

**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 .

Гребенкин Е. А. « » 2022г.

Подпись студента _____

Работа защищена « » _____ 2022г.

Проверил Воронкин Р.А. _____

(подпись)

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.
3. Дополните файл .gitignore необходимыми правилами для работы с IDE PyCharm.
4. Организуйте свой репозиторий в соответствие с моделью ветвления git-flow.
5. Создайте проект PyCharm в папке репозитория.
6. 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}")
```

```
C:\Users\dimu7\AppData\Local\Programs\Py
x = {'o', 'd', 'j', 'k', 'e'}
y = {'y', 'v', 'g', 'h', 'c', 'o', 'f'}
```

Рисунок 5 – Результат работы программы

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

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

if __name__ == "__main__":
    s = input('Введите строку: ').lower()
    vowels = {"a", "o", "y", "ы", "э", "е", "ё", "и", "ю",
              "я"}

    count = 0
    for i in s:
        if i in vowels:
            count += 1

    print(f"Количество гласный в введенной строке = {count}")
```

```
C:\Users\dimu7\AppData\Local\Progra
Введите строку: Привет как дела
5
```

Рисунок 6 – Результат работы программы

8. 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

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

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
if __name__ ==
"__main__":
    line1 =
    set(input())
    line2 =
    set(input())

    print(line1.intersection(line2))
```

```
Hello
World
{'o', 'l'}
```

Рисунок 8 – Результат работы программы

10. 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

11. 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)
    a) print('Множество B = ', b)
    b) print('Множество C = ', c)
    c) print('Множество D = ', 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)

```

```

C:\Users\dimu7\AppData\Local\Programs\Python\Python311\python.exe C:\U
Множество A = {'d', 'g', 'l', 'b', 'f', 'u'}
Множество B = {'d', 'n', 'z', 'e', 'f', 'm'}
Множество C = {'h', 'i', 'y', 'r', 'x'}
Множество D = {'k', 'a', 'x', 'e', 'f', 'r', 's'}

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

Process finished with exit code 0

```

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 = set('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
```

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`.