

# Fundamentos Java e Orientação a Objetos



Por  
**Thiago Faria**

## 5.7. Desafio: objeto this, construtores e JavaBeans

5. Orientação a Objetos - parte 2



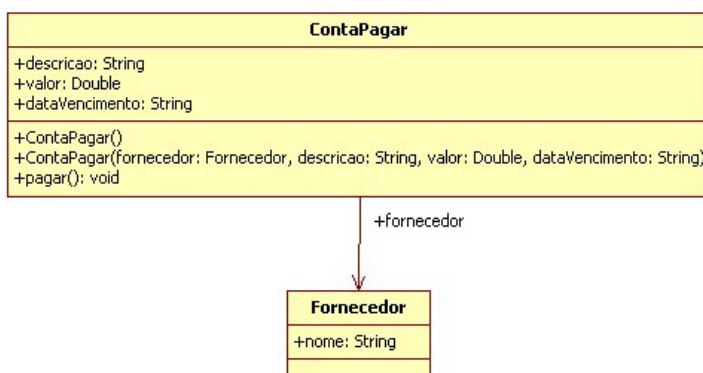
Você é um empresário da área de tecnologia da informação que tem um bom faturamento mensal, porém suas contas a pagar estão totalmente desorganizadas. A planilha Excel que você usa já não suporta mais o movimento que a empresa possui, e chegou a hora de investir no desenvolvimento de um sistema financeiro para a empresa.

Você faz questão de ser o programador desse sistema, no entanto, você pediu que seu analista de sistemas criasse um diagrama de classes simples para criar as primeiras classes do projeto.

Os requisitos passados para o analista foram:

- Uma conta a pagar possui uma descrição, valor, data de vencimento e fornecedor.
- Um fornecedor é uma entidade que possui apenas o nome.

Veja abaixo o diagrama de classes que você recebeu:



Agora você deve criar o código-fonte das classes ContaPagar e Fornecedor. Note que existem dois construtores na classe ContaPagar, sendo um construtor padrão (sem parâmetros), que não deve fazer nada, e outro construtor que recebe alguns parâmetros e deve atribuir os valores recebidos às variáveis de instância.

As duas classes são JavaBeans. Quando desenhamos diagramas de classes, não há necessidade de mostrar os métodos getters e setters de JavaBeans.

O método pagar() deve apenas exibir na tela as informações do pagamento (descrição da conta, valor, data de vencimento e nome do fornecedor).

Quando finalizar a criação das classes ContaPagar e Fornecedor, crie uma outra classe chamada Principal com o código-fonte abaixo:

```

public class Principal {

    public static void main(String[] args) {

        Fornecedor imobiliaria = new Fornecedor();
        imobiliaria.setNome("Casa & Cia Negócios Imobiliários");

        Fornecedor mercado = new Fornecedor();
        mercado.setNome("Mercado do João");

        ContaPagar conta1 = new ContaPagar();
        conta1.setDescricao("Aluguel da matriz");
        conta1.setValor(1230d);
        conta1.setDataVencimento("10/05/2012");
        conta1.setFornecedor(imobiliaria);

        ContaPagar conta2 = new ContaPagar(mercado, "Compras do mês", 390d, "19/05/2012");

        ContaPagar conta3 = new ContaPagar(mercado, "Aluguel da filial", 700d, "11/05/2012");

        conta1.pagar();
        conta2.pagar();
        conta3.pagar();

    }

}
    
```

Agora compile e execute a classe Principal. Evite ver a resolução do desafio antes de resolver os problemas sozinho.

Bom trabalho! :)

[Acesse o código-fonte desta aula](#)

### Comentários sobre esta aula

Fabiano Martins - 26/03/2012 às 13:16



Os atributos: descricao , valor, DataNascimento no diagrama deveriam esta com sinal de - pois sao atributos privados e nao com + que significa publico, esta correto ?



**Normandes Júnior** INSTRUCTOR - 26/03/2012 às 20:14

Sim Fabiano, você está certo. O mais correto seriam os atributos no diagrama estar com o sinal de -. Boa observação.



**Manuel Monteiro** - 20/02/2012 às 13:04

Porque é que os atributos nos metodos set e get tem de ser sempre private ?



**Normandes Júnior** INSTRUCTOR - 21/02/2012 às 08:22

Se você assistir novamente a aula "5.5 Encapsulamento e modificadores de acesso public e private" verá que se você deixa o acesso livre aos atributos do objeto, alguém que for utilizá-lo poderá fornecer qualquer valor. No exemplo da aula, configuramos um valor negativo para um ar condicionado!!!

