

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ**

**ФЕДЕРАЦИИ**

**Федеральное государственное автономное  
образовательное учреждение высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций**

**«Работа с кортежами в языке Python»**

**Отчет по лабораторной работе № 2.7**

**по дисциплине «Основы программной инженерии»**

Выполнил студент группы ПИЖ-б-о-21-1

Кучеренко С. Ю. « » 2022г.

Подпись студента \_\_\_\_\_

Работа защищена « » \_\_\_\_\_ 2022г.

Проверил Воронкин Р.А. \_\_\_\_\_

(подпись)

Ставрополь 2022

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

**Выполнение работы:**

1. Изучить теоретический материал работы.
2. Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и язык программирования Python.

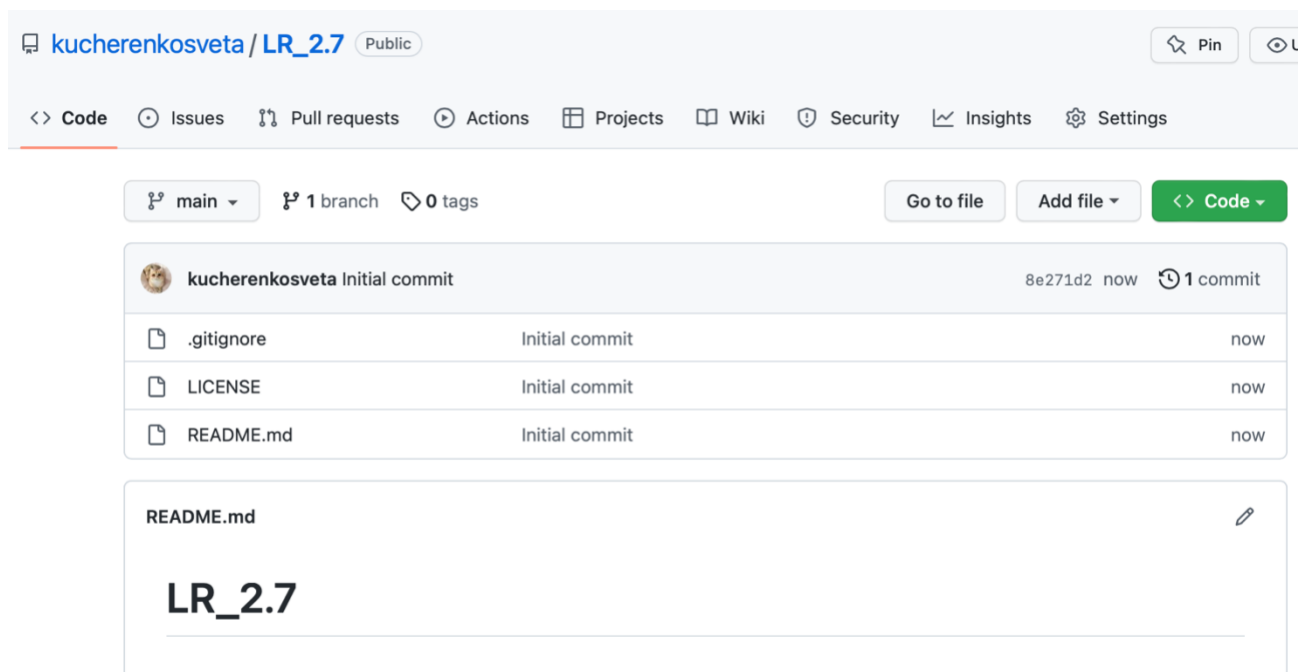


Рисунок 1 – Создание репозитория

3. Выполните клонирование созданного репозитория.

```
[(base) svetik@MacBook-Air-Svetik Laba10 % git clone https://github.com/kucherenk]
osveta/LR_2.7.git
Cloning into 'LR_2.7'...
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.
(base) svetik@MacBook-Air-Svetik Laba10 %
```

