

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»

Выполнил:
Евдаков Евгений Владимирович
1 курс, группа ИТС-б-о-22-1,
11.03.02 «Инфокоммуникационные
технологии и системы связи»,
направленность (профиль)
«Инфокоммуникационные системы и
сети», 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: приобретение навыков по работе с множествами при написании программ с помощью языка программирования Python версии 3.x.

Ход работы:

Задание 1. Создал общедоступный репозиторий на GitHub, в котором использована лицензий MIT и язык программирования Python, также добавил файл .gitignore с необходимыми правилами. Клонировал свой репозиторий на свой компьютер.

```
C:\Users\Gaming-PC>git clone https://github.com/EvgenyEvdakov/Laba_2.7.git
Cloning into 'Laba_2.7'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.
```

Рисунок 1. Клонирование репозитория

Задание 2. Организовал свой репозиторий в соответствии с моделью ветвления git-flow, появилась новая ветка develop в которой буду выполнять дальнейшие задачи.

```
C:\Users\Gaming-PC\Laba_2.7>git flow init

which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [C:/Users/Gaming-PC/Laba_2.7/.git/hooks]
```

Рисунок 2. Модель ветвления git-flow

Задание 3. Создал проект PyCharm в папке репозитория. Приступил к работе с примером. Добавил новый файл primer.py.

Условие примера: определить результат выполнения операций над множествами. Считать элементы множества строками.

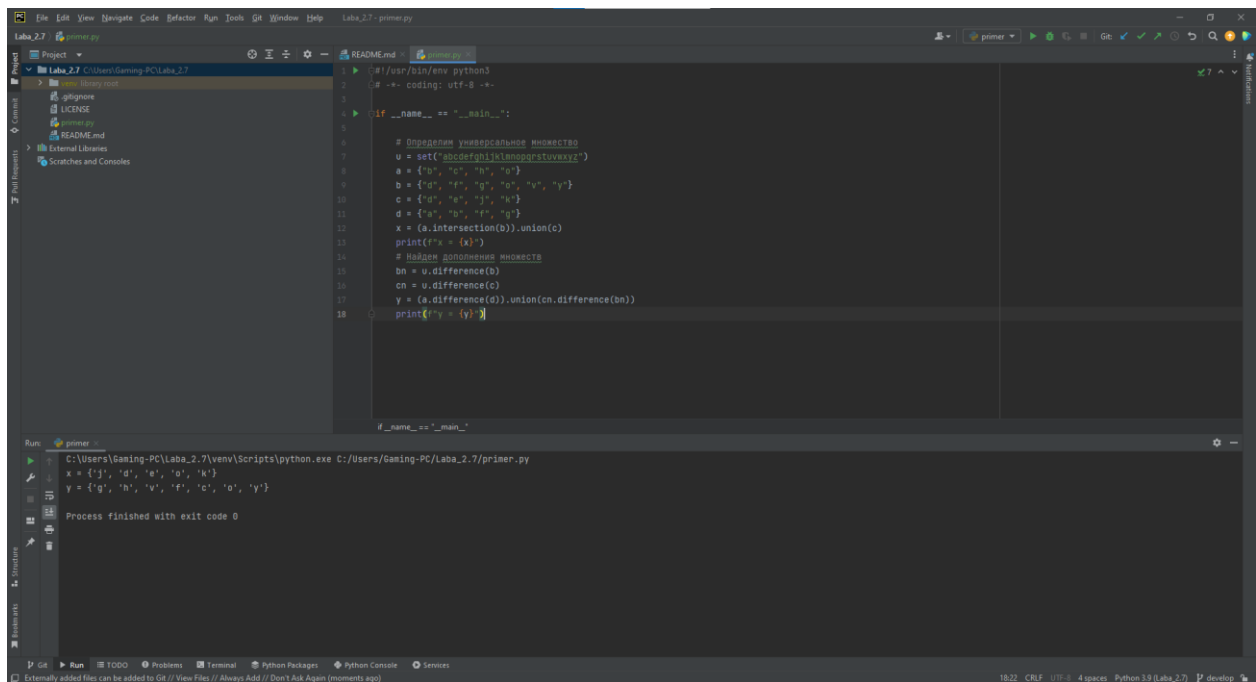


Рисунок 3. Реализация примера

Задание 4. Создал новый файл под названием `ObshZadanie1`. Приступил к работе с общим заданием №1.

Условие примера: подсчитайте количество гласных в строке, введенной с клавиатуры с использованием множеств.

