Objetos

1

Orientação a Objetos

Histórico

- •Linguagem SIMULA (Anos 60):
 - Derivada do Algol, desenvolvida no Centro Norueguês de Computação, por Kristen Nygaard e Ole-Johan Dahl
 - Utilizada em simulações, foi a pioneira no uso de classes e subclasses
- Linguagem SMALLTALK (Anos 70):
 - A primeira linguagem puramente orientada a objetos.
 Foi desenvolvida por cientistas do Xerox Palo Alto
 Research Center (Xerox PARC)
 - -Cada elemento do SmallTalk é tratado como um objeto

Histórico

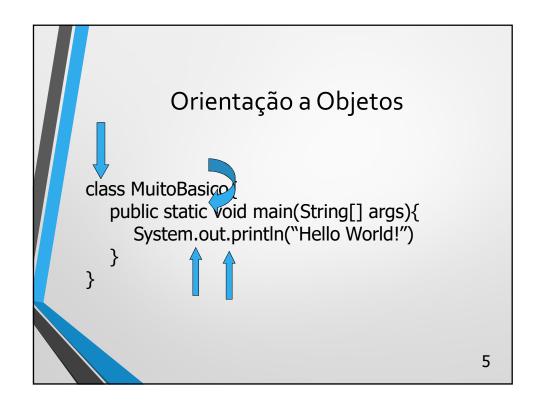
- •C++, dos laboratórios da AT&T Bell, criada por Bjarne Stroustrup, uma expansão da Linguagem C, muito popular e conhecida.
- **Object Pascal**, uma expansão do Pascal. Atualmente a empresa Embarcadero é que tem ditado o padrão da linguagem, através do seu ambiente de desenvolvimento *Delphi*.
- •Java, criada pela Sun, semelhante ao C++, mas totalmente adaptada para OO. Facilidades para o desenvolvimento de aplicações na Internet.

Orientação a Objetos **Mas antes de falar em objetos...**

O Java é uma linguagem orientada a objeto pura

Isto porque....

Até o mais simples dos programas exige conceitos de orientação a objeto





Orientação a Objetos Exemplos da vida real também são objetos:

- Cadeira
- Mesa
- Computador
- Pessoa

7

Orientação a Objetos

Qualquer objeto tem:

- Características (Atributos)
- Ações (Métodos)

Orientação a Objetos Exemplo de uma CADEÍRA

Atributos:

- •? •Altura
- •? •Peso

Fornecedor

Preço

- •? •Material da Base
- Durabilidade
- •? •Material do Assento •Data da Compra

•? •Cor

Quais as mais importantes?

9

Orientação a Objetos

Exemplo de uma CADEIRA

Métodos:

- Montar
- Desmontar
- Ocupar
- Desocupar
- Quebrar
 - Pintar
 - Reformar

Abstração é a habilidade de conseguir retirar de uma situação com várias características, apenas o que é interessante representar naquele momento.

É utilizada para a definição de entidades do mundo real.

11

Classe

Orientação a Objetos

Conceito: é um modelo para múltiplos objetos com características semelhantes.

Classes personificam todas as características de um grupo particular de objetos.

Classe

Ao escrever um programa em linguagem orientada a objetos, você não define objetos reais. Você define "classes" que irão gerar objetos.

13

Orientação a Objetos **Exemplo de Classe**

Precisamos montar uma classe que defina uma Pessoa.

Orientação a Objetos **Exemplo de Classe**

Atributos:

- Nome
- •Idade
- Sexo
- Profissão

Podem haver outros????

15

Orientação a Objetos

Atributos de Estado – são atributos que definem um status ou situação atual do objeto.

Maioridade(Maior/Menor)

A classe *Pessoa* serve como um modelo *Abstrato* para o conceito dos futuros objetos Pessoa.

Antes que você possa representar uma Pessoa:

Representação *Concreta* da classe *Pessoa*

17

Orientação a Objetos

A criação de uma representação
concreta se dá quando existe uma *Instância* da classe abstrata.

Ao criar uma *Instância*, você está criando um objeto real e que pode ser manipulado.

Orientação a Objetos Quantos objetos de uma determinada classe podem ser criados?

R: Quantas pessoas existem no mundo?

