INTRODUÇÃO OO

Classe: Estrutura abstrata para definir os objetos. Um molde.

Atributo: Característica de uma classe. Caracterizam um objeto. São chamados de atributos ou características ou campo ou propriedade de um objeto. Quando acionamos a palavra-chave new e o Java instancia o objeto, todos (exceto os de referência) os campos são zerados. Zero é o valor default de vários tipos numéricos, como int, double e long. No caso do tipo boolean o valor é false.

- Não é uma boa prática manipular diretamente o atributo. Sempre utilizar os métodos.
- Depois de criar os atributos o eclipse gera alguns métodos (Menu Source - generate getters and setters).
- A palavra static faz com que o atributo seja da classe, e não mais do objeto (da instância). Com isso, todo o objeto criado possui acesso ao valor compartilhado desse atributo.

Objeto: É a instância ou modelo derivado da classe.

Instanciamento: é a parte concreta. A partir do instanciamento da classe é criado o objeto na memória do computador.

- Atenção: É comum ter a impressão de que a palavra-chave new nos devolve o objeto em si, e de que a variável criada contém o objeto, mas isso nunca ocorre. No Java, assim como em outras linguagens, um objeto nunca está dentro de uma variável nem o objeto está dentro de um objeto. São unicamente referências (de um lugar para outro como flechas/caminhos).

  O que têm dentro de uma variável é somente uma indicação (endereço de memória) a um objeto específico, uma referência a um objeto específico.

Método(): maneira de fazer algo. Parte comportamental do objeto.

No parênteses () adicionaremos o que está sendo recebido pelo método, ou seja, um parâmetro, declarando o tipo de variável. Depois que o método é chamado e executado ele pode devolver uma mensagem ou algo do tipo. Quando não existe qualquer tipo de retorno ao acionarmos um método, ou quando utilizamos métodos que alteram atributos, utilizamos a palavra-chave VOID.

- THIS.atributo: É a referência a quem está invocado o método(). O this, além de guardar a referência para mexermos nos atributos do objeto, para o desenvolvedor, significa que o atributo está definido na própria classe.

- RETURN: encerra a execução do método e volta para quem estava chamando.
- SUPER: Significa que subimos na hierarquia, para chamar um método ou atributo da classe mãe. *super.atributo* (usado para deixar explícito em nosso código que o atributo sendo manipulado vem de uma superclasse. Desta forma, o desenvolvedor sabe que precisará subir na hierarquia para encontrar este atributo, já que ele não está definido nesta classe).

Construtores: são elaborados visando que os objetos tenham seus atributos inicializados na própria construção. Essa estratégia evita estados inconsistentes no nosso objeto.

- Quando escrevemos “new Classe()” é executada uma rotina padrão com um construtor implícito(default), sem parâmetros. Se não existe um construtor escrito explicitamente, o construtor default, vem por padrão, zero para objetos e null para atributos do tipo referência. Usar apenas o construtor default pode gerar objetos em estado inconsistente com a regra de negócio. O construtor default do java deixa de existir a partir do momento que um construtor específico é declarado na classe.
- No Java é possível fazer a chamada de um construtor dentro de outro e faz-se isso para evitar duplicações de códigos e regras. Afinal, uma regra aplicada em um

construtor normalmente será a mesma para o outro caso. Para isso usa-se o this() passando os parâmetros correspondentes ao construtor que se queira chamar.