

**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-22-1

Сумин Никита Сергеевич.

« » _____ 2023г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

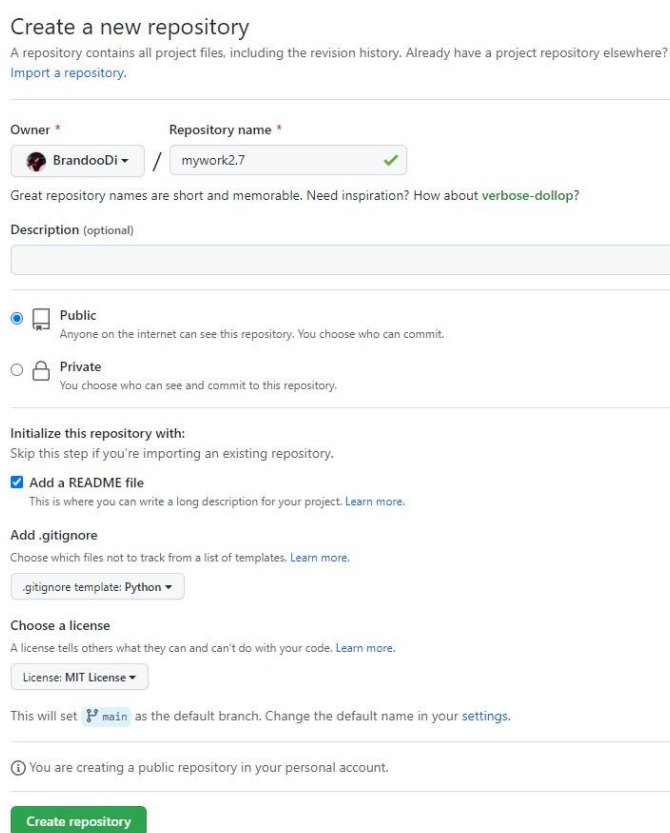
Проверил Воронкин Р.А. _____
(подпись)

Ставрополь 2023

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.



The screenshot shows the GitHub 'Create a new repository' page. At the top, it says 'Create a new repository' and provides a brief explanation of what a repository is. Below this, there are two main sections: 'Owner' and 'Repository name'. The 'Owner' is set to 'BrandooDi' and the 'Repository name' is 'mywork2.7', which is marked as valid with a green checkmark. A note below these fields states: 'Great repository names are short and memorable. Need inspiration? How about verbose-dollop?'. The 'Description (optional)' field is empty. The 'Visibility' section has two options: 'Public' (selected) and 'Private'. The 'Initialize this repository with:' section has a checkbox for 'Add a README file' which is checked. Below this is the 'Add .gitignore' section with a dropdown menu set to '.gitignore template: Python'. The 'Choose a license' section has a dropdown menu set to 'License: MIT License'. At the bottom, there is a green button labeled 'Create repository'.

Рисунок 1 - Создание репозитория

2. Выполните клонирование созданного репозитория.



```
MINGW64:/d/www/mywork2.7
user@MINGW64 ~
$ cd D:\www

user@MINGW64 /d/www
$ git clone https://github.com/BrandooDi/mywork2.7.git
Cloning into 'mywork2.7'...
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 8 (delta 1), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (8/8), done.
Resolving deltas: 100% (1/1), done.

user@MINGW64 /d/www
$ cd D:\www\mywork2.7

user@MINGW64 /d/www/mywork2.7 (main)
$ |
```

Рисунок 2 - Клонирование репозитория

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



```
user@MINGW64 /d/www/mywork2.7 (main)
$ git checkout -b develop
Switched to a new branch 'develop'

user@MINGW64 /d/www/mywork2.7 (develop)
$ |
```

Рисунок 3 - Ветвление по модели git-flow 4.

Создайте проект PyCharm в папке репозитория.

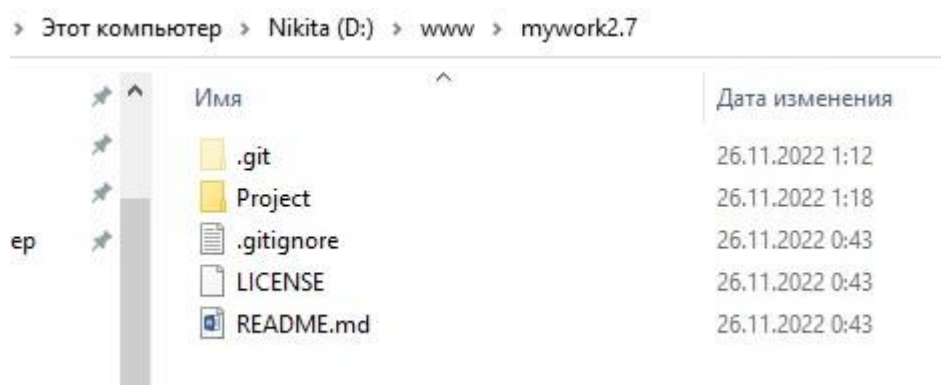


Рисунок 4 - Проект PyCharm

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

```
pr1 x
D:\www\mywork2.7\Project\venv\Scripts\python.exe D:/www/mywork2.7/Project/pr1.py
x = {'e', 'o', 'k', 'j', 'd'}
y = {'f', 'h', 'v', 'c', 'g', 'o', 'y'}

Process finished with exit code 0
```

Рисунок 5 - Результат выполнения примера

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

```
D:\www\mywork2.7\Project\venv\Scripts\python.exe D:/www/mywork2.7/Project/zd1.py
Введите строку:
А б я е н
Кол-во гласных в строке: 3

Process finished with exit code 0
|
```

Рисунок 6 - Результат выполнения задачи 1

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

```
D:\www\mywork2.7\Project\venv\Scripts\python.exe D:/www/mywork2.7/Project/zd2.py
Введите две строки:

Первое предложение
Второе предложение
Общие символы: {'о', 'и', 'п', 'л', 'е', ' ', 'н', 'р', 'ж'}

Process finished with exit code 0
|
```

Рисунок 7 - Результат выполнения задачи 2

4. Выполните индивидуальное задание (Вариант 24).

Определить результат выполнения операций над множествами.

Считать элементы множества строками.

$$A = \{a, b, f, g, i\}; \quad B = \{c, f, g, i, s, v\}; \quad C = \{a, g, h, i\}; \quad D = \{f, w, x\};$$

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

```
D:\www\mywork2.7\Project\venv\Scripts\python.exe D:/www/mywork2.7/Project/ind2.py
x = {'a', 'h', 'g', 'f', 'i'}
y = {'b', 'g', 'h', 'a', 'i'}

Process finished with exit code 0
```

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.