

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:  
Горбунов Данила Евгеньевич  
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 \_\_\_\_\_

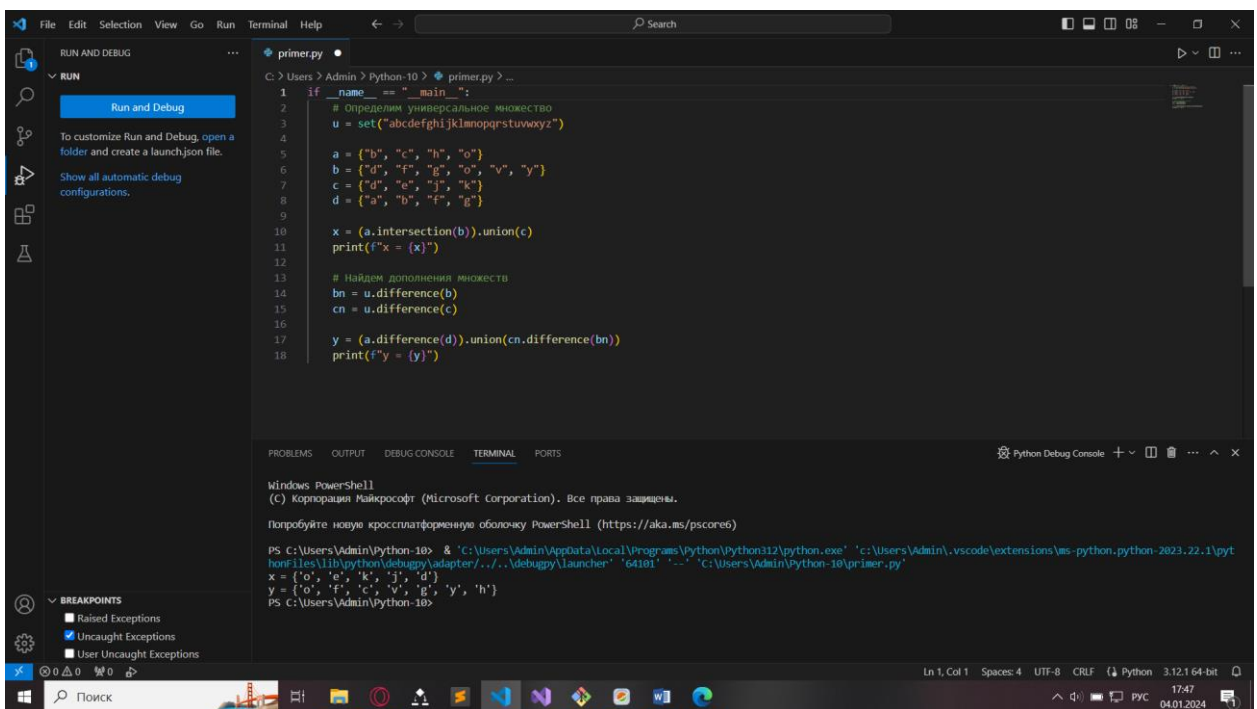
Ставрополь, 2023 г.

## 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 соответствии 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 if __name__ == "__main__":
2     # Определим универсальное множество
3     u = set("abcdefghijklmnopqrstuvwxyz")
4
5     a = {"b", "c", "h", "o"}
6     b = {"d", "f", "g", "o", "v", "y"}
7     c = {"d", "e", "j", "k"}
8     d = {"a", "b", "f", "g"}
9
10    x = (a.intersection(b)).union(c)
11    print(f"x = {x}")
12
13    # Найдём дополнения множеств
14    bn = u.difference(b)
15    cn = u.difference(c)
16
17    y = (a.difference(d)).union(cn.difference(bn))
18    print(f'y = {y}')
```

Windows PowerShell  
(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.  
