



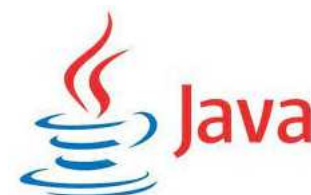
Residência em Software

Residência em Tecnologia da Informação e Comunicação

Conceitos OO: paralelo C++ e Java

Professor:

Alvaro Degas Coelho



INSTITUIÇÃO EXECUTORA



COORDENADORA



APOIO



OO em C++ e em Java

- **Classes**
 - Praticamente idêntico
 - Sintaxe

```
Class NomeDaClasse {  
    Atributos e métodos  
}
```
- **Diferenças importantes na definição de atributos e métodos públicos, protegidos e privados**
 - Em C++: uma área
 - Em Java: definidos individualmente

Exercício

- Copie uma classe em C++ da Internet
- Remova atributos e métodos (guarde-os pra se lembrar depois)
- Crie uma classe em Java (use o Eclipse)
 - Mesmo nome
- Compare os formatos

Atributos

- Em C++
 - Tipo nomeDoAtributo;
- Em Java
 - Tipo nomeDoAtributo;
- Em C++: o atributo é especificamente o dado armazenado na memória
- Em Java: nem sempre (a se ver)
 - Ponteiro implícito
- Em C++: o atributo, por default, é privado
- Em Java: também

Tipos de Dados

- Tipos primitivos em Java
 - byte, short, int, long, char, float, double, boolean
- C++ também...
- Tipos complexos em Java
 - String (em C++ é string), List, Set... veremos
- Tipos complexos em C++
 - string, vector, ...

Exercício

- Volte à classe que você copiou da Internet
- Há atributos primitivos?
 - Crie-os na sua classe Java
- Há atributos complexos (vector e string)?
 - Crie-os em sua classe Java
 - vector: ArrayList
 - String: String
- Se a classe que você escolheu ficou sem graça ou complicada, procure outra no mesmo lugar
 - Google

Métodos

- Em C++: Funções
 - Tipo de dado nomeDaFuncao(parâmetros) { ... };
- Em Java:
 - Tipo de dado nomeDaFuncao(parâmetros) { ... };
- Em c++: um construtor é um método especial
 - NomeDaClasse(parâmetros) {...};
- Em Java: um construtor é um método especial
 - NomeDaClasse(parâmetros) {...};
- Invocação
 - Em java a declaração de um objeto é feita forçando a alocação dele na memória
 - Executando o construtor
 - NomeDaClasse Objeto = new NomeDaClasse(parâmetro);

Exercício

- Volte à classe que você copiou da Internet
- Há métodos (que não mandem mensagens a outras classes)?
 - Crie-os na sua classe Java
- Há Construtor?
 - Crie-os em sua classe Java
- Se a classe que você escolheu ficou sem graça ou complicada, procure outra no mesmo lugar
 - Google

Visibilidade

- Em C++
 - Métodos e atributos podem ser
 - public, protected ou private
 - Default (em caso de omissão): public
- Em Java
 - Métodos e atributos podem ser
 - public, protected ou private
 - Default (em caso de omissão): public
- Em C++: área dos públicos e área dos private (protected)
- Em Java: uma cláusula antes do atributo ou do método
 - Indicando se é public, private ou protected

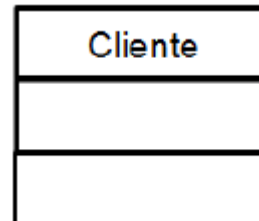
Exercício

- Volte à classe que você copiou da Internet
 - Há atributos privados? Há métodos privados?
 - Há atributos public? Há métodos public?
 - Há atributos protected? Há métodos protected?
- Crie-os com as mesmas configurações em sua classe Java
- Se a classe que você escolheu ficou sem graça ou complicada, procure outra no mesmo lugar
 - Google

Entrada e Saída de Dados

- Considere o código abaixo
 - `Scanner entrada = new Scanner(System.in);`
 - `System.out.println("Digite seu nome e Pressione <Enter>");`
 - `String mensagem = entrada.nextLine();`
 - `System.out.println("Oi " + mensagem);`
- Crie um método main na sua classe
 - Dica: digite main e pressione ctrl + space
- Copie este código em seu método main
- Execute sua classe (Run as Java Application)

Exercício



- Criar a classe Cliente acima
 - Atributos: nome, idade, cpf
 - Métodos: gets e sets dos atributos
 - Um construtor que recebe o nome e o cpf como parâmetro)
 - A idade inicia com 0
- Crie um método main
 - Solicite que digitem o nome e o cep
 - Crie uma variável (objeto) do tipo Cliente (passe o nome e o cpf como parâmetro)
 - Mostre os dados (use os gets)
 - Solicite alteração dos dados (use os sets)