

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития  
Кафедра инфокоммуникаций

**ОТЧЕТ**  
**ПО ЛАБОРАТОРНОЙ РАБОТЕ №10**  
**дисциплины «Программирование на Python»**

Выполнил:  
Данилецкий Дмитрий Витальевич  
2 курс, группа ИВТ-б-о-22-1,  
09.03.01 «Информатика и  
вычислительная техника»,  
направленность (профиль)  
«Программное обеспечение средств  
вычислительной техники и  
автоматизированных систем», очная  
форма обучения

---

(подпись)

Руководитель практики:  
Воронкин Р. А., канд. технических  
наук, доцент кафедры  
инфокоммуникаций

---

(подпись)

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

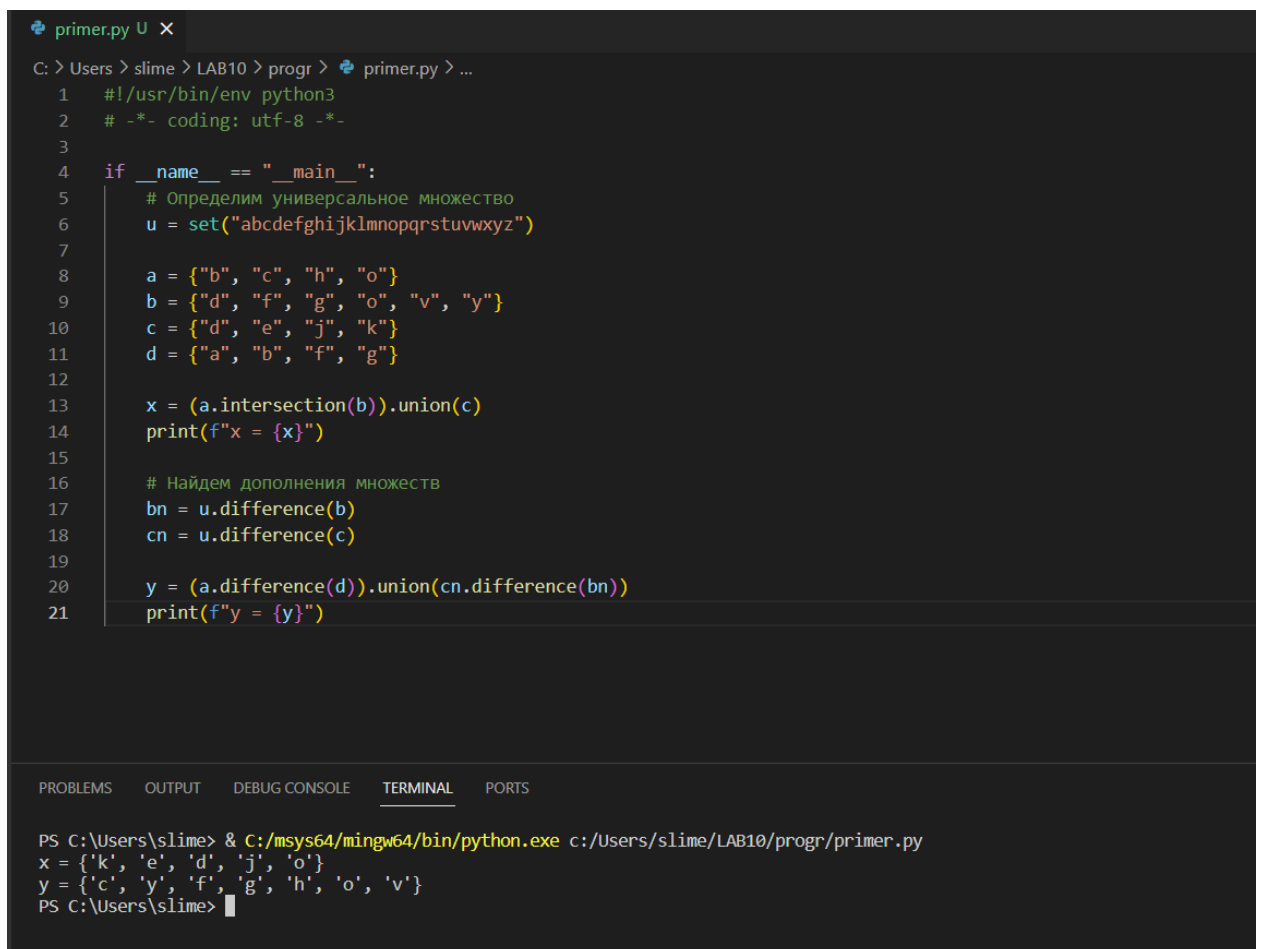
Ставрополь, 2023 г.

