

Aula 3 - Introdução a classes e objetos

Prof. Me. Rodrigo Brito Battilana

Roteiro

- Conceitos de variáveis
- Recebendo uma entrada do teclado
- Classes, objetos, métodos e variáveis de instância
 - Classes
 - Declarando uma nova classe
 - Objetos
 - Instanciando um objeto de uma classe
- Compilando um aplicativo com múltiplas classes

Roteiro

- Modificadores de acesso
 - Public
 - Private
 - Protected
- Introdução aos Métodos
- Referências
- Exercícios

Conceitos de variáveis

- As variáveis são utilizadas para que alguns valores possam ser utilizados durante a execução de um programa.
- Geralmente resultados de operações aritméticas, valores externos ao programa e etc...
- As variáveis podem ser de 2 tipos: **primitivos** e **referenciadas**

Conceitos de variáveis

- **Tipos primitivos**

Os tipos primitivos são descritos como características de uma determinada variável.

- Os tipos primitivos são:

boolean

Int

Float

Double

Char

Byte

Short

Long

Conceitos de variáveis

- Todos os tipos primitivos possuem uma declaração direta, como por exemplo: **int x**.
- As variáveis locais não possuem nenhum valor inicial, sendo para os tipos numéricos o número '0' como padrão e para o boolean 'true' ou 'false'.

Conceitos de variáveis

- **Tipos por referência**

- Todos os tipos não primitivos são tipos por referência.
- Os programas utilizam as variáveis de tipos por referência para armazenar as localizações de objetos na memória do computador.
- As variáveis de instância de tipo por referência são inicializadas por padrão com o valor **null**.

Recebendo uma entrada do teclado

- Para receber entradas via teclado é necessário realizar uma importação de um pacote do java.
- Para importar as bibliotecas (libs) utilizamos o **import**.
- As bibliotecas são classes que devem ser importadas para o arquivo (classe) onde será utilizado.

Recebendo uma entrada do teclado


- A biblioteca que utiliza a entrada de teclado é o ***scanner*** que é uma classe do pacote **java.util**.
- *Para importar a classe do scanner é necessário utilizar a sintaxe de importação.*

Ex:

Import java.util.Scanner // importa a biblioteca do scanner



Recebendo uma entrada do teclado

- Para se trabalhar com a classe do Scanner, são necessários algumas observações.
 - Por se tratar de uma biblioteca pronta, algumas funções (métodos) já estão prontas e podem ser acessados diretamente da classe.
- 

Recebendo uma entrada do teclado

- Alguns dos métodos mais utilizados do Scanner:
 - `nextInt`
 - `nextDouble`
 - `nextLine`

Recebendo uma entrada do teclado

- O método `nextInt` de `Scanner` retorna um valor `int`.
- O valor padrão para uma variável de instância do tipo `int` é 0.

Recebendo uma entrada do teclado

- O método `nextDouble` de `Scanner` retorna um valor `double`.
- O especificador de formato `%f` é utilizado para gerar saída de valores de tipo `float` ou `double`. Já o especificador de formato `%.2f` especifica que dois dígitos da precisão devem ser gerados à direita do ponto decimal no número de ponto flutuante.
- O valor padrão para uma variável de instância do tipo `double` é `0.0`, e o valor padrão para uma variável de instância do tipo `int` é `0`.

Recebendo uma entrada do teclado

- O método `Scanner nextLine` lê os caracteres até um caractere de nova linha ser encontrado, depois retorna os caracteres como um método `String`.
- O valor padrão para uma variável de instância do tipo `String` é `null`.

Classes, objetos e variáveis de instância

- **Classes**

- Classes são estruturas (arquivos) que contém atributos e/ou métodos e possuem características em comum.

Ex: Classe carro

- Todos os carros possuem portas, janelas, rodas, motor e etc...
- Essas características são chamadas de atributos dentro da classe.

Classes, objetos e variáveis de instância

- Cada classe que você cria torna-se um novo tipo que pode ser utilizado para declarar variáveis e elaborar objetos.
- É possível declarar novos tipos de classe conforme necessário; essa é uma razão pela qual o Java é conhecido como uma linguagem extensível.

