

# Лабораторная работа 2.7 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

Множеством в языке программирования Python называется неупорядоченная совокупность уникальных значений. В качестве элементов этого набора данных могут выступать любые неизменяемые объекты, такие как числа, символы, строки. В отличие от массивов и списков, порядок следования значений не учитывается при обработке его содержимого. Над одним, а также несколькими множествами можно выполнять ряд операций, благодаря функциям стандартной библиотеки языка программирования Python.

## Создание

Перед тем как начать работу с множеством, необходимо для начала его создать. Сделать это можно, просто присвоив переменной последовательность значений, выделив их фигурными скобками. Следующий пример показывает код, в котором создается множество целых чисел под названием `a`, после функция `print` выводит на экран его содержимое.

```
a = {1, 2, 0, 1, 3, 2}
print(a)

{0, 1, 2, 3}
```

Как можно заметить, все элементы полученной последовательности являются уникальными, без повторений. Существует и другой способ создания множеств, который подразумевает использование вызова `set`. Аргументом этой функции может быть набор неких данных или даже строка с текстом, как это показано в следующем примере.

```
a = set('data')
print(a)

{'d', 'a', 't'}
```

В результате выполнения этого кода, программа заполняет новое множество уникальными символами из входной строки. Содержимое набора также выводится на экран.

## Использование

Обычно используется для следующих операций:

- Проверка, есть ли данное значение в множестве. Для этого используется `in`.

```
a = {0, 1, 2, 3}
print(2 in a)

True
```

- Наоборот, проверка отсутствия. Используется `not in`.

```
a = {0, 1, 2, 3}
print(2 not in a)

False
```

- Перебор всех элементов.

```
for a in {0, 1, 2}:
    print(a)

0
1
2
```

## Set Comprehensions

Для создания множества можно в Python воспользоваться генератором, позволяющих заполнять списки, а также другие наборы данных с учетом неких условий.

Следующий код демонстрирует генерацию множества `a` с циклом `for` для нескольких чисел.

```
a = {i for i in [1, 2, 0, 1, 3, 2]}
print(a)

{0, 1, 2, 3}
```

Как и прежде, метод `print` показывает содержимое полученного набора значений, где нет повторений, поскольку одинаковые числа автоматически удаляются.

## Изменение множеств

Для управления содержимым множеств в языке Python присутствуют специальные методы, дающие возможность добавлять и удалять отдельные элементы.

### Получение размера

Узнать точное количество элементов, входящих в состав множества, поможет метод `len`, принимающий в качестве аргумента набор данных. Функция `print` выводит результат.

```
a = {0, 1, 2, 3}
print(len(a))

4
```

### Добавление элемента

Чтобы внести новые значения, потребуется вызывать метод `add`. Аргументом в данном случае будет добавляемый элемент последовательности. В примере кода на Python добавим в множество элемент со значением 4.

```
a = {0, 1, 2, 3}
a.add(4)
print(a)

{0, 1, 2, 3, 4}
```

### Удаление элемента

Для удаления элементов из множества используются следующие функции в Python (кроме очистки, которая будет рассмотрена ниже):

- `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.

Избавиться от лишних значений в наборе данных с помощью `remove`. В качестве входного параметра здесь выступает элемент, который нужно удалить (в примере удалим число со значением 3).

```
a = {0, 1, 2, 3}
a.remove(3)
print(a)

{0, 1, 2}
```

По поводу функции `pop` хотелось бы отметить, что так как множества не упорядочены, то удалится случайный элемент, который будет находиться в памяти первым. Но если они в памяти хранятся в отсортированном виде, что не факт, то скорее всего будет удален элемент с наименьшим значением. Но на это не стоит рассчитывать.

### Полная очистка

Иногда необходимо полностью убрать все элементы. Чтобы не удалять каждый элемент отдельно, используется метод `clear`, не принимающий аргументов. Если вывести содержимое после этой операции, на экране появится только его название.

```
a = {0, 1, 2, 3}
a.clear()
print(a)

set()
```

В результате получили пустое множество.

## Порядок элементов в множестве

Порядок следования элементов не учитывается. Поэтому нет смысла говорить о сортировке множеств в Python 3.

Но с другой стороны все дело обстоит не со всем так. Для быстрого поиска элемента, желательно их хранить в памяти в упорядоченном виде.