Попробуйте новую кроссплатформенную оболочку PowerShell (https://aka.ms/powershell)  
PS C:\Users\Admin\Python-10> & 'C:\Users\Admin\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\Admin\..\.vscode\extensions\ms-python.python-2023.22.1\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\_launcher' '64101' '--' 'C:\Users\Admin\Python-10\primer.py'  
x = {'b', 'c', 'h', 'o'}  
y = {'o', 'f', 'c', 'v', 'g', 'y', 'h'}  
PS C:\Users\Admin\Python-10>

Рисунок 1. Результат работы программы из примера 1

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

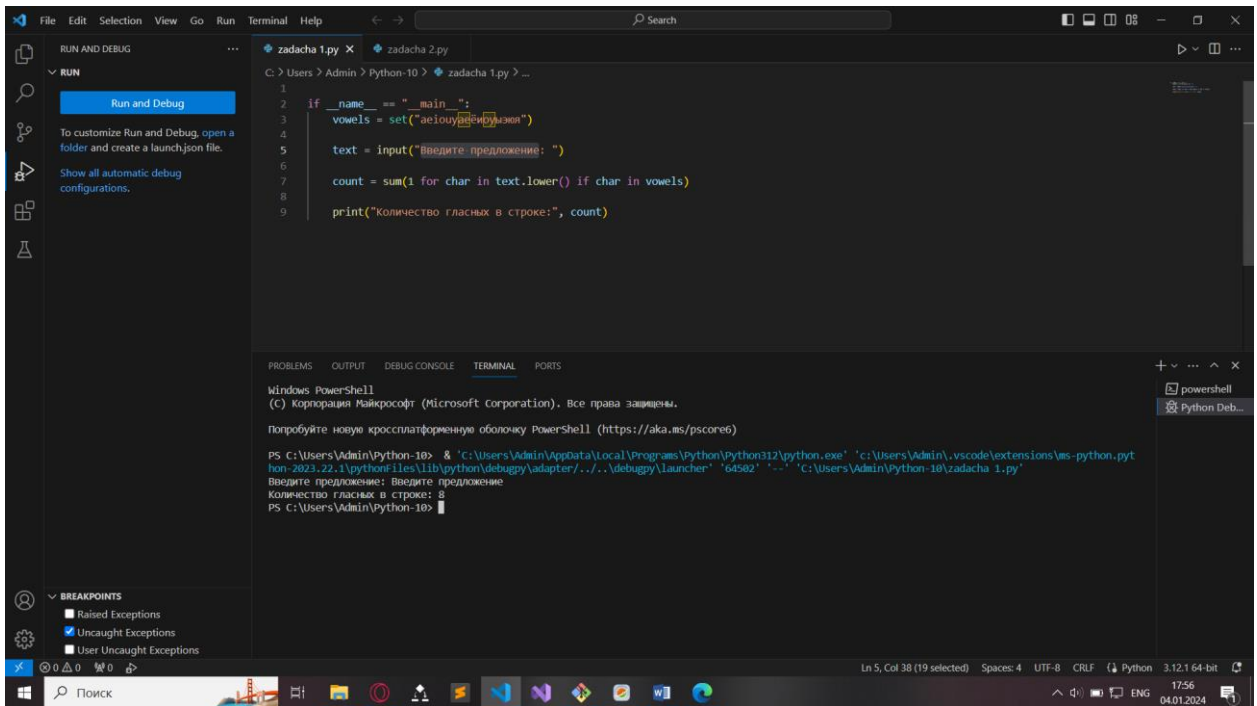


Рисунок 2. Результат работы программы из задачи 1

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

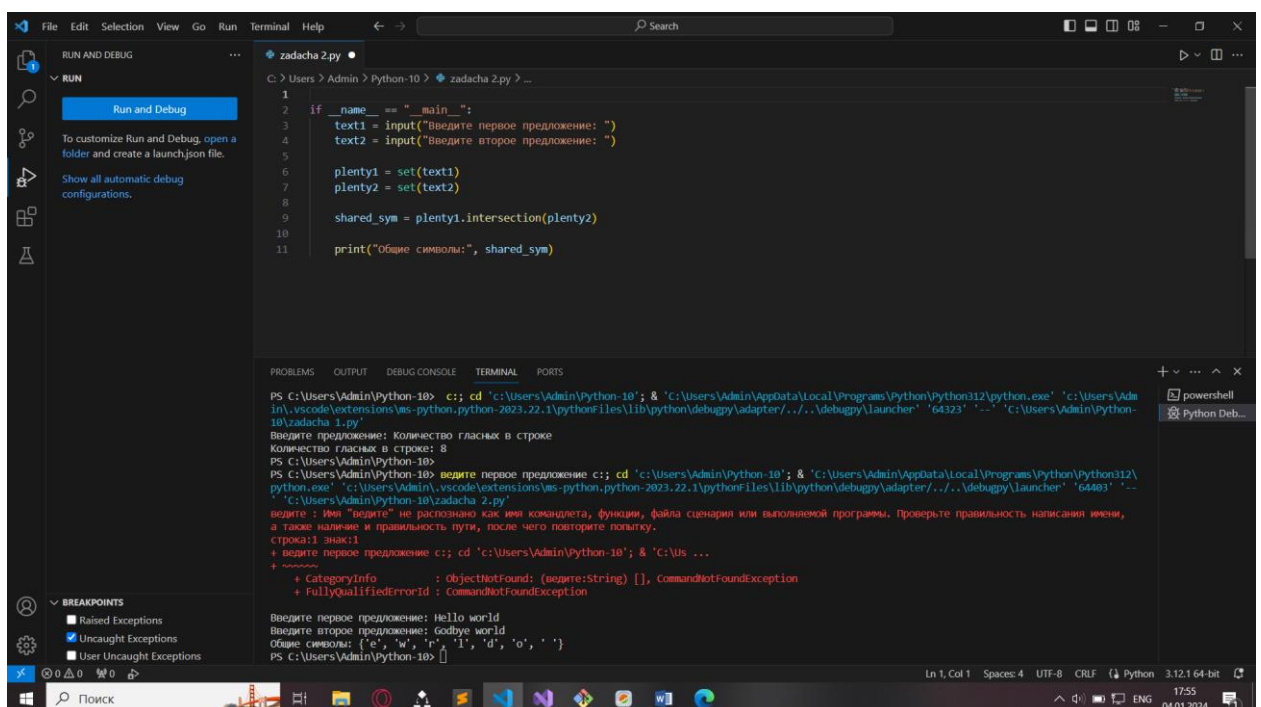


Рисунок 3. Результат работы программы из задачи 2

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

