

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:
Гайчук Дарья Дмитриевна
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.-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.

Порядок выполнения работы

1. Создала общедоступный репозиторий на GitHub, в котором будет использоваться лицензия MIT.

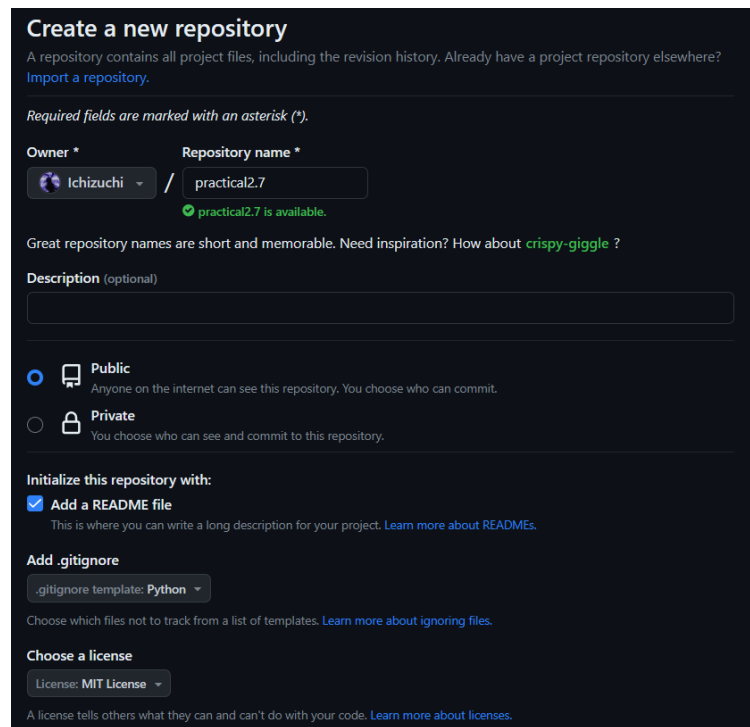


Рисунок 1. Новый репозиторий

2. Скопировала репозиторий на свой компьютер.

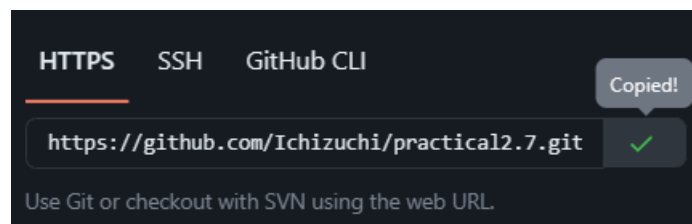


Рисунок 2. Клонирование репозитория

3. Использовала систему ветвления git-flow

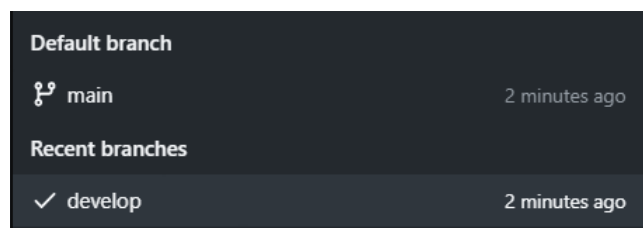
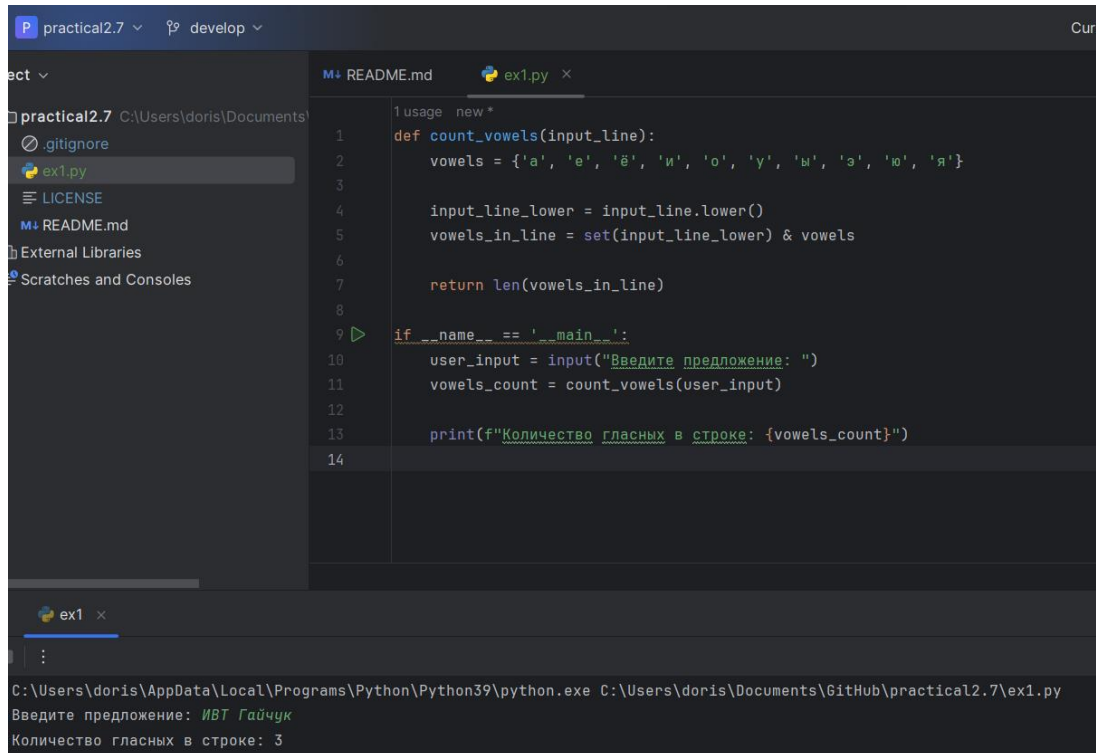