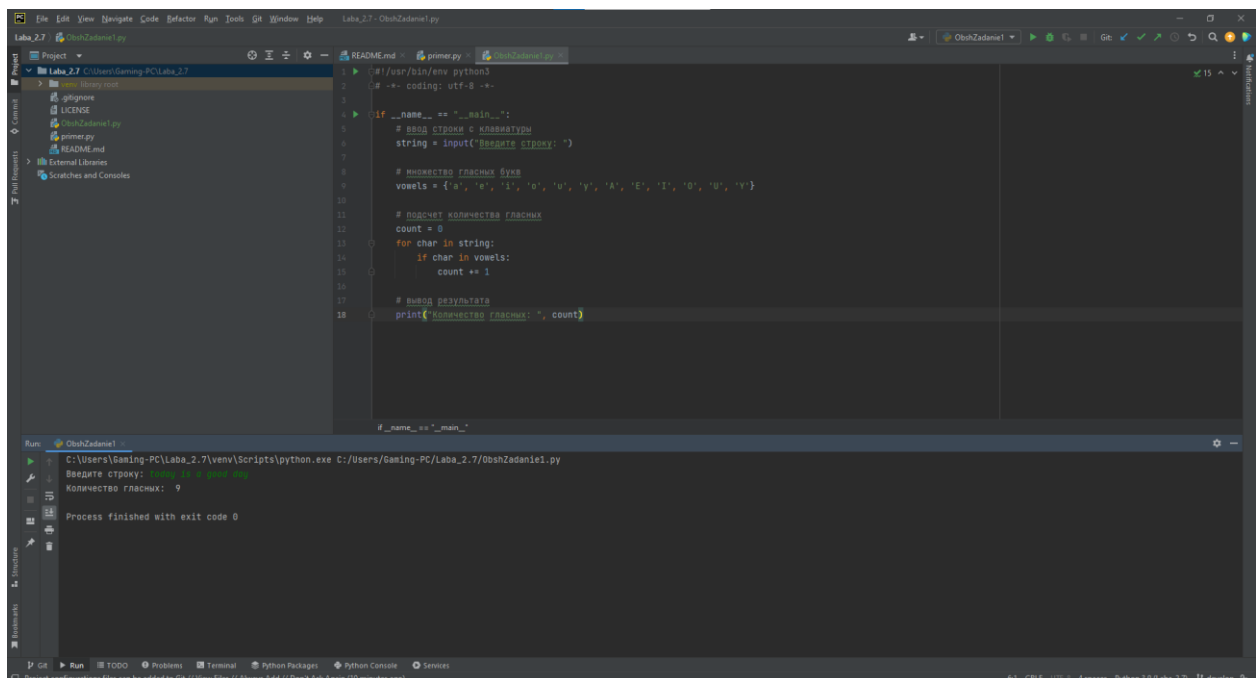


Рисунок 4. Реализация общего задания 1

Задание 5. Создал новый файл под названием `ObshZadanie2`. Приступил к работе с общим заданием №2.

Условие примера: определите общие символы в двух строках, введенных с клавиатуры.