## Тема: Работа с множествами в языке Python

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

### Ход работы

1. Создал общедоступный репозиторий на GitHub, в котором использована лицензия MIT и язык программирования Python. Выполнил клонирование созданного репозитория.
2. Дополнил файл .gitignore необходимыми правилами.
3. Организовал созданный репозиторий в соответствие с моделью ветвления git-flow.
4. Проработал пример лабораторной работы. Создал для него отдельный модуль языка Python. Привел в отчете скриншоты результата выполнения программы примера.



```
primer.py U X
C: > Users > slime > LAB10 > progr > primer.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  if __name__ == "__main__":
5      # Определим универсальное множество
6      u = set("abcdefghijklmnopqrstuvwxyz")
7
8      a = {"b", "c", "h", "o"}
9      b = {"d", "f", "g", "o", "v", "y"}
10     c = {"d", "e", "j", "k"}
11     d = {"a", "b", "f", "g"}
12
13     x = (a.intersection(b)).union(c)
14     print(f"x = {x}")
15
16     # Найдем дополнения множеств
17     bn = u.difference(b)
18     cn = u.difference(c)
19
20     y = (a.difference(d)).union(cn.difference(bn))
21     print(f"y = {y}")

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\Users\slime> & C:/msys64/mingw64/bin/python.exe c:/Users/slime/LAB10/progr/primer.py
x = {'k', 'e', 'd', 'j', 'o'}
y = {'c', 'y', 'f', 'g', 'h', 'o', 'v'}
PS C:\Users\slime>
```

Рисунок 1. Результат работы программы из примера 1

5. Решил следующую задачу: подсчитайте количество гласных в строке, введенной с клавиатуры с использованием множеств.

```
C: > Users > slime > LAB10 > progr > zadacha1.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  if __name__ == "__main__":
5      vowels = set("aeiouyаеёиоуеюя")
6
7      text = input("Введите предложение: ")
8
9      count = sum(1 for char in text.lower() if char in vowels)
10
11     print("Количество гласных в строке:", count)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\slime> & C:/msys64/mingw64/bin/python.exe c:/Users/slime/LAB10/progr/zadacha1.py
Введите предложение: Работа с множествами
Количество гласных в строке: 7
PS C:\Users\slime> & C:/msys64/mingw64/bin/python.exe c:/Users/slime/LAB10/progr/zadacha1.py
Введите предложение: Bad weather today
Количество гласных в строке: 7
PS C:\Users\slime> & C:/msys64/mingw64/bin/python.exe c:/Users/slime/LAB10/progr/zadacha1.py
Введите предложение: Hello world
Количество гласных в строке: 3
PS C:\Users\slime>
```

Рисунок 2. Результат работы программы из задачи 1

6. Решил следующую задачу: определите общие символы в двух строках, введенных с клавиатуры.

```
C: > Users > slime > LAB10 > progr > zadanie2.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  if __name__ == "__main__":
5      text1 = input("Введите первое предложение: ")
6      text2 = input("Введите второе предложение: ")
7
8      plenty1 = set(text1)
9      plenty2 = set(text2)
10
11     shared_sym = plenty1.intersection(plenty2)
12
13     print("Общие символы:", shared_sym)
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

```
PS C:\Users\slime> & C:/msys64/mingw64/bin/python.exe c:/Users/slime/LAB10/progr/zadanie2.py
Введите первое предложение: Сковородка
Введите второе предложение: куртка
Общие символы: {'a', 'p', 'к'}
PS C:\Users\slime> & C:/msys64/mingw64/bin/python.exe c:/Users/slime/LAB10/progr/zadanie2.py
Введите первое предложение: I wanna eat bread
Введите второе предложение: apple on the table
Общие символы: {'e', 'n', 'b', 'a', 't', ' '}
PS C:\Users\slime> █
```

Рисунок 3. Результат работы программы из задачи 2

7. Выполнил индивидуальное задание, согласно варианту 8. Привёл в отчете скриншот работы программы. Задание: определить результат выполнения операций над множествами. Считать элементы множества строками, проверить результаты вручную.

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