Рисунок 3. Ветка develop

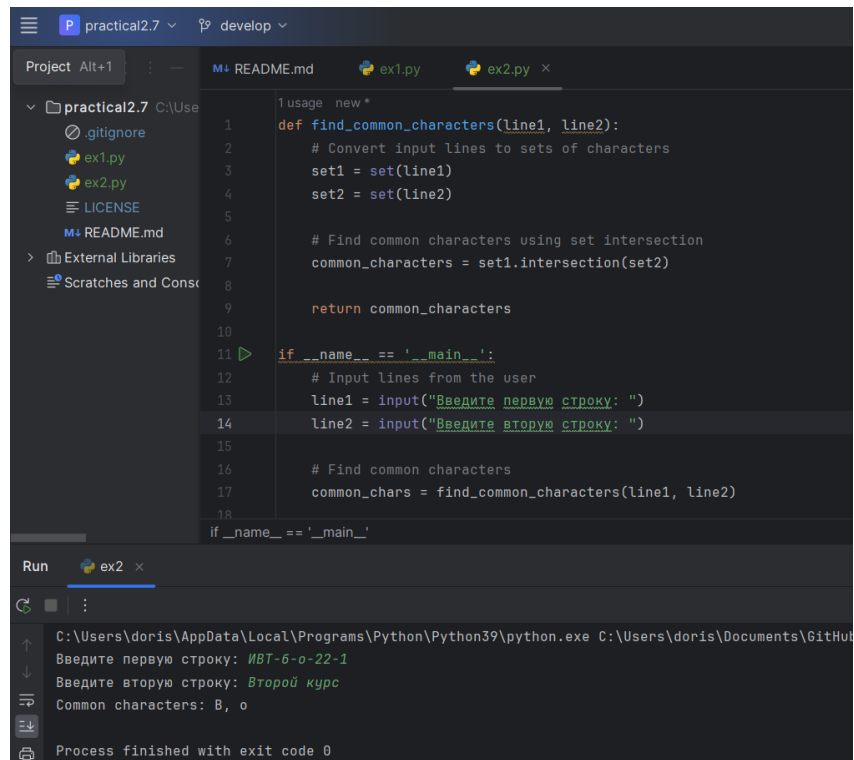
4. Задача №1. Условие: подсчитайте количество гласных в строке, введенной с клавиатуры с использованием множеств.



```
practical2.7 develop
M+ README.md ex1.py x
1 usage new *
2 def count_vowels(input_line):
3     vowels = {'a', 'e', 'ё', 'и', 'о', 'у', 'ы', 'э', 'ю', 'я'}
4
5     input_line_lower = input_line.lower()
6     vowels_in_line = set(input_line_lower) & vowels
7
8     return len(vowels_in_line)
9
10 if __name__ == '__main__':
11     user_input = input("Введите предложение: ")
12     vowels_count = count_vowels(user_input)
13
14     print(f"Количество гласных в строке: {vowels_count}")
15
ex1 x
C:\Users\doris\AppData\Local\Programs\Python\Python39\python.exe C:\Users\doris\Documents\GitHub\practical2.7\ex1.py
Введите предложение: ИБТ Гайчук
Количество гласных в строке: 3
```

