

Resumo Capítulo 1

Convenções de código

Classes e interfaces: iniciam com letra maiúscula.

Identificadores legais começam com: uma letra, underscore (_) ou o cifrão (\$)

Declaração de classe

<code>[package]0..1</code> <code>[import]0..*</code> <code>[class]0..*</code> <code>[public class]0..1</code> <code>[class]0..*</code>	<ul style="list-style-type: none">- Se houver uma classe pública, o nome do arquivo deve ser o mesmo da classe pública.- Só pode haver uma classe pública em um arquivo-fonte.
--	---

Modificador Protected

<code>package com.A;</code> <code>public class C {</code> <code>protected int a = 1;</code> <code>}</code>	<code>package com.C;</code> <code>import com.A.*;</code> <code>public class D extends C{</code> <code>a = 5 // válido (também seria se fosse mesmo pacote)</code> <code>C c = new C();</code> <code>c.a = 4; // erro compilação</code> <code>}</code>
---	---

Enum

<pre>// Overriding de métodos enum Teste{ // ... C(3){ public int getQ() { return 3; } // apenas para o enum de valor C }; public int getQ(){ return 2; } // lembrar do .values } // método values() retorna cada item do Enum for (Teste t : Teste.values()){ ... }</pre>

Interfaces

- métodos: public e abstract. variáveis: public, static, final.
- pode estender +1 interface, nunca 1 classe
- são implicitamente abstract
- métodos não podem ser static

<pre>public abstract algumaInteface{} ↓ (opcional)</pre>
--

Modificadores

public class	Classe vista por todos
class	(default) visível somente no pacote
final class / final método	Classe não pode ser estendida. Método não pode ser subscrito
Argumento final	Não pode se alterado no método
Método abstract	Só pode existir em método abstract

Inicialização

Tanto variáveis static como de instâncias são inicializadas com valor default.

Inicialização de arrays

```
int a[] = new int[3];
int []b[] = new int[4][5];
int c[] = {2,3,4};
int []d[][] = new int[4][5][6];
d[3][4][5] = 4;

// inicialização anônima
// não é declarado o tamanho no new
int[]e[] = new int[][]{ {1,2}, {56,9} };

// mesmo efeito da inicialização anônima
int[]f[] = { {1,2}, {56,9} };

// compile mas gera ArrayIndexOutOfBoundsException
d[3][4][6] = 3;
```