# МИНИCTEPCTBO НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙ- СКОЙ 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 |
| Криворот В.Г. « » 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 ра- боты с IDE PyCharm с ЯП Python, выбрал лицензию MIT, клонировал его на компьютер и организовал в соответствии с моделью ветвления git-flow.

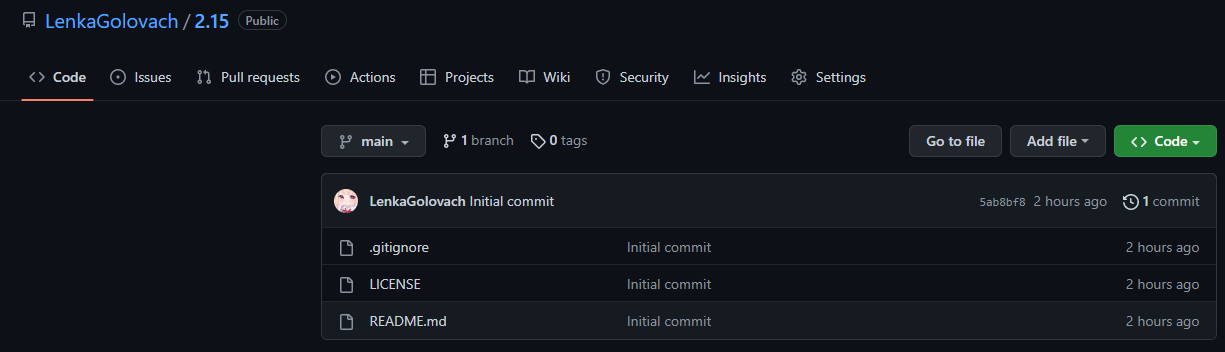


Рисунок 1.1 – Созданный репозиторий

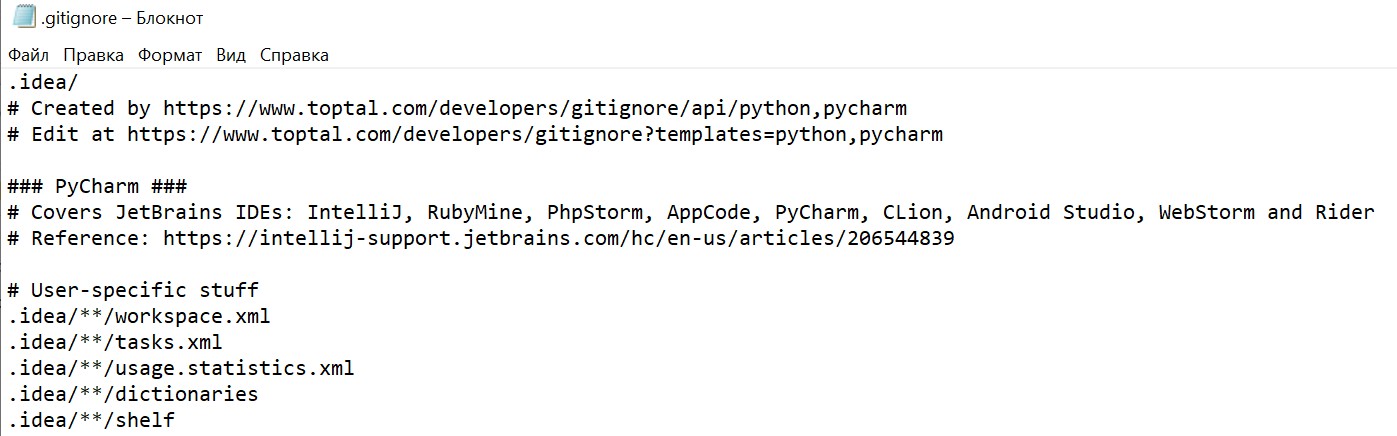


Рисунок 1.2 – Дополнил правила в .gitignore

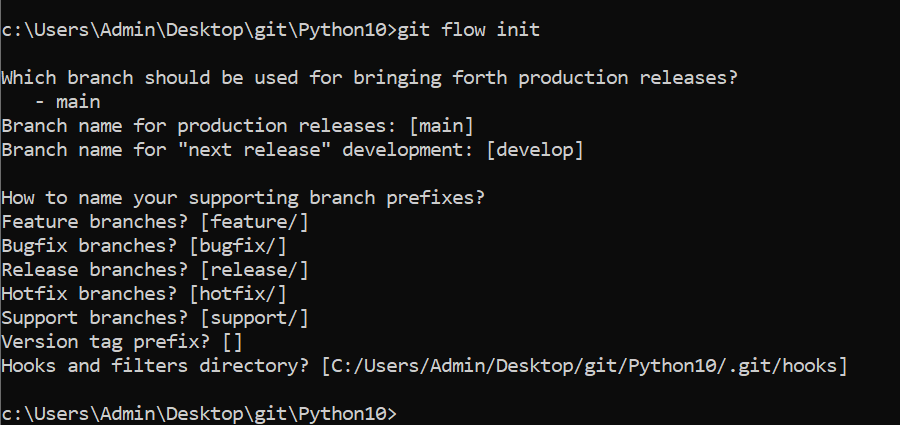


Рисунок 1.3 – Организация репозитория в соответствии с моделью ветвления git-flow

ЛР.

