

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 ПРАКТИЧЕСКОЙ РАБОТЕ №2.7
дисциплины «Программирование на Python»

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

(подпись)

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

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

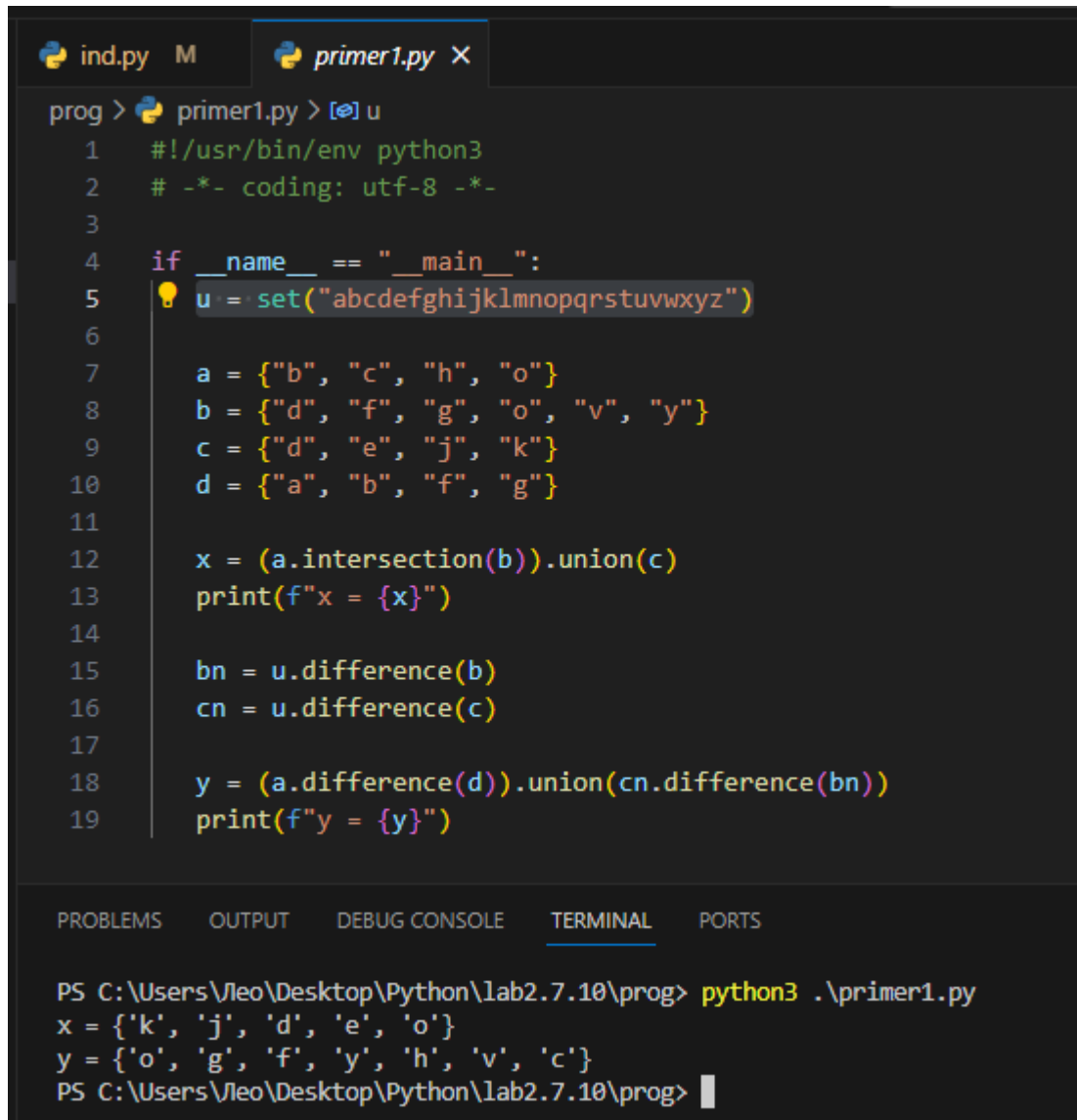
Ставрополь, 2023 г.

