

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:
Говоров Егор Юрьевич
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 _____

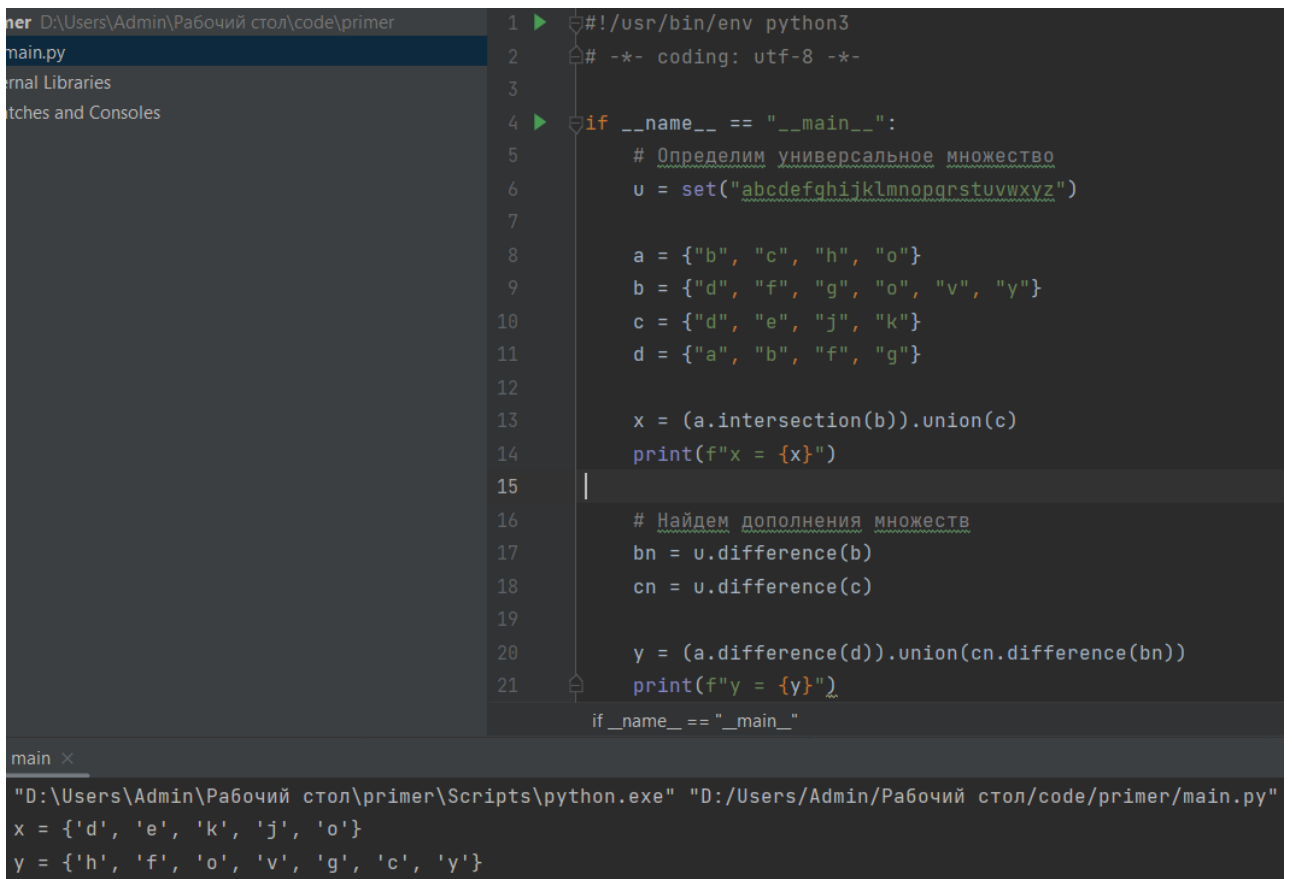
Ставрополь, 2023 г.

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.



