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 №2.15

Тема: «Работа с файлами в языке Python»

Выполнил студент группы

ИВТ-б-о-21-1

Артемьев А.В. « »_____20__г.

Подпись студента_____

Работа защищена « »_____20__г.

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

_____
(подпись)

Ставрополь 2022

Цель работы: приобретение навыков по работе с текстовыми файлами при написании программ с помощью языка программирования Python версии 3.x, изучение основных методов модуля os для работы с файловой системой, получение аргументов командной строки.

Ход работы:

1. Создал репозиторий в GitHub, дополнил правила в .gitignore для работы с IDE PyCharm с ЯП Python, выбрал лицензию MIT, клонировал его на компьютер и организовал в соответствии с моделью ветвления git-flow.

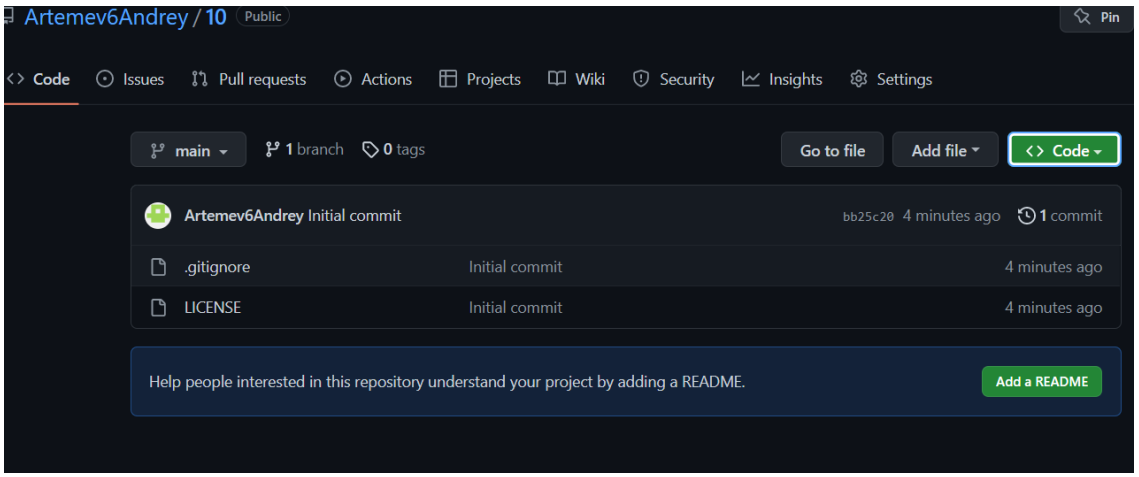

Рисунок 1.1 – Созданный репозиторий

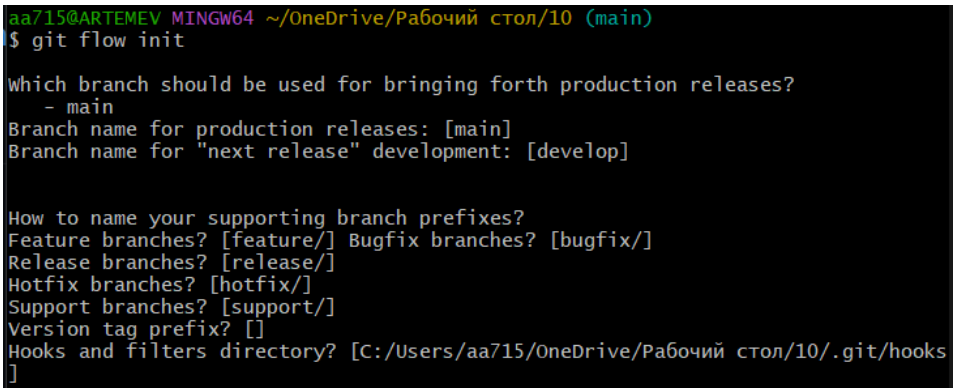

Рисунок 1.2 – Организация репозитория в соответствии с моделью ветвления git-flow

2. Создал проект Pycharm в папке репозитория, проработал примеры ЛР.

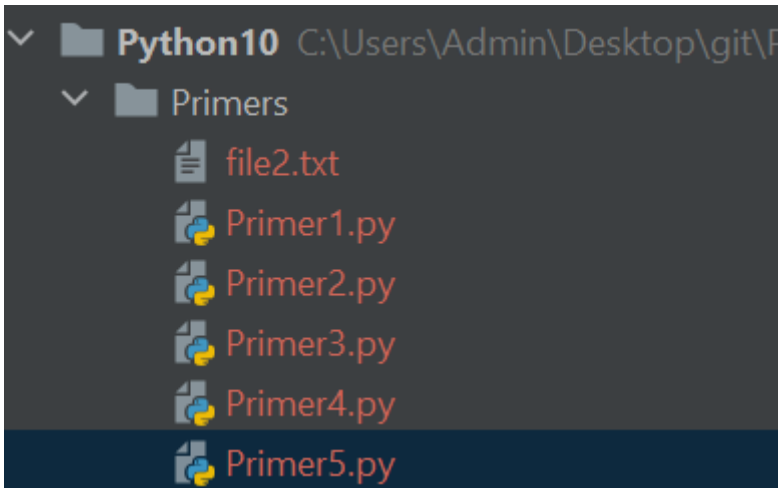

Рисунок 2 – Созданный проект

3. Индивидуальные задания.