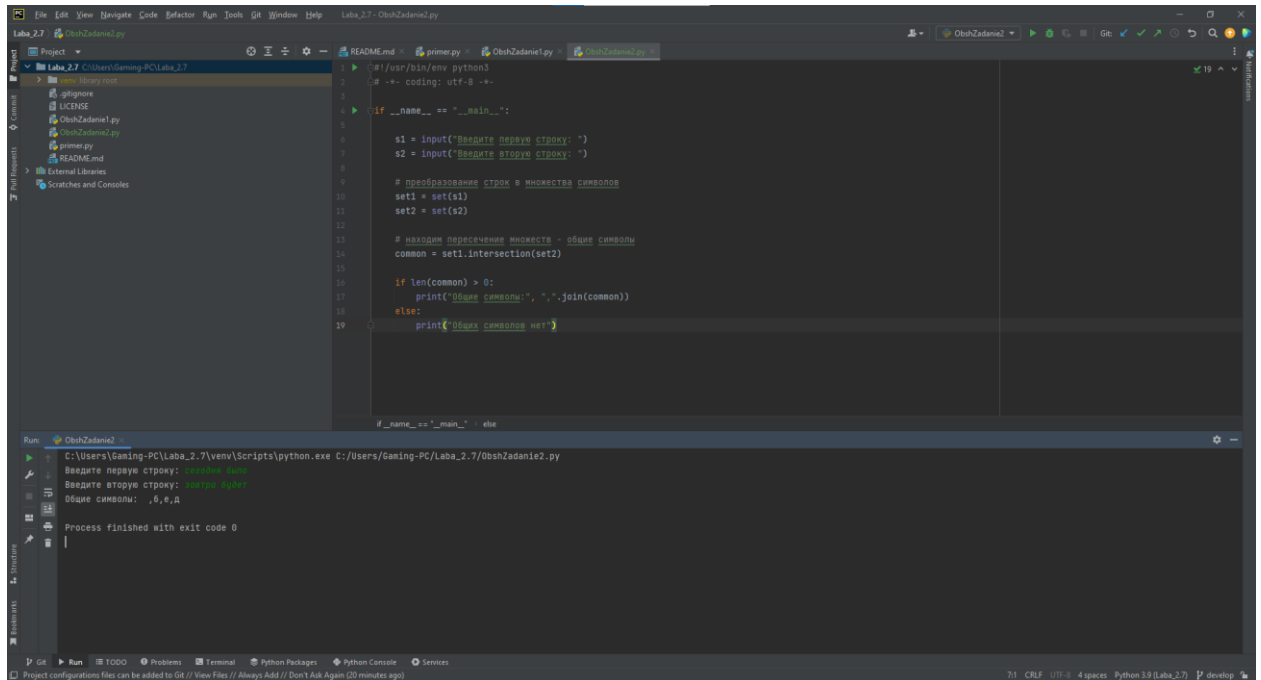


Рисунок 5. Реализация общего задания 2

Задание 6.

Индивидуальное задание

Вариант 9 (по списку группы)

Создал новый файл под названием individual.py.

Условие задания: определить результат выполнения операций над множествами. Считать элементы множества строками.

$$A = \{a, e, f, i\}; \quad B = \{a, b, k, n\}; \quad C = \{e, f, n, o, w, x\}; \quad D = \{a, d, e, o, p, t, u\};$$
$$X = (A \cup B) \cap D; \quad Y = (\bar{A} \cap \bar{B}) / (C \cup D).$$

9.

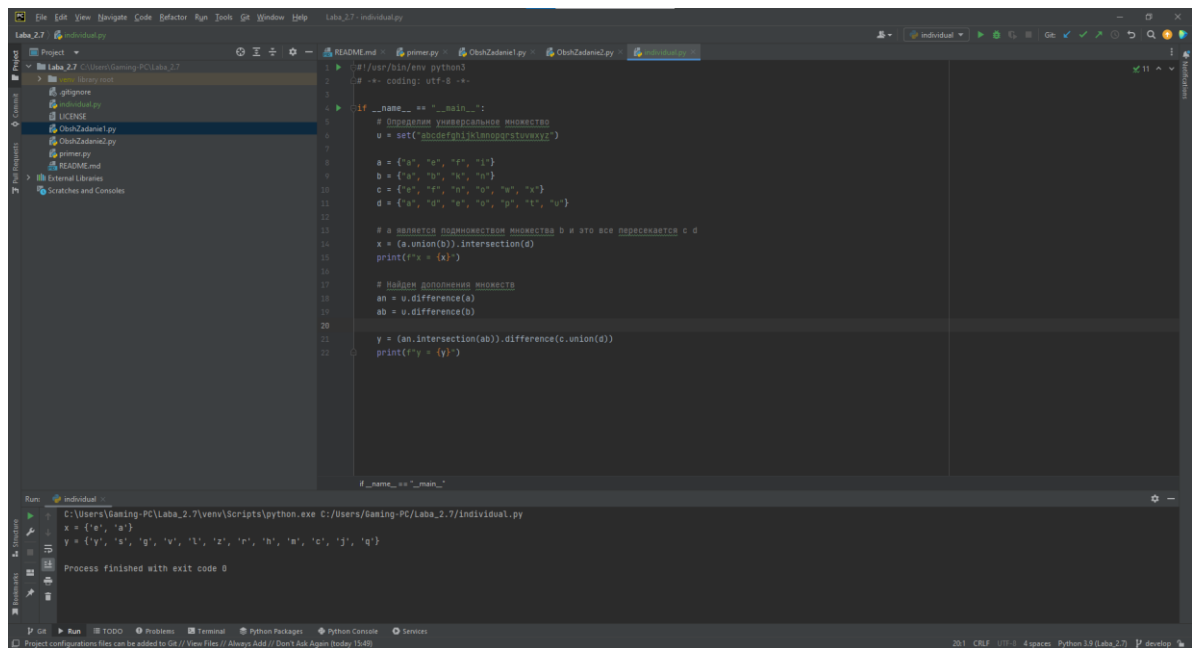


Рисунок 6. Программа индивидуального задания

Задание 7.

После выполнения работы на ветке develop, слил ее с веткой main и отправил изменения на удаленный сервер.

```
C:\Users\Gaming-PC\Laba_2.7>git merge develop
Updating 801b49b..a6328bb
Fast-forward
 ObshZadanie1.py | 18 ++++++
 ObshZadanie2.py | 19 ++++++
 individual.py   | 22 ++++++
 primer.py      | 18 ++++++
 4 files changed, 77 insertions(+)
 create mode 100644 ObshZadanie1.py
 create mode 100644 ObshZadanie2.py
 create mode 100644 individual.py
 create mode 100644 primer.py
```

Рисунок 7. Слияние ветки develop и main

Ссылка: https://github.com/EvgenyEvdaikov/Laba_2.7

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 = 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 и 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:

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

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

3. 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

Вывод: приобрел навыки по работе с множествами при написании программ с помощью языка программирования Python версии 3.x.