

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ
ФЕДЕРАЦИИ**
**Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

Институт цифрового развития

ОТЧЁТ

по лабораторной работе №2.15

Дисциплина: «Программирование на Python»

Тема: «Работа с файлами в языке Python»

Выполнил: студент 2 курса

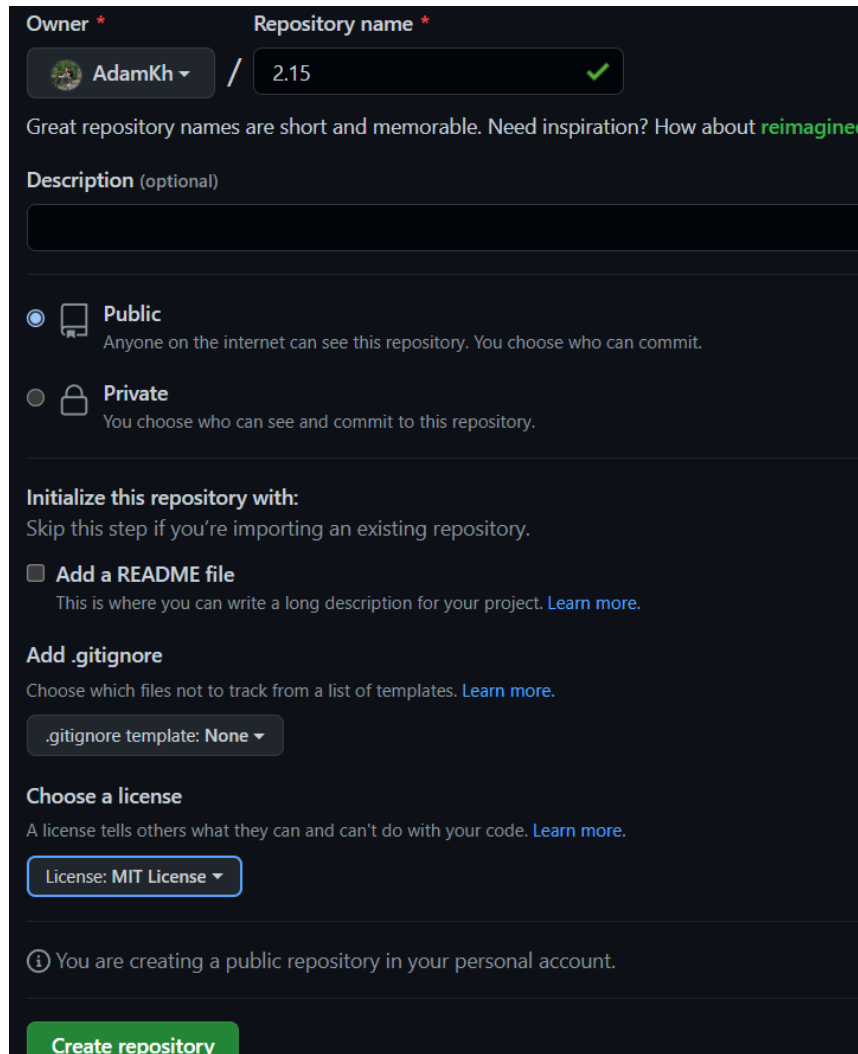
группы ИВТ-б-о-21-1

Хашиев Адам Мухарбекович

Ставрополь 2022

Выполнение работы:

1. Создал репозиторий в GitHub «rep 2.6» в который добавил .gitignore, который дополнил правила для работы с IDE PyCharm с ЯП Python, выбрал лицензию MIT, клонировал его на лок. сервер и организовал в соответствии с моделью ветвления git-flow.



The screenshot shows the GitHub 'Create repository' form. At the top, the 'Owner' is 'AdamKh' and the 'Repository name' is '2.15', with a green checkmark indicating it's valid. Below this is a 'Description (optional)' text area. The 'Public' option is selected with a radio button, and the 'Private' option is unselected. Under 'Initialize this repository with:', the 'Add a README file' checkbox is unchecked. The 'Add .gitignore' section shows a dropdown menu set to '.gitignore template: None'. The 'Choose a license' section shows a dropdown menu set to 'License: MIT License'. At the bottom, there is a green 'Create repository' button and a note: 'You are creating a public repository in your personal account.'

