



Пример 3: Программа "Words"

- Цель примера

Продемонстрировать процесс создания простейшей программы с использованием `Set`.

- Введение

Структура данных `Set` отлично подходит в случае, если нам нужно запретить повторение дублирующихся элементов без разработки собственной логики.

- Практическое руководство

Рассмотрим программу, которая демонстрирует базовые методы интерфейса `Set`. Самая популярная реализация этого интерфейса – `HashSet`.

- Шаг 1.

Создадим класс `Main`, где создадим `Set` и заполним его именами фруктов при помощи метода `add()`:

```
public class Main {  
  
    public static void main(String[] args) {  
  
        Set<String> words = new HashSet<>();  
  
        System.out.println("words = " + words);  
  
        words.add("Apple");  
        words.add("Mango");  
        words.add("Pineapple");  
        words.add("Watermelon");  
  
        System.out.println("words = " + words);  
  
    }  
  
}
```

Результат работы программы:

```
words = [] words = [Apple, Watermelon, Mango, Pineapple]
```

Информация: Обратите внимание, что в `Set` необходимо указывать только хранимый тип данных.

- Шаг 2.

Добавим еще несколько элементов, среди них дубликаты того, что уже есть в `Set` :

```
public class Main {  
  
    public static void main(String[] args) {  
  
        Set<String> words = new HashSet<>();  
  
        System.out.println("words = " + words);  
  
        words.add("Apple");  
        words.add("Mango");  
        words.add("Pineapple");  
        words.add("Watermelon");  
  
        System.out.println("words = " + words);  
  
        System.out.println("words.size() = " + words.size());  
  
        words.add("Apple");  
        words.add("Pear");  
        words.add("Pineapple");  
        words.add("Orange");  
        words.add("Watermelon");  
  
        System.out.println("words = " + words);  
  
        System.out.println("words.size() = " + words.size());  
  
    }  
  
}
```

Результат работы программы:

```
words = [] words = [Apple, Watermelon, Mango, Pineapple] words.size() = 4 words = [Apple, Pear, Watermelon, Mango,  
Pineapple, Orange] words.size() = 6
```

!! Примечание: Дубликат в `Set` определяется при помощи методов `equals()` и `hashCode()` . За счет этого можно решать, какие объекты являются дубликатами, а какие – нет. Например, строки `apple` и `Apple` являются разными, следовательно обе строки можно добавить в один `Set` .

- Шаг 3.

Проверим содержимое множества, удалим единственный элемент и всю коллекцию сразу:

```
public class Main {  
  
    public static void main(String[] args) {  
  
        Set<String> words = new HashSet<>();  
  
        System.out.println("words = " + words);
```

```

words.add("Apple");
words.add("Mango");
words.add("Pineapple");
words.add("Watermelon");

System.out.println("words = " + words);

System.out.println("words.size() = " + words.size());

words.add("Apple");
words.add("Pear");
words.add("Pineapple");
words.add("Orange");
words.add("Watermelon");

System.out.println("words = " + words);

System.out.println("words.size() = " + words.size());

System.out.println();
System.out.println("-- Contains & Remove --");

System.out.println("words.contains(\"apple\") = " + words.contains("apple"));
System.out.println("words.contains(\"apple\") = " + words.contains("Apple"));

words.remove("Mango");

System.out.println("words = " + words);
System.out.println("words.size() = " + words.size());

words.clear();

System.out.println("words = " + words);
System.out.println("words.size() = " + words.size());

}

}

```

Результат работы программы:

```

words = [] words = [Apple, Watermelon, Mango, Pineapple] words.size() = 4 words = [Apple, Pear, Watermelon, Mango,
Pineapple, Orange] words.size() = 6
-- Contains & Remove -- words.contains("apple") = false words.contains("apple") = true words = [Apple, Pear,
Watermelon, Pineapple, Orange] words.size() = 5 words = [] words.size() = 0

```

• Рекомендации:

- Запустить программу и сравнить результаты;
- Изменить тип хранимых данных на свой тип и проверить, как определяются дубликаты;