

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 РАБОТЕ №2.17**  
**дисциплины «Основы кроссплатформенного программирования»**

Выполнил:

Арзютов Иван Владиславович

2 курс, группа ИТС-б-о-22-1,

11.03.02 «Инфокоммуникационные  
технологии и системы связи»,  
направленность (профиль)  
«Инфокоммуникационные системы  
и

сети», очная форма обучения

---

(подпись)

Проверил:

Воронкин Р.А., канд. тех. наук, доцент,  
доцент кафедры инфокоммуникаций

---

(подпись)

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

Ставрополь, 2023 г

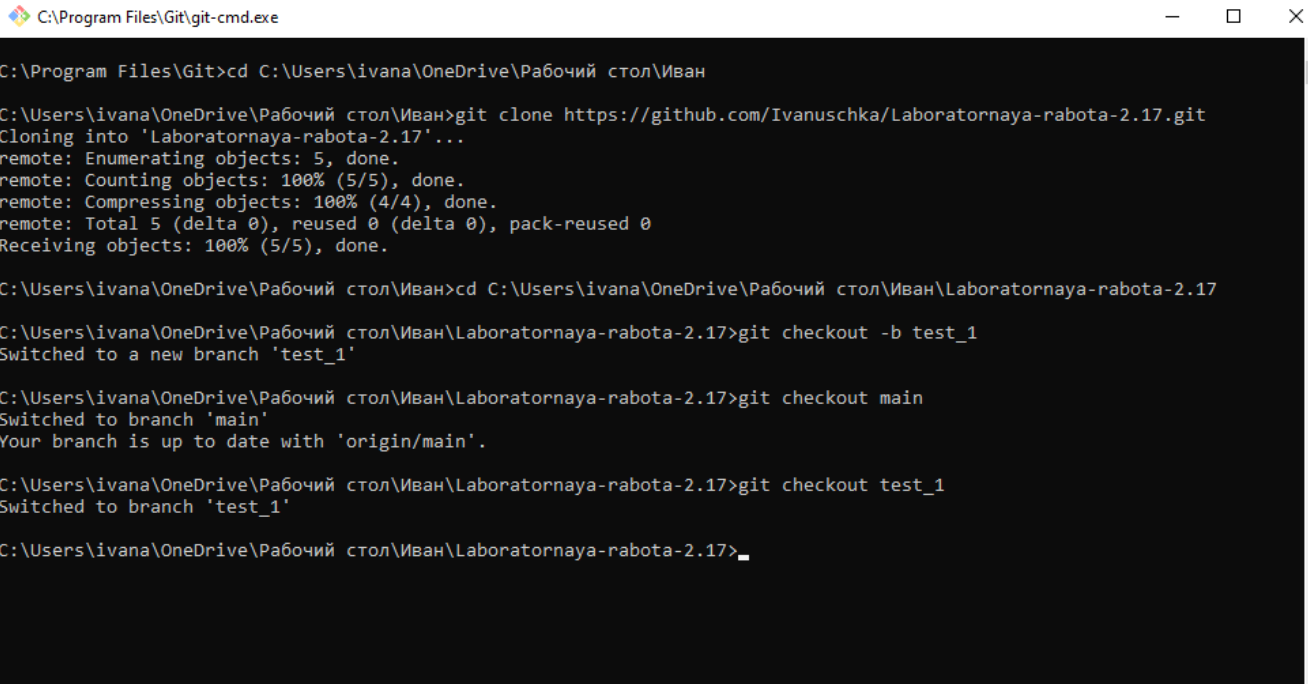
Тема: работа с данными формата JSON в языке Python.

Цель работы: приобретение навыков по работе с данными формата JSON с помощью языка программирования Python версии 3.x.

### Порядок выполнения работы:

Задание 1. Изучил теоретический материал работы, создал общедоступный репозиторий на GitHub, в котором использована лицензий MIT и язык программирования Python, также добавил файл .gitignore с необходимыми правилами.

Клонировал репозиторий на свой компьютер. В ходе данной лабораторной работы работал с моделью ветвления git-flow.



```
C:\Program Files\Git\git-cmd.exe
C:\Program Files\Git>cd C:\Users\ivana\OneDrive\Рабочий стол\Иван
C:\Users\ivana\OneDrive\Рабочий стол\Иван>git clone https://github.com/Ivanuschka/Laboratornaya-rabota-2.17.git
Cloning into 'Laboratornaya-rabota-2.17'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.
C:\Users\ivana\OneDrive\Рабочий стол\Иван>cd C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornaya-rabota-2.17
C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornaya-rabota-2.17>git checkout -b test_1
Switched to a new branch 'test_1'
C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornaya-rabota-2.17>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornaya-rabota-2.17>git checkout test_1
Switched to branch 'test_1'
C:\Users\ivana\OneDrive\Рабочий стол\Иван\Laboratornaya-rabota-2.17>_
```

Рисунок 1. Клонирование и модель ветвления git-flow

## Работа с примером №1.

Для примера 1 лабораторной работы 2.16 разработайте интерфейс командной строки.