Рисунок 4. Программа и ее результат

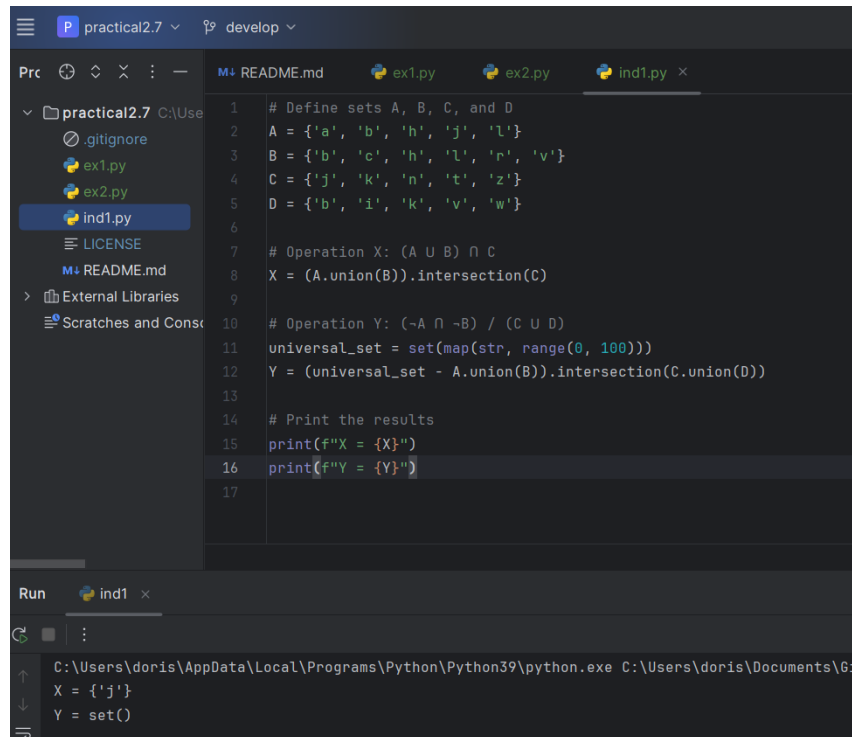
5. Задача №1. Условие: определите общие символы в двух строках, введенных с клавиатуры.



```
practical2.7 develop
Project Alt+1 M+ README.md ex1.py ex2.py x
1 usage new *
2 def find_common_characters(line1, line2):
3     # Convert input lines to sets of characters
4     set1 = set(line1)
5     set2 = set(line2)
6
7     # Find common characters using set intersection
8     common_characters = set1.intersection(set2)
9
10    return common_characters
11
12 if __name__ == '__main__':
13     # Input lines from the user
14     line1 = input("Введите первую строку: ")
15     line2 = input("Введите вторую строку: ")
16
17     # Find common characters
18     common_chars = find_common_characters(line1, line2)
19
20     if __name__ == '__main__':
21         print(common_chars)
22
Run ex2 x
C:\Users\doris\AppData\Local\Programs\Python\Python39\python.exe C:\Users\doris\Documents\GitHub\practical2.7\ex2.py
Введите первую строку: ИБТ-6-0-22-1
Введите вторую строку: Второй курс
Common characters: Б, о
Process finished with exit code 0
```

Рисунок 5. Программа и ее результат

6. Индивидуальное задание, вариант №4: Напишите программу для определения результата выполнения операций над множествами. Считать элементы множества строками. Проверить результаты вручную. значения и положения: $X = (A \cup B) \cap C$; $Y = (\bar{A} \cap \bar{B}) / (C \cup D)$, $A = \{a, b, h, j, l\}$; $B = \{b, c, h, l, r, v\}$; $C = \{j, k, n, t, z\}$; $D = \{b, i, k, v, w\}$;



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project named 'practical2.7' with files 'ex1.py', 'ex2.py', and 'ind1.py'. The code editor shows the content of 'ind1.py'.