Рисунок 1.1 Создание репозитория

```
C:\Users\adamkh>cd C:\Users\adamkh\Desktop\3sem\Python\2.15
C:\Users\adamkh\Desktop\3sem\Python\2.15>git clone https://github.com/AdamKh/2.15.git
Cloning into '2.15'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
```

Рисунок 1.2 Клонирование репозитория

```

C:\Users\adamkh\Desktop\3sem\Python\2.15\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/adamkh/Desktop/3sem/Python/2.15/2.15/.git/hooks]

```

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



```

.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,
# 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

```

Рисунок 1.4 Изменение .gitignore

2. Создал проект PyCharm в папке репозитория активировал виртуальное окружение, проработал примеры ЛР.

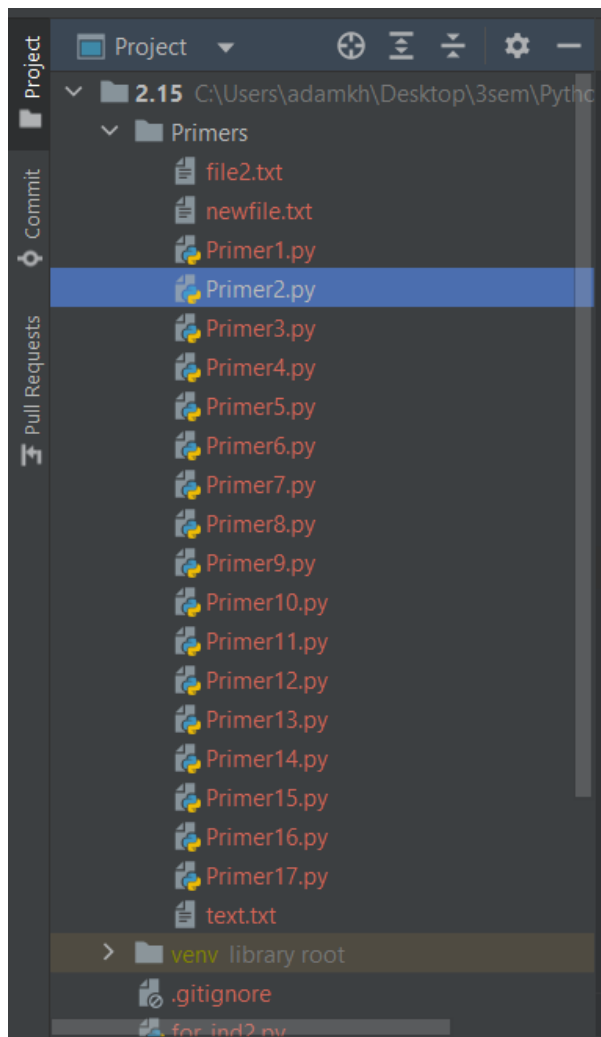


Рисунок 2.1 Создание проекта в PyCharm

```
C:\Users\adamkh\Desktop\3sem\Python\2.15\2.15>.\venv\Scripts\activate  
(venv) C:\Users\adamkh\Desktop\3sem\Python\2.15\2.15>
```

Рисунок 2.2 Активация ВО

3. (15 вариант). Выполнил индивидуальные задания №1, 2 и 3.
Перенаправил пакетные зависимости в файл requirements.txt

```
Вопрос?
Какой твой любимый цвет?
Сколько стоит этот товар?
От куда?

Красный!
Ответ!
Только попробуй!

Process finished with exit code 0
```

Рисунок 3.1 Вывод программы индивидуального задания №1

```
(venv) C:\Users\adamkh\Desktop\3sem\Python\2.15\2.15>python ind2.py for_ind2.py asd.py 2.py ind1.py
Номер строки - 2 , имя файла - for_ind2.py , функция - asd
Номер строки - 7 , имя файла - for_ind2.py , функция - asd2
Номер строки - 10 , имя файла - for_ind2.py , функция - asd3
Номер строки - 15 , имя файла - for_ind2.py , функция - asd5
Номер строки - 20 , имя файла - for_ind2.py , функция - asd7
Не удалось открыть файл с именем asd.py
Не удалось открыть файл с именем 2.py
(venv) C:\Users\adamkh\Desktop\3sem\Python\2.15\2.15>
```

