PROGRAMAÇÃO ORIENTADA A OBJETOS

Introdução a Classes, Objetos, Métodos e Strings
(parte 2)

Prof. Emanoel Barreiros

Tipos

- Tipos em Java são divididos em dois grupos: primitivos e por referência
- Tipos primitivos em Java:
  - **boolean**: valores possíveis são **true** ou **false**
  - **byte**: representa um inteiro de 8 bits com sinal em complemento de dois. Pode armazenar valores entre 127 e -128 (inclusive). Pode ser usado no lugar de inteiros que não vão armazenar valores acima do range do tipo
  - **short**: representa um inteiro de 16 bits com sinal em complemento de dois. Pode armazenar valores entre -32.768 e 32.767 (inclusive). Pode ser usado no lugar de inteiros que não vão armazenar valores acima do range do tipo

Tipos

- Tipos primitivos em Java (cont.):
  - **int**: representa um inteiro de 32 bits com sinal em complemento de dois. Pode armazenar valores entre -2.147.483.648 e 2.147.483.647 (inclusive).
  - **long**: representa um inteiro de 64 bits com sinal em complemento de dois. Pode armazenar valores entre -9.223.372.036.854.775.808 e 9.223.372.036.854.775.807 (inclusive).
  - **float**: representa um valor de ponto flutuante de precisão simples de 32 bits. Não use este tipo para valores precisos como moeda.
  - **double**: representa um valor de ponto flutuante de precisão simples de 64 bits. Não use este tipo para valores precisos como moeda.
  - **char**: representa um caractere unicode de 16 bits

Tipos primitivos

- Uma variável de tipo primitivo pode armazenar apenas um valor daquele tipo por vez
- Variáveis de instância de tipo primitivo são automaticamente inicializadas com seus valores padrão
- Variáveis do tipo char, short, int, float, double e float são inicializadas com 0 (zero)
- Variáveis do tipo boolean são inicializadas com **false**

Tipos por referência

- Variáveis de tipos por referência são usada para armazenar a localização de um objeto na memória
- Diz-se que essa variável referencia um objeto
- Objetos referenciados podem conter várias variáveis de instância
- Variáveis de instância de tipo por referência são inicializadas como o valor null
- null uma palavra chave que indica que a variável não aponta para nada
- Métodos só podem ser invocados a partir de variáveis que apontam para objetos, logo, de tipo por referência

Inicializando objetos com construtores

- No nosso exemplo da classe LivroNotas, o atributo nomeCurso é inicializado com null
- E se desejássemos inicializar um objeto do tipo LivroNotas com o nome de um curso desde o início?
- Todas as classes possuem o que chamamos de construtor, um método especial usado para inicializar objetos
- A criação de objetos só ocorre através da invocação de algum construtor
- O operador new solicita memória ao sistema e invoca o construtor para inicialização do objeto

Construtores

- Construtores DEVEM ter o mesmo nome da classe
- O compilador fornece o construtor padrão para todas as classes, a não ser que você, programador, o defina um outro construtor explicitamente
- Quando a classe possui a apenas o construtor padrão suas variáveis de instância são inicializadas com seus valores padrão
- Você pode declarar outros construtores para inicialização personalizada dos objetos
- Vamos ver na prática?

Construtores

- Como mostrado, construtores podem ter parâmetros
- Ao serem invocados com o comando new, todos os argumentos do construtor devem ser passados, ou teremos erros de compilação

Exemplo

- Altere a classe LivroNotas, criada na aula passada, para que o seu construtor receba um argumento do tipo String e o atribua o valor recebido ao atributo nomeCurso
- Altere a classe LivroNotasTeste para que ela possa utilizar a classe LivroNotas após sua alteração
- Crie mais de um objeto do tipo LivroNotas, inicialize-os com valores diferentes e veja que cada um guarda seu valor individualmente para o atributo nomeCurso

Números de ponto flutuante

- Muitas vezes não é possível usar apenas valores inteiros como tipos das variáveis
- O Java fornece os tipos float e double para representar números decimais
- Lembre-se que números float e double não são tão precisos, mas podem ser tranquilamente usados na maioria dos casos (ex: a altura de uma pessoa ou seu peso em kg)
- Números de ponto flutuante são sempre aproximações, pois o computador só consegue armazenar uma quantidade finita de casas decimais

Exercício

- Vamos modelar uma classe Conta, representando uma conta bancária
- A classe conta precisa armazenar o valor atual do saldo da conta (use float ou double mesmo, não tem problema nesse exercício)
- O objeto deve ser inicializado com o saldo inicial da conta
- Crie métodos para creditar e debitar valores da conta, bem como retornar o saldo
- Crie uma classe chamada ContaTeste, nela declare o método main, crie e manipule alguns objetos Conta, creditando, debitando e exibindo o saldo das mesmas após as manipulações