

Enums



Enum, é um tipo especial de classe, onde os objetos são previamente criados, imutáveis e disponíveis por toda aplicação.

Usamos Enum, quando o nosso modelo de negócio contém objetos de mesmo contexto, que já existem de forma pré-estabelecida com a certeza de não haver tanta alteração de valores.

Exemplos:

Grau de Escolaridade: Analfabeto, Fundamental, Médio, Superior;

Estado Civil: Solteiro, Casado, Divorciado, Viúvo;

Estados Brasileiros: São Paulo, Rio de Janeiro, Piauí, Maranhão.

⚠ Não confunda uma lista de constantes com enum.

Enquanto que uma constante é uma variável de tipo com valor imutável, enum é um conjunto de objetos já pre-definidos na aplicação.

Como um enum é um conjunto de objetos, logo, estes objetos podem conter atributos e métodos. Veja o exemplo de um enum, para disponibilizar os quatro estados brasileiros citados acima, contendo informações de: Nome, Sigla e um método que pega o nome do de cada estado e já retorna para todo maiúsculo.

```
// Criando o enum EstadoBrasileiro para ser usado em toda a aplicação.
public enum EstadoBrasileiro {
    SAO_PAULO ("SP","São Paulo"),
    RIO_JANEIRO ("RJ", "Rio de Janeiro"),
    PIAUI ("PI", "Piauí"),
    MARANHAO ("MA","Maranhão") ;

    private String nome;
    private String sigla;

    private EstadoBrasileiro(String sigla, String nome) {
        this.sigla = sigla;
        this.nome = nome;
    }
    public String getSigla() {
        return sigla;
    }
    public String getNome() {
        return nome;
    }
    public String getNomeMaiusculo() {
        return nome.toUpperCase();
    }
}
```



Boas práticas para criar objetos Enum

- As opções (objetos), devem ser descritos em caixa alta separados por underline (), ex.: OPCAO_UM, OPCAO_DOIS;
- Após as opções, deve-se encerrar com ponto e vírgula ";" ;
- Um enum é como uma classe, logo, poderá ter atributos e métodos tranquilamente;
- Os valores dos atributos, devem já ser definidos após cada opção, dentro de parênteses como se fosse um `new` ;
- O construtor deve ser privado;
- Não é comum um enum possuir o recurso `setter` (alteração de propriedade), somente os métodos `getters` correspondentes.

Agora **NÃO** precisaremos, criar objetos que representam cada estado, toda vez que precisarmos destas informações, basta usar o **enum** acima e escolher a opção (objeto), já pré-definido em qualquer parte do nosso sistema.

```
// qualquer classe do sistema poderá obter os objetos de EstadoBrasileiro
public class SistemaIbge {
    public static void main(String[] args) {
        //imprimindo os estados existentes no enum
        for(EstadoBrasileiro uf: EstadoBrasileiro.values() ) {
            System.out.println(uf.getSigla() + "-" + uf.getNomeMaiusculo());
        }

        //selecionando um estado a partir das opções disponíveis
        EstadoBrasileiro ufSelecionado = EstadoBrasileiro.PIAUI;

        System.out.println("O estado selecionado foi: " + ufSelecionado.getNome());
    }
}
```

valueOf

asd



Programação Orientada a Objetos - Previous
Construtores

Next - Programação Orientada a Objetos
UML



Last modified 1mo ago