

**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

Дисциплина: «Программирование на Python»

Тема: «Работа с множествами в языке Python»

Выполнил: студент 2 курса

группы ИВТ-б-о-21-1

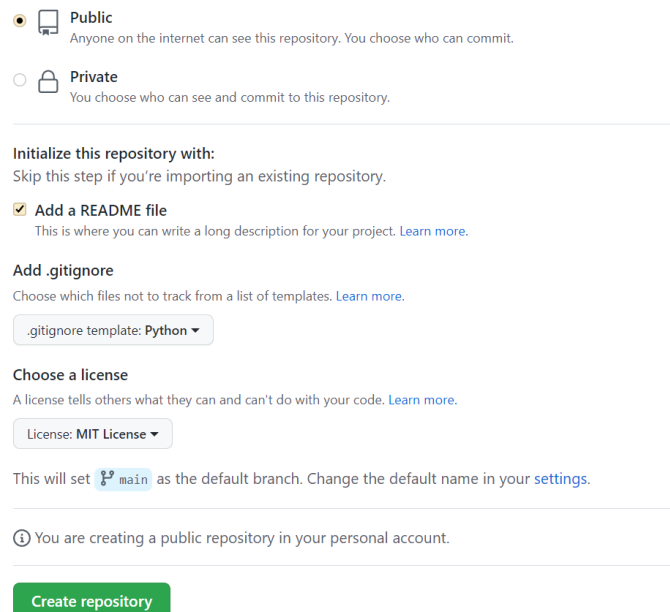
Уланбекова Айканыш Уланбековна

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. Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и язык программирования Python.



☒ **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 public repository in your personal account.

Create repository

Рисунок 1. Создание репозитория

2. Выполните клонирование созданного репозитория.

```
C:\Users\User>cd C:\Users\User\Desktop\2 кypc Python\lab 10
C:\Users\User\Desktop\2 кypc Python\lab 10>git clone https://github.com/aikanyshkauanbekova/lab-10.git
Cloning into 'lab-10'...
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), 4.50 KiB | 1.50 MiB/s, done.
Resolving deltas: 100% (1/1), done.
C:\Users\User\Desktop\2 кypc Python\lab 10>
```

Рисунок 2. Клонирование репозитория

3. Дополните файл .gitignore необходимыми правилами для работы с IDE PyCharm.

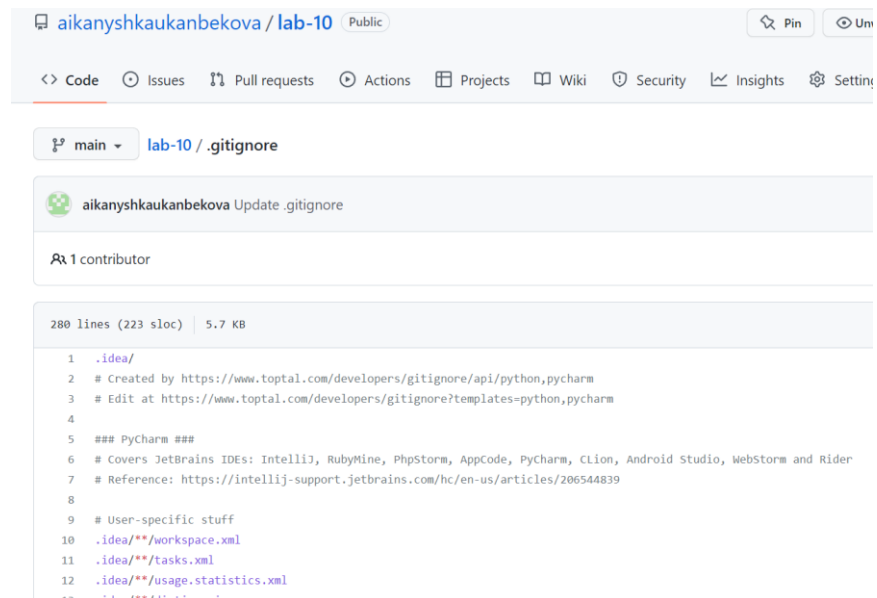


Рисунок 3. Дополнение файла .gitignore

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

```
C:\Users\User\Desktop\2 кypc Python\lab 10\lab-10>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/User/Desktop/2 кypc Python/lab 10/lab-10/.git/hooks]
C:\Users\User\Desktop\2 кypc Python\lab 10\lab-10>
```

