



Orientação a Objetos

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.

3

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

4

Orientação a Objetos

```
class MuitoBasico {
    public static void main(String[] args){
        System.out.println("Hello World!")
    }
}
```

5

Orientação a Objetos

Mas afinal, o que é um objeto?



6

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**)

8

Orientação a Objetos

Exemplo de uma CADEIRA

Atributos:

- ? •Altura
- ? •Peso
- ? •Material da Base
- ? •Material do Assento
- ? •Cor
- Preço
- Fornecedor
- Durabilidade
- Data da Compra

Quais as mais importantes?

9

Orientação a Objetos

Exemplo de uma CADEIRA

Métodos:

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

10

Orientação a Objetos

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

Orientação a Objetos

Classe

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

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

12

Orientação a 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.

14

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)

16

Orientação a Objetos

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.

18

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"

20

Orientação a Objetos

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.

22

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

24

Orientação a Objetos

Enfim, uma classe!

```
class Pessoa{  
    public String nome;  
    public int idade;  
    public char sexo;  
    public String profissao;  
}
```

Lembre que o Java é Case Sensitive

25

Orientação a Objetos

Observações

Nome da classe base

Atributos

Modificadores

```
class Pessoa{  
    public String nome;  
    public int idade;  
    public char sexo;  
    public String profissao;  
}
```

26

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.

28

Orientação a Objetos

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 no 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"

30

Orientação a Objetos

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

32

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.

34

Orientação a Objetos

Instanciação

```
Pessoa P1 = new Pessoa();  
Pessoa P2 = new Pessoa();  
Pessoa P3 = new Pessoa();
```

1 objeto

2 objetos

3 objetos

35

Orientação a Objetos

Instanciação

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

36

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*)

38