МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

Основы кроссплатформенного программирования

Отчет по лабораторной работе №2.17

Тема: «Разработка приложений с интерфейсом командной строки (CLI) в Python3»

Выполнил студент группы

ИВТ-б-о-21-1

Назаров Н.Ю. « »_____20__г.

Подпись студента_____

Работа защищена « »_____20__г.

Проверил доцент
Кафедры инфокоммуникаций, старший преподаватель
Воронкин Р.А.

_____
(подпись)

Ставрополь 2022

Цель работы: приобретение построения приложений с интерфейсом командной строки с помощью языка программирования Python версии 3.x.

Ход работы:

```
c:\Users\AdMin\Desktop\2.17>cd /d c:\users\admin\desktop

c:\Users\AdMin\Desktop>git clone https://github.com/NikitaNazarov179/2.17.git
Cloning into '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.
```

Рисунок 1 – клонирование репозитория

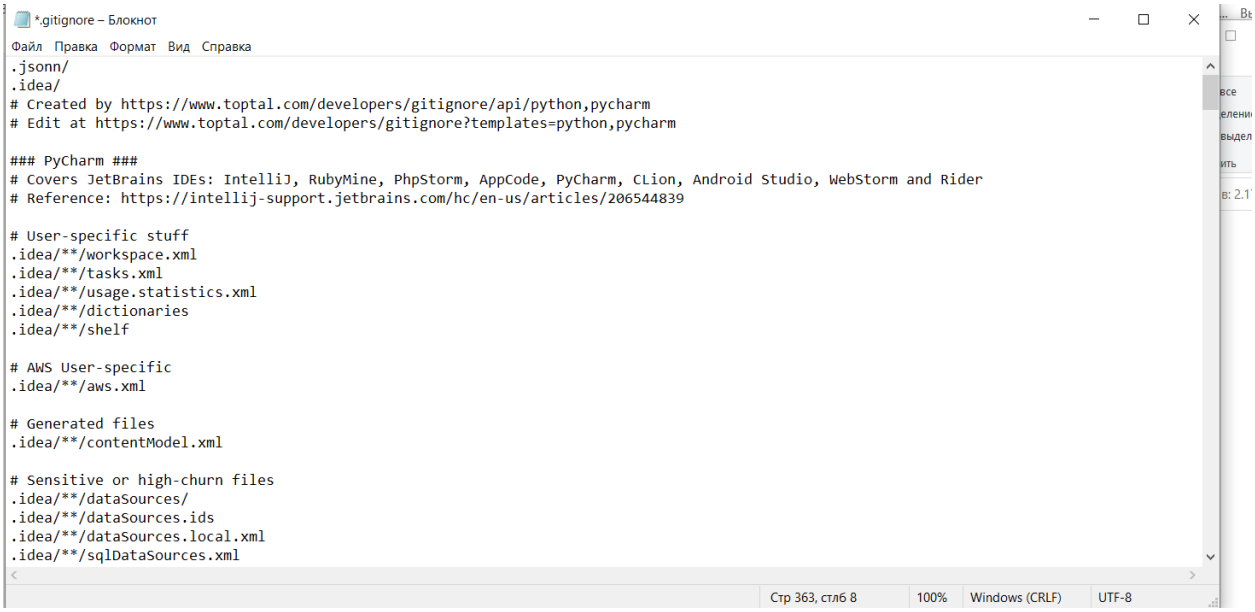

Рисунок 2 – отредактировал gitignore

```
c:\Users\AdMin\Desktop\2.17>git flow init

Which branch should be used for bringing forth production releases?
   - main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [C:/Users/AdMin/Desktop/2.17/.git/hooks]
```

