

**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 №2.7**

Дисциплина: «Программирование на Python»

Тема: «Работа с множествами в  
языке Python»

Выполнил: студент 2 курса

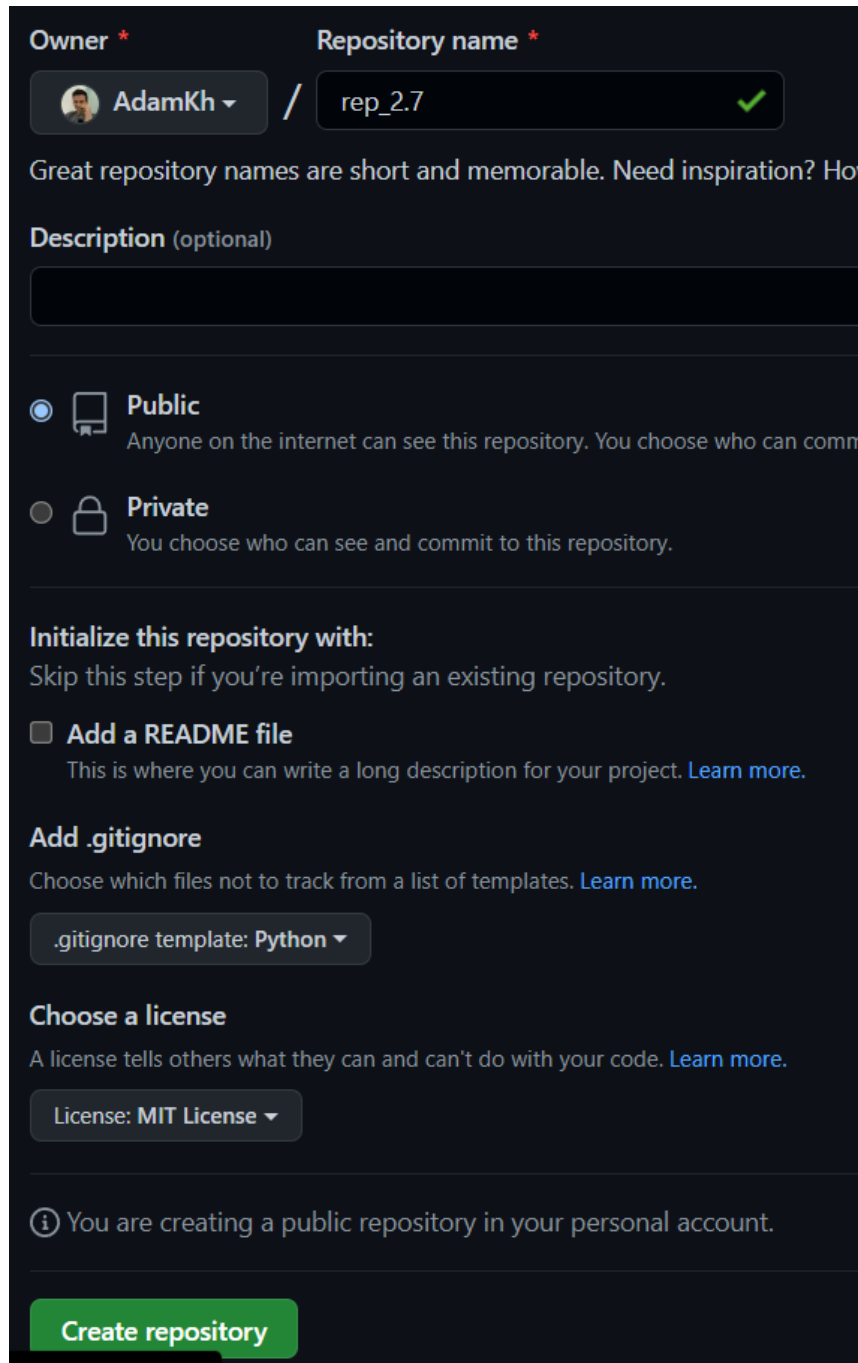
группы ИВТ-б-о-21-1

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.



