

Abstração, static e final

Prof. Pedro Pongelupe



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
Departamento de Ciência da Computação

Sumário

- 1 Abstração
- 2 static
- 3 Membros finais
 - Exemplo: membros finais
 - Atributo final não constante
- 4 Destrutores



Abstração

O quê é??

- "Capacidade de enxergar uma operação complexa de uma forma simplificada."
- "Visualização ou representação de uma entidade que inclui somente os atributos de importância em um contexto particular."(Sebesta, 2000)



Abstração

Representação

Vamos pensar num módulo Usuário.

- Quais atributos?
- Quais métodos?



Abstração: Representação

Peraiiii

Um módulo deve prover uma boa abstração da função pela qual é responsável. Antes de pensar no módulo Pessoa. Vamos pensar em qual contexto esse módulo será utilizado:

- Sistema de doação de sangue
- Sistema de gerenciamento de leitos de um hospital
- Sistema de gerenciamento de salas de coworking
- App para gerenciamento de receitas digitais



Abstração: operação simplificada

- Conceito da caixa-preta:
 - Entrada e saída bem conhecidas
 - Detalhes ocultos

Ideia Principal

Não é necessário saber detalhes do funcionamento de um objeto para utilizá-lo

Como funciona classe Scanner do Java?



Como funciona a classe Scanner?

```
310 public final class Scanner implements Iterator<String>, Closeable {  
311  
312     // Internal buffer used to hold input  
313     private CharBuffer buf;
```

```
3027•         Pattern forName(String name) {  
3050     }  
3051 }
```

Como funciona a classe Scanner? Não precisamos saber!!!!





Abstração: operação simplificada

Mas mesmo assim podemos usar esses métodos tranquilamente

- `Scanner s = new Scanner(System.in);`
- `s.nextLine();`
- `s.hasNext();`



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 variável¹.

¹Na verdade, em Java, membro é inicializado quando a classe é carregada.



Membros estáticos

Quais as principais utilidades?

- Implementação de contadores e identificadores de autoincremento
- Definição de constantes.

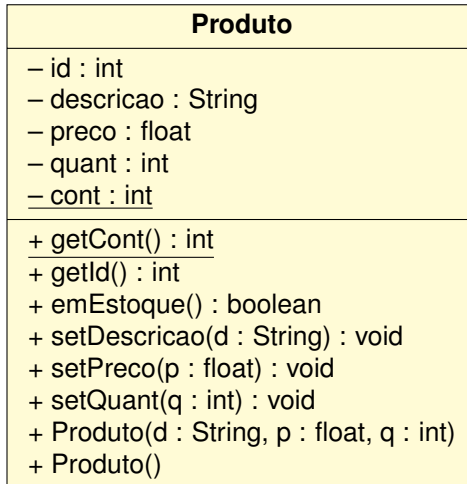


Membros estáticos

Exemplos no Eclipse (Classe Aluno)



Diagrama UML (sem *getters*)





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.

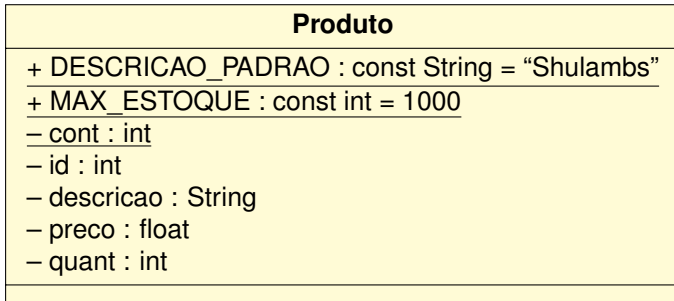


Classe Produto: atributo final

Exemplos no Eclipse (Classe SalaDeEstudos)



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"};
```



Destrutores

O quê é??

Métodos especiais invocados quando um objeto é finalizado (recolhido pelo coletor de lixo).

- Só há um destrutor por classe.
- Um destrutor não tem parâmetros.
- Não deve ser chamado diretamente



Coletor de Lixo - Garbage Collector

Quando o objeto está apto para ser coletado?

Métodos especiais invocados quando um objeto é finalizado (recolhido pelo coletor de lixo). Não é mais acessado por nenhuma referência (ou referências circulares)

- O coletor de lixo é autônomo.
- `System.gc()`

Como isso nos ajuda

- Eliminam a necessidade de se desalocar memória explicitamente.
- Eliminam o vazamento de memória
- Eliminam referências pendentes (*dangling pointer*)



Coletor de Lixo - Garbage Collector

Quando coletor de lixo passa?

- *finalize()*
- Java 9: Classe *Cleaner* e método *clean()*



Obrigado!!

Muito obrigado pela atenção! Alguma dúvida? Bora praticar!!!

"Um passo para trás, para depois dar dois à frente"