$$A = \{a, f, I, n, o\}; \quad B = \{f, g, o, p, z\}; \quad C = \{i, j, u, w\}; \quad D = \{f, h, n, t, u, y, z\};$$

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

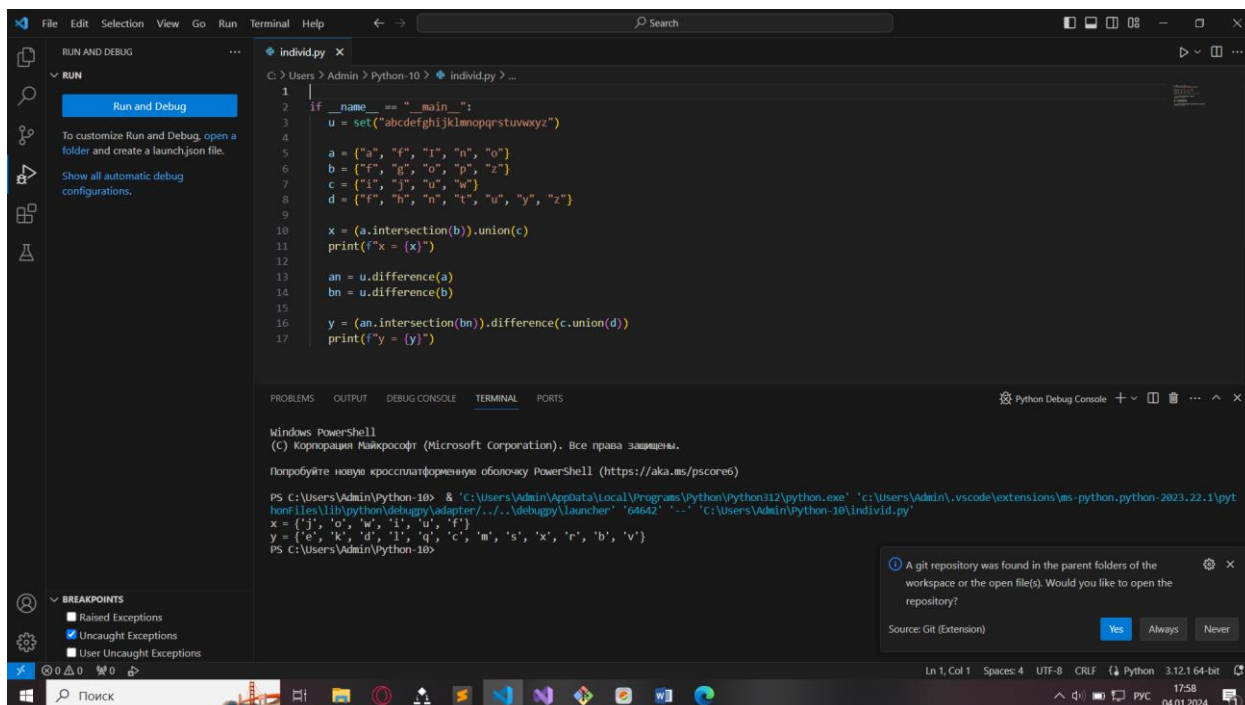


Рисунок 4. Результат работы программы из индивидуального задания

## 8. Проверка работы программы:

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

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

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

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

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

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

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

$$Y = \{i, j, u, w, f, h, n, t, y, z\}$$

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

## 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()`.

### 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 `-`.

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.