Owner \* / Repository name \*

AdamKh / rep\_2.7 ✓

Great repository names are short and memorable. Need inspiration? How to choose a name

Description (optional)

☒ Public  
Anyone on the internet can see this repository. You choose who can commit to it.

☐ Private  
You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☐ Add a README file  
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore

Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: Python

Choose a license

A license tells others what they can and can't do with your code. [Learn more.](#)

License: MIT License

*i* You are creating a public repository in your personal account.

Create repository

Рисунок 1.1 Создание репозитория

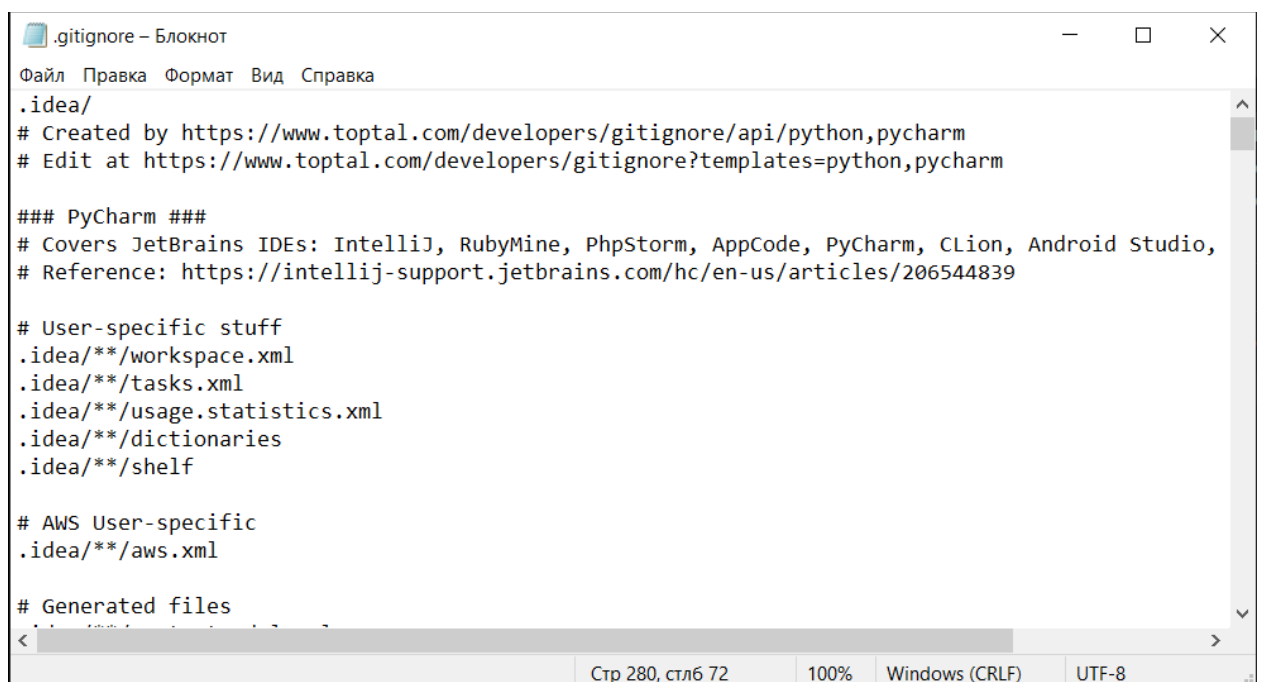
```
C:\Users\adamkh\Desktop\3 семестр\Программирование на Python\2.7>git clone https://github.com/AdamKh/rep_2.7.git
Cloning into 'rep_2.7'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.
C:\Users\adamkh\Desktop\3 семестр\Программирование на Python\2.7>_
```

Рисунок 1.2 Клонирование репозитория

```
C:\Users\adamkh\Desktop\3 семестр\Программирование на Python\2.7\rep_2.7>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/3 семестр/Программирование на Python/2.7/rep_2.7/.git/hooks]
```

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



```
.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

# Generated files
```

Рисунок 1.4 Изменение .gitignore

2. Создал проект PyCharm в папке репозитория и проработал примеры ЛР.