```
C: > Users > slime > LAB10 > progr > ind.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  if __name__ == "__main__":
5      u = set("abcdefghijklmnopqrstuvwxyz")
6
7      a = {"a", "f", "i", "n", "o"}
8      b = {"f", "g", "o", "p", "z"}
9      c = {"i", "j", "u", "w"}
10     d = {"f", "h", "n", "t", "u", "y", "z"}
11
12     x = (a.intersection(b)).union(c)
13     print(f"x = {x}")
14
15     an = u.difference(a)
16     bn = u.difference(b)
17
18     y = (an.intersection(bn)).difference(c.union(d))
19     print(f"y = {y}")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\slime> & C:/msys64/mingw64/bin/python.exe c:/Users/slme/LAB10/progr/ind.py
x = {'w', 'o', 'j', 'u', 'f', 'i'}
y = {'d', 'q', 'x', 'm', 'b', 'r', 'v', 'k', 'e', 's', 'l', 'c'}
PS C:\Users\slime>
```

Рисунок 4. Результат работы программы из индивидуального задания

#### 8. Проверка работы программы:

$$X = (A \cap B) \cup C$$

$$A \cap B = \{f, o\}$$

$$(A \cap B) \cup C = \{f, o, i, j, u, w\}$$

$$Y = (\overline{A} \cap \overline{B}) \setminus (C \cup D)$$

$$\overline{A} = \{b, c, d, e, g, h, j, k, l, m, p, q, r, s, t, u, v, w, x, y, z\}$$

$$\overline{B} = \{a, b, c, d, e, i, h, j, k, l, n, m, q, r, s, t, u, v, w, x, y\}$$

$$(\overline{A} \cap \overline{B}) = \{b, c, d, e, h, j, k, l, m, q, r, s, t, u, v, w, x, y\}$$

$$(C \cup D) = \{i, j, u, w, f, h, n, t, y, z\}$$

$$(\overline{A} \cap \overline{B}) \setminus (C \cup D) = \{b, c, d, e, k, l, m, q, r, s, v, x\}$$

## Контрольные вопросы

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

В языке Python множество (set) – это неупорядоченная коллекция уникальных элементов. Множества в Python поддерживают операции объединения, пересечения, разности и другие операции, свойственные математическим множествам.

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

В Python множества можно создать с помощью фигурных скобок {} или с помощью функции set(). Множества могут содержать только уникальные элементы, поэтому дублирующиеся элементы будут автоматически удалены при создании множества.

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

В Python можно проверить присутствие или отсутствие элемента в множестве с помощью операторов in и not in.

### 4. Как выполнить перебор элементов множества?

В Python можно выполнить перебор элементов множества с помощью цикла for.

### 5. Что такое set comprehension?

Set comprehension (генератор множества) — это способ создания множества на основе итерации и условий. Он аналогичен list comprehension и использует синтаксис с фигурными скобками {}.

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

В Python можно выполнить добавление элемента во множество с помощью метода add().

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

В Python можно выполнить удаление одного элемента из множества с помощью метода remove() или discard(). Например:

```
my_set.remove(2) # Удаление элемента 2
```

```
my_set.discard(4) # Попытка удаления элемента 4 (если элемента нет, то ничего не происходит)
```

Чтобы удалить все элементы из множества, можно использовать метод `clear()`. Например:

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

В Python основные операции над множествами выполняются с помощью соответствующих методов или операторов: Объединение множеств: можно использовать метод `union()` или оператор `|`. Пересечение множеств: можно использовать метод `intersection()` или оператор `&`. Разность множеств: можно использовать метод `difference()` или оператор `-`. Например:

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

В Python можно определить, что некоторое множество является надмножеством или подмножеством другого множества с помощью методов `issuperset()` и `issubset()`.

10. Каково назначение множеств `frozenset`?

Множество `frozenset` в Python является неизменяемым (`immutable`) аналогом обычного множества `set`. Это означает, что после создания `frozenset` нельзя изменять его содержимое, добавлять или удалять элементы. `frozenset` полезен, когда требуется использовать множество в качестве ключа словаря или элемента другого множества, так как `frozenset` является хешируемым (`hashable`), в отличие от обычного множества `set`.

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

В Python можно осуществить преобразование множества в строку, список и словарь с помощью соответствующих функций и методов. Преобразование в строку: Можно использовать функцию `str()` или метод `join()` для преобразования множества в строку. Преобразование в список: Можно использовать функцию `list()` для преобразования множества в список. Преобразование в словарь: Можно использовать функцию `dict.fromkeys()` для

преобразования множества в словарь, где элементы множества станут ключами словаря.

Вывод: в ходе выполнения работы были приобретены навыки по работе с множествами при написании программ с помощью языка программирования.