В начале рассмотрим, что будет с элементами разных типов данных в одном множестве. Такие элементы не должны сортироваться. Если мы будем выводить элементы с помощью команды print, то они выводятся примерно следующим образом:

```
a = {0, 1, 12, 'b', 'ab', 3, 2, 'a'}
print(a)

{0, 1, 'b', 3, 2, 12, 'ab', 'a'}
```

Как видим, у нас вывелись не отсортированные значения, если повторить запуск, то порядок будет меняться. Но это только в том случае, если перемешаны элементы разного типа.

Посмотрим, что будет, если попытаемся вывести только числа:

```
a = {0, 1, 12, 3, 2}
print(a)

{0, 1, 2, 3, 12}
```

Все элементы выведены упорядоченно. Теперь посмотрим что будет если преобразовать в список:

```
a = {0, 1, 12, 3, 2}
b = list(a)
print(b)

[0, 1, 2, 3, 12]
```

Аналогично, в список значения записались отсортированными по возрастанию.

**Получается, что элементы хранятся в памяти в упорядоченном виде, если они одного типа.** Но лучше не стоит на это рассчитывать, алгоритмы Python могут поменяться.

Если вам нужно получить отсортированный список из множества, лучше воспользоваться для верности функцией sort. Элементы будут точно отсортированы. Ваш код будет понятен для других.

## Операции над множествами

Помимо различных манипуляций с элементами множеств, существуют еще и операции над ними, позволяющие одной строчкой кода выполнять сложные преобразования.

Рассмотрим операции с множествами доступные в Python 3.

### Объединение

Чтобы объединить все элементы двух разных множеств, стоит воспользоваться методом union на одном из объектов. Следующий пример демонстрирует работу данной функции, где создается последовательность чисел под именем c.

```
a = {0, 1, 2, 3}
b = {4, 3, 2, 1}
c = a.union(b)
print(c)

{0, 1, 2, 3, 4}
```

## Добавление

Чтобы добавить все элементы из одного множества к другому, необходимо вызывать метод `update` на первом объекте. Таким образом можно перенести уникальные данные из одного набора чисел в другой, как это показано в следующем примере.

```
a = {0, 1, 2, 3}
b = {4, 3, 2, 1}
a.update(b)
print(a)

{0, 1, 2, 3, 4}
```

## Пересечение

Чтобы найти общие элементы для двух разных множеств, следует применить функцию `intersection`, принимающую в качестве аргумента один из наборов данных. Код, приведенный ниже, создает новую последовательность чисел из пересечения двух множеств в Python 3.

```
a = {0, 1, 2, 3}
b = {4, 3, 2, 1}
c = a.intersection(b)
print(c)

{1, 2, 3}
```

## Разность

Чтобы вычислить разность для двух разных множеств, необходимо воспользоваться методом `difference`. Функция позволяет найти элементы, уникальные для второго набора данных, которых в нем нет. Следующий код демонстрирует эту операцию.

```
a = {0, 1, 2, 3}
b = {4, 3, 2, 1}
c = a.difference(b)
print(c)

{0}
```

## Отношения между множествами

Для определения подмножеств и надмножеств существуют специальные функции, возвращающие `True` или `False` в зависимости от результата выполнения. Для определения принадлежности элемента к множеству используется оператор `in`. Для того, чтобы проверить, что элемент не входит в множество используется оператор `not in`.

### Определение подмножества

Чтобы выяснить, является ли множество `a` подмножеством `b`, стоит попробовать вывести на экран результат выполнения метода `issubset`, как в следующем примере. Так как не все элементы набора чисел `a` присутствуют в `b`, функция вернет `False`.

```
a = {0, 1, 2, 3, 4}
b = {3, 2, 1}
print(a.issubset(b))

False
```

### Определение надмножества

Чтобы узнать, является ли множество `a` надмножеством `b`, необходимо вызвать метод `issuperset` и вывести результат его работы на экран. Поскольку все элементы набора чисел `b` присутствуют в `a`, функция возвращает `True`.

```
a = {0, 1, 2, 3, 4}
b = {3, 2, 1}
print(a.issuperset(b))

True
```

## Тип frozenset

Множество, содержимое которого не поддается изменению имеет тип `frozenset`. Значения из этого набора нельзя удалить, как и добавить новые. В следующем примере демонстрируется создание при помощи стандартной функции.

```
a = frozenset({"hello", "world"})
print(a)

frozenset({'hello', 'world'})
```

Поскольку содержимое `frozenset` должно всегда оставаться статичным, перечень функций, с которыми такое множество может взаимодействовать, имеет ограничения.