1. **Создал проект Pycharm в папке репозитория**, проработал примеры

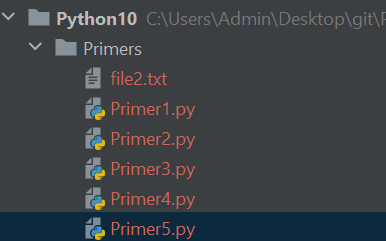


Рисунок 2 – Созданный проект

# Индивидуальные задания.

**Индивидуальное задание №1.** В – 1.

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

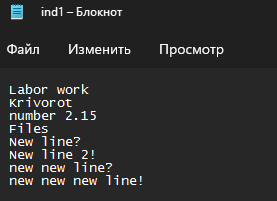


Рисунок 3 – Текст в файле

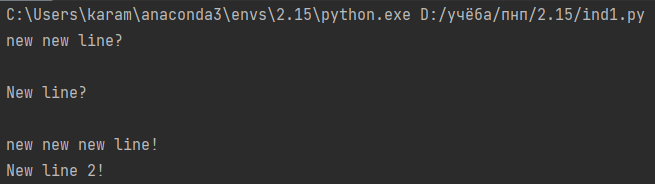


Рисунок 4 – Результат работы программы

# 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 про- грамму на Python, имитирующую поведение этой утилиты. Если файла, ука- занного пользователем, не существует, или не задан аргумент командной строки, необходимо вывести соответствующее сообщение об ошибке.

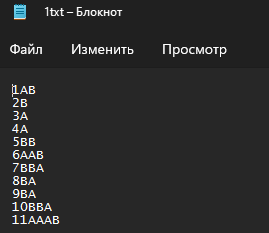


Рисунок 5 – Содержимое текстового файла для задания

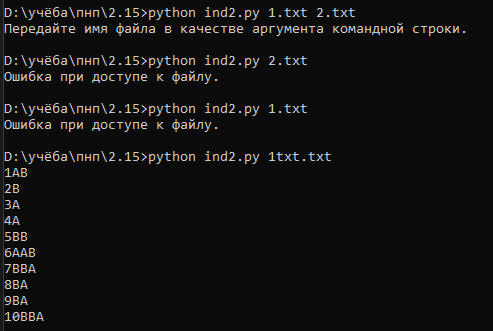


Рисунок 6 – Все возможные варианты вывода индивидуального задания №2

**Индивидуальное задание №3.** Самостоятельно подберите или приду- майте задачу для работы с изученными функциями модуля os. Приведите ре- шение этой задачи.

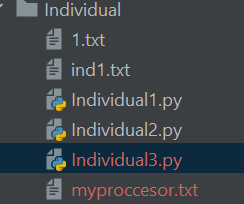
**Задача:** создать новый текстовый файл, затем требуется узнать число ядер процессора, а потом записать их в созданный файл, изменить имя файла на myproccesor.txt если его еще не существует, если он существует, то выдать соответствующее сообщение.

Рисунок 7 – Созданный файл в папке с заданиями

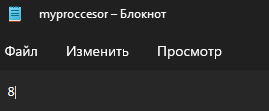


Рисунок 8 – Результат выполнения программы

**Вывод:** в результате выполнения лабораторной работы были приобре- тены 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 вер- нет нам всю строку. file = open("text.txt", 'r', encoding = 'utf-8').

# Как открыть файл в языке Python только для записи?

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

# Как прочитать данные из файла в языке Python?

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

# Как записать данные в файл в языке Python?

Запись данных в файл. Записать данные в файл можно с помощью ме- тода write().

# 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().

# 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 конструк- циям

# Изучите самостоятельно документацию Python по работе с фай- лами. 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, та- кой способ является основным для того, чтобы быстро просмотреть результат свой работы

# 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, со- здавая 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) - переименовы- вает файл или директорию из src в dst.

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

os.replace (src, dst, \*, src\_dir\_fd=None, dst\_dir\_fd=None) - переименовы- вает из src в dst с принудительной заменой.

os.rmdir (path, \*, dir\_fd=None) - удаляет пустую директорию.

os.removedirs (path) - удаляет директорию, затем пытается удалить роди- тельские 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 из- менения файла. Либо times - кортеж (время доступа в секундах, время измене- ния в секундах), либо ns - кортеж (время доступа в наносекундах, время изме- нения в наносекундах).

os.walk (top, topdown=True, onerror = None, followlinks=False) – генера- ция 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).