

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 №2.7
дисциплины «Основы кроссплатформенного программирования»
Вариант ____

Выполнил:

Быковская Стефания Станиславовна
1 курс, группа ИТС-б-0-22-1,
11.03.02 «Инфокоммуникационные
технологии и системы связи», направ-
ленность (профиль) «Инфокоммуника-
ционные системы и сети», очная форма
обучения

(подпись)

Руководитель практики:

Воронкин Р.А. доцент кафедры инфо-
коммуникаций

(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 Python версии 3.x.

Порядок выполнения работы:

Пример 1. Определить результат выполнения операций над множествами. Считать элементы множества строками.

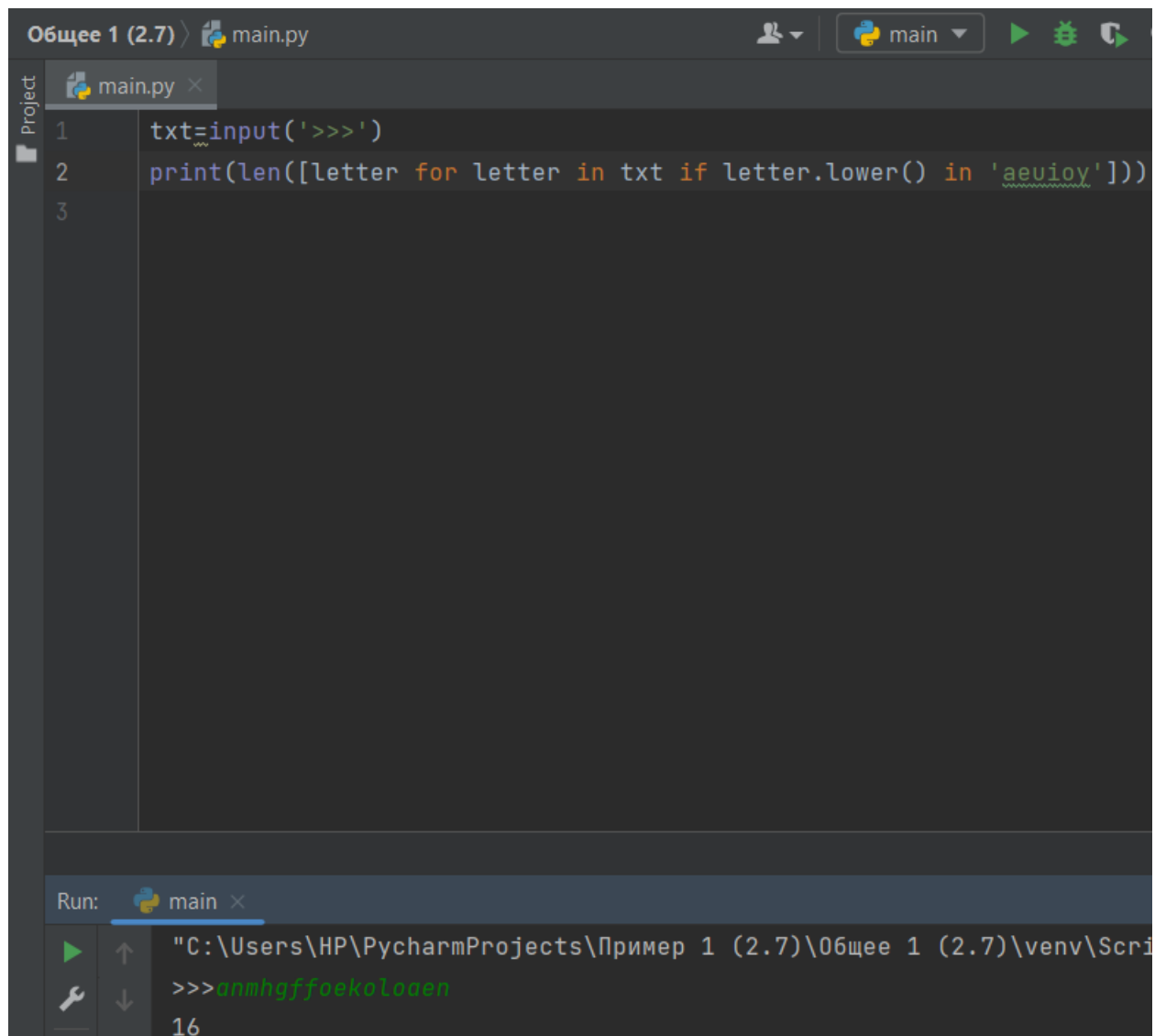
$$A = \{b, c, h, o\}; \quad B = \{d, f, g, o, v, y\}; \quad C = \{d, e, j, k\}; \quad D = \{a, b, f, g\}; \quad X = (A \cap B) \cup C; \quad Y = (A/D) \cup (\bar{C}/\bar{B}).$$

```
Пример 1 (2.7) > main.py
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3  if __name__ == "__main__":
4      # Определим универсальное множество
5      u = set("abcdefghijklmnopqrstuvwxyz")
6      a = {"b", "c", "h", "o"}
7      b = {"d", "f", "g", "o", "v", "y"}
8      c = {"d", "e", "j", "k"}
9      d = {"a", "b", "f", "g"}
10     x = (a.intersection(b)).union(c)
11     print(f"x = {x}")
12     # Найдем дополнения множеств
13     bn = u.difference(b)
14     cn = u.difference(c)
15     y = (a.difference(d)).union(cn.difference(bn))
16     print(f"y = {y}")

if __name__ == "__main__"
Run: main x
"C:\Users\HP\PycharmProjects\Пример 1 (2.7)\Пример 1 (2.7)\venv
x = {'o', 'j', 'd', 'e', 'k'}
y = {'c', 'g', 'y', 'o', 'h', 'f', 'v'}
```