Рисунок 3 – организовал репозиторий в соответствии с git flow

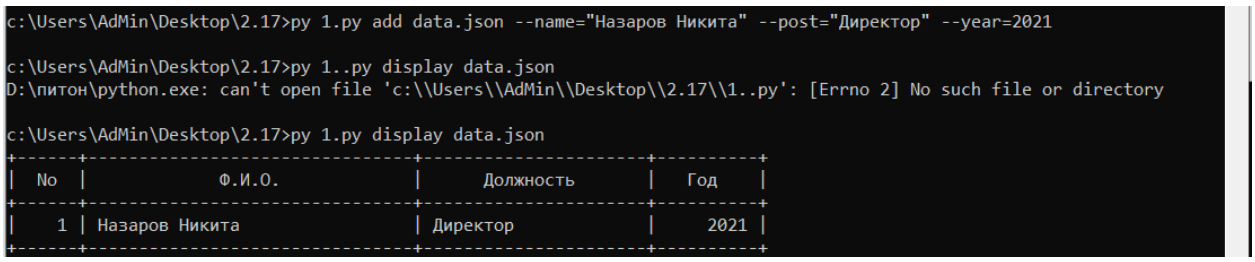

Рисунок 4 – проработал 1 пример

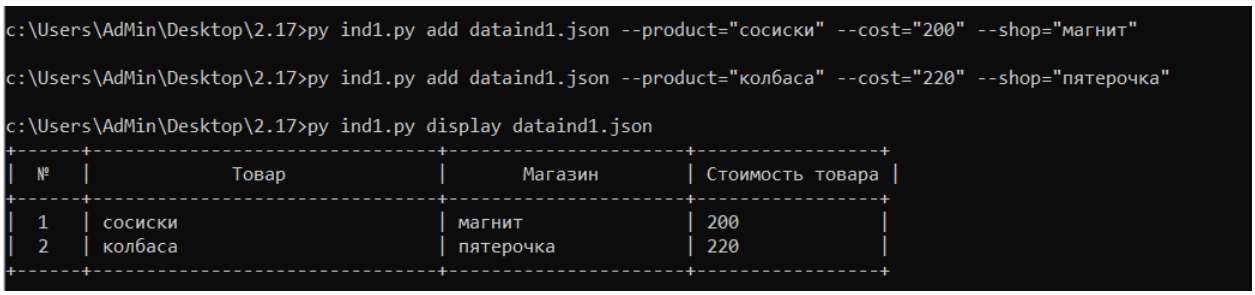

Рисунок 5 – выполнил 1 индивидуальное задание

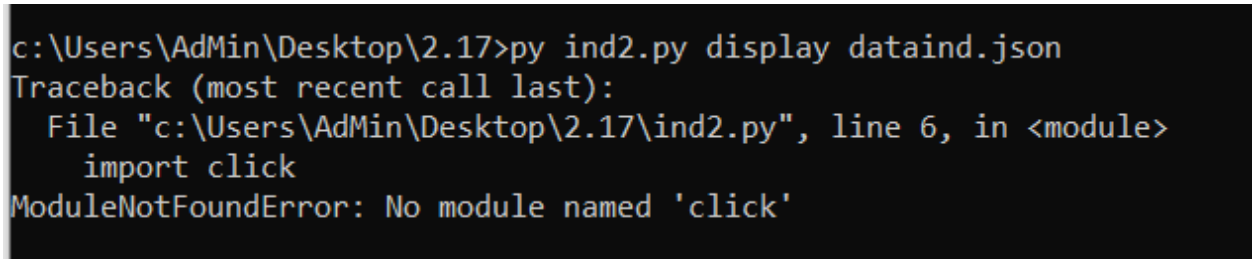

Рисунок 6 – выполнил задание повышенной сложности

Ответы на контрольные вопросы:

1. В чем отличие терминала и консоли?

Терминал (от лат. terminus — граница) — устройство или ПО, выступа- ющее посредником между человеком и вычислительной системой.

Обычно данный термин используется, когда точка доступа к системе вы- несена в отдельное физическое устройство и предоставляет свой пользова- тельский интерфейс на основе внутреннего интерфейса (например, сетевых протоколов).

Консоль console — исторически реализация терминала с клавиатурой и текстовым дисплеем. В настоящее время это слово часто используется

как си-ноним сеанса работы или окна оболочки командной строки. В том же смыслеиногда применяется и слово “терминал”.

2. Что такое консольное приложение?

Консольное приложение console application — вид ПО, разработанный срасчётом на работу внутри оболочки командной строки, т.е. опирающийся натекстовый ввод-вывод.

3. Какие существуют средства языка программирования Python дляпостроения приложений командной строки?

Python 3 поддерживает несколько различных способов обработки аргу- ментов командной строки.

Встроенный способ – использовать модуль sys. С точки зрения имен и использования, он имеет прямое отношение к библиотеке C (libc). Второй спо-соб – это модуль getopt, который обрабатывает как короткие, так и длинные параметры, включая оценку значений параметров.

4. Какие особенности построение CLI с использованием модуля sys?

Это базовый модуль, который с самого начала поставлялся с Python. Онespecиспользует подход, очень похожий на библиотеку C, с использованием argc иargv для доступа к аргументам.

Модуль sys реализует аргументы командной строки в простой структуресписка с именем sys.argv

5. Какие особенности построение CLI с использованием модуля getopt?

Как вы могли заметить ранее, модуль sys разбивает строку командной строки только на отдельные фасеты. Модуль getopt в Python идет немного дальше и расширяет разделение входной строки проверкой параметров.

Основанный на функции C getopt, он позволяет использовать как

корот-кие, так и длинные варианты, включая присвоение значений.

6. Какие особенности построение CLI с использованием модуля argparse?

Начиная с версий Python 2.7 и Python 3.2, в набор стандартных библио- тек была включена библиотека argparse для обработки аргументов (парамет- ров, ключей) командной строки.

Для начала рассмотрим, что интересного предлагает argparse:

- анализ аргументов sys.argv;
- конвертирование строковых аргументов в объекты вашей программы и работа с ними;
- форматирование и вывод информативных подсказок.