Java

Классы перечислений

Лекция #17

Пустовалова О.Г.
доцент. каф. мат.мод.
ИММиКН ЮФУ

Содержание

Перечисления

Класс java.lang.Enum

Перечисления - Enum

Enum — это Java-класс, который предоставляет ограниченный набор значений-объектов.

- Перечисление в Java относится к типу класса, но перечисление НЕ может наследоваться от другого класса и НЕ может быть суперклассом.

- Все перечисления автоматически наследуют от класса **java.lang.Enum**

Объявление перечислений. Внутри класса

Перечисления могут быть объявлены: **внутри класса**

```
public class Main {
    public enum myEnum {
        WINTER,
        SUMMER,
        SPRING,
        AUTUMN;
    }
    public static void main(String[] args) {
        System.out.println(myEnum.SUMMER);
    }
}
```

рекомендуется писать перечисления большими буквами

SUMMER

Объявление перечислений. switch

```
public class Main {
    enum myEnum {
        WINTER,
        SUMMER,
        SPRING,
        AUTUMN;
    }
    public static void main(String[] args) {

        myEnum arg = myEnum.SPRING;

        switch (arg)
        {
            case WINTER:
                System.out.println("It's winter!"); break;
            case SUMMER:
                System.out.println("It's summer!"); break;
            case SPRING:
                System.out.println("It's spring!"); break;
            case AUTUMN:
                System.out.println("It's autumn!"); break;
        }    }}
```

объект типа **enum**

Объявление перечислений. Цикл foreach. values()

```
public class Main {
    enum myEnum {
        WINTER,
        SUMMER,
        SPRING,
        AUTUMN;
    }
    public static void main(String[] args) {

        for (myEnum x : myEnum.values()) {
            System.out.println(x);
        }
    }
}
```

values() - возвращает **массив**, содержащий список констант перечислимого типа.

```
WINTER
SUMMER
SPRING
AUTUMN
```

Объявление перечислений. valueOf()

```
public class Main {
    enum myEnum {
        WINTER,
        SUMMER,
        SPRING,
        AUTUMN;
    }
    public static void main(String[] args) {

        System.out.println(myEnum.valueOf("SUMMER"));
    }
}
```

valueOf()

- получение значения перечисления по его строковому представлению

SUMMER

Объявление перечислений. ordinal()

```
public class Main {
    enum myEnum {
        WINTER,  // 0
        SUMMER,  // 1
        SPRING,  // 2
        AUTUMN;  // 3
    }
    public static void main(String[] args) {

System.out.println(myEnum.valueOf("SUMMER").ordinal());
    }
}
```

ordinal() - возвращает порядковый номер

1

Объявление перечислений

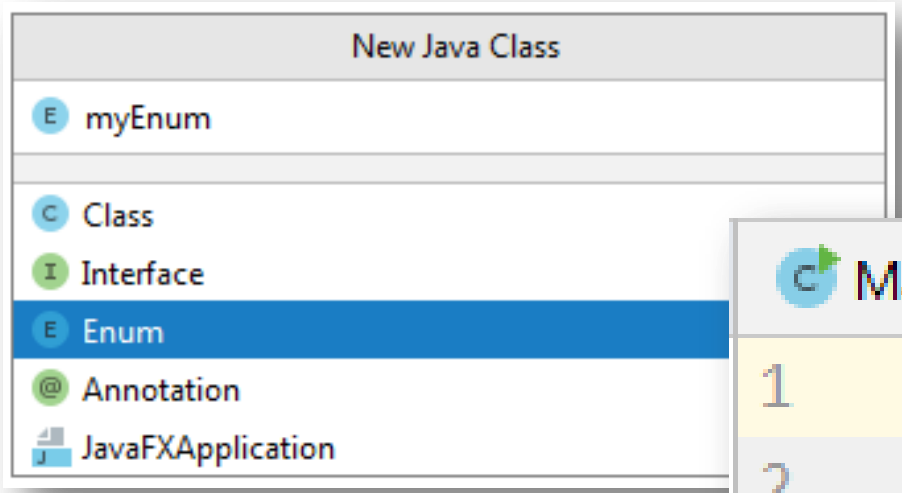
- Перечисления **НЕ могут** быть объявлены в методе

```
3   public void myMethod(){
4       enum myEnum {
5           WINTER,
6           SUMMER,
7           SPRING,
8           AUTUMN;
9
10      }
11  }
```

```
C:\Users\Oleandr\IdeaProjects\MyEnum\src\Main.java
Error:(4, 5)  java: enum types must not be local
```

Объявление перечислений. Отдельный класс

- Перечисления могут быть объявлены: **отдельным классом**



Main.java myEnum.java

```
1  public enum myEnum {
2      WINTER,
3      SUMMER,
4      SPRING,
5      AUTUMN;
6  }
```

Объявление перечислений. Отдельный класс

Main.java myEnum.java

```
1  public enum myEnum {
2      WINTER,
3      SUMMER,
4      SPRING,
5      AUTUMN;
6  }
```

Main.java myEnum.java

```
1  public class Main {
2      public static void main(String[] args) {
3
4          System.out.println(myEnum.SUMMER);
5      }
6  }
```

Конструкторы, поля и методы в классе перечислений

```
enum Color {

    RED("#FF0000"), BLUE("#0000FF"), GREEN("#00FF00");

    private String code;

    Color(String code) {
        this.code = code;
    }

    public String getCode() {
        return code;
    }
}
```

Перечисления, как и обычные классы, могут определять конструкторы, поля и методы.