Рисунок 4. Организован модель ветвления git flow

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

```

1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 ▶ if __name__ == "__main__":
5     # Определим универсальное множество
6     u = set("abcdefghijklmnopqrstuvwxyz")
7     a = {"b", "c", "h", "o"}
8     b = {"d", "f", "g", "o", "v", "y"}
9     c = {"d", "e", "j", "k"}
10    d = {"a", "b", "f", "g"}
11    x = (a.intersection(b)).union(c)
12    print(f"x = {x}")
13    # Найдем дополнения множеств
14    bn = u.difference(b)
15    cn = u.difference(c)
16    y = (a.difference(d)).union(cn.difference(bn))
17    print(f"y = {y}")

```

Рисунок 5. Пример лаб работы

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

```

▶ #!/usr/bin/env python3
# -*- coding: utf-8 -*-

▶ if __name__ == "__main__":
    word = str(input('Введите слово --> ')).lower()
    count = 0
    vowels = set("аоэеиыуёюя")
    for letter in word:
        if letter in vowels:
            count += 1
    print(f"Количество гласных букв в слове {word} --> {count}")

```

Рисунок 7. Выполненное задание 1

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

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-

if __name__ == "__main__":
    str1 = str(input('Введите что-нибудь --> ')).lower()
    str2 = str(input('Введите что-нибудь ещё раз --> ')).lower()

    common_letters = set(str1) & set(str2)

    print(common_letters)

```

Рисунок 8. Выполненное задание 2

Индивидуальное задание

9. $A = \{a, e, f, i\}; \quad B = \{a, b, k, n\}; \quad C = \{e, f, n, o, w, x\}; \quad D = \{a, d, e, o, p, t, u\};$
 $X = (A \cup B) \cap D; \quad Y = (\bar{A} \cap \bar{B}) / (C \cup D).$

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 if __name__ == "__main__":
5     u = set("abcdefghijklmnopqrstuvwxyz")
6     A = {"a", "e", "f", "i"}
7     B = {"a", "b", "k", "n"}
8     C = {"e", "f", "n", "o", "w", "x"}
9     D = {"a", "d", "e", "o", "p", "t", "u"}
10
11     x = (A.union(B)).intersection(D)
12     print(f"x = {x}")
13
14     An = u.difference(A)
15     Bn = u.difference(B)
16
17     y = (An.intersection(Bn)).difference(C.union(D))
18     print(f"y = {y}")
```

Рисунки 9. Выполненное индивидуальное задание

8. Сделала коммит, выполнил слияние с веткой main, и запустил изменения в уд. репозиторий.

```
C:\Users\User>cd C:\Users\User\Desktop\2 кypc Python\lab 10\lab-10
C:\Users\User\Desktop\2 кypc Python\lab 10\lab-10>git add .
C:\Users\User\Desktop\2 кypc Python\lab 10\lab-10>git commit -m "new"
[main a64afda] new
4 files changed, 56 insertions(+)
create mode 100644 indiv.py
create mode 100644 primer 1.py
create mode 100644 zadanie 1.py
create mode 100644 zadanie 2.py
C:\Users\User\Desktop\2 кypc Python\lab 10\lab-10>git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 1.52 KiB | 777.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/aikanyshkaukanbekova/lab-10.git
6f06a3b..a64afda main -> main
C:\Users\User\Desktop\2 кypc Python\lab 10\lab-10>_
```

Рисунок 10. Сохранения

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?

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?

При помощи команд `in/not in` соответственно.

4. Как выполнить перебор элементов множества?

```
For I in {set} Print(i)
```

5. Что такое set comprehension?

Генератор, позволяющий заполнять списки, а также другие наборы с учётом некоторых условий.

6. Как выполнить добавление элемента во множество?

При помощи тега `add`.

7. Как выполнить удаление одного или всех элементов множества?

Есть 3 тега одиночного удаления элемента: `remove`, `pop`, `discard`. Чтобы очистить множество полностью используют команду `clear`.

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

Объединение при помощи тега `union` или знака `|`. Пересечение при помощи тега `intersection` или знака `&`. Разность при помощи тега `difference` или знака `-`.

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

Определение подмножества: тег `issubset`. Определение надмножества: тег `issuperset`.

10. Каково назначение множеств frozenset?

Значения из этого набора нельзя удалить, как и добавить новые.

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

В строку при помощи тега `join`; в список при помощи тега `list`; а в словарь при помощи тега `dict`.