Тема: Работа с множествами в языке Python

Цель: Приобретение навыков по работе с множествами при написании программ с помощью языка программирования

Порядок выполнения работы:

1. Создал файл (primer1.py), в котором проработал пример 1, в нем производятся различные операции над множествами: объединение, пересечение, инверсия



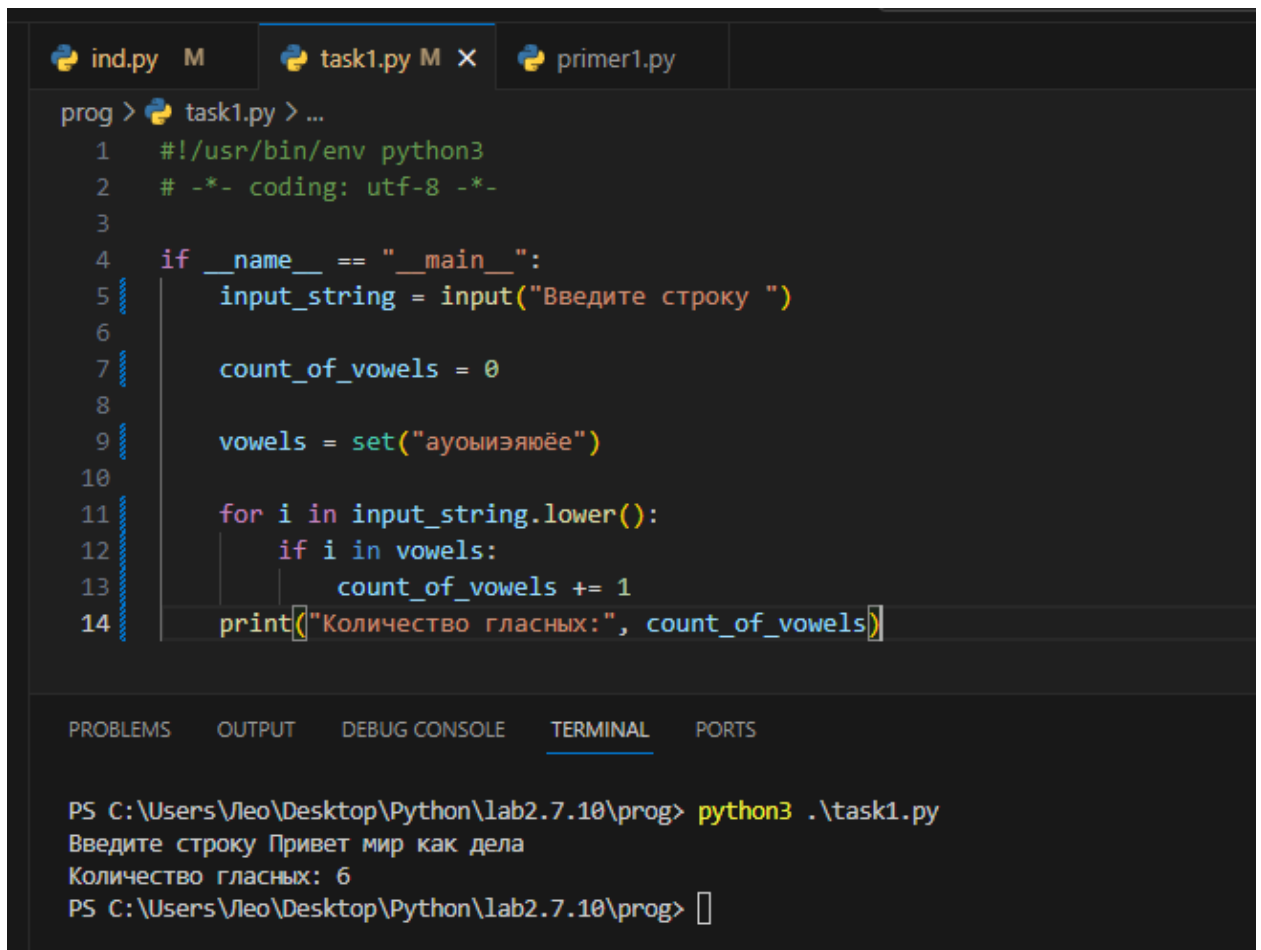
```
prog > primer1.py > [u] u
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  if __name__ == "__main__":
5      u = set("abcdefghijklmnopqrstuvwxyz")
6
7      a = {"b", "c", "h", "o"}
8      b = {"d", "f", "g", "o", "v", "y"}
9      c = {"d", "e", "j", "k"}
10     d = {"a", "b", "f", "g"}
11
12     x = (a.intersection(b)).union(c)
13     print(f"x = {x}")
14
15     bn = u.difference(b)
16     cn = u.difference(c)
17
18     y = (a.difference(d)).union(cn.difference(bn))
19     print(f"y = {y}")

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\Leo\Desktop\Python\lab2.7.10\prog> python3 .\primer1.py
x = {'k', 'j', 'd', 'e', 'o'}
y = {'o', 'g', 'f', 'y', 'h', 'v', 'c'}
PS C:\Users\Leo\Desktop\Python\lab2.7.10\prog>
```