Индивидуальное задание №1. В – 1.

Написать программу, которая считывает из текстового файла три предложения и выводит их в обратном порядке.

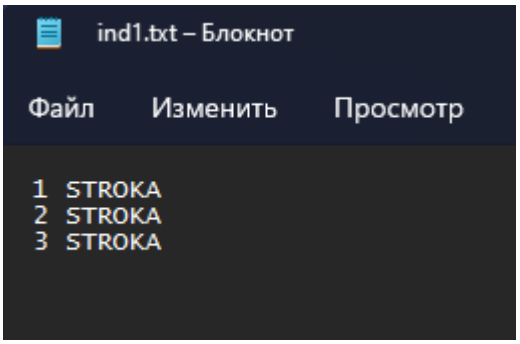

Рисунок 3 – Текст в файле

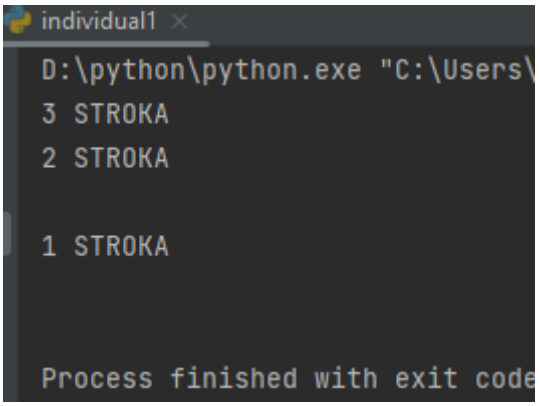

Рисунок 4 – Результат работы программы

Индивидуальное задание №2. В – 1.

В операционных системах на базе Unix обычно присутствует утилита с названием head. Она выводит первые десять строк содержимого файла, имя которого передается в качестве аргумента командной строки. Напишите программу на Python, имитирующую поведение этой утилиты. Если файла, указанного пользователем, не существует, или не задан аргумент командной строки, необходимо вывести соответствующее сообщение об ошибке.

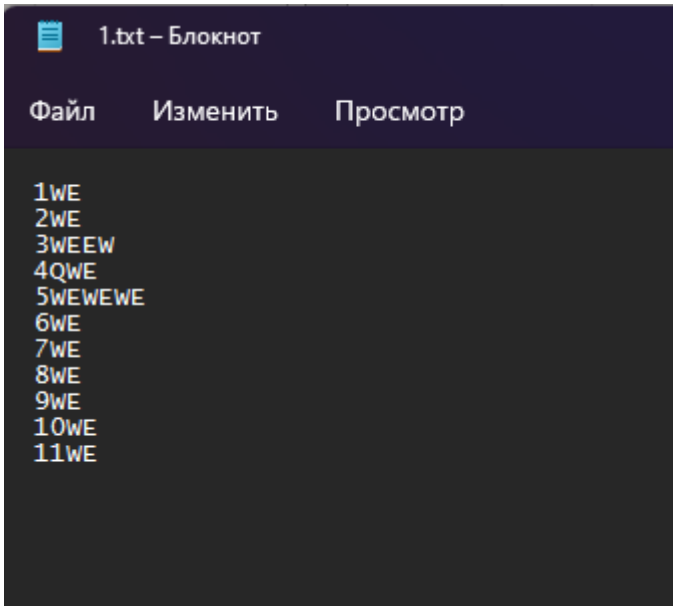

Рисунок 5 – Содержимое текстового файла для задания

```
C:\Users\aa715\OneDrive\Рабочий стол\10\ind>python individual2.py 1.txt
1WE
2WE
3WEEW
4QWE
5WEWEWE
6WE
7WE
8WE
9WE
10WE
```

Рисунок 6 – Все возможные варианты вывода индивидуального задания №2

Индивидуальное задание №3. Самостоятельно подберите или придумайте задачу для работы с изученными функциями модуля os. Приведите решение этой задачи.

Задача: создать новый текстовый файл, затем требуется узнать число ядер процессора, а потом записать их в созданный файл, изменить имя файла на myproccesor.txt если его еще не существует, если он существует, то выдать соответствующее сообщение.