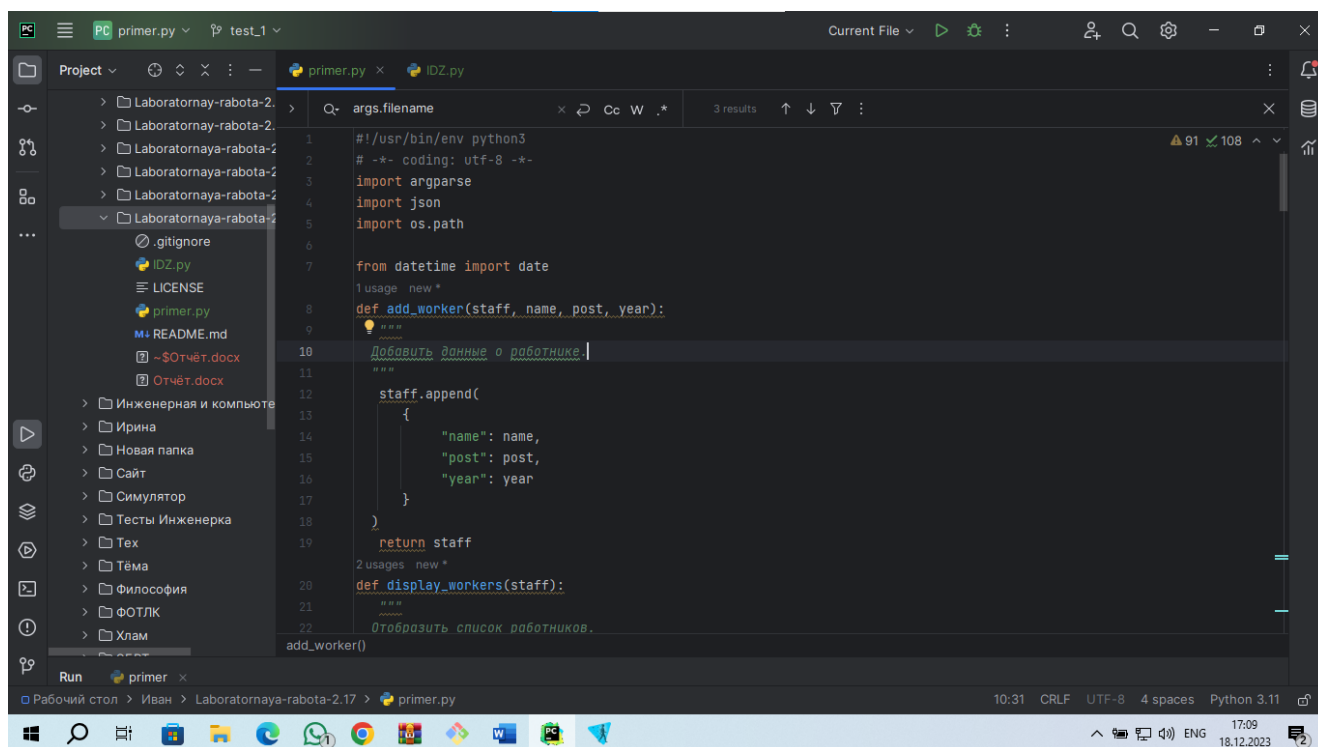


Рисунок 2. Работа с примером №1.

Выполнение индивидуального задания.

Для своего варианта лабораторной работы 2.16 необходимо дополнительно реализовать интерфейс командной строки (CLI).

Условие задания: использовать словарь, содержащий следующие ключи: фамилия и инициалы; номер группы; успеваемость (список из пяти элементов). Написать программу, выполняющую следующие действия: ввод с клавиатуры данных в список, состоящий из словарей заданной структуры; записи должны быть упорядочены по возрастанию номера группы; вывод на дисплей фамилий и номеров групп для всех студентов, включенных в массив, если средний балл студента больше 4.0; если таких студентов нет, вывести соответствующее сообщение.

