

Orientação a Objetos Enumerações

Profa. Karen Selbach Borges



Enumerações

- Vem para facilitar a manipulação de constantes numéricas.

Antes

```
public abstract class Tipo{  
    public static final int JANEIRO = 1;  
    public static final int FEVEREIRO = 2;  
    public static final int MARCO = 3;  
    public static final int ABRIL= 4;  
    public static final int MAIO= 5;  
    public static final int JUNHO= 6;  
    public static final int JULHO= 7;  
    public static final int AGOSTO= 8;  
    ....  
}
```



Depois ...

```
enum Mes {  
    JANEIRO, FEVEREIRO, MARCO, ABRIL, MAIO,  
    JUNHO, JULHO, AGOSTO, SETEMBRO, OUTUBRO,  
    NOVENBRO, DEZEMBRO  
}
```



Importante saber

- Tipos enumerados herdam da classe Enum, que possui alguns métodos úteis como `name()` e `ordinal()`, que retornam o nome da constante e sua posição na estrutura respectivamente.

```
enum Mes {
    JANEIRO, FEVEREIRO, MARCO, ABRIL, MAIO,
    JUNHO, JULHO, AGOSTO, SETEMBRO, OUTUBRO,
    NOVENBRO, DEZEMBRO
}

public class Principal{
    public static void main(String[] args){
        Mes m = Mes.JANEIRO;
        System.out.println("mes = " + m.name());
        System.out.println("mes ordinal = "+ m.ordinal());
    }
}
```

```
mes = JANEIRO
mes ordinal = 0
```




Importante saber

- As enumerações permitem o acréscimo de métodos próprios

```
enum Mes {
    JAN, FEV, MAR, ABR, MAI, JUN,
    JUL, AGO, SET, OUT, NOV, DEZ;

    public String getDescricao() {

        switch(this) {
            case JAN: return "Janeiro";
            case FEV: return "Fevereiro";
            case MAR: return "Março";
            case ABR: return "Abril";
            case MAI: return "Maio";
            case JUN: return "Junho";
            case JUL: return "Julho";
            case AGO: return "Agosto";
            case SET: return "Setembro";
            case OUT: return "Outubro";
            case NOV: return "Novembro";
            case DEZ: return "Dezembro";
            default : return "";
        }
    }
}
```

```
public class Principal{  
    public static void main(String[] args){  
        String mes = Mes.JAN.getDescricao();  
        System.out.println("Mes = " + mes);  
    }  
}
```

```
Mes = Janeiro
```



Percorrendo enumerações

- Os valores Enum tem um método estático chamado **values** que retorna uma matriz contendo todos os valores do enum na ordem em que são declarados.
- Este método é normalmente usado em combinação com o for para construir cada repetição dos valores de um tipo de enumeração.



```
public class Principal{
```

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

```
        for(Mes m : Mes.values()){
```

```
            System.out.print("Mes " + m + " = " +  
                             m.getDescricao() + "\n");
```

```
        }
```

```
    }
```

```
}
```

```
Mes JAN = Janeiro  
Mes FEV = Fevereiro  
Mes MAR = Março  
Mes ABR = Abril  
Mes MAI = Maio  
Mes JUN = Junho  
Mes JUL = Julho  
Mes AGO = Agosto  
Mes SET = Setembro  
Mes OUT = Outubro  
Mes NOV = Novembro  
Mes DEZ = Dezembro
```

Comparando enumerações

- Um Enum pode ser comparado com outro objeto através do método **equals**.

```
public class Principal{
    public static void main (String[] args){
        for(Mes m : Mes.values()){
            if (m.equals(Mes.MAR))
                System.out.print(m.getDescricao() +" marca o fim do verão e início do outono \n");
            if (m.equals(Mes.JUN))
                System.out.print(m.getDescricao() +" marca o fim do outono e início do inverno \n");
            if (m.equals(Mes.SET))
                System.out.print(m.getDescricao() +" marca o fim do inverno e início da primavera \n");
            if (m.equals(Mes.DEZ))
                System.out.print(m.getDescricao() +" marca o fim da primavera e início do verão \n");
        }
    }
}
```



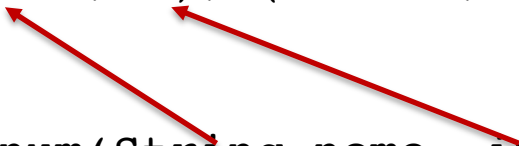
Alguns cuidados


- As instâncias dos tipos enum são criadas e nomeadas junto com a declaração da classe, sendo fixas e imutáveis (o valor é fixo).;
- Não é permitido criar novas instâncias com a palavra chave new;
- Seguindo a convenção, por serem objetos constantes e imutáveis (static final), os nomes declarados recebem todas as letras em MAIÚSCULAS;
- As instâncias dos tipos enum devem obrigatoriamente ter apenas um nome;
- Opcionalmente, a declaração da classe pode incluir variáveis de instância, construtor, métodos de instância, de classe, etc.

Construtores de Enums

- O construtor é declarado private, embora não precise de modificador private explícito
- Deve-se colocar como parâmetros do construtor as variáveis que irão conter os dados declarados na criação das constantes.

```
enum CartasEnum {  
  
    J("Valete", 11), Q("Dama", 12), K("Rei", 13), A("Ás", 14);  
  
    CartasEnum(String nome, int valor){ ... } //Construtor
```





```
enum CartasEnum {
    J("Valete",11),Q("Dama",12),K("Rei",13),A("Ás",14);

    public int valorCarta;
    public String nomeCarta;

    private CartasEnum(String nome,int valor) {
        nomeCarta = nome;
        valorCarta = valor;
    }

    public int getValor(){
        return valorCarta;
    }

    public String getNome(){
        return nomeCarta;
    }
}
```

```
public class Principal{  
    public static void main(String[] args){  
        String carta = CartasEnum.J.getNome();  
        System.out.println("Carta = " + carta);  
        int valor = CartasEnum.J.getValor();  
        System.out.println("Valor = " + valor);  
    }  
}
```

```
Carta = Valete  
Valor = 11
```