

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового  
развитияКафедра  
инфокоммуникаций

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ  
№2.16

Выполнил:

Назаров Никита Юрьевич

2 курс, группа ИВТ-б-о-21-  
1,

09.03.01 «Информатика

и вычислительная  
техника»,направленность  
(профиль)

«Программное обеспечение  
средстввычислительной техники  
и автоматизированных систем»,  
очная форма обучения

---

(подпись)

Ставрополь, 2022 г.

Выполнение работы.

```
*.gitignore – Блокнот
Файл Правка Формат Вид Справка

.idea/
# Created by https://www.toptal.com/developers/gitignore/api/python,pycharm
# Edit at https://www.toptal.com/developers/gitignore?templates=python,pycharm

### PyCharm ###
# Covers JetBrains IDEs: IntelliJ, RubyMine, PhpStorm, AppCode, PyCharm, CLion, Android Studio, WebStorm and Rider
# Reference: https://intellij-support.jetbrains.com/hc/en-us/articles/206544839

# User-specific stuff
.idea/**/workspace.xml
.idea/**/tasks.xml
.idea/**/usage.statistics.xml
.idea/**/dictionaries
.idea/**/shelf

# AWS User-specific
.idea/**/aws.xml

# Generated files
.idea/**/contentModel.xml

# Sensitive or high-churn files
.idea/**/dataSources/
.idea/**/dataSources.ids
.idea/**/dataSources.local.xml
.idea/**/sqlDataSources.xml
.idea/**/dynamic.xml
```

Рисунок 1 – редактирование gitignore

```
C:\Users\AdMin\Desktop\2.15>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]

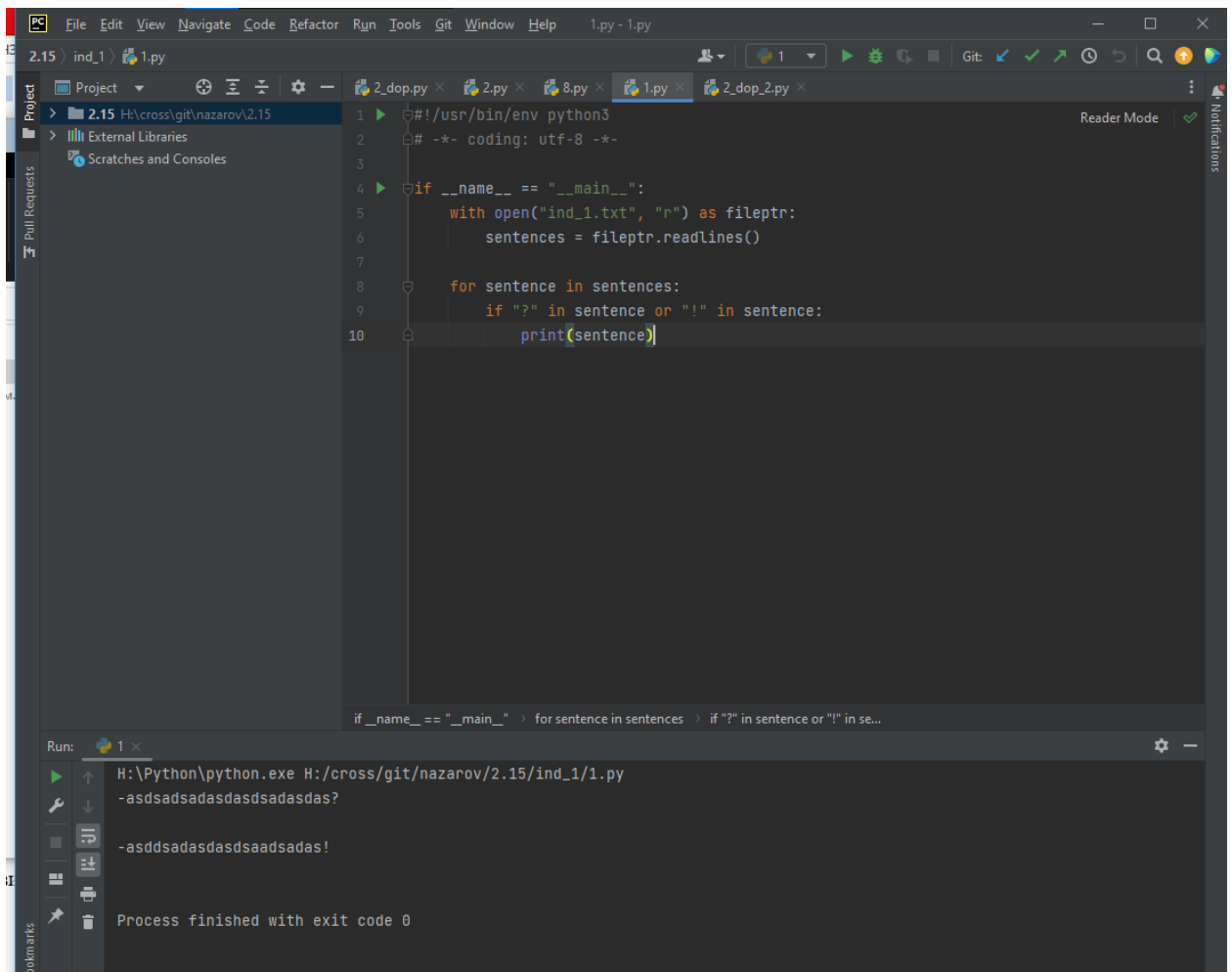
How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [C:/Users/AdMin/Desktop/2.15/.git/hooks]

C:\Users\AdMin\Desktop\2.15>
```