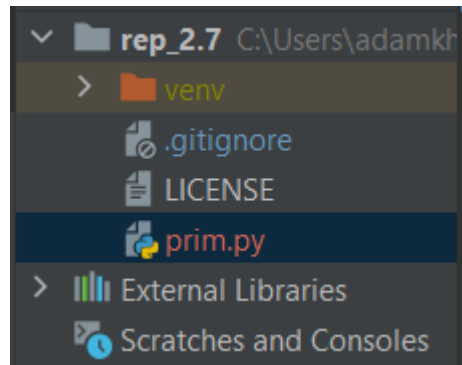


Рисунок 2.1 Создание проекта в PyCharm

```
"C:\Users\adamkh\Desktop\3 семестр\Прогр
x = {'e', 'j', 'd', 'k', 'o'}
y = {'y', 'c', 'f', 'v', 'h', 'g', 'o'}

Process finished with exit code 0
```

Рисунок 2.2 Рез-т выполнения программы

3. Выполнил задания.

Рисунок 3.1 Вывод программы задания

4. (15 вариант). Выполнил индивидуальное задание.

Определить результат выполнения операций над множествами. Считать элементы множества строками. Проверить результаты вручную.

$$A = \{c, m, n, o, q\}; \quad B = \{c, d, m, w\}; \quad C = \{m, n, q\}; \quad D = \{c, m, p\};$$

$$X = (A \cup B) \cap C; \quad Y = (A \cap \bar{B}) \cup (C/D).$$

```

c = red
m = blue
n = green
o = purple
q = ping
d = yellow
w = orange
p = white

Множество A = {'red', 'green', 'blue', 'ping', 'purple'}
Множество B = {'yellow', 'blue', 'red', 'orange'}
Множество C = {'green', 'blue', 'ping'}
Множество D = {'blue', 'white', 'red'}
Множество X = {'green', 'blue', 'ping'}
Множество Y = {'green', 'purple', 'ping'}

Process finished with exit code 0

```

Рисунок 4.1 Вывод программы индивидуального задания

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

```

C:\Users\adamkh\Desktop\3 семестр\Программирование на Python\2.7\rep_2.7>git add .
C:\Users\adamkh\Desktop\3 семестр\Программирование на Python\2.7\rep_2.7>git commit -m "added progs"
[develop 3a830de] added progs
 4 files changed, 221 insertions(+), 3 deletions(-)
 create mode 100644 ind.py
 create mode 100644 prim.py
 create mode 100644 zadaniya.py

C:\Users\adamkh\Desktop\3 семестр\Программирование на Python\2.7\rep_2.7>git push --set-upstream origin develop
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), 3.69 KiB | 1.84 MiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:   https://github.com/AdamKh/rep_2.7/pull/new/develop
remote:
To https://github.com/AdamKh/rep_2.7.git
 * [new branch]      develop -> develop
branch 'develop' set up to track 'origin/develop'.

C:\Users\adamkh\Desktop\3 семестр\Программирование на Python\2.7\rep_2.7>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

```

Рисунок 4.1 коммит и пуш изменений и переход на ветку main

```
C:\Users\adamkh\Desktop\3 семестр\Программирование на Python\2.7\rep_2.7>git merge develop
Updating 95b90b4..3a830de
Fast-forward
 .gitignore | 157 ++++++
 ind.py      | 29 ++++++
 prim.py     | 22 ++++++
 zadaniya.py | 16 ++++++
 4 files changed, 221 insertions(+), 3 deletions(-)
 create mode 100644 ind.py
 create mode 100644 prim.py
 create mode 100644 zadaniya.py
```

Рисунок 4.2 Слияние ветки main с develop

```
C:\Users\adamkh\Desktop\3 семестр\Программирование на Python\2.7\rep_2.7>git push
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/AdamKh/rep_2.7.git
 95b90b4..3a830de main -> main
```