## Преобразование множеств

Иногда возникает необходимость представления уже готовой последовательности значений в качестве совсем другого типа данных. Возможности языка позволяют конвертировать любое множество в строку, словарь или список при помощи стандартных функций.

### Строка

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. Запятая в кавычках выступает в качестве символа, разделяющего значения. Метод `type` возвращает тип данных объекта в конце приведенного кода.

```
a = {'set', 'str', 'dict', 'list'}
b = ','.join(a)
print(b)
print(type(b))

set,dict,list,str
<class 'str'>
```

### Словарь

Чтобы получить из множества словарь, следует передать функции `dict` набор из нескольких пар значений, в каждом из которых будет находиться ключ. Функция `print` демонстрирует на экране содержимое полученного объекта, а `type` отображает его тип.

```
a = {('a', 2), ('b', 4)}
b = dict(a)
print(b)
print(type(b))

{'b': 4, 'a': 2}
<class 'dict'>
```

Следует отметить, что каждый элемент для такого преобразования — кортеж состоящий из двух значений:

1. ключ будущего словаря;
2. значение, соответствующее ключу.

### Список

По аналогии с предыдущими преобразованиями можно получить список неких объектов. На этот раз используется вызов `list`, получающий в качестве аргумента множество `a`. На выходе функции `print` отображаются уникальные значения для изначального набора чисел.

```
a = {1, 2, 0, 1, 3, 2}
b = list(a)
print(b)
print(type(b))

[0, 1, 2, 3]
<class 'list'>
```

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

$$A = \{b, c, h, o\}; \quad B = \{d, f, g, o, v, y\}; \quad C = \{d, e, j, k\}; \quad D = \{a, b, f, g\}; \quad X = (A \cap B) \cup C; \quad Y = (A/D) \cup (\bar{C}/\bar{B}). \quad (1)$$

*Примечание:* в качестве универсального множества считать все строчные латинские буквы от *a* до *z*.

**Решение:** Напишем программу для решения поставленной задачи.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

if __name__ == "__main__":
    # Определим универсальное множество
    u = set("abcdefghijklmnopqrstuvwxyz")

    a = {"b", "c", "h", "o"}
    b = {"d", "f", "g", "o", "v", "y"}
    c = {"d", "e", "j", "k"}
    d = {"a", "b", "f", "g"}

    x = (a.intersection(b)).union(c)
    print(f"x = {x}")

    # Найдем дополнения множеств
    bn = u.difference(b)
    cn = u.difference(c)

    y = (a.difference(d)).union(cn.difference(bn))
    print(f"y = {y}")
```

Результаты выполнения программы:

```
x = {'k', 'd', 'o', 'j', 'e'}
y = {'g', 'f', 'o', 'c', 'y', 'h', 'v'}
```

## Аппаратура и материалы

1. Компьютерный класс общего назначения с конфигурацией ПК не хуже рекомендованной для ОС Windows 10 с подключением к глобальной сети Интернет.
2. Операционная система Windows 10.
3. Система контроля версий Git.
4. Браузер для доступа к web-сервису GitHub, рекомендован к использованию Google Chrome.
5. Дистрибутив языка программирования Python, включающий набор популярных библиотек Anaconda.
6. Интегрированная среда разработки PyCharm Community Edition.

## Указания по технике безопасности

При работе на ЭВМ без разрешения руководителя занятия запрещается:

- подавать (снимать) напряжение на ПЭВМ и электрические розетки с распределительного щита;
- включать и выключать блоки питания ПЭВМ и мониторы;
- извлекать ПЭВМ из защитного кожуха;
- устранять неисправности, возникшие в ходе выполнения лабораторной работы.

## Методика и порядок выполнения работы

1. Изучить теоретический материал работы.
2. Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и язык программирования Python.
3. Выполните клонирование созданного репозитория.
4. Дополните файл `.gitignore` необходимыми правилами для работы с IDE PyCharm.
5. Организуйте свой репозиторий в соответствие с моделью ветвления git-flow.
6. Создайте проект PyCharm в папке репозитория.
7. Проработайте примеры лабораторной работы. Создайте для них отдельный модуль языка Python. Зафиксируйте изменения в репозитории.
8. Решите задачу: подсчитайте количество гласных в строке, введенной с клавиатуры с использованием множеств.
9. Зафиксируйте сделанные изменения в репозитории.
10. Решите задачу: определите общие символы в двух строках, введенных с клавиатуры.
11. Зафиксируйте сделанные изменения в репозитории.
12. Добавьте отчет по лабораторной работе в *формате PDF* в папку *doc* репозитория. Зафиксируйте изменения.
13. Выполните слияние ветки для разработки с веткой *master/main*.
14. Отправьте сделанные изменения на сервер GitHub.
15. Отправьте адрес репозитория GitHub на электронный адрес преподавателя.

## Индивидуальное задание

Определить результат выполнения операций над множествами. Считать элементы множества строками. Проверить результаты вручную. Номер варианта задания необходимо получить у преподавателя.

1. 
$$\begin{aligned} A &= \{b, e, f, k, t\}; \quad B = \{f, i, j, p, y\}; \quad C = \{j, k, l, y\}; \quad D = \{i, j, s, t, u, y, z\}; \\ X &= (A \cap C) \cup (B \cap C); \quad Y = (A \cap \bar{B}) \cup (D/C). \end{aligned} \quad (2)$$
2. 
$$\begin{aligned} A &= \{b, c, h, i, j\}; \quad B = \{e, h, i, s, w\}; \quad C = \{a, b, j, k, l, m\}; \quad D = \{a, h, i, w, x\}; \\ X &= (A/C) \cap \bar{B}; \quad Y = (A \cap \bar{B}) \cup (C/D). \end{aligned} \quad (3)$$
3. 
$$\begin{aligned} A &= \{a, h, m, o, r\}; \quad B = \{j, k, o, u, y\}; \quad C = \{g, h, j\}; \quad D = \{g, j, q\}; \\ X &= (A \cap C) \cup (D \cap B); \quad Y = (A \cap B) \cup (D/C). \end{aligned} \quad (4)$$
4. 
$$\begin{aligned} A &= \{a, b, h, j, l\}; \quad B = \{b, c, h, l, r, v\}; \quad C = \{j, k, n, t, z\}; \quad D = \{b, i, k, v, w\}; \\ X &= (A \cup B) \cap C; \quad Y = (\bar{A} \cap \bar{B}) / (C \cup D). \end{aligned} \quad (5)$$
5. 