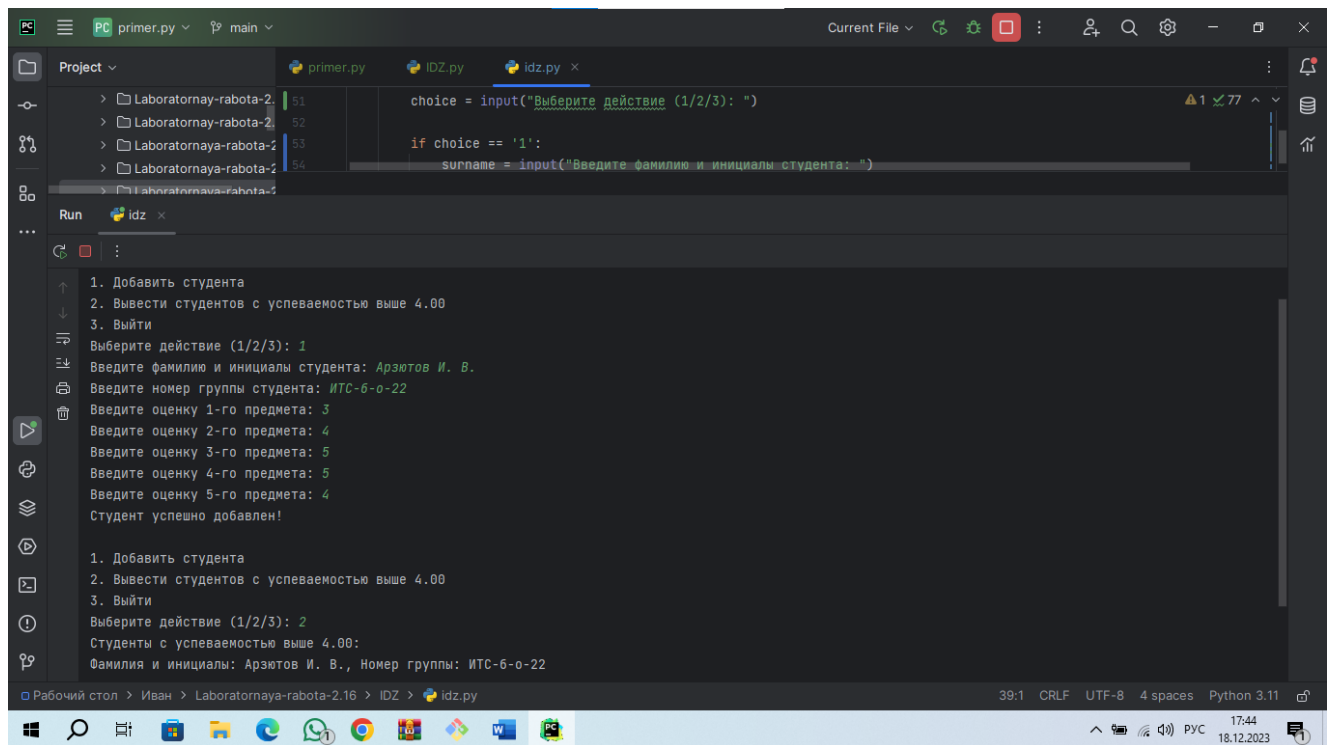


Рисунок 3. Выполнение индивидуального задания

Слил ветку tets\_1 с веткой main и отправил на удаленный сервер

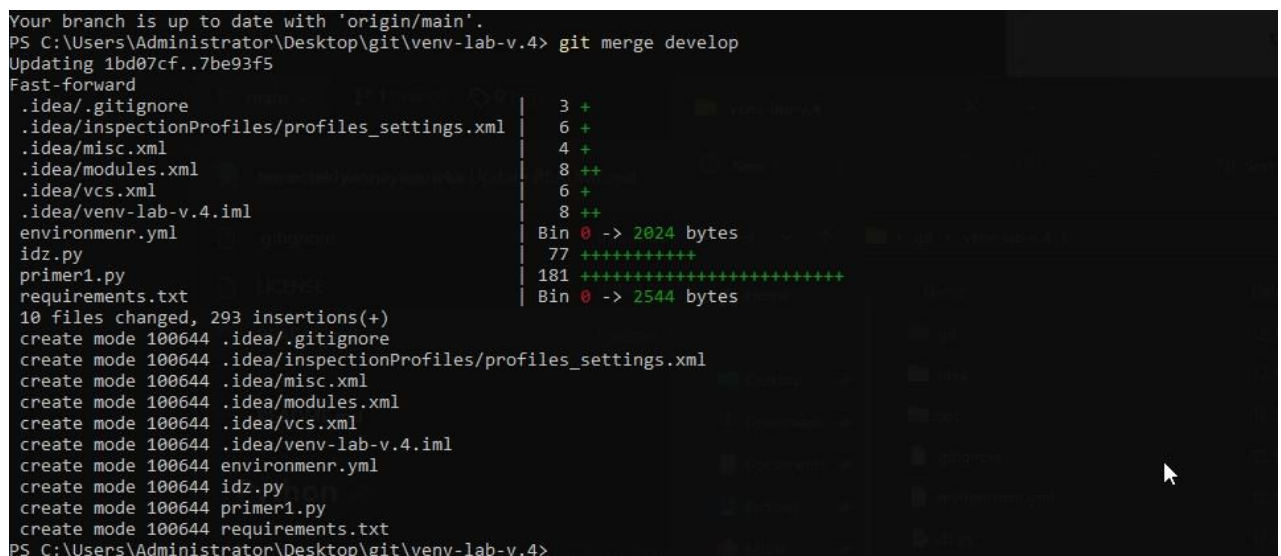


Рисунок 4. Слил ветки

## Ответы на контрольные вопросы

1. В чем отличие терминала и консоли?

Терминал (Terminal):

В общем смысле, терминал - это программа, предоставляющая текстовый интерфейс для взаимодействия с операционной системой. Это может быть командная строка в графической среде, также известная как терминал в Unix-подобных системах.

В контексте Python, "терминал" может означать окно командной строки, в котором вы запускаете скрипты Python.

Консоль (Console):

В Python термин "консоль" часто используется для обозначения интерактивной оболочки Python (REPL - Read-Eval-Print Loop), где вы можете вводить команды Python непосредственно и видеть результаты выполнения.

В графической среде Windows, "консоль" может также относиться к окну командной строки (Command Prompt) или PowerShell.

## 2. Что такое консольное приложение?

Консольное приложение (или текстовое приложение) — это приложение, взаимодействие с пользователем которого осуществляется через текстовый интерфейс командной строки. В отличие от графических приложений, консольные приложения не используют графический пользовательский интерфейс (GUI) и обычно работают в текстовом режиме.

## 3. Какие существуют средства языка программирования Python для построения приложений командной строки?

Модуль `argparse`: Этот модуль предоставляет инструменты для анализа аргументов командной строки. Он позволяет определять, какие аргументы ожидаются при запуске программы и как они должны обрабатываться.

Модуль `sys`: Модуль `sys` предоставляет доступ к некоторым переменным и функциям, связанным с интерпретатором Python. Например, `sys.argv` содержит список аргументов командной строки, переданных скрипту.

Модуль `click`: Это библиотека, которая облегчает создание красивых и удобных в использовании интерфейсов командной строки.