Рисунок 4.3 Пуш изменений на удаленный сервер

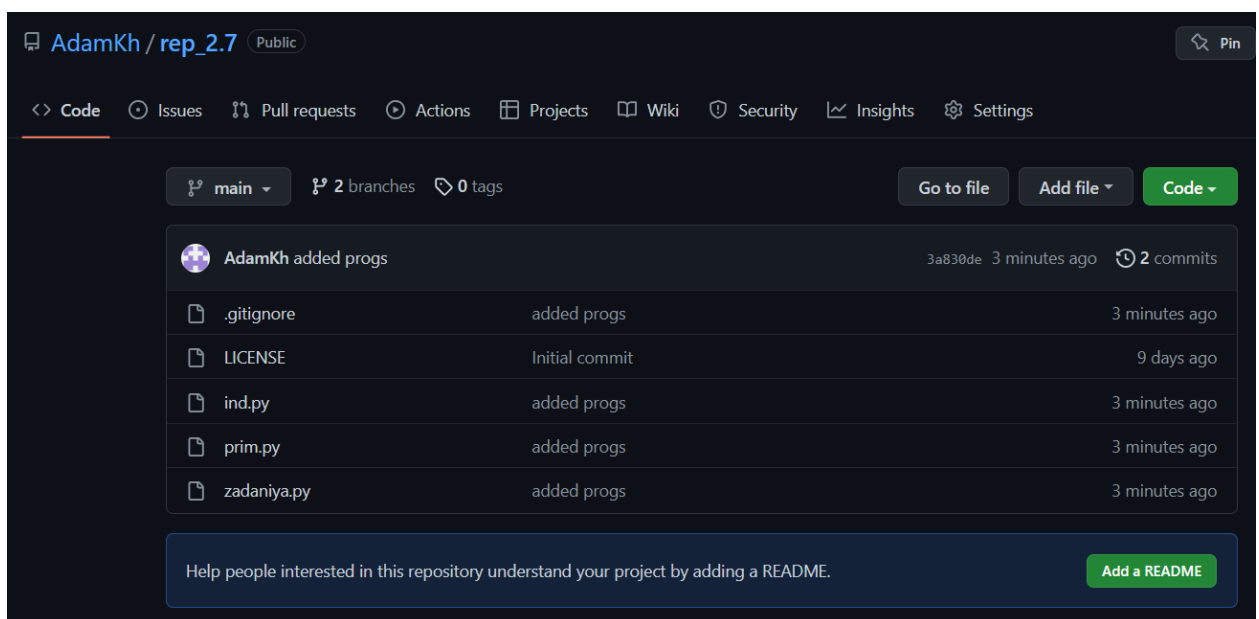


Рисунок 4.4 Изменения на удаленном сервере

**Контр. вопросы и ответы на них:**

**1. Что такое множества в языке Python?**

Это неупорядоченная совокупность уникальных элементов.

**2. Как осуществляется создание множества в Python?**

С помощью фигурных скобок. Пример: `a = {a, b, c, d}`

**3. Как проверить присутствие/отсутствие элемента в множестве?**

`<элемент> in <множество>` или `<элемент> not in <множество>`

**4. Как выполнить перебор элементов множества?**

С помощью цикла `for`

**5. Что такое set comprehension?**

Это метод для создания множеств из других итерируемых объектов

**6. Как выполнить добавление элемента во множество?**

С помощью метода `add()`

**7. Как выполнить удаление одного или всех элементов множества?**

Удаление одного элемента производится с помощью метода `remove()`, а удаление при помощи метода `clear()`

**8. Как выполнить основные операции над множествами:**

**объединение, пересечение, разность?**

Объединение: `union()`

Пересечение: `intersection()`

Разность: `difference()`

**9. Как определить, что некоторое множество является надмножеством или подмножеством другого множества?**

С помощью методов `issubset()` и `issuperset()`

**10. Каково назначение множеств `frozenset`?**

Множество, созданное с помощью этого ключевого слова нельзя изменять.

**11. Как осуществляется преобразование множеств в строку, список, словарь?**

С помощью методов `dict()` и `list()`