

**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  
Урусов Максим Андреевич

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.

**Create a new repository**

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

**Owner \*** MaksimUrusov / **Repository name \*** Lab2.7 ✓

Great repository names are short and memorable. Need inspiration? How about **crispy-potato**?

**Description (optional)**

☐ **Public**  
Anyone on the internet can see this repository. You choose who can commit.

☒ **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 ▾

This will set **main** as the default branch. Change the default name in your [settings](#).

You are creating a private repository in your personal account.

**Create repository**

Рисунок 1.1 Создание репозитория

```
C:\Users\den-n>git clone https://github.com/MaksimUrusov/Laba2.6.git
Cloning into 'Laba2.6'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.
```

Рисунок 1.2 Клонирование репозитория

```
C:\Users\den-n\Laba2.6>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/den-n/Laba2.6/.git/hooks]

C:\Users\den-n\Laba2.6>
```

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

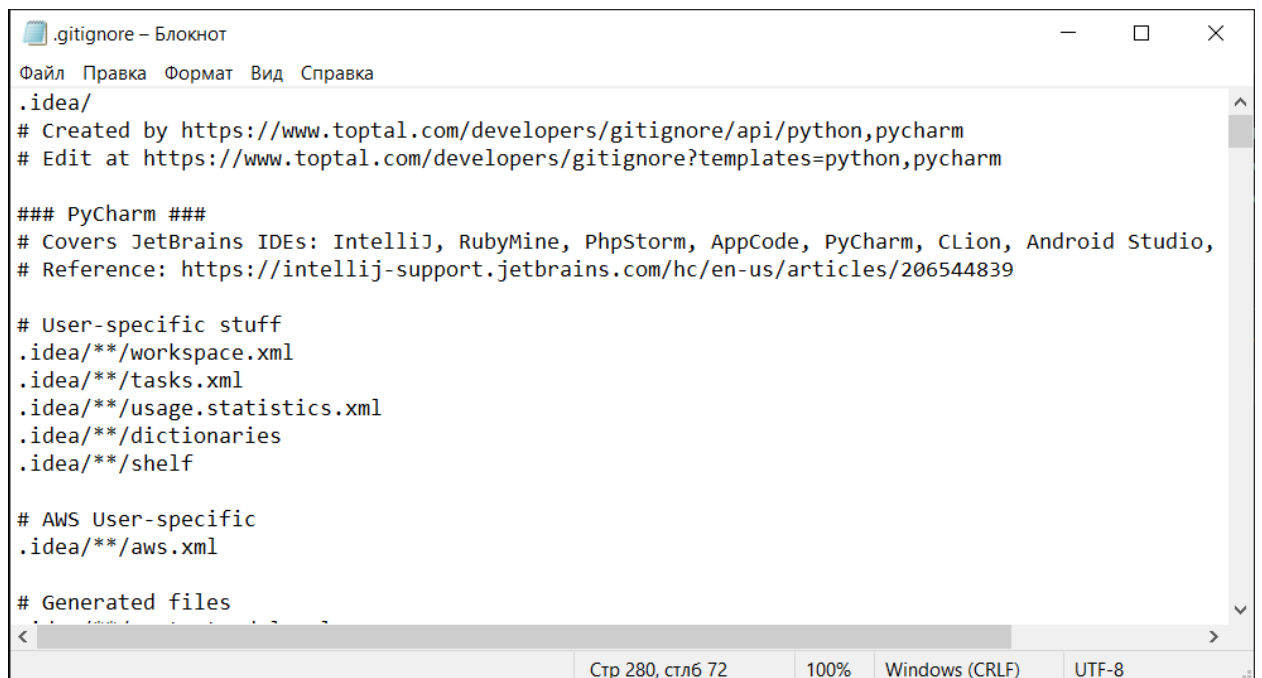


Рисунок 1.4 Изменение .gitignore

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

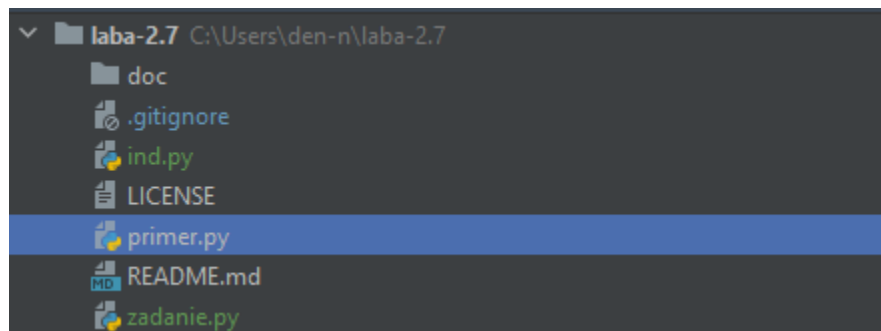


Рисунок 2.1 Создание проекта в PyCharm

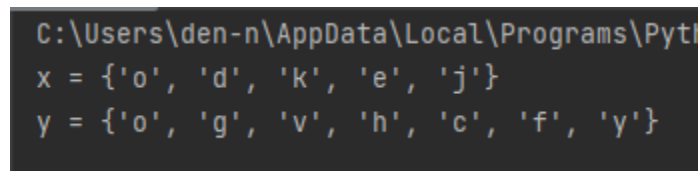


Рисунок 2.2 Рез-т выполнения программы

3. Выполнил задания.

Рисунок 3.1 Вывод программы задания

4. (18 вариант). Выполнил индивидуальное задание.

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

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

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

```
Множество A = {'6', '5', '4', '3', '2', '1'}
Множество B = {'7', '0', '3', '2', '8', '9'}
Множество C = {'12', '13', '14', '11', '10'}
Множество D = {'12', '16', '7', '13', '3', '17', '15'}

Множество X = set()
Множество Y = {'12', '11', '10', '13', '14'}

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\den-n\laba-2.7>git add .
C:\Users\den-n\laba-2.7>git commit -m "test"
[develop d11dbaf] test
 4 files changed, 228 insertions(+), 3 deletions(-)
 create mode 100644 ind.py
 create mode 100644 primer.py
 create mode 100644 zadanie.py
```

```
C:\Users\den-n\laba-2.7>git commit -m 1
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

C:\Users\den-n\laba-2.7>git push
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), 3.84 KiB | 1.92 MiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/MaksimUrusov/laba-2.7.git
 560785f..d11dbaf  main -> main
```

Рисунок 4.1 коммит и пуш изменений и переход на ветку main

```

C:\Users\den-n\laba-2.7>git merge develop
Updating 560785f..d11dbaf
Fast-forward
 .gitignore | 157 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 ind.py      | 39  ++++++
 primer.py   | 18  ++++++
 zadanie.py  | 17  ++++++
 4 files changed, 228 insertions(+), 3 deletions(-)
 create mode 100644 ind.py
 create mode 100644 primer.py
 create mode 100644 zadanie.py

```

Рисунок 4.2 Слияние ветки main с develop

```

C:\Users\den-n\laba-2.7>git push --set-upstream origin develop
Total 0 (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/MaksimUrusov/laba-2.7/pull/new/develop
remote:
To https://github.com/MaksimUrusov/laba-2.7.git
 * [new branch]      develop -> develop
branch 'develop' set up to track 'origin/develop'.

```

Рисунок 4.3 Пуш изменений на удаленный сервер

**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()`