Рисунок 1 – Выполнение примера №1.

Общее задание 1. Подсчитать количество гласных в строке.



The screenshot shows the PyCharm IDE interface. The top toolbar includes icons for running, debugging, and other IDE functions. The main editor window displays a Python script in a file named `main.py`. The script consists of two lines of code: `txt=input('>>>')` and `print(len([letter for letter in txt if letter.lower() in 'aeuioy']))`. The left sidebar shows the project structure. At the bottom, the 'Run' console shows the execution of the script, with the input string `>>>anmhgffoekoloen` and the output `16`.

```
Общее 1 (2.7) > main.py
main.py x
1 txt=input('>>>')
2 print(len([letter for letter in txt if letter.lower() in 'aeuioy']))
3
Run: main x
"C:\Users\HP\PycharmProjects\Пример 1 (2.7)\Общее 1 (2.7)\venv\Scripts\python.exe"
>>>anmhgffoekoloen
16
```

Рисунок 2 – Выполнение общего задания №1.

Общее задание №2. Определите общие символы в двух строках введенных с клавиатуры.

The screenshot shows the PyCharm IDE interface. The top toolbar indicates the current file is 'main.py' in a project named 'Общее 2 (2.7)'. The left sidebar shows the 'Project' view with 'main.py' selected. The main editor displays the following Python code:

```
1
2 str1 = input("Введите первую строку: ")
3 str2 = input("Введите вторую строку: ")
4 simb = set(str1) & set(str2)
5 if simb:
6     print("Общие символы: " + "".join(simb))
7 else:
8     print("Общих символов нет")
```

Below the code editor, the 'Run' tab is active, showing the execution output. The output path is 'C:\Users\HP\PycharmProjects\Пример 1 (2.7)\Общее 2 (2.7)\'. The input and output are as follows:

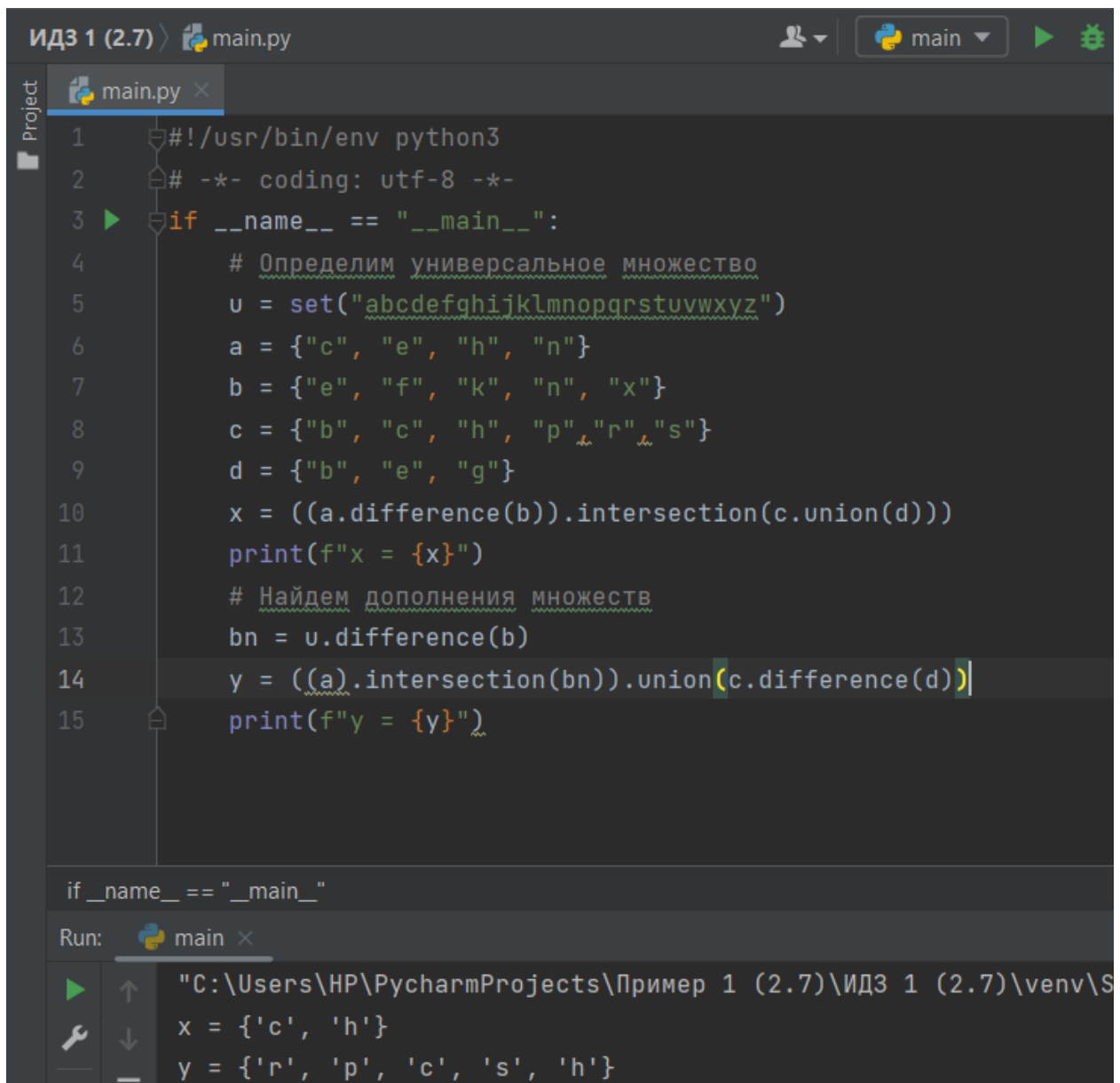
```
Введите первую строку: catcatcatcat
Введите вторую строку: dogcatdogcat
Общие символы: cta
```

Рисунок 3 – Выполнение общего задания №2.

Индивидуальное задание

Вариант 5.