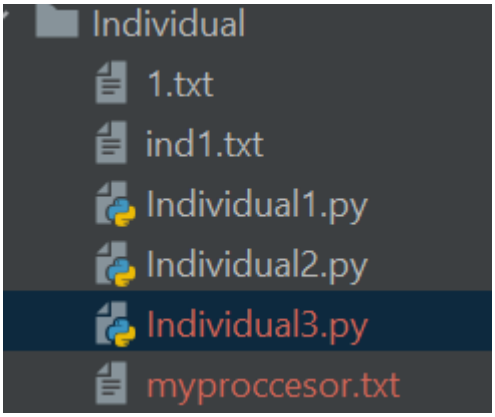

Рисунок 7 – Созданный файл в папке с заданиями

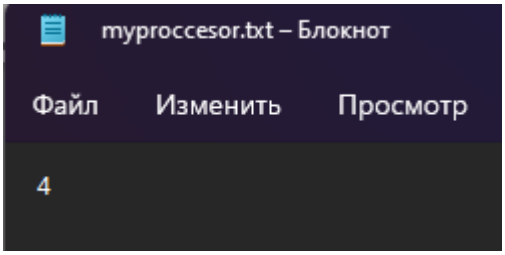

Рисунок 8 – Результат выполнения программы

Вывод: в результате выполнения лабораторной работы были приобретены практические навыки и теоретические сведения по работе с текстовыми файлами при написании программ с помощью языка программирования Python версии 3.x, а также изучены основные методы модуля os для работы с файловой системой, получением аргументов командной строки.

Ответы на контрольные вопросы:

1. Как открыть файл в языке Python только для чтения?

Чтобы открыть файл для чтения, мы используем режим r. Для чтения мы воспользуемся функцией read(size), если параметр size не указан, функция вернет нам всю строку. file = open("text.txt", 'r', encoding = 'utf-8').

2. Как открыть файл в языке Python только для записи?

В Python открытие файлов выполняется с помощью функции open(), которой передается два аргумента - имя файла и режим. Файл может быть открыт в режиме чтения, записи, добавления.

3. Как прочитать данные из файла в языке Python?

Чтение данных из файла осуществляется с помощью методов read(размер) и readline(). Метод read(размер) считывает из файла определенное количество символов, переданное в качестве аргумента.

4. Как записать данные в файл в языке Python?

Запись данных в файл. Записать данные в файл можно с помощью метода write().

5. Как закрыть файл в языке Python?

После того, как мы открыли файл, и выполнили все нужные операции, нам необходимо его закрыть. Для закрытия файла используется функция close().

6. Изучите самостоятельно работу конструкции with ... as. Каково ее назначение в языке?

Конструкция with ... as используется для оборачивания выполнения блока инструкций менеджером контекста. Если в конструкции with - as было несколько выражений, то это эквивалентно нескольким вложенным конструкциям

7. Изучите самостоятельно документацию Python по работе с файлами. Какие помимо рассмотренных существуют методы записи/чтения информации из файла?

Один из самых распространенных способов вывести данные в Python – это напечатать их в консоли. Если вы находитесь на этапе изучения языка, такой способ является основным для того, чтобы быстро просмотреть результат свой работы

8. Какие существуют, помимо рассмотренных, функции модуля os для работы с файловой системой?

os.chdir(path) - смена текущей директории.

os.chmod (path, mode, *, dir_fd=None, follow_symlinks=True) - смена прав доступа к объекту (mode - восьмеричное число).

os.chown (path, uid, gid, *, dir_fd=None, follow_symlinks=True) - меняет id владельца и группы (Unix).

os.getcwd() - текущая рабочая директория.

os.link (src, dst, *, src_dir_fd=None, dst_dir_fd=None,

follow_symlinks=True) - создаёт жёсткую ссылку.

os.listdir (path=".") - список файлов и директорий в папке.

os.mkdir (path, mode=0o777, *, dir_fd=None) - создаёт директорию.

OSError, если директория существует.

os.makedirs (path, mode=0o777, exist_ok=False) - создаёт директорию, создавая при этом промежуточные директории.

os.remove (path, *, dir_fd=None) - удаляет путь к файлу.

os.rename (src, dst, *, src_dir_fd=None, dst_dir_fd=None) - переименовывает файл или директорию из src в dst.

os.renames (old, new) - переименовывает old в new, создавая промежуточные директории.

os.replace (src, dst, *, src_dir_fd=None, dst_dir_fd=None) - переименовывает из src в dst с принудительной заменой.

os.rmdir (path, *, dir_fd=None) - удаляет пустую директорию.

os.removedirs (path) - удаляет директорию, затем пытается удалить родительские директории, и удаляет их рекурсивно, пока они пусты.

os.sync() - записывает все данные на диск (Unix).

os.truncate (path, length) - обрезает файл до длины length.

os.utime (path, times=None, *, ns=None, dir_fd=None,

follow_symlinks=True) - модификация времени последнего доступа и изменения файла. Либо times - кортеж (время доступа в секундах, время изменения в секундах), либо ns - кортеж (время доступа в наносекундах, время изменения в наносекундах).

os.walk (top, topdown=True, onerror = None, followlinks=False) – генерация имён файлов в дереве каталогов, сверху вниз (если topdown равен True), либо снизу вверх (если False). Для каждого каталога функция walk возвращает кортеж (путь к каталогу, список каталогов, список файлов).