Модуль `subprocess`: Этот модуль позволяет запускать другие программы из Python и взаимодействовать с ними. Он может быть полезен для выполнения внешних команд из командной строки.

## 4. Какие особенности построение CLI с использованием модуля `sys` ?

Построение интерфейса командной строки (CLI) с использованием модуля `sys` включает в себя работу с аргументами командной строки, переданными скрипту. Основные шаги включают в себя:

Импорт модуля `sys`:

Обработка аргументов: Аргументы командной строки доступны в списке `sys.argv`. Этот список содержит имя скрипта (индекс 0) и все переданные аргументы.

Обработка флагов и значений: Модуль `sys` не предоставляет специальных инструментов для обработки флагов и значений. Запуск из командной строки: Ваш скрипт может быть запущен из командной строки с аргумент.

#### 5. Какие особенности построение CLI с использованием модуля `getopt` ?

Модуль `getopt` - это более старый и менее удобный способ обработки аргументов командной строки по сравнению с более современным модулем `argparse`. Однако, если вам нужна простая и легкая в использовании альтернатива, `getopt` может быть полезным.

1. Импорт модуля `getopt`
2. Определение параметров командной строки
3. Обработка ошибок
4. Запуск из командной строки

#### 6. Какие особенности построение CLI с использованием модуля `argparse`

Модуль `argparse` предоставляет более мощные и удобные средства для построения интерфейса командной строки (CLI) по сравнению с `getopt` или простым использованием `sys.argv`. Вот основные особенности построения CLI с использованием модуля `argparse`:

1. Импорт модуля `argparse`
2. Создание объекта парсера
3. Добавление аргументов
4. Парсинг аргументов
5. Использование аргументов
6. Запуск из командной строки
7. Поддержка справки и документации

**Вывод:** приобрел навыки построения приложений с интерфейсом командной строки с помощью языка программирования Python версии 3.x.