Classes, objetos e variáveis de instância

- Toda declaração de classe que inicia com o **modificador de acesso public** deve ser armazenada em um arquivo que tem o mesmo nome que a classe e termina com a extensão de arquivo .java.
- Cada declaração de classe contém a palavra-chave **class** seguida imediatamente do nome da classe.

Ex: `public class NomeDaClasse`

Classes, objetos e variáveis de instância

- Os nomes de classe, método e variável são identificadores. Por convenção, todos usam nomes na notação **CamelCase**.
- Um objeto tem atributos que são implementados como variáveis de instância que eles mantêm ao longo de sua vida.
- Toda variável que representa uma característica na classe é chamado de **atributo**

Classes, objetos e variáveis de instância

- Todos os atributos devem possuir um **modificador de acesso**, um **tipo** e um nome (identificador).
- Os modificadores de acesso vão permitir que os atributos possam ser utilizados pelas classes do programa
- O tipo deverá ser primitivo ou por referência
- O nome da variável será o identificador no programa.

Classes, objetos e variáveis de instância

- **Objetos**

- Todos os elementos que as classes representam são chamados de **objetos** e estes podem ser chamados (instanciados) por outras classes.

Ex: classe Carro → cria um objeto carro a partir de uma instância.

Classes, objetos e variáveis de instância

- Para realizar uma instância de uma classe é necessário utilizar a palavra reservada **new**.

Ex: NomeDaClasse nomeDaVariavel = **new** NomeDaClasse()

Compilando um aplicativo com múltiplas classes

- Para compilar no console um programa com diversas classes, será necessário seguir alguns passos.

1 - Entrar no diretório do arquivo com os códigos fontes.

Execute o comando: **javac** nomeDoArquivo1.java
nomeDoArquivo2.java

Onde: os arquivos devem ser separados por espaço.

Podem ser passados quantos arquivos forem necessários.

Compilando um aplicativo com múltiplas classes

- Para compilar vários arquivos de uma única vez também é possível utilizar um caractere curinga (*).

Ex: **javac *.java**

Onde: o * representa todos os arquivos com a extensão .java

Modificadores de acesso

- **Public**
- As declarações de classes são precedidas pela palavra-chave **public**, que é um modificador de acesso.
- As variáveis, métodos e classes declarados com o modificador de acesso **public** são acessíveis a todas as demais classes dos pacotes.

Modificadores de acesso

- **Private**
- A maioria das declarações de variável de instância é precedida pela palavra-chave **private**, que é um modificador de acesso.
- As variáveis ou métodos declarados com o modificador de acesso private só são acessíveis a métodos da classe em que são declarados.

Modificadores de acesso

- **Protected**

- As declarações de variáveis, métodos e classes podem ser precedidas pela palavra-chave **protected**, que é um modificador de acesso.
- As variáveis ou métodos declarados com o modificador de acesso **protected** são acessíveis às demais classes do mesmo pacote em que são declarados.

Introdução aos Métodos

- Um método é uma função que pertence a uma determinada classe.
- Os métodos podem ser declarados dentro da classe para ocultar uma determinada funcionalidade dos olhos do usuário.
- Com o uso dos métodos as classes podem ter maior flexibilidade nas execuções.

Introdução aos Métodos

- Os métodos para serem declarados, necessitam conter um **tipo** e um **nome**.
- Modificadores de acesso também podem ser aplicados, bem como parâmetros.

Ex: `tipo nomeDaFuncao() { // código aqui }`

Introdução aos Métodos

- Os tipos podem ser:
 - **void:** Sem retorno
 - **Int:** Retorno inteiro
 - **Float:** Retorno ponto flutuante
 - **Double:** Retorno ponto flutuante com precisão dupla
 - **String:** Retorno de textos
 - **Bool:** Retorno de verdadeiro ou falso

Referências

- Deitei, Harvey M. **Java C o m o Programar**, 8ªed tradução Edson Furmankiewicz ; São Paulo, Editora: Pearson Prentice Hall, 2010.