Configurando os atributos como private, você poderá se necessário, utilizar os métodos "set" e "get" para ter maior controle sobre o valor do atributo.



**Manuel Monteiro** - 20/02/2012 às 13:03

Boas

Gostei mto deste desafio porque conseguir fazer o mesmo sozinho sem ver a resolucao dos instrutores, mas voltou novamente a frisar na necessidade de ter uma lista de mais exercicios .

Precisamos de mais exercicios.

Compartilhe esta aula com seus amigos

[Twitter](#) [Facebook](#)

## 1. Introdução

[1.1. Como aprender Java?](#) 5m 50s GRÁTIS

[1.2. A história do Java](#) 2m 46s GRÁTIS

[1.3. As plataformas Java e como elas evoluem](#) 10m 31s GRÁTIS

[1.4. Máquina virtual Java](#) 8m 45s GRÁTIS

[1.5. Baixando, instalando e configurando a JDK](#) 7m 59s GRÁTIS

[1.6. Exercício: instalação da JDK](#) GRÁTIS

## 2. Fundamentos da linguagem

[2.1. Codificando, compilando e executando o programa "oi mundo"](#) 13m 10s GRÁTIS

[2.2. Exercício: codificando um primeiro programa](#) GRÁTIS

[2.3. Comentários](#) 3m 3s GRÁTIS

[2.4. Sequências de escape](#) 5m 14s GRÁTIS

[2.5. Palavras reservadas](#) 3m 32s GRÁTIS

[2.6. Convenções de código](#) 2m 28s GRÁTIS

[2.7. Trabalhando com variáveis](#) 6m 18s GRÁTIS

[2.8. Nomeando variáveis](#) 5m 42s GRÁTIS

[2.9. Operadores aritméticos](#) 9m 36s GRÁTIS

[2.10. Exercício: variáveis e operadores aritméticos](#) GRÁTIS

[2.11. Tipos primitivos](#) 12m 0s GRÁTIS

[2.12. Outros operadores de atribuição](#) 4m 43s GRÁTIS

[2.13. Conversão de tipos primitivos](#) 12m 39s GRÁTIS

[2.14. Promoção aritmética](#) 6m 25s GRÁTIS

[2.15. Exercício: tipos primitivos e outros operadores de atribuição](#) GRÁTIS

[2.16. Trabalhando com strings](#) 7m 5s GRÁTIS

[2.17. Recebendo entrada de dados](#) 7m 41s GRÁTIS

[2.18. Operadores de comparação e igualdade](#) 6m 40s GRÁTIS

[2.19. Estruturas de controle if, else if e else](#) 12m 23s GRÁTIS

[2.20. Exercício: Strings, entrada de dados, operadores de comparação e if else](#) GRÁTIS

[2.21. Escopo de variáveis](#) 6m 3s GRÁTIS

[2.22. Operadores lógicos](#) 15m 13s GRÁTIS

[2.23. Exercício: operadores lógicos](#) GRÁTIS

[2.24. Estrutura de controle switch](#) 7m 10s GRÁTIS

[2.25. Operador ternário](#) 6m 49s GRÁTIS

[2.26. Operadores de incremento e decremento](#) 8m 11s GRÁTIS

[2.27. Estrutura de controle while](#) 5m 45s GRÁTIS

[2.28. Estrutura de controle do-while](#) 3m 47s GRÁTIS

[2.29. Estrutura de controle for](#) 4m 15s GRÁTIS

[2.30. Cláusulas break e continue](#) 7m 2s GRÁTIS

[2.31. Exercício: operador ternário, decremento e estruturas de repetição](#) GRÁTIS