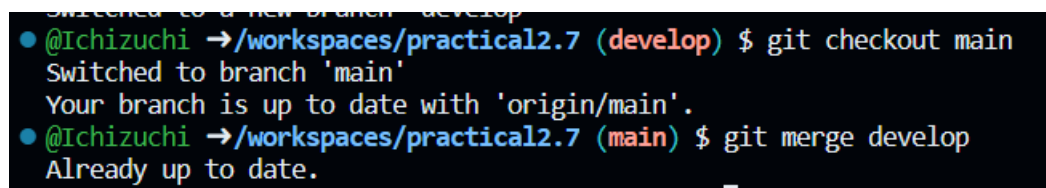
```
1 # Define sets A, B, C, and D
2 A = {'a', 'b', 'h', 'j', 'l'}
3 B = {'b', 'c', 'h', 'l', 'r', 'v'}
4 C = {'j', 'k', 'n', 't', 'z'}
5 D = {'b', 'i', 'k', 'v', 'w'}
6
7 # Operation X: (A U B) ∩ C
8 X = (A.union(B)).intersection(C)
9
10 # Operation Y: (~A ∩ ~B) / (C U D)
11 universal_set = set(map(str, range(0, 100)))
12 Y = (universal_set - A.union(B)).intersection(C.union(D))
13
14 # Print the results
15 print(f"X = {X}")
16 print(f"Y = {Y}")
17
```

The output window shows the results of the script:

```
Run ind1 x
C:\Users\doris\AppData\Local\Programs\Python\Python39\python.exe C:\Users\doris\Documents\G3
X = {'j'}
Y = set()
```

Рисунок 6. Программа и ее результат

7. Слила ветку develop с веткой main и отправил на удаленный сервер – Github.



The screenshot shows a terminal window with the following commands and output:

```
Switched to a new branch 'develop'
• @Ichizuchi → /workspaces/practical2.7 (develop) $ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
• @Ichizuchi → /workspaces/practical2.7 (main) $ git merge develop
Already up to date.
```

Рисунок 7. Слияние веток

1. Что такое множества в языке Python?

Множеством в языке программирования Python называется неупорядоченная совокупность уникальных значений.

2. Как осуществляется создание множеств в Python?

Создать множество можно, просто присвоив переменной последовательность значений, выделив их фигурными скобками или с помощью метода `set`.

3. Как проверить присутствие/отсутствие элемента в множестве?

Для проверки присутствия используется `in`, для проверки отсутствия `not in`.

4. Как выполнить перебор элементов множества?

```
for a in set: print(a)
```

5. Что такое `set comprehension`?

Генератор, позволяющий заполнять списки, а также другие наборы данных с учетом неких условий.

6. Как выполнить добавление элемента во множество?

Чтобы внести новые значения, потребуется вызывать метод `add`. Аргументом в данном случае будет добавляемый элемент последовательности.

7. Как выполнить удаление одного или всех элементов множества?

Для удаления элементов из множества используются следующие функции в Python: `remove` — удаление элемента с генерацией исключения в случае, если такого элемента нет; `discard` — удаление элемента без генерации исключения, если элемент отсутствует; `pop` — удаление первого элемента, генерируется исключение при попытке удаления из пустого множества. Иногда необходимо полностью убрать все элементы. Чтобы не удалять каждый элемент отдельно, используется метод `clear()`, не принимающий аргументов.

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

Чтобы объединить все элементы двух разных множеств, стоит воспользоваться методом `union()` на одном из объектов. Чтобы найти общие элементы для двух разных множеств, следует применить функцию `intersection()`, принимающую в качестве аргумента один из наборов данных.

Чтобы вычислить разность для двух разных множеств, необходимо воспользоваться методом `difference()`.

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

Чтобы выяснить, является ли множество `a` подмножеством `b`, стоит попробовать вывести на экран результат выполнения метода `issubset`. Чтобы узнать, является ли множество `a` надмножеством `b`, необходимо вызвать метод `issuperset` и вывести результат его работы на экран.

10. Каково назначение множеств `frozenset` ?

Множество, содержимое которого не поддается изменению имеет тип `frozenset`. Значения из этого набора нельзя удалить, как и добавить новые. Поскольку содержимое `frozenset` должно всегда оставаться статичным, перечень функций, с которыми такое множество может взаимодействовать, имеет ограничения.

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

Для преобразования множества в строку используется конкатенация текстовых значений, которую обеспечивает функция `join()`. В этом случае ее аргументом является набор данных в виде нескольких строк. Чтобы получить из множества словарь, следует передать функции `dict` набор из нескольких пар значений, в каждом из которых будет находиться ключ. По аналогии с предыдущими преобразованиями можно получить список неких объектов. На этот раз используется вызов `list`, получающий в качестве аргумента множество `a`.

Вывод: в результате выполнения работы были приобретены навыки по работе с множествами при написании программ с помощью языка программирования Python версии 3.x.