

**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: «Основы программной инженерии»
Тема: «Работа с множествами в языке Python»**

**Выполнил:
студент 2 курса
группы Пиж-б-о-21-1
Логвинов Иван
Васильевич**

Ставрополь 2022

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

1. Был создан репозиторий в Github в который были добавлены правила gitignore для работы IDE PyCharm, была выбрана лицензия MIT, сам репозиторий был клонирован на локальный сервер и был организован в соответствии с моделью ветвления git-flow.

Рисунок 1 – Создание репозитория

```

#!/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\Иван\AppData\Local\Programs\Python
x = {'e', 'k', 'j', 'o', 'd'}
y = {'h', 'v', 'c', 'g', 'o', 'f', 'y'}

```

Рисунок 5 – Результат работы программы

3. Выполнил задания.

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

Код программы:

```

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

if __name__ == "__main__":
    # Все согл
    a = set("bcdfghilmnpqrstvwxyz")
    mark = set(".,!,:;'\`\" ")

    # Ввод строки

```

```

x = set(input("Введите строку: ").lower())

# Находим все гласные в строке
gl = x.difference(a)
gl = gl.difference(mark)

count = len(gl)

print(f"Все гласные буквы из введенной строки: = {gl}")
print(f"Кол-во гласных букв: = {count}")

```

```

C:\Users\Иван\AppData\Local\Programs\Python\Python39\pyt
Введите строку: regeydfhj
Все гласные буквы из введенной строки: = {'y', 'j', 'e'}
Кол-во гласных букв: = 3

```

Рисунок 6 – Результат работы программы

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

Код программы:

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-
if __name__ == "__main__":
    # Ввод строки
    x1 = set(input("Введите первую строку: ").lower())
    x2 = set(input("Введите вторую строку: ").lower())

    mark = set(".?;!,:;'\`\" ")

    gl = x1.intersection(x2)
    gl = gl.difference(mark)

    count = len(gl)

    print(f"Общие символы: = {gl}")
    print(f"Кол-во общих символов: = {count}")

```

```

C:\Users\Иван\AppData\Local\Programs\Python\Pyt
Введите первую строку: i go home
Введите вторую строку: i want to buy home
Общие символы: = {'i', 'e', 'h', 'o', 'm'}
Кол-во общих символов: = 5

```

Рисунок 7 – Результат работы программы

Индивидуальное задание

$$A = \{c, g, h, k, y\}; \quad B = \{a, b, k, n, u\}; \quad C = \{i, j, o, y, z\}; \quad D = \{a, b, f, g, y, z\};$$

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

$$A \cup B = \{c, g, n, b, u, a, k, y, h\}$$

$$X = (A \cup B) \cap D = \{a, b, g, y\}$$

$$\bar{A} = \{t, p, n, a, x, w, e, l, s, q, d, b, m, z, v, o, f, r, i, j, u\}$$

$$\bar{B} = \{t, p, x, w, y, e, g, l, h, s, q, d, m, z, v, o, c, f, r, i, j\}$$

$$\bar{C} = \{t, p, n, a, x, w, e, g, l, h, s, k, q, d, b, m, v, c, f, r, u\}$$

$$\bar{A} \cap B = \{a, b, f, z\}$$

$$\bar{C} / \bar{B} = \{n, b, k, u, a\}$$

$$Y = (\bar{A} \cap D) \cup (\bar{C} / \bar{B}) = \{n, b, k, u, a, f, z\}$$

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

if __name__ == "__main__":

    # Universe
    u = set("abcdefghijklmnopqrstuvwxyz")

    # Set Data
    a = {"c", "g", "h", "k", "y"}
    b = {"a", "b", "k", "n", "u"}
    c = {"i", "j", "o", "y", "z"}
    d = {"a", "b", "f", "g", "y", "z"}

    # Definition X
    x = d.intersection(a.union(b))
    print(f'X = {x}')

    # Inverses for a b and c
    ne_a = u.difference(a)
    ne_b = u.difference(b)
    ne_c = u.difference(c)

    # Definition Y
    y = (ne_a.intersection(d)).union(ne_c.difference(ne_b))
    print(f'Y = {y}')
```

```
C:\Users\Иван\AppData\Local\Programs\Python
X = {'y', 'a', 'g', 'b'}
Y = {'n', 'k', 'u', 'a', 'z', 'b', 'f'}

Process finished with exit code 0
```

Рисунок 8 – Результат работы программы

Контрольные вопросы:

1. Что такое словари в языке Python?

Словари в Python – это изменяемые отображения ссылок на объекты, доступные по ключу.

2. Может ли функция len() быть использована при работе со словарями?

Функция len() возвращает длину (количество элементов) в объекте. Аргумент может быть последовательностью, такой как строка, байты, кортеж, список или диапазон или коллекцией (такой как словарь, множество или неизменяемое множество).

3. Какие методы обхода словарей Вам известны?

Самый очевидный вариант обхода словаря — это попытаться напрямую запустить цикл for по объекту словаря, так же как мы делаем это со списками, кортежами, строками и любыми другими итерируемыми объектами. for something in currencies: print(something)

4. Какими способами можно получить значения из словаря по ключу?

С помощью метода .get()

5. Какими способами можно установить значение в словаре по ключу?

С помощью функции dict.update()

6. Что такое словарь включений?

Словарь включений аналогичен списковым включениям, за исключением того, что он создаёт объект словаря вместо списка.

7. Самостоятельно изучите возможности функции zip() приведите

примеры ее использования.

Функция `zip()` в Python создает итератор, который объединяет элементы из нескольких источников данных. Эта функция работает со списками, кортежами, множествами и словарями для создания списков или кортежей, включающих все эти данные. Предположим, что есть список имен и номером сотрудников, и их нужно объединить в массив кортежей. Для этого можно использовать функцию `zip()`. Вот пример программы, которая делает именно это:

```
employee_numbers = [2, 9, 18, 28]
employee_names = ["Дима", "Марина", "Андрей", "Никита"]
zipped_values = zip(employee_names, employee_numbers)
zipped_list = list(zipped_values)
print(zipped_list)
```

Функция `zip` возвращает следующее:

```
[('Дима', 2), ('Марина', 9), ('Андрей', 18), ('Никита', 28)]
```

8. Самостоятельно изучите возможности модуля `datetime`. Каким функционалом по работе с датой и временем обладает этот модуль? `Datetime` — важный элемент любой программы, написанной на Python. Этот модуль позволяет управлять датами и временем, представляя их в таком виде, в котором пользователи смогут их понимать.

`datetime` включает различные компоненты. Так, он состоит из объектов следующих типов:

- ☐ `date` — хранит дату
- ☐ `time` — хранит время
- ☐ `datetime` — хранит дату и время

Как получить текущие дату и время?

```
import datetime
dt_now = datetime.datetime.now()
```

```
print(dt_now)
```

Результат:

2022-09-11 15:43:32.249588

Получить текущую дату:

```
from datetime import date
```

```
current_date = date.today()
```

```
print(current_date)
```

Результат:

2022-09-11

Получить текущее время:

```
import datetime
```

```
current_date_time = datetime.datetime.now()
```

```
current_time = current_date_time.time()
```

```
print(current_time)
```

Результат:

15:51:05.627643