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



The screenshot shows the PyCharm IDE interface. The top toolbar includes a Python icon, a dropdown menu with 'main', and a green run button. The left sidebar shows a 'Project' view with 'main.py' selected. The main editor displays the following Python code:

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3  if __name__ == "__main__":
4      # Определим универсальное множество
5      u = set("abcdefghijklmnopqrstuvwxyz")
6      a = {"c", "e", "h", "n"}
7      b = {"e", "f", "k", "n", "x"}
8      c = {"b", "c", "h", "p", "r", "s"}
9      d = {"b", "e", "g"}
10     x = ((a.difference(b)).intersection(c.union(d)))
11     print(f"x = {x}")
12     # Найдем дополнения множеств
13     bn = u.difference(b)
14     y = ((a.intersection(bn)).union(c.difference(d)))
15     print(f"y = {y}")
```

Below the code editor, the 'Run' tab is active, showing the execution path and the output:

```
Run: main
"C:\Users\HP\PycharmProjects\Пример 1 (2.7)\ИДЗ 1 (2.7)\venv\Scripts\python.exe"
x = {'c', 'h'}
y = {'r', 'p', 'c', 's', 'h'}
```

Рисунок 4 – Выполнение ИДЗ.

Ответы на контрольные вопросы

1) Что такое множества в языке Python?

Множества в языке Python - это неупорядоченные коллекции уникальных элементов, где каждый элемент может присутствовать только один раз. В качестве элементов этого набора данных могут выступать любые неизменяемые объекты, такие как числа, символы, строки.

2) Как осуществляется создание множеств в Python?

Создание множества осуществляется с помощью фигурных скобок {} или функции set(). Например, $a = \{1, 2, 3\}$ или $b = \text{set}([4, 5, 6])$.

3) Как проверить присутствие/отсутствие элемента в множестве?

Для проверки присутствия/отсутствия элемента в множестве в Python используется ключевое слово `in`.

4) Как выполнить перебор элементов множества?

Перебор элементов множества можно выполнить с помощью цикла `for`.

5) Что такое `set comprehension`?

`Set comprehension` (генератор множеств) - это способ создания заполнения множества в Python одной строки кода.

6) Как выполнить добавление элемента во множество?

Для добавления элемента во множество в Python используется метод `add()`.

7) Как выполнить удаление одного или всех элементов множества?

Удаление элемента из множества можно выполнить с помощью метода `remove()`, который удаляет заданный элемент, или метода `discard()`, который также удаляет элемент, но не вызывает исключение, если элемент не найден.

Удаление всех элементов множества может быть выполнено с помощью метода `clear()`.

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

Основные операции над множествами в Python:

— Объединение (`union`) - создает новое множество, содержащее все элементы из двух или более множеств.

— Пересечение (`intersection`) - создает новое множество, содержащее только те элементы, которые присутствуют в обоих множествах.

— Разность (`difference`) - создает новое множество, состоящее из элементов первого множества, которые не присутствуют во втором множестве.

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

Для определения, является ли некоторое множество подмножеством или надмножеством другого множества в Python, можно воспользоваться методами `issubset()` и `issuperset()`. Метод `issubset()` проверяет, является ли одно

множество подмножеством другого. Если все элементы первого множества присутствуют во втором множестве, то метод вернет True, иначе False. Метод `issuperset()` проверяет, является ли одно множество надмножеством другого. Если все элементы второго множества присутствуют в первом множестве, то метод вернет True, иначе False.

10) Каково назначение множеств `frozenset`?

Множества `frozenset` являются неизменяемыми (`immutable`) множествами, то есть их элементы нельзя добавлять, удалять или изменять после создания. Они используются, когда требуется иметь неизменяемое множество, которое можно использовать как элемент словаря или ключ в другом множестве.

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

Для преобразования множества в строку, список или словарь в языке Python можно использовать следующие функции:

- `str()` - для преобразования множества в строку;
- `list()` - для преобразования множества в список;
- `dict.fromkeys()` - для преобразования множества в словарь

Выводы: приобрела навыки по работе с множествами при написании программ с помощью языка программирования Python 3.x.