Рисунок 1 – Результат выполнения программы primer1.py

2. Создал файл (task1.py), в котором решил задачу: подсчитайте количество гласных в строке, введенной с клавиатуры с использованием множеств.



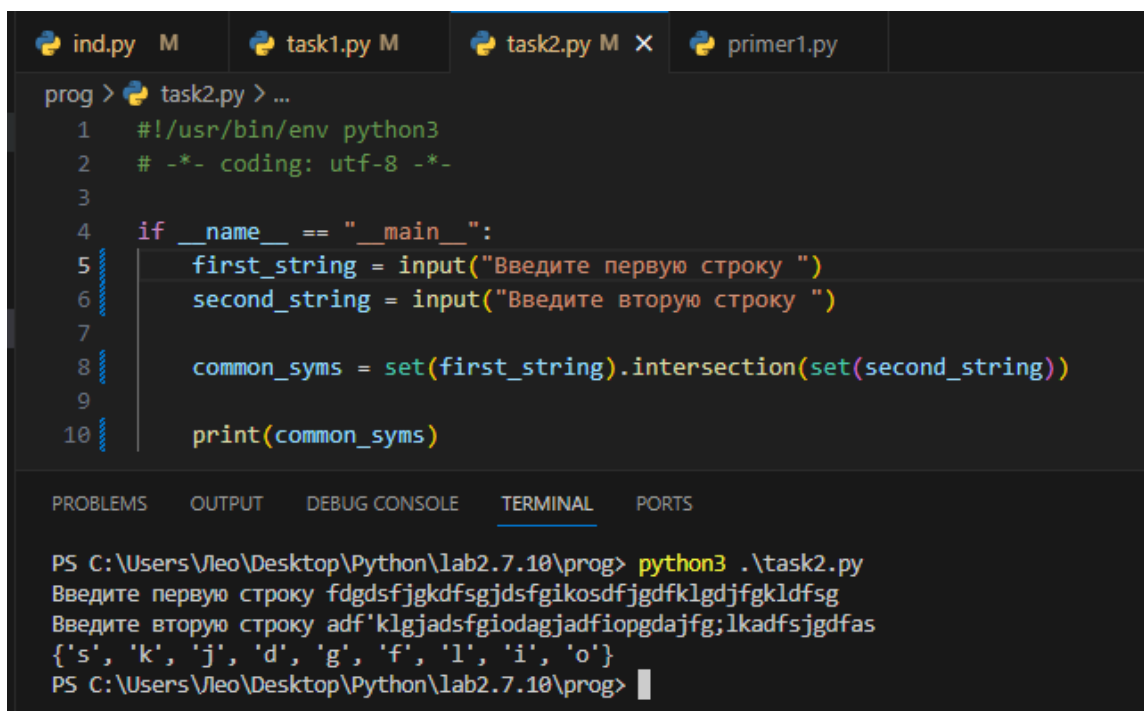
```
ind.py M task1.py M X primer1.py
prog > task1.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  if __name__ == "__main__":
5      input_string = input("Введите строку ")
6
7      count_of_vowels = 0
8
9      vowels = set("ауоыиэяюёе")
10
11     for i in input_string.lower():
12         if i in vowels:
13             count_of_vowels += 1
14     print("Количество гласных:", count_of_vowels)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\Leo\Desktop\Python\lab2.7.10\prog> python3 .\task1.py
Введите строку Привет мир как дела
Количество гласных: 6
PS C:\Users\Leo\Desktop\Python\lab2.7.10\prog>

Рисунок 2 – Результат выполнения программы task1.py

3. Создал файл (task2.py), в котором решил задачу: определите общие символы в двух строках, введенных с клавиатуры.



