



Пример 2: Программа "Book"

- Цель примера

Продемонстрировать возможные методы для сравнения объектов между собой.

- Поэтапное создание программы

Рано или поздно программист сталкивается с необходимостью сравнить объекты между собой. Важно помнить, что равенство объектов определяется по разным критериям:

1. "Ссылочное" равенство (`==`) – когда две переменные (ссылки) указывают на один и тот же объект в памяти ("*heap memory*");
2. "Структурное" равенство (`.equals()`) – когда два объекта частично или полностью равны между собой по свойствам;

В обоих случаях результатом операции сравнения будет `boolean`, результат которого можно использовать в условных выражениях.

⚠ **Важно:** Для каждого класса необходимо переопределять метод `equals()`, чтобы конкретизировать набор свойств, по которым объекты должны совпадать.

⚠ **Важно:** Даже если объекты полностью структурно идентичны, нету гарантии того что это один и тот же объект с точки зрения ссылки.

⚠ **Важно:** При структурном сравнении объектов учитываются только свойства. Набор методов в сравнении не участвует.

В рамках примера рассмотрим простой класс – "книга" `Book`, который обладает следующими свойствами:

- Название
- Автор
- Жанр

– Шаг 1.

Создадим класс `Book` в соответствии с характеристиками из условия:

```
public class Book {  
  
    private String title;  
    private String author;  
    private String genre;  
  
    public Book(String title, String author, String genre) {  
        this.title = title;  
        this.author = author;  
        this.genre = genre;  
    }  
}
```

```

    }

    public String getTitle() {
        return title;
    }

    public String getAuthor() {
        return author;
    }

    public String getGenre() {
        return genre;
    }
}

```

- Шаг 2.

Продemonстрируем работу операций сравнения:

```

public class BookDemo {

    public static void main(String[] args) {

        Book book1 = new Book("Head First Java", "O'Reilly", "Technology");
        Book book2 = new Book("Head First Java", "O'Reilly", "Technology");

        System.out.println("book1 == book1 = " + (book1 == book1));
        System.out.println("book1 == book2 = " + (book1 == book2));
        System.out.println("book1.equals(book1) = " + book1.equals(book1));
        System.out.println("book1.equals(book2) = " + book1.equals(book2));

    }

}

```

Результат работы программы:

```

book1 == book1 = true book1 == book2 = false book1.equals(book1) = true book1.equals(book2) = false

```

Как видно, сравнение книги самой с собой (`book1 == book1`) будет `true` , поскольку это один и тот же объект в памяти.

Объекты (`book1 == book2`) не равны, поскольку это разные объекты в памяти.

Структурно первая книга равна самой себе, поскольку по умолчанию метод `equals()` сравнивает ссылочно.

Первая и вторая книги не равны структурно, несмотря на одинаковое состояние, потому что метод `equals()` не переопределен.

- Шаг 3.

Переопределим метод `equals()` в классе `Book` :

```

import java.util.Objects;

```

```

public class Book {

    private String title;
    private String author;
    private String genre;

    public Book(String title, String author, String genre) {
        this.title = title;
        this.author = author;
        this.genre = genre;
    }

    public String getTitle() {
        return title;
    }

    public String getAuthor() {
        return author;
    }

    public String getGenre() {
        return genre;
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;
        Book book = (Book) o;
        return Objects.equals(title, book.title) &&
            Objects.equals(author, book.author) &&
            Objects.equals(genre, book.genre);
    }

}

```

– Шаг 4.

Запустим демонстрацию и сравним результат:

```

public class BookDemo {

    public static void main(String[] args) {

        Book book1 = new Book("Head First Java", "O'Reilly", "Technology");
        Book book2 = new Book("Head First Java", "O'Reilly", "Technology");

        System.out.println("book1 == book1 = " + (book1 == book1));
        System.out.println("book1 == book2 = " + (book1 == book2));
        System.out.println("book1.equals(book1) = " + book1.equals(book1));
        System.out.println("book1.equals(book2) = " + book1.equals(book2));

    }

}

```

Результат работы программы:

```
book1 == book1 = true book1 == book2 = false book1.equals(book1) = true book1.equals(book2) = true
```

Теперь, как видно, обе книги равны также структурно, поскольку их свойства идентичны.

• Рекомендации:

- Запустить программу и сравнить результаты;
- Переопределить метод `equals()` таким образом, чтобы при учитывались не все свойства;
- Попробовать создать переменную, которая содержит ссылку на один из объектов и проверить их равенство;