19

Orientação a Objetos Mesmo que duas pessoas tenham os mesmos dados, mesmo assim elas são duas pessoas distintas. O que acontece com uma não tem necessariamente relação com a outra.

Cada Instância tem "vida própria"

Os objetos podem se comunicar entre si através de MENSAGENS.

Essas Mensagens podem ser de qualquer tipo. Valores numéricos, String's, estados ou até mesmo outros Objetos.

21

Orientação a Objetos **Métodos**

Definição: são funções definidas DENTRO de classes e que trabalham em INSTÂNCIAS dessas classes.

Orientação a Objetos **Métodos**

Um método determina o que as instâncias daquela classe fazem quando seu estado interno se modifica ou quando aquela instância é solicitada a fazer algo.

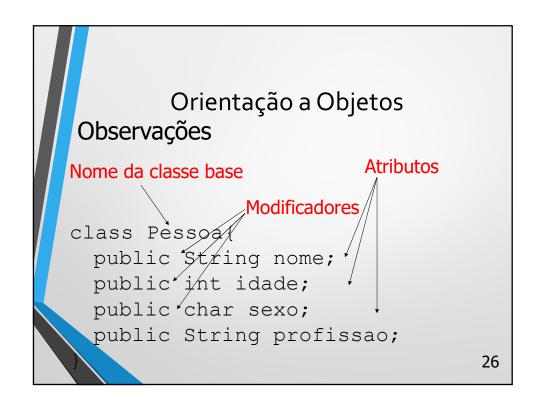
23

Orientação a Objetos Alguns métodos de uma pessoa:

- •Trocar de Idade
- •Ingressar em uma Profissão

```
Orientação a Objetos
Enfim, uma classe!

class Pessoa{
  public String nome;
  public int idade;
  public char sexo;
  public String profissao;
}
embre que o Java é Case Sensitive
```



Orientação a Objetos Esta é uma classe composta apenas por atributos. Não foram adicionados métodos.

Uma classe sem métodos pode ser comparada ao *Record* do Pascal ou *Struct* do C++.

Uma classe sem atributos pode ser utilizada como biblioteca.

27

Orientação a Objetos

Uma característica importante de uma class abstrata é o fato dele não conter um método chamado main, ou seja, não tem programa principal e por consequência não poderá ser executada sozinha.

Inserindo métodos.

Para que seja adicionado um método a *class* abstrata, devemos escrever método(s) que faça(m) algum tipo de manipulação nos atributos da classe.

Importante: para que este método faça parte da *class* abstrata, ele deve ser escrito corpo da *class*.

29

Orientação a Objetos

Pode haver **validação** dos valores antes da ação.

Validações garantem a integridade dos valores.

Esta validação pode ser feita na própria classe ou em outra camada.

Para os exemplos de aula a validação será realizada nos métodos "set"

set, get, is e toString

Você pode chamar os seus métodos como bem entender.

Entretanto, quando você trabalha em Java, algumas **regras de nomenclatura** são extremamente comuns.

31

Orientação a Objetos

set, get, is e toString

O Java em seus métodos usa uma regra para dar nomes a seus métodos. São eles:

- set
- •get
- •is
- toString

Orientação a Objetos **set, get, is e toString**

- •set para atribuir valores
- •get para retornar valores
- •is para retornar valores booleanos
- •**toString** métodos que retornam todo o conteúdo de um objeto.

É um exercício interessante o uso desses mesmos nomes em nossas classes abstratas.

33

Orientação a Objetos **Instanciação**

A instanciação de uma classe abstrata, ou seja, a criação de um objeto, é feita com o comando *new*.

Cada *new* dado em uma classe abstrata cria um objeto distinto.

```
Orientação a Objetos
Instanciação

1 objeto
Pessoa P1 = new Pessoa();
Pessoa P2 = new Pessoa();
Pessoa P3 = new Pessoa();
3 objetos
```

Orientação a Objetos **Instanciação**

Para que você possa testar algo, entretanto, você deve ter escrito, em algum ponto, um programa principal (main)

Orientação a Objetos **Construtores:**

São métodos especiais que são chamados automaticamente quando uma classe é instanciada (no new).

Tem por objetivo atribuir valores iniciais a uma instância.

37

Orientação a Objetos

São especiais por:

- •São chamadas automaticamente
- •Possuírem o mesmo nome da classe
- •Não possuírem retorno (nem void)