Рисунок 3.2 Вывод программы индивидуального задания №2

```
(venv) C:\Users\adamkh\Desktop\3sem\Python\2.15\2.15>python ind3.py for asd weq
Ошибка. Слишком много параметров.

(venv) C:\Users\adamkh\Desktop\3sem\Python\2.15\2.15>python ind3.py for asd
Ошибка. Неизвестный параметр 'for'

(venv) C:\Users\adamkh\Desktop\3sem\Python\2.15\2.15>python ind3.py --name asd
Привет, asd!

(venv) C:\Users\adamkh\Desktop\3sem\Python\2.15\2.15>
```

Рисунок 3.3 Вывод программы индивидуального задания №3

```
(venv) C:\Users\adamkh\Desktop\3sem\Python\2.15\2.15>pip freeze > requirements.txt
(venv) C:\Users\adamkh\Desktop\3sem\Python\2.15\2.15>
```

Рисунок 3.4 Перенаправление пакетных зависимостей в файл (сохранение)

4. Сделал коммит, выполнил слияние с веткой main, и запустил изменения в уд. репозиторий.

```

C:\rep_2.6>git add .

C:\rep_2.6>git commit -m "added programs + modidied .gitignore"
[develop 2582c62] added programs + modidied .gitignore
 4 files changed, 379 insertions(+), 3 deletions(-)
 create mode 100644 ind.py
 create mode 100644 prim.py
 create mode 100644 zadaniya.py

C:\rep_2.6>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

C:\rep_2.6>

```

```

C:\rep_2.6>git push
Everything up-to-date

```

Рисунок 4.1 коммит и пуш изменений и переход на ветку main

```

C:\rep_2.6>git merge develop
Updating 5d4b8d1..2582c62
Fast-forward
 .gitignore | 157 +++++
--
 ind.py      | 105 +++++
 prim.py     | 99  +++++
 zadaniya.py | 21  +++++
 4 files changed, 379 insertions(+), 3 deletions(-)
 create mode 100644 ind.py
 create mode 100644 prim.py
 create mode 100644 zadaniya.py

C:\rep_2.6>_

```

Рисунок 4.2 Слияние ветки main с develop

```

C:\rep_2.6>git push
info: please complete authentication in your browser...
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 5.16 KiB | 2.58 MiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/AdamKh/rep_2.6.git
 5d4b8d1..2582c62  main -> main

```

Рисунок 4.3 Пуш изменений на удаленный сервер

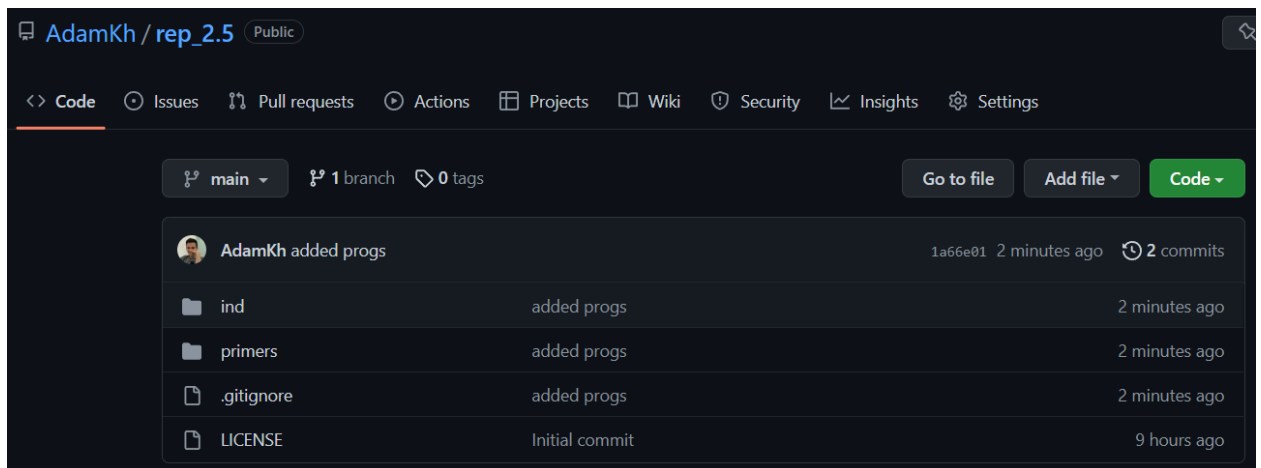


Рисунок 4.4 Изменения на удаленном сервере

Контр. вопросы и ответы на них:

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).