

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»

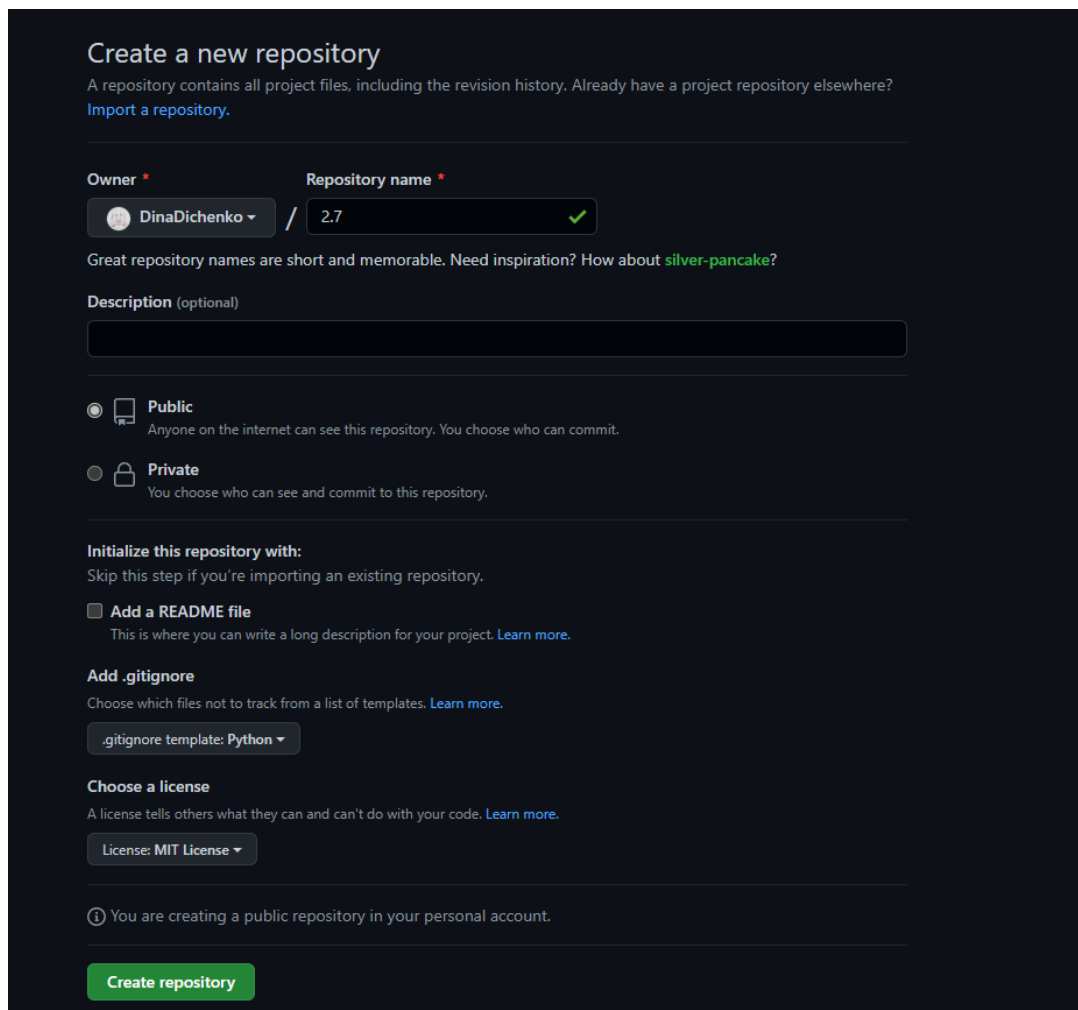
Выполнила: студентка 1 курса,
группы ИВТ-б-о-21-1
Диченко Дина Алексеевна

Ставрополь 2022

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

Практическая часть:

1. Создала общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и язык программирования Python. Выполнила клонирование созданного репозитория. Дополнила файл .gitignore необходимыми правилами для работы с IDE PyCharm. Организовала свой репозиторий в соответствие 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 **DinaDichenko** / Repository name **2.7** ✓

Great repository names are short and memorable. Need inspiration? How about **silver-pancake**?

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**

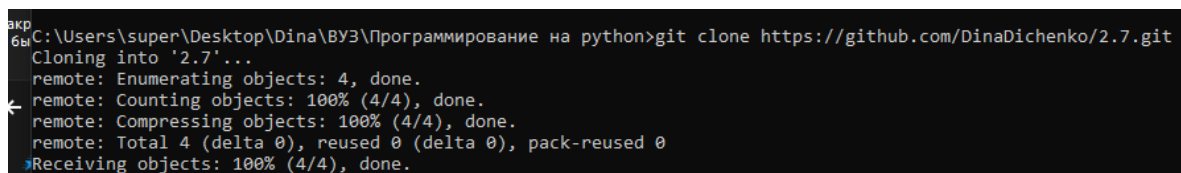
Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

License: **MIT License**

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

Create repository

Рисунок 1. Создание репозитория



```
C:\Users\super\Desktop\Dina\ВУЗ\Программирование на python>git clone https://github.com/DinaDichenko/2.7.git
Cloning into '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.
```

