebook.sort(key=lambda x: x['номер телефона'][:3]) # Сортировка по первым трем цифрам номера телефона
12
13 def display_record(phonebook, last_name):
14     for record in phonebook:
15         if record['фамилия'] == last_name:
16             print(f"Фамилия: {record['фамилия']}")
17             print(f"Имя: {record['имя']}")
18             print(f"Номер телефона: {record['номер телефона']}")
19             print(f"Дата рождения: {'.'.join(map(str, record['дата рождения']))}")
20             return
21     print(f"Запись с фамилией '{last_name}' не найдена.")
22
23 def main():
24     phonebook = []
25
26     while True:
27         print("\n1. Добавить запись")
28         print("2. Найти запись по фамилии")
29         print("3. Выйти")
30
31         choice = input("Выберите действие (1/2/3): ")
32
33         if choice == '1':
34             last_name = input("Введите фамилию: ")
35             first_name = input("Введите имя: ")
36             phone_number = input("Введите номер телефона: ")
37             birth_date = list(map(int, input("Введите дату рождения через пробел (дд мм гggg): ").split()))
38
39             add_record(phonebook, last_name, first_name, phone_number, birth_date)
40             print("Запись добавлена успешно.")
41
42         elif choice == '2':
43             last_name = input("Введите фамилию для поиска: ")
44             display_record(phonebook, last_name)

```

Рисунок 10 – Решить задачу: Использовать словарь, содержащий следующие ключи: фамилия, имя; номер телефона; дата рождения (список из трех чисел).

Написать программу, выполняющую следующие действия: ввод с клавиатуры данных в список, состоящий из словарей заданной структуры; записи должны быть упорядочены по трем первым цифрам номера телефона; вывод на экран информации о человеке, чья фамилия введена с клавиатуры; если такого нет, выдать на дисплей соответствующее сообщение.

```
ПРОБЛЕМЫ Выходные данные КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ
PS C:\Users\maxik\OneDrive\Desktop\Don-t-cry-pussy> & C:\Users\maxik\AppData\Local\Programs\Python\Python39\python.exe c:\Users\maxik\OneDrive\Desktop\Don-t-cry-pussy\ind1.py

1. Добавить запись
2. Найти запись по фамилии
3. Выйти
Выберите действие (1/2/3): 1
Введите фамилию: Яблоновский
Введите имя: Дмитрий
Введите номер телефона: +79888406997
Введите дату рождения через пробел (дд мм гггг): 07 10 2004
Запись добавлена успешно.

1. Добавить запись
2. Найти запись по фамилии
3. Выйти
Выберите действие (1/2/3): 1
Введите фамилию: Шехов
Введите имя: Владислав
Введите номер телефона: Федорович
Введите дату рождения через пробел (дд мм гггг): 22 10 2004
Запись добавлена успешно.

1. Добавить запись
2. Найти запись по фамилии
3. Выйти
Выберите действие (1/2/3): 2
Введите фамилию для поиска: Фва
Запись с фамилией "Фва" не найдена.

1. Добавить запись
2. Найти запись по фамилии
3. Выйти
Выберите действие (1/2/3): 2
Введите фамилию для поиска: Яблоновский
Фамилия: Яблоновский
Имя: Дмитрий
Номер телефона: +79888406997
Дата рождения: 7.10.2004

1. Добавить запись
2. Найти запись по фамилии
3. Выйти
Выберите действие (1/2/3): 5
Некорректный ввод. Пожалуйста, выберите 1, 2 или 3.

1. Добавить запись
2. Найти запись по фамилии
3. Выйти
Выберите действие (1/2/3): 3
PS C:\Users\maxik\OneDrive\Desktop\Don-t-cry-pussy>
```

Рисунок 11 – Результат выполнения кода.

```
ПРОБЛЕМЫ (6) Выходные данные КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ GITLENS
PS C:\Users\maxik\OneDrive\Desktop\Don-t-cry-pussy> git add .
PS C:\Users\maxik\OneDrive\Desktop\Don-t-cry-pussy> git commit -m "last commit"
[dev 4be373f] last commit
4 files changed, 164 insertions(+)
create mode 100644 ind1.py
create mode 100644 task1.py
create mode 100644 task2.py
create mode 100644 task3.py
PS C:\Users\maxik\OneDrive\Desktop\Don-t-cry-pussy> git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
PS C:\Users\maxik\OneDrive\Desktop\Don-t-cry-pussy> git merge dev
Updating 3b37a81..4be373f
Fast-forward
 ind1.py | 53 ++++++
 task1.py | 88 ++++++
 task2.py | 15 ++++++
 task3.py | 8 +++++
4 files changed, 164 insertions(+)
create mode 100644 ind1.py
create mode 100644 task1.py
create mode 100644 task2.py
create mode 100644 task3.py
PS C:\Users\maxik\OneDrive\Desktop\Don-t-cry-pussy> git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 2.92 KiB | 2.92 MiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/NeTRooo/Don-t-cry-pussy.git
 3b37a81..4be373f  main -> main
PS C:\Users\maxik\OneDrive\Desktop\Don-t-cry-pussy> git checkout dev
Switched to branch 'dev'
Your branch is ahead of 'origin/dev' by 1 commit.
(use "git push" to publish your local commits)
PS C:\Users\maxik\OneDrive\Desktop\Don-t-cry-pussy> git push
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/NeTRooo/Don-t-cry-pussy.git
 3b37a81..4be373f  dev -> dev
PS C:\Users\maxik\OneDrive\Desktop\Don-t-cry-pussy>
```

Рисунок 12 – Создание и отправка коммита

Ссылка на репозиторий: <https://github.com/NeTRooo/Don-t-cry-pussy>



## Ответы на контрольные вопросы

1. Что такое списки в языке Python?

Списки в языке Python - это упорядоченные изменяемые коллекции элементов.

2. Каково назначение кортежей в языке Python?

Кортежи в языке Python используются для создания неизменяемых упорядоченных коллекций элементов.

3. Как осуществляется создание кортежей?

Кортежи создаются с использованием круглых скобок, например, `my_tuple = (1, 2, 3)`.

4. Как осуществляется доступ к элементам кортежа?

Доступ к элементам кортежа осуществляется по индексу, например, `element = my_tuple[0]`.

5. Зачем нужна распаковка (деструктуризация) кортежа?

Распаковка кортежа (деструктуризация) позволяет присваивать значения элементам кортежа одной строкой.

6. Какую роль играют кортежи в множественном присваивании?

Кортежи играют ключевую роль в множественном присваивании, где значения присваиваются сразу нескольким переменным.

7. Как выбрать элементы кортежа с помощью среза?

Выбор элементов кортежа с помощью среза осуществляется, например, `subset = my_tuple[1:3]`.

8. Как выполняется конкатенация и повторение кортежей?

Конкатенация кортежей выполняется оператором `+`, а повторение - оператором `*`.

9. Как выполняется обход элементов кортежа?

Обход элементов кортежа осуществляется с использованием цикла, например, `for item in my_tuple:`.

10. Как проверить принадлежность элемента кортежу.

Принадлежность элемента кортежу можно проверить с использованием оператора `in`

11. Какие методы работы с кортежами Вам известны?

Некоторые методы работы с кортежами включают `count()` для подсчета элементов и `index()` для поиска индекса элемента.

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

Да, функции агрегации, такие как `len()`, `sum()`, могут использоваться с кортежами.

13. Как создать кортеж с помощью спискового включения?

Кортеж можно создать с помощью спискового включения, например, `my_tuple = tuple(x for x in my_list)`.