Конструкторы, поля и методы в классе перечислений

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

        System.out.println(Color.RED.getCode());        // #FF0000
        System.out.println(Color.GREEN.getCode());      // #00FF00
    }
}
```

Пример 2. Часть 1

```
public enum Month {

    JANUARY("Январь", 31),
    FEBRUARY("Февраль", 28),
    MARCH("Март", 31),
    APRIL("Апрель", 30),
    MAY("Май", 31),
    JUNE("Июнь", 30),
    JULY("Июль", 31),
    AUGUST("Август", 31),
    SEPTEMBER("Сентябрь", 30),
    OCTOBER("Октябрь", 31),
    NOVEMBER("Ноябрь", 30),
    DECEMBER("Декабрь", 31);
```

Пример 2. Часть 2

```
    private String name;
    private int daysCount;

    Month(String name, int daysCount) {
        this.name = name;
        this.daysCount = daysCount;
    }

    public static Month[] getWinterMonths() {

        return new Month[]{DECEMBER, JANUARY, FEBRUARY};
    }

    @Override
    public String toString() {
        return "Month{" +
                "name='" + name + '\'' +
                ", daysCount=" + daysCount +
                '}';
    }}
```

Пример 2. Часть 3

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

        for (Month m: Month.values()) {
            System.out.println(m.toString());
        }
    }
}
```

```
Month{name='Январь', daysCount=31}
Month{name='Февраль', daysCount=28}
Month{name='Март', daysCount=31}
Month{name='Апрель', daysCount=30}
Month{name='Май', daysCount=31}
Month{name='Июнь', daysCount=30}
Month{name='Июль', daysCount=31}
Month{name='Август', daysCount=31}
Month{name='Сентябрь', daysCount=30}
Month{name='Октябрь', daysCount=31}
Month{name='Ноябрь', daysCount=30}
Month{name='Декабрь', daysCount=31}
```

Методы класса java.lang.Enum

Enum переопределяет базовые методы класса Object

Enum переопределяет базовые методы класса **Object.**

- **ordinal() -** значение, которое обозначает **позицию** константы в списке констант перечислимого типа
- **compareTo() - сравнение** порядковых значений двух констант одного и того же перечислимого типа
- **equals() - сравнение** на равенство константы перечисления с любым другим объектом
- **values() -** возвращает **массив**, содержащий список констант перечислимого типа.
- **valueOf() -** возвращает **константу** перечислимого типа.

Класс java.lang.Enum. equals & compare

equals

```
System.out.println(Month.DECEMBER.equals(Month.DECEMBER)); // true

System.out.println(Month.DECEMBER.equals(Month.JULY));     // false
```

compare

```
System.out.println(Month.MARCH.compareTo(Month.JANUARY)); // 2

System.out.println(Month.JANUARY.compareTo(Month.MARCH)); // -2

System.out.println(Month.MARCH.compareTo(Month.MARCH));   // 0
```

Класс java.lang.Enum. ==

```
enum Color { BLACK, WHITE };
enum Chiral { LEFT, RIGHT };


// Компилируется
if (Color.BLACK.equals(Chiral.LEFT));

// НЕ КОМПИЛИРУЕТСЯ!!! Несовместимые типы!
if (Color.BLACK == Chiral.LEFT);
```

Сравнения через == проверяются на соответствия типов во время компиляции (это хорошо, т.к. поможет вовремя выявить ошибку)

Класс java.lang.Enum. Переопределение методов

```
@Override
    public String toString() {
        return "Month{" +
                "name='" + name + '\'' +
                ", daysCount=" + daysCount +
                '}';
    }}
```

Для перечислений можно переопределять методы.

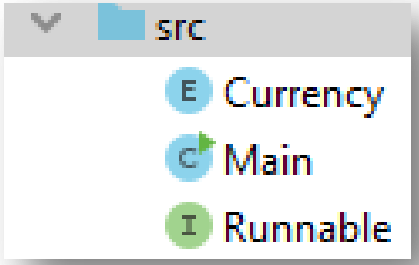
```
JANUARY
FEBRUARY
MARCH
APRIL
MAY
JUNE
JULY
AUGUST
SEPTEMBER
OCTOBER
NOVEMBER
DECEMBER
```

@Override

```
Month{name='Январь', daysCount=31}
Month{name='Февраль', daysCount=28}
Month{name='Март', daysCount=31}
Month{name='Апрель', daysCount=30}
Month{name='Май', daysCount=31}
Month{name='Июнь', daysCount=30}
Month{name='Июль', daysCount=31}
Month{name='Август', daysCount=31}
Month{name='Сентябрь', daysCount=30}
Month{name='Октябрь', daysCount=31}
Month{name='Ноябрь', daysCount=30}
Month{name='Декабрь', daysCount=31}
```

Класс java.lang.Enum. Перечисления и интерфейсы

- Перечисления **не могут наследовать другие классы, но могут реализовывать интерфейсы**.




```
public interface Runnable {
    public void run();
}
```

Класс java.lang.Enum. Перечисления и интерфейсы

```
public enum Currency implements Runnable {

    PENNY(1), NICKLE(5), DIME(10), QUARTER(25);
    private int value;

    Currency(int value) {
        this.value = value;
    }

    @Override
    public void run() {
        System.out.println("Перечисления в Java
                            могут реализовывать интерфейсы");
    }
}
```

Класс java.lang.Enum. Перечисления и интерфейсы

```
public class Main {

    public static void main(String[] args) {

        for (Currency p: Currency.values()) {

            System.out.println(p);

        }

    }

}
```

```
PENNY
NICKLE
DIME
QUARTER
```

Спасибо за внимание!