Рисунок 2 – организация репозитория в соответствии с моделью git-flow

Имя файла	Дата и время	Тип файла	Размер
1.py	26.12.2022 23:39	Python File	1 КБ
2.py	26.12.2022 23:39	Python File	1 КБ
3.py	26.12.2022 23:39	Python File	1 КБ
4.py	26.12.2022 23:39	Python File	1 КБ
5.py	26.12.2022 23:39	Python File	1 КБ
6.py	26.12.2022 23:39	Python File	1 КБ
7.py	26.12.2022 23:39	Python File	1 КБ
8.py	26.12.2022 23:39	Python File	1 КБ
9.py	26.12.2022 23:39	Python File	1 КБ
10.py	26.12.2022 23:39	Python File	1 КБ
11.py	26.12.2022 23:39	Python File	1 КБ
12.py	26.12.2022 23:39	Python File	1 КБ
13.py	26.12.2022 23:39	Python File	1 КБ
14.py	26.12.2022 23:39	Python File	1 КБ
15.py	26.12.2022 23:39	Python File	1 КБ
16.py	26.12.2022 23:39	Python File	1 КБ
17.py	26.12.2022 23:39	Python File	1 КБ
newfile.txt	26.12.2022 23:39	Текстовый докум...	0 КБ
text.txt	26.12.2022 23:39	Текстовый докум...	1 КБ

Рисунок 3 – проработал примеры, представленные в работе



```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 if __name__ == "__main__":
5     with open("ind_1.txt", "r") as fileptr:
6         sentences = fileptr.readlines()
7
8     for sentence in sentences:
9         if "?" in sentence or "!" in sentence:
10             print(sentence)
```

Run: 1 x

```
H:\Python\python.exe H:/cross/git/nazarov/2.15/ind_1/1.py
-asdsadsadsadsadsadsads?
-asddsadsadsadsadsads!
Process finished with exit code 0
```