[2.32. Introdução e instalação do Eclipse IDE](#) 13m 40s GRÁTIS

[2.33. Depurando códigos com o Eclipse](#) 8m 43s GRÁTIS

[2.34. Exercício: instalando o Eclipse IDE](#) GRÁTIS

## 3. Orientação a Objetos - parte 1

[3.1. O que é POO?](#) 2m 57s GRÁTIS

[3.2. Classes e objetos](#) 5m 16s GRÁTIS

[3.3. Criando uma classe com atributos](#) 2m 48s GRÁTIS

[3.4. Instanciando objetos](#) 7m 59s GRÁTIS

[3.5. Acessando atributos de objetos](#) 8m 32s GRÁTIS

[3.6. Exercício: instanciando e acessando atributos do objeto](#) GRÁTIS

[3.7. Composição de objetos](#) 9m 28s GRÁTIS

[3.8. Valores padrão](#) 5m 59s GRÁTIS

[3.9. Variáveis referenciam objetos](#) 9m 22s GRÁTIS

[3.10. Criando, nomeando e chamando métodos](#) 8m 2s GRÁTIS

[3.11. Métodos com retorno](#) 11m 13s GRÁTIS

[3.12. Passando argumentos para métodos](#) 5m 25s GRÁTIS

3.13. Argumentos por valor ou referência

7m0sGRÁTIS

3.14. Exercício: composição de objetos e chamada de métodos

GRÁTIS

4. Wrappers, boxing e arrays

4.1. Wrappers do java.lang

3m31sGRÁTIS

4.2. Boxing

6m47sGRÁTIS

4.3. Desafio: wrappers e boxing

GRÁTIS

4.4. Trabalhando com arrays

16m37sGRÁTIS

4.5. Exercício: arrays

GRÁTIS

5. Orientação a Objetos - parte 2

5.1. Introdução à UML e diagrama de classes

7m31sGRÁTIS

5.4. Construtores

11m43sGRÁTIS

5.7. Desafio: objeto this, construtores e JavaBeans

GRÁTIS

5.10. Modificadores static e final

12m40sGRÁTIS

5.13. Desafio: pacotes e enumerações

GRÁTIS

5.16. Sobreposição

7m48sGRÁTIS

5.19. Exercício: sobrecarga

GRÁTIS

5.22. Desafio: polimorfismo e classes abstratas

GRÁTIS

5.2. Desafio: diagrama de classes

GRÁTIS

5.5. Encapsulamento e modificadores de acesso public e private

11m7sGRÁTIS

5.8. Organizando os projetos em pacotes

11m51sGRÁTIS

5.11. Desafio: static e final

GRÁTIS

5.14. Herança e modificador protected

10m42sGRÁTIS

5.17. Desafio: herança e sobreposição

GRÁTIS

5.20. Polimorfismo, casting de objetos e instanceof

18m49sGRÁTIS

5.23. Interfaces

11m49sGRÁTIS

5.3. O objeto this

8m18sGRÁTIS

5.6. Criando JavaBeans

8m40sGRÁTIS

5.9. Modificador de acesso default

6m55sGRÁTIS

5.12. Enumerações

17m26sGRÁTIS

5.15. Classe java.lang.Object

4m13sGRÁTIS

5.18. Sobrecarga

7m48sGRÁTIS

5.21. Classes abstratas

9m49sGRÁTIS

5.24. Exercício: interfaces e polimorfismo

GRÁTIS

6. Tópicos avançados

6.1. Coleta de lixo

8m40sGRÁTIS

6.4. Tratando e lançando exceções

29m12sGRÁTIS

6.7. Trabalhando com datas

19m28sGRÁTIS

6.10. Desafio: números

GRÁTIS

6.13. Arquivos JAR

6m19sGRÁTIS

6.16. Desafio: javadoc

GRÁTIS

6.2. Classe java.lang.Math

16m6sGRÁTIS

6.5. Desafio: exceções

GRÁTIS

6.8. Desafio: datas

GRÁTIS

6.11. Collections Framework

22m25sGRÁTIS

6.14. Exercício: arquivos JAR

GRÁTIS

6.17. Próximos passos

4m8sGRÁTIS

6.3. Desafio: classe java.lang.Math

GRÁTIS

6.6. Classes String, StringBuffer e StringBuilder

8m26sGRÁTIS

6.9. Trabalhando com números

9m12sGRÁTIS

6.12. Desafio: collections

GRÁTIS

6.15. Documentação javadoc

9m55sGRÁTIS

6.18. Conclusão

2m6sGRÁTIS

Cursos online

Depoimentos de alunos

Sobre nós

Cursos presenciais

Instrutores

Fale conosco

Apostilas gratuitas

Trabalhe conosco

AlgaWorks Softwares, Treinamentos e Serviços Ltda  
Av. Afonso Pena, 3538, Átrio Business Center  
CEP. 38400-710 - Uberlândia/MG - Brasil  
Tel. +55 (34) 8400-6931 - comercial@algaworks.com