

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

**Кафедра инфокоммуникаций  
«Работа с множествами в языке Python»**

**Отчет по лабораторной работе № 2.7  
по дисциплине «Программирование на Python»**

Выполнил студент группы ИВТ-б-о-21-1

Шайдеров Дмитрий Викторович.

«20 » сентября 2022г.

Подпись студента \_\_\_\_\_

Работа защищена «    » \_\_\_\_\_ 20\_\_ г.

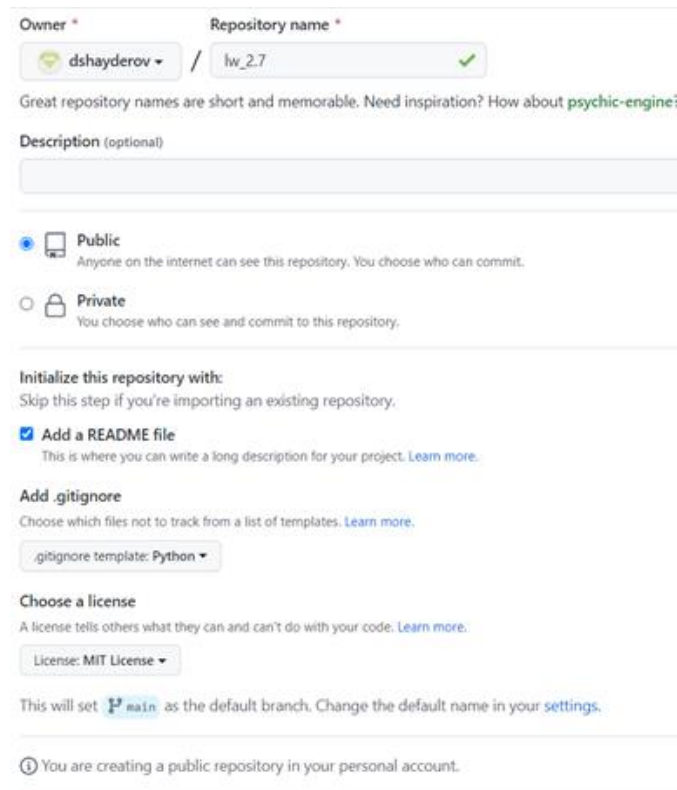
Проверил Воронкин Р.А. \_\_\_\_\_  
(подпись)

Ставрополь 2022

**Цель работы:** приобретение навыков по работе с множествами при написании программ с помощью языка программирования Python версии 3.x.

**Порядок выполнения работы:**

1. Создал общедоступный репозиторий на GitHub, в котором использована лицензия MIT и язык программирования Python.



Owner \* dshayderov / Repository name \* lw\_2.7

Great repository names are short and memorable. Need inspiration? How about [psychic-engine](#)?

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](#).

① You are creating a public repository in your personal account.

Рисунок 1 - Создание репозитория

2. Выполните клонирование созданного репозитория.

```
Asus@LAPTOP-09A994CG MINGW64 ~  
$ cd "c:\My projects\3"  
  
Asus@LAPTOP-09A994CG MINGW64 /c/My projects/3  
$ git clone https://github.com/dshayderov/lw_2.7.git  
Cloning into 'lw_2.7'...  
remote: Enumerating objects: 11, done.  
remote: Counting objects: 100% (11/11), done.  
remote: Compressing objects: 100% (10/10), done.  
remote: Total 11 (delta 2), reused 0 (delta 0), pack-reused 0  
Receiving objects: 100% (11/11), done.  
Resolving deltas: 100% (2/2), done.
```

Рисунок 2 - Клонирование репозитория

3. Организуйте свой репозиторий в соответствие с моделью ветвления git-flow.

```

Asus@LAPTOP-09A994CG MINGW64 /c/My projects/3
$ cd "C:\My projects\3\lw_2.7"

Asus@LAPTOP-09A994CG MINGW64 /c/My projects/3/lw_2.7 (main)
$ git checkout -b develop
Switched to a new branch 'develop'

```

Рисунок 3 - Ветвление по модели git-flow

4. Создайте проект PyCharm в папке репозитория.

Этот компьютер > OS (C:) > My projects > 3 > lw_2.7 >				
Имя	Дата изменения	Тип	Размер	
Project	26.09.2022 10:14	Папка с файлами		
.gitignore	26.09.2022 10:12	Файл "GITIGNORE"	2 КБ	
LICENSE	26.09.2022 10:12	Файл	2 КБ	
README.md	26.09.2022 10:12	Файл "MD"	1 КБ	

Рисунок 4 - Проект PyCharm

1. Проработайте пример лабораторной работы. Создайте для него отдельный модуль языка Python. Зафиксируйте изменения в репозитории.

```

primer x
"C:\My projects\3\lw_2.7\Project\venv\Scripts\python.exe"
x = {'k', 'e', 'd', 'o', 'j'}
y = {'h', 'v', 'g', 'c', 'o', 'y', 'f'}

```

Рисунок 5 - Результат выполнения примера

2. Решите задачу: подсчитайте количество гласных в строке, введенной с клавиатуры с использованием множеств.