$$\begin{aligned} A &= \{c, e, h, n\}; \quad B = \{e, f, k, n, x\}; \quad C = \{b, c, h, p, r, s\}; \quad D = b, e, g; \\ X &= (A/B) \cap (C \cup D); \quad Y = (A \cap \bar{B}) \cup (C/D). \end{aligned} \quad (6)$$
6. 
$$\begin{aligned} A &= \{a, d, k, l, o, s\}; \quad B = \{d, e, k, s, u, x\}; \quad C = \{o, p, w\}; \quad D = \{d, n, r, y, z\}; \\ X &= (A/B) \cup (C \cap D); \quad Y = (\bar{A} \cap \bar{B}) / (C \cup D). \end{aligned} \quad (7)$$
7. 
$$\begin{aligned} A &= \{b, f, g, m, o\}; \quad B = \{b, g, h, l, u\}; \quad C = \{e, f, m\}; \quad D = \{e, g, l, p, q, u, v\}; \\ X &= (A \cap C) \cup B; \quad Y = (A \cap \bar{B}) \cup (C/D). \end{aligned} \quad (8)$$
8. 
$$\begin{aligned} A &= \{a, f, I, n, o\}; \quad B = \{f, g, o, p, z\}; \quad C = \{i, j, u, w\}; \quad D = \{f, h, n, t, u, y, z\}; \\ X &= (A \cap B) \cup C; \quad Y = (\bar{A} \cap \bar{B}) / (C \cup D). \end{aligned} \quad (9)$$
9. 
$$\begin{aligned} A &= \{a, e, f, i\}; \quad B = \{a, b, k, n\}; \quad C = \{e, f, n, o, w, x\}; \quad D = \{a, d, e, o, p, t, u\}; \\ X &= (A \cup B) \cap D; \quad Y = (\bar{A} \cap \bar{B}) / (C \cup D). \end{aligned} \quad (10)$$
10. 
$$\begin{aligned} A &= \{a, b, h, k, o, r\}; \quad B = \{b, g, h, l, s\}; \quad C = \{k, l, z\}; \quad D = \{g, j, p, q, u, v\}; \\ X &= (A \cap C) \cup B; \quad Y = (\bar{A} \cap \bar{B}) / (C \cup D). \end{aligned} \quad (11)$$
11. 
$$\begin{aligned} A &= \{a, h, k\}; \quad B = \{c, d, h, p, r\}; \quad C = \{h, i, s\}; \quad D = \{c, g, j, v, w\}; \\ X &= (A \cup B) \cap C; \quad Y = (\bar{A} \cap \bar{B}) / (C \cup D). \end{aligned} \quad (12)$$
12. 
$$\begin{aligned} A &= \{b, k, n, o, q\}; \quad B = \{a, b, k, u\}; \quad C = \{o, p\}; \quad D = \{a, m, n, y, z\}; \\ X &= (A \cup B) \cap D; \quad Y = (\bar{A} \cap D) \cup (C/B). \end{aligned} \quad (13)$$
13. 
$$\begin{aligned} A &= \{a, b, g, k, m, p\}; \quad B = \{b, e, f, l, r\}; \quad C = \{k, l, w, x\}; \quad D = \{e, j, o, p, q, u, v\}; \\ X &= (A/B) \cap (C \cup D); \quad Y = (\bar{A} \cap \bar{B}) / (C \cup D). \end{aligned} \quad (14)$$
14. 
$$\begin{aligned} A &= \{b, e, g, h, k, s\}; \quad B = \{c, g, p, q\}; \quad C = \{f, g, s, x, y, z\}; \quad D = \{a, c, d, , g, u, v, z\}; \\ X &= (A \cup B) \cap C; \quad Y = (\bar{A} \cap D) \cup (C/B). \end{aligned} \quad (15)$$
15. 
$$\begin{aligned} A &= \{c, m, n, o, q\}; \quad B = \{c, d, m, w\}; \quad C = \{m, n, q\}; \quad D = \{c, m, p\}; \\ X &= (A \cup B) \cap C; \quad Y = (A \cap \bar{B}) \cup (C/D). \end{aligned} \quad (16)$$
16. 
$$\begin{aligned} A &= \{b, d, f, g, l, u\}; \quad B = \{d, e, f, m, n, z\}; \quad C = \{h, i, r, x, y\}; \quad D = \{a, e, f, k, r, s, x\}; \\ X &= (A/B) \cap (C \cup D); \quad Y = (\bar{A} \cap D) \cup (C/B). \end{aligned} \quad (17)$$
17. 
$$\begin{aligned} A &= \{b, d, l, p\}; \quad B = \{b, d, e, l, p, x\}; \quad C = \{k, l, p, t\}; \quad D = \{d, k, o, p, q, u, v\}; \\ X &= (A/B) \cap (C \cup D); \quad Y = (A \cap \bar{B}) \cup (C/D). \end{aligned} \quad (18)$$
18. 
$$\begin{aligned} A &= \{b, c, g, I, w\}; \quad B = \{e, g, h, q, w\}; \quad C = \{c, d, k, l, y\}; \quad D = \{a, g, h, u, v, z\}; \\ X &= (A \cap C) \cup B; \quad Y = (\bar{A} \cap D) \cup (C/B). \end{aligned} \quad (19)$$
19. 
$$\begin{aligned} A &= \{a, b, f, g, i\}; \quad B = \{c, f, g, i, s, v\}; \quad C = \{a, g, h, i\}; \quad D = \{f, w, x\}; \\ X &= (A \cap B) \cup C; \quad Y = (A \cap \bar{B}) \cup (C/D). \end{aligned} \quad (20)$$