Рисунок 2 – Клонирование репозитория

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

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

```
[(base) svetik@MacBook-Air-Svetik LR_2.7 % 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? [/Users/svetik/Desktop/Laba10/LR_2.7/.git/hooks]
[(base) svetik@MacBook-Air-Svetik LR_2.7 % _
```

Рисунок 3 – Организация репозитория в соответствии с моделью git-flow

## 6. Создайте проект PyCharm в папке репозитория.

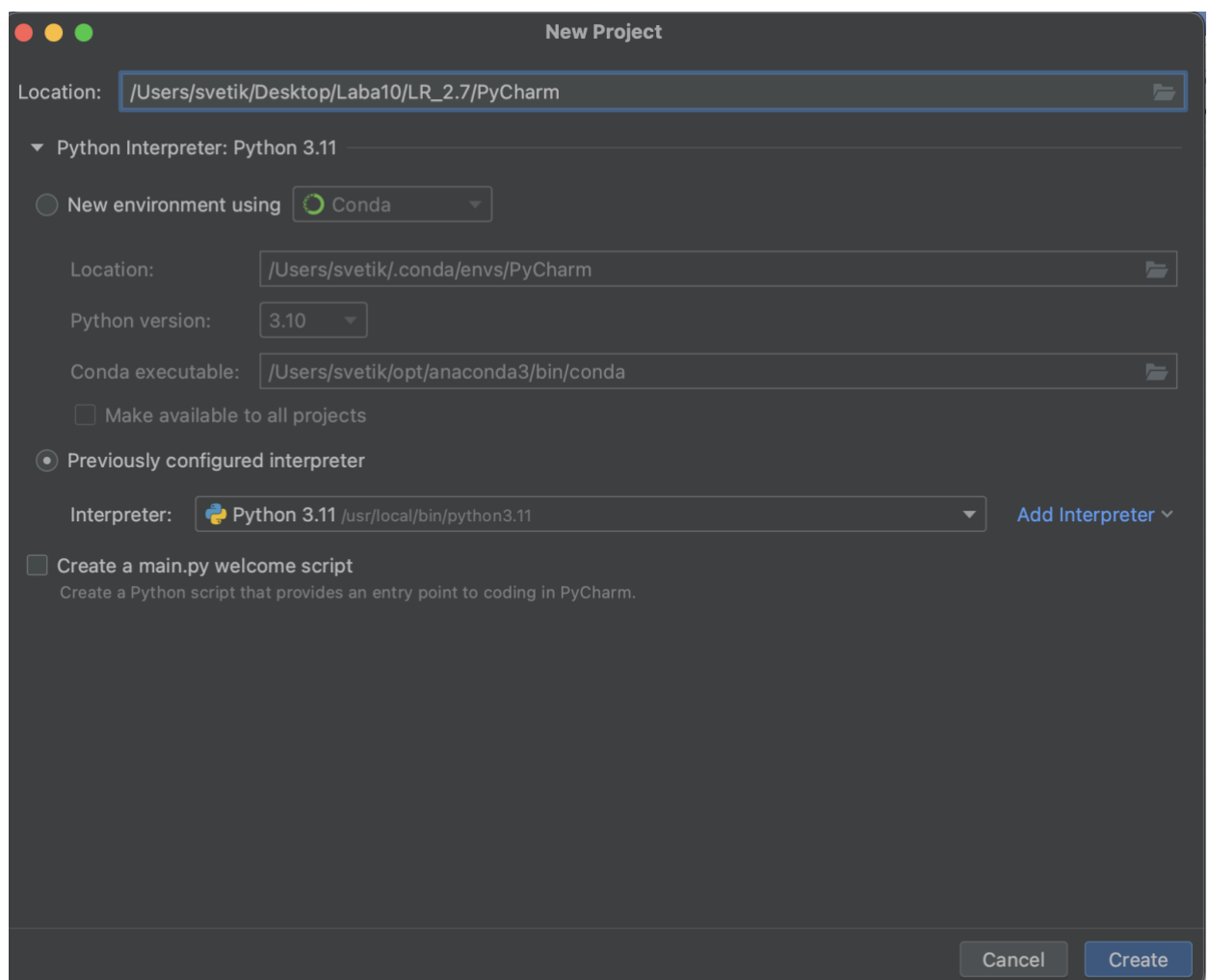


Рисунок 4 – Создание проекта PyCharm в папке репозитория

7. Проработайте примеры лабораторной работы. Создайте для каждого примера отдельный модуль языка Python. Зафиксируйте изменения в репозитории.