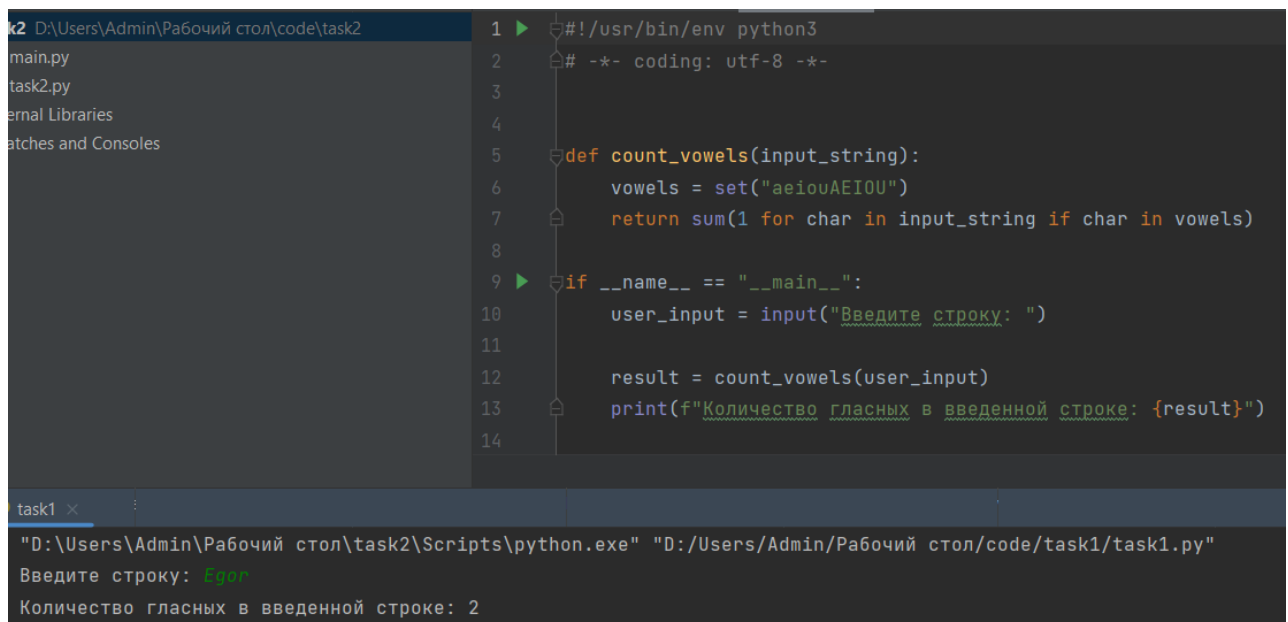
```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  if __name__ == "__main__":
5      # Определим универсальное множество
6      u = set("abcdefghijklmnopqrstuvwxyz")
7
8      a = {"b", "c", "h", "o"}
9      b = {"d", "f", "g", "o", "v", "y"}
10     c = {"d", "e", "j", "k"}
11     d = {"a", "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}")
22
23 if __name__ == "__main__":
```

main x

```
"D:\Users\Admin\Рабочий стол\primer\Scripts\python.exe" "D:/Users/Admin/Рабочий стол/code/primer/main.py"
x = {'d', 'e', 'k', 'j', 'o'}
y = {'h', 'f', 'o', 'v', 'g', 'c', 'y'}
```

Рисунок 1. Результат работы программы из примера 1

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



```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4
5  def count_vowels(input_string):
6      vowels = set("aeiouAEIOU")
7      return sum(1 for char in input_string if char in vowels)
8
9  if __name__ == "__main__":
10     user_input = input("Введите строку: ")
11
12     result = count_vowels(user_input)
13     print(f"Количество гласных в введенной строке: {result}")
14
```

task1 x

"D:\Users\Admin\Рабочий стол\task2\Scripts\python.exe" "D:/Users/Admin/Рабочий стол/code/task1/task1.py"