Рисунок 4 – выполнил 1 индивидуальное задание

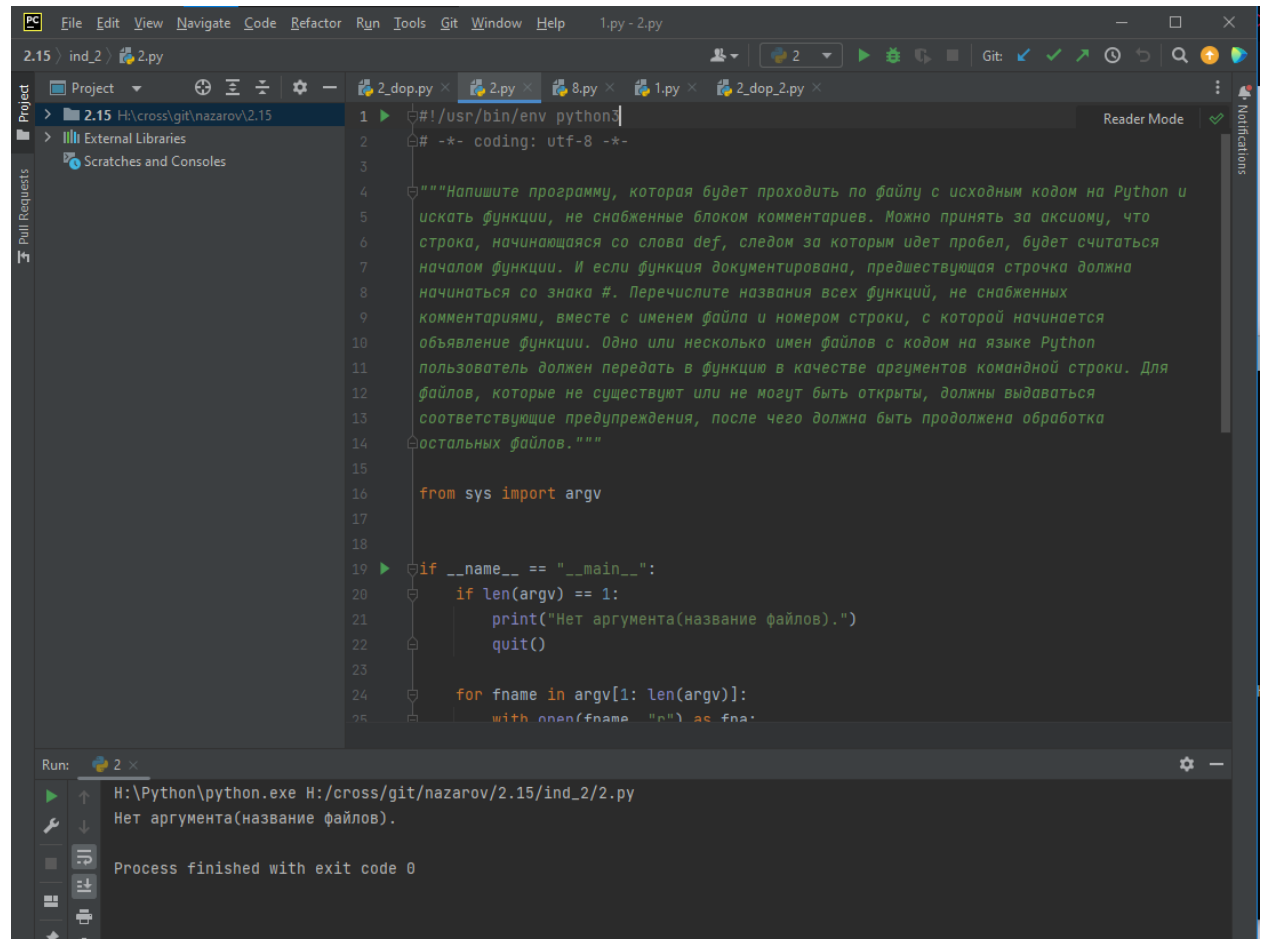


Рисунок 5 – выполнил индивидуальное задание 2.

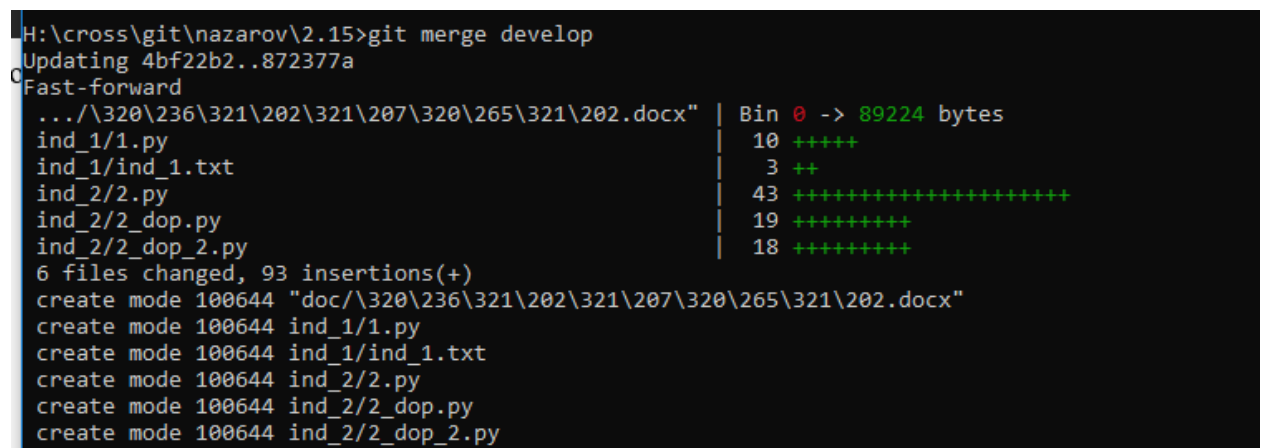


Рисунок 6 – выполнил слияние веток

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 своей 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.rename (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).