

APIs Java: Enum e Tipos genéricos

POO29004 – Engenharia de Telecomunicações

Prof. Emerson Ribeiro de Mello

<http://docente.ifsc.edu.br/mello/poo>

29 DE MAIO DE 2020



**INSTITUTO
FEDERAL**
Santa Catarina

Câmpus
São José

Enum

- O tipo Enum é usado para **definir um conjunto fixo de constantes**
 - Exemplo: estações do ano, dias da semana, meses do ano, planetas do sistema solar
- Antes da introdução do Enum no JDK1.5, um padrão de projeto era declarar um grupo de constantes inteiras

```
1 public class Teste{  
2     public static final int DOMINGO = 0;  
3     public static final int SEGUNDA = 1;  
4     public static final int TERCA    = 2;  
5     // .....  
6     public void aulas(){  
7         int aulasPoo[] = {SEGUNDA, TERCA};  
8     }  
9 }
```



Enum para dias da semana

```
10 // Arquivo DiasDaSemana.java
11 public enum DiasDaSemana{
12     DOMINGO, SEGUNDA, TERCA, QUARTA, QUINTA, SEXTA, SABADO
13 }
```

```
14 // Arquivo Teste.java
15 public class Teste{
16     public void aulas(){
17         DiasDaSemana aulasPoo[] = {DiasDaSemana.SEGUNDA, DiasDaSemana.TERCA};
18     }
19 }
```

Definição

Classes que exportam uma única instância para cada constante enumerada via um atributo público, estático e final.



Enum para cores da Shell

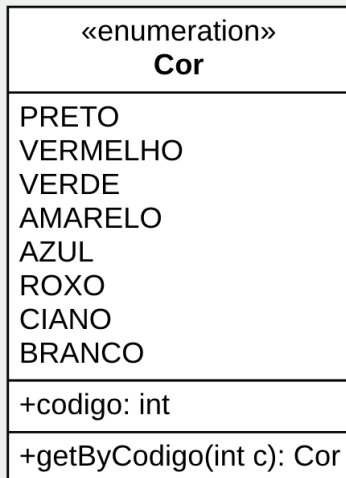
```
20 public enum Cor{
21     PRETO(0), VERMELHO(1), VERDE(2), AMARELO(3),
22     AZUL(4), ROXO(5), CIANO(6), BRANCO(7);
23
24     public final int codigo;
25
26     Cor(int c){
27         this.codigo = c;
28     };
29
30     public static Cor getByCodigo(int c){
31         for (Cor cor: Cor.values()){
32             if (c == cor.codigo){
33                 return cor;
34             }
35         }
36         throw new IllegalArgumentException("código inválido");
37     }
38 }
```



Enum para cores da Shell

```
39 public class Teste{
40
41     public static void main(String[] args) {
42         Cor corDoFundo = Cor.getByCodigo(0);
43         Cor corDoTexto = Cor.getByCodigo(7);
44
45         System.out.println("Cor do fundo: " + corDoFundo);
46
47         switch(corDoFundo){
48             case PRETO:
49                 switch(corDoTexto){
50                     case PRETO:
51                         System.out.println("não me parece uma boa combinação");
52                         break;
53                     case BRANCO:
54                         System.out.println("me parece uma boa combinação");
55                         break;
56                 }
57             }
58 }
```





- Fazer um Enum para os planetas do sistema solar
- Para cada planeta é importante guardar qual a sua posição em relação ao sol (e.g. A Terra está na posição 3)



Tipos genéricos em Java

Classe capaz de armazenar instâncias de qualquer classe

```
59 public class Caixa{  
60     private Object dado;  
61  
62     public void set(Object obj){  
63         this.dado = obj;  
64     }  
65  
66     public Object getDado(){ return this.dado;}  
67 }
```

```
68 public class Principal{  
69     public static void main(String[] args){  
70         Caixa c = new Caixa();  
71         String s = "Olá mundo";  
72         c.set(s);  
73         String outra = (String) c.getDado(); // typecasting obrigatório  
74     }  
75 }
```



Classe capaz de armazenar instâncias de qualquer classe

```
Caixa c = new Caixa();
Caixa d = new Caixa();
String s = "Olá mundo";
Pessoa p = new Pessoa("Joao");
c.set(s); // OK
d.set(p); // OK
String nova = (String) c.getDado();
// só apresentará erro durante a execução
String outra = (String) d.getDado(); // ERRO durante execução!!
```

- É mais fácil detectar *bugs* em tempo de compilação do que em tempo de execução.



Classe capaz de armazenar instâncias de qualquer classe

```
Caixa c = new Caixa();  
Caixa d = new Caixa();  
String s = "Olá mundo";  
Pessoa p = new Pessoa("Joao");  
c.set(s); // OK  
d.set(p); // OK  
String nova = (String) c.getDado();  
// só apresentará erro durante a execução  
String outra = (String) d.getDado(); // ERRO durante execução!!
```

- É mais fácil detectar *bugs* em tempo de compilação do que em tempo de execução.

Tipos genéricos

Possibilitam que bugs possam ser detectados já na fase de compilação



- Permite que tipos (Classes ou Interfaces) sejam passados como parâmetros durante a definição de Classes, Interfaces ou métodos
- **Parâmetros de tipos permite reutilizar a mesma classe** diante de diferentes entradas
- Entradas de parâmetros formais são obrigatoriamente valores

```
public void metodo(int parametro){ .... }  
....  
objeto.metodo(123); // parâmetro deve ser um valor
```

- **Entradas de parâmetros de tipos** são tipos
 - Exemplo de tipos: String, Pessoa, Carro, etc.



Reescrevendo classe Caixa para usar tipos genéricos

```
public class Caixa<T>{  
    private T dado;  
  
    public void set(T obj){  
        this.dado = obj;  
    }  
  
    public T getDado(){ return this.dado;}  
}
```

```
public class Principal{  
    public static void main(String[] args){  
        Caixa<String> c = new Caixa<>();  
        String s = "Olá mundo";  
        c.set(s);  
        String outra = c.getDado(); // não precisa do typecasting  
    }  
}
```



Reescrevendo classe Caixa para usar tipos genéricos

```
public class Caixa<T>{  
    private T dado;  
  
    public void set(T obj){  
        this.dado = obj;  
    }  
  
    public T getDado(){ return this.dado;}  
}
```

```
Caixa<String> c = new Caixa<>();  
  
Pessoa p = new Pessoa("Joao");  
  
c.set(p); // erro de compilação, tipos errados
```



Parâmetro de tipo limitado

- É possível restringir quais tipos são aceitos
 - Somente as classes ou interfaces da hierarquia seriam permitidas

```
public class Caixa<T extends Personagem>{  
    private T personagem;  
    .....  
}
```

```
public class Principal{  
    public static void main(String[] args){  
  
        // ok, pois Aldeao herda de Personagem  
        Caixa<Aldeao> c = new Caixa<>();  
  
        Caixa<String> n = new Caixa<>();// erro!  
    }  
}
```



- Nomes dos parâmetros de tipos devem ser uma única letra maiúscula
 - E – para elementos
 - T – para tipos
 - K – para chaves
 - V – para valores
 - N – para números
- A API Collections do Java faz uso do nome *E*
- Veja mais informações na documentação oficial
<https://docs.oracle.com/javase/tutorial/java/generics>