```
ind.py M task1.py M task2.py M X primer1.py
prog > task2.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  if __name__ == "__main__":
5      first_string = input("Введите первую строку ")
6      second_string = input("Введите вторую строку ")
7
8      common_syms = set(first_string).intersection(set(second_string))
9
10     print(common_syms)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\Leo\Desktop\Python\lab2.7.10\prog> python3 .\task2.py
Введите первую строку fdgdsfjgkdfsgjdsfgikosdfjgdfklgdjfgklfdsg
Введите вторую строку adf'klgjadsfgiodagjadfiopgdajfg;lkadfsjgdfas
{', 's', 'k', 'j', 'd', 'g', 'f', 'l', 'i', 'o'}
PS C:\Users\Leo\Desktop\Python\lab2.7.10\prog>

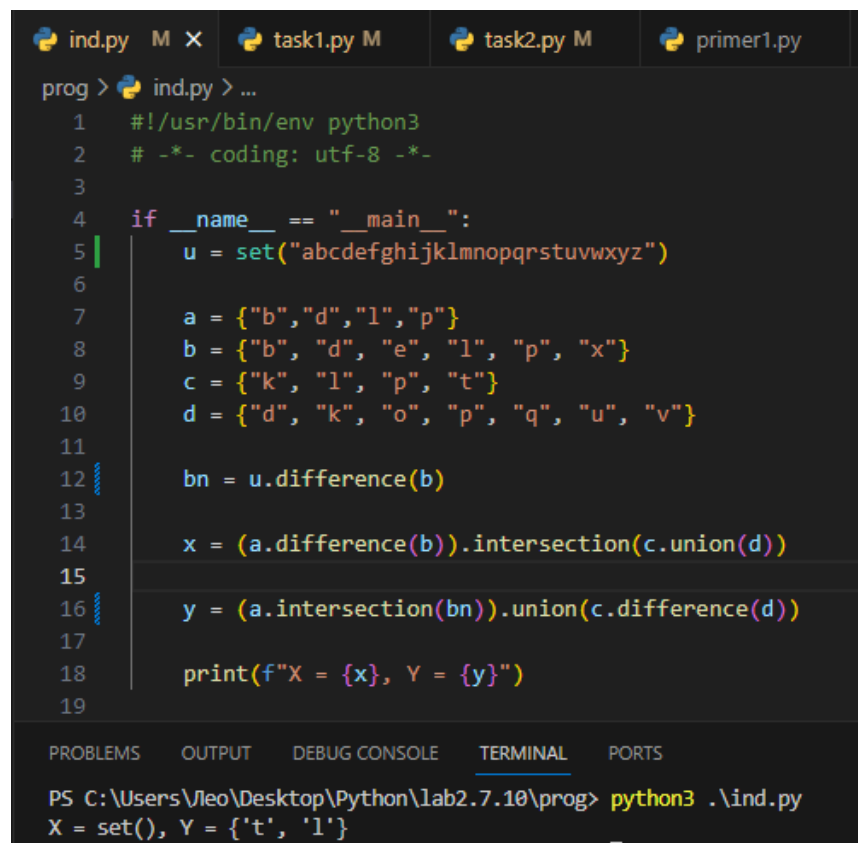
Рисунок 3 – Результат выполнения программы task2.py