Пример 1. Определить результат выполнения операций над множествами. Считать элементы множества строками.

Пример 1. Определить результат выполнения операций над множествами. Считать элементы множества строками.

$$A = \{b, c, h, o\}; \quad B = \{d, f, g, o, v, y\}; \quad C = \{d, e, j, k\}; \quad D = \{a, b, f, g\}; \quad X = (A \cap B) \cup C; \quad Y = (A/D) \cup (\bar{C}/\bar{B}). \quad (1)$$

Примечание: в качестве универсального множества считать все строчные латинские буквы от a до z.

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

if __name__ == "__main__":
    # Определим универсальное множество
    u = set("abcdefghijklmnopqrstuvwxyz")

    a = {"b", "c", "h", "o"}
    b = {"d", "f", "g", "o", "v", "y"}
    c = {"d", "e", "j", "k"}
    d = {"a", "b", "f", "g"}

    x = (a.intersection(b)).union(c)
    print(f"x = {x}")

    # Найдем дополнения множеств
    bn = u.difference(b)
    cn = u.difference(c)

    y = (a.difference(d)).union(cn.difference(bn))
    print(f"y = {y}")
```

```
/usr/local/bin/python3.11 /Users/svetik/...
x = {'o', 'e', 'd', 'k', 'j'}
y = {'o', 'f', 'c', 'h', 'v', 'y', 'g'}

Process finished with exit code 0
```

Рисунок 5 – Результат работы программы

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

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

```

if __name__ == "__main__":
    s = input("Enter line: ")
    vowels = {'e', 'y', 'u', 'i', 'o', 'a'}
    n = 0
    for i in s:
        if i in vowels:
            n += 1
    print(f"Number of vowels per line: {n}")

```

10.

```

/usr/local/bin/python3.11 /Users/sv
Enter line: eguhi]
Number of vowels per line: 3

Process finished with exit code 0

```

Рисунок 6 – Результат работы программы

11. Зафиксируйте сделанные изменения в репозитории.

```

[(base) svetik@MacBook-Air-Svetik LR_2.7 % git add .
[(base) svetik@MacBook-Air-Svetik LR_2.7 % git status
On branch develop
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   PyCharm/.idea/.gitignore
    new file:   PyCharm/.idea/PyCharm.iml
    new file:   PyCharm/.idea/inspectionProfiles/profiles_settings.xml
    new file:   PyCharm/.idea/misc.xml
    new file:   PyCharm/.idea/modules.xml
    new file:   PyCharm/.idea/vcs.xml
    new file:   PyCharm/ind1.py
    new file:   PyCharm/pr1.py

```

Рисунок 7 – Фиксация изменений в репозитории

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

```

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

if __name__ == "__main__":
    s1 = set(input("Enter line №1: "))
    s2 = set(input("Enter line №2: "))
    line = ', '.join(s1.intersection(s2))
    print(f"common characters in two lines: {line}")

```

14.

```

Enter line №1: Hello
Enter line №2: World
common characters in two lines: l, o

Process finished with exit code 0

```

Рисунок 8 – Результат работы программы

15. Зафиксируйте сделанные изменения в репозитории.

