1. Что такое перечисления?

В разработке бывают ситуации, когда есть потребность в использовании набора **констант, которые логически связаны между собой.**

Например, если вы пишете приложение для покупки авиабилетов, вам нужно хранить набор кодов аэропортов. Использовать для этого базу данных нет большой потребности, так как это просто группа строк, а вот хранить их в формате перечислений очень удобно для быстрого доступа к ним.

Именно для таких случаев в пятой версии Java были введены перечисления.

> Чтобы создать перечисление, необходимо воспользоваться **ключевым словом enum**, а внутри фигурных скобок через запятую перечислить нужные константы.

Самым стандартным примером для того, чтобы познакомиться с перечислениями, являются дни недели.

```
// Перечисления можно создать как отдельный "класс",
// только при создании нужно выбрать enum
enum Day {
   MONDAY,
   TUESDAY,
   WEDNESDAY,
   THURSDAY,
   FRIDAY,
   SATURDAY,
   SUNDAY
}
```

Для реализации списка дней недели можно было бы создать отдельный класс и наполнить его переменными типа

`String` . Однако в этом случае значения класса не были бы защищены: туда можно было бы добавить что-то лишнее.

Когда-то эту проблему решали с помощью создания класса с набором публичных объектов и приватным конструктором. Однако такой подход куда менее удобен,

чем появившиеся в Java 1.5 перечисления.

> Перечисления максимально лаконичны, а главное — имеют свой особенный набор методов, о которых мы поговорим далее.

Поскольку перечисления по своей сути представляют из себя отдельный тип, мы можем просто создать переменную данного типа и присвоить ей одно из значений нашего перечисления.

Создадим класс, в котором создадим переменную типа нашего перечисления.

```
public class Program {
    public static void main(String[] args) {
        // Переменную ссылаем на конкретную константу
        // через имя перечисления
        Day current = Day.MONDAY;
        System.out.println(current); // MONDAY
    }
}

enum Day {
    MONDAY,
    TUESDAY,
    WEDNESDAY,
    THURSDAY,
    FRIDAY,
    SATURDAY,
    SUNDAY
}
```

Использование перечислений в операторе switch

> Одним из основных свойств перечислений является то, что **их очень удобно использовать для хранения строковых данных** — ввиду быстрого доступа к этим строкам. При этом к константам перечисления можно будет обратиться через **оператор switch**.

Мы можем создать перечисление и положить в него определенные данные, например жанры книг. Также мы можем создать класс `Book` и объявить в нем, помимо остальных полей, переменную, которая имеет тип нашего перечисления.

Теперь благодаря оператору `switch` мы можем определить, к какому жанру относится та или иная книга.

> В выражениях ветвей case **имена констант указываются без уточнения имени их перечисляемого типа**, так как тип перечисления в операторе switch уже неявно **задает тип enum для операторов case**.

Создадим класс, который содержит поле типа нашего перечисления.

```
class Book {
    String name;
    Genre genre;
    String author;

    Book(String name, String author, Genre genre) {
        this.genre = genre;
        this.name = name;
        this.author = author;
    }
}

// Создадим перечисление
enum Genre {
    HORROR,
    FICTION,
    FANTASY,
    FAIRY_TALE
}
```

Создадим класс с методом `main` .

```java
public class Test {
   public static void main(String[] args) {

      // Создадим объект класса Book
      Book book = new Book("Dracula", "Bram Stoker", Genre.HORROR);
      System.out.println("Book " + book.name + " has a type " + book.genre);

      // Объявим конструкцию switch
      // В параметре указываем поле, объекты которого будем перебирать
      switch(book.genre){
         // В блоках case указываем варианты реализаций enum'а
         case FICTION:
            System.out.println("Fiction");
            break;
         case HORROR:
            System.out.println("Horror");
            break;
         case FAIRY_TALE:
            System.out.println("Fairy tale");
            break;
         case FANTASY:
            System.out.println("Fantasy");
            break;
      }
   }
}
```