Индивидуальное задание:

Определить результат выполнения операций над множествами. Считать элементы множества строками. Проверить результаты вручную.

$$A = \{b, d, l, p\}; \quad B = \{b, d, e, l, p, x\} \quad C = \{k, l, p, t\}; \quad D = \{d, k, o, p, q, u, v\};$$
$$X = (A/B) \cap (C \cup D); \quad Y = (A \cap \bar{B}) \cup (C/D).$$

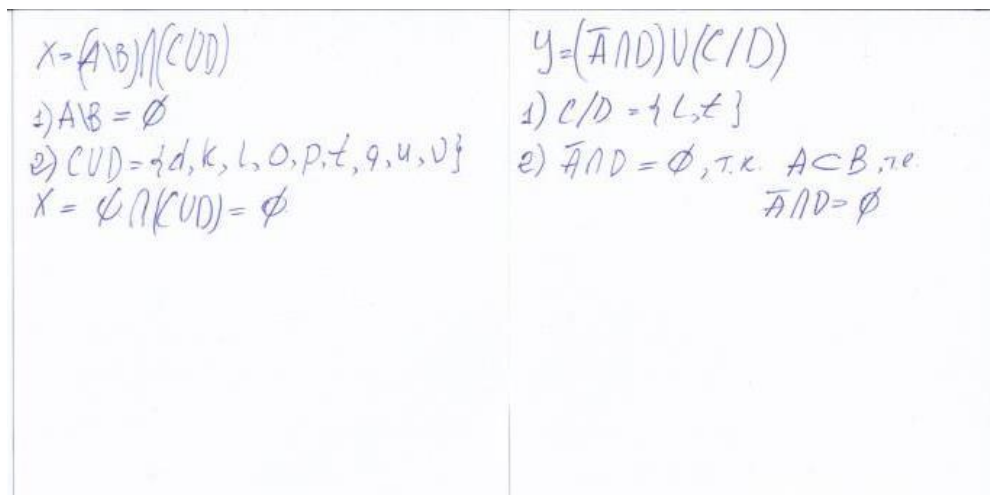
Рисунок 4 – Задание



```
ind.py M X task1.py M task2.py M primer1.py
prog > ind.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  if __name__ == "__main__":
5      u = set("abcdefghijklmnopqrstuvwxyz")
6
7      a = {"b", "d", "l", "p"}
8      b = {"b", "d", "e", "l", "p", "x"}
9      c = {"k", "l", "p", "t"}
10     d = {"d", "k", "o", "p", "q", "u", "v"}
11
12     bn = u.difference(b)
13
14     x = (a.difference(b)).intersection(c.union(d))
15
16     y = (a.intersection(bn)).union(c.difference(d))
17
18     print(f"X = {x}, Y = {y}")
19
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\leo\Desktop\Python\lab2.7.10\prog> python3 .\ind.py
X = set(), Y = {'t', 'l'}
```

Рисунок 5 – Результат выполнения программы ind.py

Проверка работы программы:



$X = (A/B) \cap (C \cup D)$ <p>1) $A/B = \emptyset$ 2) $C \cup D = \{d, k, l, o, p, t, q, u, v\}$ $X = \emptyset \cap (C \cup D) = \emptyset$</p>	$Y = (\bar{A} \cap D) \cup (C/D)$ <p>1) $C/D = \{l, t\}$ 2) $\bar{A} \cap D = \emptyset$, т.к. $A \subset B$, т.е. $\bar{A} \cap D = \emptyset$</p>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Рисунок 6 – Результат решения вручную

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

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.