```

(base) svetik@MacBook-Air-Svetik LR_2.7 % git add .
(base) svetik@MacBook-Air-Svetik LR_2.7 % git commit -m '/'
[develop 1830c7e] /
 9 files changed, 75 insertions(+)
 create mode 100644 PyCharm/.idea/.gitignore
 create mode 100644 PyCharm/.idea/PyCharm.iml
 create mode 100644 PyCharm/.idea/inspectionProfiles/profiles_settings.xml
 create mode 100644 PyCharm/.idea/misc.xml
 create mode 100644 PyCharm/.idea/modules.xml
 create mode 100644 PyCharm/.idea/vcs.xml
 create mode 100644 PyCharm/ind1.py
 create mode 100644 PyCharm/ind2.py
 create mode 100644 PyCharm/pr1.py
(base) svetik@MacBook-Air-Svetik LR_2.7 %

```

Рисунок 9 – Фиксация изменений в репозитории

16. Выполните индивидуальные задания, согласно своему варианту.

Вариант 16.

$$A = \{b, d, f, g, l, u\}; \quad B = \{d, e, f, m, n, z\}; \quad C = \{h, i, r, x, y\}; \quad D = \{a, e, f, k, r, s, x\};$$

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

```

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

if __name__ == '__main__':

    a = {'b', 'd', 'f', 'g', 'l', 'u'}
    b = {'d', 'e', 'f', 'm', 'n', 'z'}
    c = {'h', 'i', 'r', 'x', 'y'}
    d = {'a', 'e', 'f', 'k', 'r', 's', 'x'}
    u = set("abcdefghijklmnopqrstuvwxyz")
    n_a = u.difference(a)
    print('Множество A = ', a)
    print('Множество B = ', b)
    print('Множество C = ', c)

```

```

print('Множество D = ', d)

x = (a.difference(b)).intersection(c.union(d))
print('\nМножество X = ', x)

y = (n_a.intersection(d)).union(c.difference(b))
print('Множество Y = ', y)

```

```

/usr/local/bin/python3.11 /Users/svetik/Desktop/Laba10/LR_2.7/
Множество A = {'u', 'b', 'l', 'd', 'f', 'g'}
Множество B = {'m', 'd', 'f', 'z', 'e', 'n'}
Множество C = {'h', 'y', 'x', 'r', 'i'}
Множество D = {'x', 'f', 'e', 'r', 'k', 'a', 's'}

Множество X = set()
Множество Y = {'y', 'k', 'i', 'h', 'x', 'e', 'r', 'a', 's'}

```

Рисунок 10 – Результат работы программы

## Вопросы для защиты работы:

### 1. Что такое множества в языке Python?

Множеством в языке программирования Python называется неупорядоченная совокупность уникальных значений. В качестве элементов этого набора данных могут выступать любые неизменяемые объекты, такие как числа, символы, строки.

### 2. Как осуществляется создание множеств в Python?

Присвоить переменной последовательность значений, выделив их фигурными скобками:  $a = \{1, 3, 5, 9\}$  Вызвать set:  $a = \text{set}(\text{'stroka'})$

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

Для этого можно воспользоваться «in» или «not in»:

```

a = {1, 3, 5, 9} print(2
in a) // false

```

#### **4. Как выполнить перебор элементов множества?**

С помощью цикла for:

```
for i in {1, 2, 3}    print(i) //1
2 3
```

#### **5. Что такое set comprehension?**

Это генератор, позволяющий заполнять списки, а также другие наборы данных с учетом неких условий, например:

```
a = {i for i in [1, 2, 3, 1, 2, 3, 0]} print(a)
// {0, 1, 2, 3}
```

#### **6. Как выполнить добавление элемента во множество?**

Необходимо использовать метод add. Например: a =

```
{1, 3, 5, 9} a.add(6)
```

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

Для удаления элемента можно воспользоваться несколькими функциями:

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.

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

```
a = {1, 3, 5, 7} b
= {1, 4, 6, 7, 8}
```

Объединение: c = a.union(b)

Пересечение: c = a.intersection(b)

Разность: c = a.difference(b)

Все команды возвращают множества



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

Для этого можно воспользоваться следующими командами (возвращают true/false):

a.issubset(b) – подмножество

a.issuperset(b) – надмножество

**10. Каково назначение множеств frozenset?**

Это множества, которые нельзя изменить (ни удалить, ни 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 (например, элементы множества – кортежи из двух элементов).

Для получения списка используется вызов list, получающий в качестве аргумента множество a.