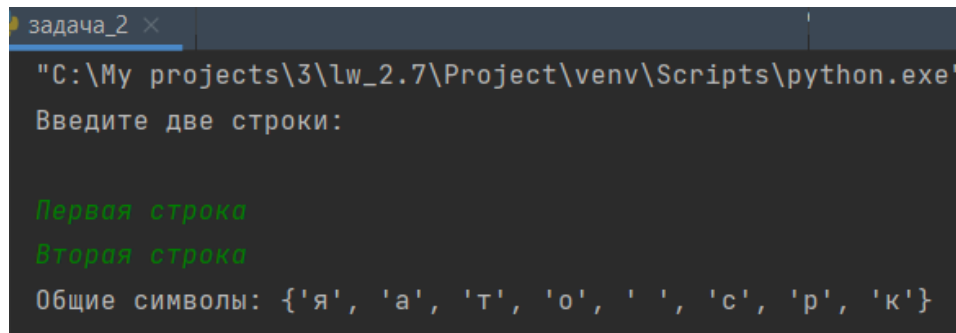
```

задача_1 x
"C:\My projects\3\lw_2.7\Project\venv\Scripts\python.exe"
Введите строку:
Подный зал людей
Кол-во гласных в строке: 5

```

Рисунок 6 - Результат выполнения задачи 1

3. Решите задачу: определите общие символы в двух строках, введенных с клавиатуры.



```
задача_2 x
"C:\My projects\3\lw_2.7\Project\venv\Scripts\python.exe"
Введите две строки:

Первая строка
Вторая строка
Общие символы: {'я', 'а', 'т', 'о', ' ', 'с', 'р', 'к'}
```

Рисунок 7 - Результат выполнения задачи 2

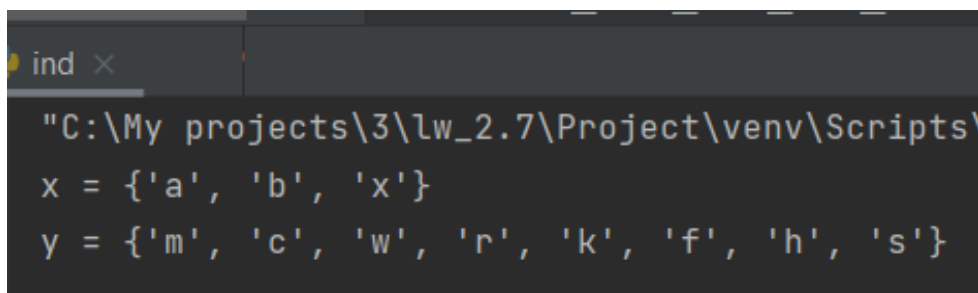
4. Выполните индивидуальное задание (Вариант 24).

Определить результат выполнения операций над множествами.  
Считать элементы множества строками.

$A = \{a, b, d, i, x\}; B = \{d, e, h, i, n, u\}; C = \{e, f, m, n\};$

$D = \{a, c, h, k, r, s, w, x\};$

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



```
ind x
"C:\My projects\3\lw_2.7\Project\venv\Scripts\
x = {'a', 'b', 'x'}
y = {'m', 'c', 'w', 'r', 'k', 'f', 'h', 's'}
```

Рисунок 8 - Результат выполнения индивидуального задания

## Контрольные вопросы

1. Что такое множества в языке Python?

Множеством в языке программирования Python называется неупорядоченная совокупность уникальных значений.

2. Как осуществляется создание множеств в Python?

Перед тем как начать работу с множеством, необходимо для начала его создать. Сделать это можно, просто присвоив переменной последовательность значений, выделив их фигурными скобками. Существует и другой способ создания множеств, который подразумевает использование вызова `set`.

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

Для проверки присутствия используется `in`, для проверки отсутствия – `not in`.

### **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?**

Чтобы внести новые значения, потребуется вызывать метод `add`.

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

Для удаления элементов из множества используются следующие функции в Python:

`remove` — удаление элемента с генерацией исключения в случае, если такого элемента нет;

`discard` — удаление элемента без генерации исключения, если элемент отсутствует;

`pop` — удаление первого элемента, генерируется исключение при попытке удаления из пустого множества.

Иногда необходимо полностью убрать все элементы. Чтобы не удалять каждый элемент отдельно, используется метод `clear`, не принимающий аргументов.

### **8. Как выполняются основные операции над множествами: объединение, пересечение, разность?**

Чтобы объединить все элементы двух разных множеств, стоит воспользоваться методом `union` на одном из объектов.

Чтобы найти общие элементы для двух разных множеств, следует применить функцию `intersection`, принимающую в качестве аргумента один из наборов данных.

Чтобы вычислить разность для двух разных множеств, необходимо воспользоваться методом `difference`.

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

Чтобы выяснить, является ли множество `a` подмножеством `b`, стоит попробовать вывести на экран результат выполнения метода `issubset`.

Чтобы узнать, является ли множество `a` надмножеством `b`, необходимо вызвать метод `issuperset` и вывести результат его работы на экран.

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

Множество, содержимое которого не поддается изменению имеет тип `frozenset`. Значения из этого набора нельзя удалить, как и добавить новые.

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

Для преобразования множества в строку используется конкатенация текстовых значений, которую обеспечивает функция `join`. В этом случае ее аргументом является набор данных в виде нескольких строк.

Чтобы получить из множества словарь, следует передать функции `dict` набор из нескольких пар значений, в каждом из которых будет находиться ключ.

По аналогии с предыдущими преобразованиями можно получить список неких объектов. На этот раз используется вызов `list`, получающий в качестве аргумента множество `a`.

**Вывод:** были приобретены навыки по работе с множествами при написании программ с помощью языка программирования Python версии 3.x.