20. 
$$\begin{aligned} A &= \{c, g, h, k, y\}; \quad B = \{a, b, k, n, u\}; \quad C = \{i, j, o, y, z\}; \quad D = \{a, b, f, g, y, z\}; \\ X &= (A \cup B) \cap D; \quad Y = (\bar{A} \cap D) \cup (\bar{C}/\bar{B}). \end{aligned} \quad (21)$$
21. 
$$\begin{aligned} A &= \{c, g, h, i, j\}; \quad B = \{c, d, i, o, s\}; \quad C = \{i, j, r, z\}; \quad D = \{b, c, f, i, w, x\}; \\ X &= (A \cup B) \cap C; \quad Y = (A/D) \cup (\bar{C}/\bar{B}). \end{aligned} \quad (22)$$
22. 
$$\begin{aligned} A &= \{b, d, j, n, t, v\}; \quad B = \{f, g, j, r, t, x\}; \quad C = \{o, p, x\}; \quad D = \{a, f, m, s, x, y\}; \\ X &= (A \cap B) \cup C; \quad Y = (\bar{A} \cap D) \cup (C/B). \end{aligned} \quad (23)$$
23. 
$$\begin{aligned} A &= \{c, f, g, k\}; \quad B = \{e, f, g, m, q\}; \quad C = \{h, i, r, w, x\}; \quad D = \{b, e, j, u, v, z\}; \\ X &= (A/B) \cap (C \cup D); \quad Y = (A/D) \cup (\bar{C}/\bar{B}). \end{aligned} \quad (24)$$
24. 
$$\begin{aligned} A &= \{a, b, d, I, x\}; \quad B = \{d, e, h, i, n, u\}; \quad C = \{e, f, m, n\}; \quad D = \{a, c, h, k, r, s, w, x\}; \\ X &= (A/C) \cap \bar{B}; \quad Y = (\bar{A} \cap D) \cup (C/B). \end{aligned} \quad (25)$$
25. 
$$\begin{aligned} A &= \{a, e, g, o, p\}; \quad B = \{e, h, i, o, u\}; \quad C = \{g, h, p, s, t, w\}; \quad D = \{f, h, n, s, t, x, y\}; \\ X &= (A/C) \cap \bar{B}; \quad Y = (\bar{A} \cap \bar{B}) / (C \cup D). \end{aligned} \quad (26)$$

25.  $A = (A \cup C) \cap D; \quad I = (A \cap D) / (C \cup D).$
- $A = \{c, d, k, l, m, z\}; \quad B = \{b, c, d, n, w\}; \quad C = \{m, n, y\}; \quad D = \{b, j, l, r, s, w, x\};$  (27)
26.  $X = (A \cup D) \cap C; \quad Y = (A/D) \cup (\bar{C}/\bar{B}).$
- $A = \{a, b, c, d, e, r\}; \quad B = \{b, c, d, f, n, y\}; \quad C = \{b, c, h, k, l, s\}; \quad D = \{a, b, r, s, w, x\};$  (28)
27.  $X = (A \cup D) \cap C; \quad Y = (\bar{A} \cap D) \cup (C/B).$
- $A = \{c, f, h, l, o\}; \quad B = \{d, e, f, p, w\}; \quad C = \{j, k\}; \quad D = \{b, d, g, k, t, u, y, z\};$  (29)
28.  $X = (A/B) \cap (C/D); \quad Y = (A/D) \cup (\bar{C}/\bar{B}).$
- $A = \{a, b, c, e, t\}; \quad B = \{b, c, d, e, m, u\}; \quad C = \{b, c, f, g, h, u\}; \quad D = \{a, d, q, r, v, w\};$  (30)
29.  $X = (A/B) \cup (D/C); \quad Y = (\bar{A} \cap D) \cup (C/B).$

## Содержание отчета и его форма

Отчет по лабораторной работе оформляется электронно в формате PDF, должен содержать ответы на контрольные вопросы, ссылку на репозиторий с которым выполнялась работа, скриншоты IDE PyCharm, скриншоты результатов работы программ.

## Вопросы для защиты работы

1. Что такое множества в языке Python?
2. Как осуществляется создание множеств в Python?
3. Как проверить присутствие/отсутствие элемента в множестве?
4. Как выполнить перебор элементов множества?
5. Что такое set comprehension?
6. Как выполнить добавление элемента во множество?
7. Как выполнить удаление одного или всех элементов множества?
8. Как выполняются основные операции над множествами: объединение, пересечение, разность?
9. Как определить, что некоторое множество является надмножеством или подмножеством другого множества?
10. Каково назначение множеств `frozenset`?
11. Как осуществляется преобразование множеств в строку, список, словарь?