Введите строку: **Егор**

Количество гласных в введенной строке: 2

Рисунок 2. Результат работы программы из задачи 1

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

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 if __name__ == "__main__":
5     string1 = input("Введите первую строку: ")
6     string2 = input("Введите вторую строку: ")
7
8     set1 = set(string1)
9     set2 = set(string2)
10
11     common_characters = set1.intersection(set2)
12
13     if common_characters:
14         print(f"Общие символы: {' '.join(common_characters)}")
15     else:
16         print("Общих символов нет.")
```

main x

"D:\Users\Admin\Рабочий стол\task2\Scripts\python.exe" "D:/Users/Admin/Рабочий стол/code/task2/main.py"

Введите первую строку: **Егор**

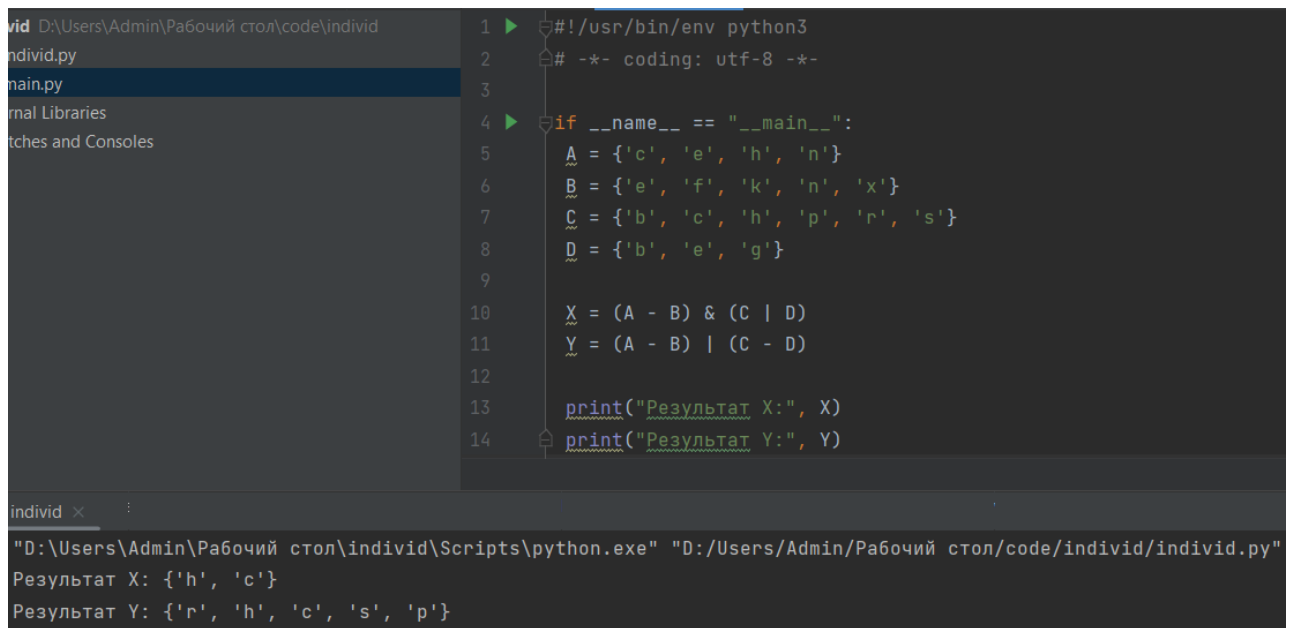
Введите вторую строку: **Егор**

Общие символы: г, р, о, Е

Рисунок 3. Результат работы программы из задачи 2

$$A = \{c, e, h, n\}; \quad B = \{e, f, k, n, x\}; \quad C = \{b, c, h, p, r, s\}; \quad D = \{b, e, g\};$$
$$X = (A/B) \cap (C \cup D); \quad Y = (A \cap \bar{B}) \cup (C/D).$$

7. Выполнил индивидуальное задание, согласно варианту 5. Привёл в отчете скриншот работы программы. Задание: определить результат выполнения операций над множествами. Считать элементы множества строками, проверить результаты вручную.



The screenshot shows a Python IDE with a file named `individ.py`. The script defines four sets A, B, C, and D, and calculates two new sets X and Y using set operations. The output of the script is displayed in the console.

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 if __name__ == "__main__":
5     A = {'c', 'e', 'h', 'n'}
6     B = {'e', 'f', 'k', 'n', 'x'}
7     C = {'b', 'c', 'h', 'p', 'r', 's'}
8     D = {'b', 'e', 'g'}
9
10    X = (A - B) & (C | D)
11    Y = (A - B) | (C - D)
12
13    print("Результат X:", X)
14    print("Результат Y:", Y)
```

Output in the console:

```
"D:\Users\Admin\Рабочий стол\individ\Scripts\python.exe" "D:/Users/Admin/Рабочий стол/code/individ/individ.py"
Результат X: {'h', 'c'}
Результат Y: {'r', 'h', 'c', 's', 'p'}
```

Рисунок 4. Результат работы программы из индивидуального задания

8. Проверка работы программы:

$$A = \{c, e, h, n\}$$

$$B = \{e, f, k, n, x\}$$

$$C = \{b, c, h, p, r, s\}$$

$$D = \{b, e, g\}$$

$$1. A/B = \{c, h\}$$

$$2. C \cup D = \{b, c, e, g, h, p, r, s\}$$

$$3. \neg B = \{a, b, c, d, g, h, i, j, l, m, o, p, q, r, s, t, u, v, w, y, z\}$$

$$4. A \cap \neg B = \{c, h\}$$

$$5. C/D = \{c, h, p, r, s\}$$

$$6. X = \{c, h\} \cap \{b, c, e, g, h, p, r, s\} = \{c, h\}$$

$$7. Y = \{c, h\} \cup \{c, h, p, r, s\} = \{c, h, p, r, s\}$$

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.