

# Membros estáticos e finais

Prof. Hugo de Paula



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS  
Departamento de Ciência da Computação

# Sumário

- 1 Membros estáticos
- 2 Exemplo: Identificador de Produtos
- 3 Membros finais
  - Exemplo: membros finais
  - Atributo final não constante



# Membros estáticos

## Membro estático

Membro com tempo de vida global e escopo local.

- São atributos ou métodos comuns a todos os objetos de uma classe.
- Membros de classe.
  - Compartilhado por todos os objetos daquela classe.
  - Primeiro objeto inicializa a variável<sup>1</sup>.

---

<sup>1</sup>Na verdade, em Java, membro é inicializado quando a classe é carregada.



# Membros estáticos

- Úteis para implementar contadores ou identificadores de autoincremento.
- Podem ser usados para definir constantes.
  - Como a variável é compartilhada por todos os objetos de uma classe, a utilização de membros estáticos constantes pode permitir grande economia de memória.
- Modificador de acesso **static**.



# Exemplo: Identificador estático de Produtos

```
class Produto {  
    ...  
    private int id;  
    private static int cont = 0;  
  
    public static int getCont() {  
        return cont;  
    }  
    public int getId() {  
        return id;  
    }  
    ...  
    public Produto(String d, float p, int q) {  
        setDescricao(d);  
        setPreco(p);  
        setQuant(q);  
  
        id = ++cont;  
    }  
    public Produto() {  
        descricao = "Novo Produto";  
        preco = 0.01F;  
        quant = 0;  
  
        id = ++cont;  
    }  
}
```



# Exemplo: Identificador estático de Produtos

```
class Aplicacao {  
    public static void main(String args[]) {  
  
        System.out.println("\nCont. prods: " + Produto.getCont());  
  
        Produto p1 = new Produto();  
        System.out.println("\nCont. prods: " + Produto.getCont());  
  
        Produto p2 = new Produto("Shulambs", 1.99F, 200);  
        System.out.println("\nCont. prods: " + Produto.getCont());  
        ...  
  
        System.out.println("Produto id: " + p1.getId());  
        System.out.println("Descrição: " + p1.getDescricao());  
        System.out.println("Preço: " + p1.getPreco());  
        System.out.println("Estoque: " + p1.getQuant());  
  
        System.out.println("Produto id: " + p2.getId());  
        System.out.println("Descrição: " + p2.getDescricao());  
        System.out.println("Preço: " + p2.getPreco());  
        System.out.println("Estoque: " + p2.getQuant());  
  
    }  
}
```



# Membros finais

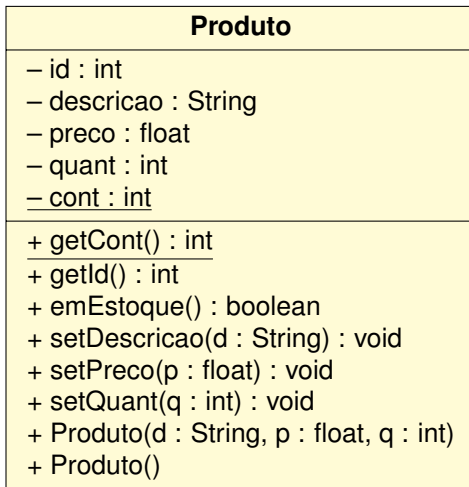
## Membro final

Podem ser definidos/inicializados apenas uma vez.

- São atributos, métodos ou classes, em geral, constantes.
- Modificador de acesso **final**.
- Sua função varia dependendo do tipo de membro. No caso de variáveis, define constantes.



# Classe Produto: diagrama UML (sem *getters*)







# Classe Produto: atributo final

```
class Produto {  
    public static final String DESCRICAO_PADRAO = "Shulambs";  
    public static final int MAX_ESTOQUE = 1000;  
    ...  
  
    public void setQuant(int q) {  
        if (q >= 0 && q <= MAX_ESTOQUE)  
            quant = q;  
    }  
    ...  
  
    public Produto() {  
        descricao = DESCRICAO_PADRAO;  
        preco = 0.01F;  
        quant = 0;  
  
        id = ++cont;  
    }  
}
```

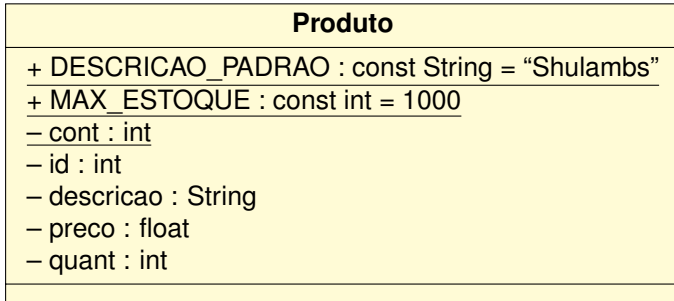


# Classe Produto: atributo final

```
class Aplicacao {  
    public static void main(String args[]) {  
  
        Produto p1 = new Produto();  
        Produto p2 = new Produto("Shulams", 1.99F, 600);  
  
        int nProd = Integer.parseInt(  
            JOptionPane.showInputDialog(null,  
                "Digite quantos produtos deseja adicionar ao estoque:",  
                "Controle de estoque",  
                JOptionPane.OK_CANCEL_OPTION));  
  
        if ((p2.getQuant() + novosProdutos) > Produto.MAX_ESTOQUE) {  
            JOptionPane.showMessageDialog(null,  
                "Estourou o limite máximo do estoque.",  
                "Erro alterando estoque", JOptionPane.ERROR_MESSAGE);  
        } else {  
            p2.setQuant(p2.getQuant() + novosProdutos);  
        }  
    }  
}
```



# Classe Produto: diagrama UML (sem métodos)





# Atributo final não constante

- Constantes são atributos **static final**.
- São imutáveis:
  - Não faz sentido um atributo imutável ser armazenado em nível de instância.
- Se o estado do objeto pode mudar, então não é constante.



## Exemplo: atributo final não constante

// Constantes

```
static final int NUMERO = 42;  
static final ImmutableList<String> MESES =  
    ImmutableList.of("janeiro", "fevereiro", "março");
```

// Não constantes

```
static String naoFinal = "atributo não final";  
final String naoStatic = "atributo não estático";  
static final String[] arranjoNaoVazio =  
    {"esses", "valores", "podem", "mudar"};
```