

Вопросы по теме Перечисление:

1. Может ли Enum реализовывать (implement) интерфейс?
2. Может ли Enum наследовать (extends) класс?
3. Можем ли мы переопределить метод toString() для Enum? Что будет, если не будем переопределять?
4. Можем ли мы создать конструктор внутри Enum?
5. Какая разница сравнивать Enum при помощи == или метода equals()?
6. Можно ли использовать Enum с TreeSet или TreeMap в Java?

Перечисление (Enum).

Перечисление (Enum) - это специальный тип данных, который позволяет переменной быть набором predefined констант. Переменная должна быть равна одному из predefined для нее значений.

Перечисление – это класс, который неявно расширяет (наследуется от) класс java.lang.Enum. Отсюда и вытекают, что перечисление, как и любой другой класс:

– может реализовывать интерфейсы, но не может наследоваться от других классов (так как уже наследуется от java.lang.Enum);

– конструктор java.lang.Enum выглядит так:

```
protected Enum(String name, int ordinal) {  
    this.name = name;  
    this.ordinal = ordinal; }
```

Это единственный конструктор. Программист не может его вызвать. Он предназначен для использования кодом, выдаваемым компилятором в ответ на объявления класса Enum.

name – имя константы перечисления

ordinal – порядковый номер константы перечисления

– может переопределять метод toString() (по умолчанию этот метод возвращает переменную name (имя константы));

– можем создать свой собственный конструктор, только с модификатором доступа private или package-private;

Поскольку существует только один экземпляр каждой константы перечисления, разрешается использовать оператор == вместо метода equals() при сравнении двух ссылок на объекты, если известно, что хотя бы одна из них ссылается на константу перечисления. Метод equals() определен в классе java.lang.Enum следующим образом:

```
public final boolean equals(Object other) { return this==other; }
```

Перечисление (Enum) наследует интерфейс Comparable, который является главным требованием для использования в упорядоченных коллекциях, поэтому он может использоваться с TreeSet и TreeMap. Также может использоваться с оператором выбора switch case.