Рисунок 2. Клонирование репозитория

```
# 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, WebStorm and Rider
# 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
.idea/**/contentModel.xml

# Sensitive or high-churn files
.idea/**/dataSources/
.idea/**/dataSources.ids
.idea/**/dataSources.local.xml
.idea/**/sqlDataSources.xml
.idea/**/dynamic.xml
.idea/**/uiDesigner.xml
.idea/**/dbnavigator.xml
```

Рисунок 3. Изменение файла .gitignore

```
C:\Users\super\Desktop\Dina\ВУЗ\Программирование на python\2.7>git flow init
Which branch should be used for bringing forth production releases?
- develop
- main
Branch name for production releases: [main] main
Which branch should be used for integration of the "next release"?
- develop
Branch name for "next release" development: [develop] develop
How to name your supporting branch prefixes?
Feature branches? [feature/] fea
Bugfix branches? [bugfix/] bug
Release branches? [release/] rel
Hotfix branches? [hotfix/] hot
Support branches? [support/] sup
Version tag prefix? [] pre
Hooks and filters directory? [C:/Users/super/Desktop/Dina/ВУЗ/Программирование на python/2.7/.git/hooks] hoo
```

Рисунок 4. Организация репозитория в соответствии с git flow

2. Проработала пример лабораторной работы

```
C:\Users\aleks\Desktop\DiDi\2\venv\Scripts\python.exe C:/Users/aleks/Desktop/DiDi/2/main.py
x = {'e', 'k', 'j', 'd', 'o'}
y = {'c', 'y', 'o', 'v', 'g', 'f', 'h'}

Process finished with exit code 0
```

Рисунок 5. Результат работы программы

3. Решила задачу: подсчитайте количество гласных в строке, введенной с клавиатуры с использованием множеств.

```
C:\Users\aleks\Desktop\DiDi\2\venv\Scripts\python.exe C:/Users/
Введите множество: Шоколад
3
Process finished with exit code 0
```

Рисунок 6. Результат выполнения программы

4. Решила задачу: определите общие символы в двух строках,

введенных с клавиатуры.

```
C:\Users\aleks\Desktop\DiDi\2\venv\Scripts\python.exe C:/Users/aleks/Desktop/DiDi/2/venv/Z2.py
Введите первое множество: fghj
Введите второе множество: afgt
{'f', 'g'}

Process finished with exit code 0
```

Рисунок 7. Результат выполнения программы

5. Выполнила индивидуальное задание (вариант 8).

$$\begin{aligned} A &= \{a, f, I, n, o\}; & B &= \{f, g, o, p, z\}; & C &= \{i, j, u, w\}; & D &= \{f, h, n, t, u, y, z\}; \\ 8. & & X &= (A \cap B) \cup C; & Y &= (\bar{A} \cap \bar{B}) / (C \cup D). \end{aligned} \quad (9)$$

Рисунок 8. Индивидуальное задание

```
"C:\Users\super\Desktop\Dina\BY3\Программирование на python\Lab2.6\prog\venv\Script
X= {'o', 'f', 'i', 'w', 'j', 'u'}
Y= set()
```

Рисунок 9. Результат выполнения программы

Ответы на 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?

Создать множество можно можно, просто присвоив переменной последовательность значений, выделив их фигурными скобками или с помощью метода 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?

for a in {0, 1, 2}:

print(a)

5. Что такое set comprehension?

Генератор, позволяющий заполнять списки, а также другие наборы данных с учетом неких условий.

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 (кроме очистки, которая будет рассмотрена ниже):

`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. Если вывести содержимое после этой операции, на экране появится только его название.

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`, как в следующем примере. Так как не все элементы набора чисел присутствуют в

b, функция вернет False.

Чтобы узнать, является ли множество a надмножеством b, необходимо вызвать метод `issuperset` и вывести результат его работы на экран. Поскольку все элементы набора чисел b присутствуют в a, функция возвращает True.

10. Каково назначение множеств `frozenset` ?

Множество, содержимое которого не поддается изменению имеет тип `frozenset`. Значения из этого набора нельзя удалить, как и добавить новые. Поскольку содержимое `frozenset` должно всегда оставаться статичным, перечень функций, с которыми такое множество может взаимодействовать, имеет ограничения.

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. Запятая в кавычках выступает в качестве символа, разделяющего значения. Метод `type` возвращает тип данных объекта в конце приведенного кода.

Чтобы получить из множества словарь, следует передать функции `dict` набор из нескольких пар значений, в каждом из которых будет находиться ключ. Функция `print` демонстрирует на экране содержимое полученного объекта, а `type` отображает его тип.

По аналогии с предыдущими преобразованиями можно получить список неких объектов. На этот раз используется вызов `list`, получающий в качестве аргумента множество a. На выходе функции `print` отображаются уникальные значения для изначального набора чисел.

Вывод: в результате выполнения практического задания были приобретены навыки по работе с множествами при написании программ с помощью языка программирования Python версии 3.x.