Fundamentos Java e Orientação a Objetos

Por
Thiago Faria

5.13. Desafio: pacotes e enumerações

Vamos continuar o desafio anterior, que você começou a desenvolver as classes de um sistema financeiro.

O analista de sistemas responsável por estruturar as classes do software enviou para você o seguinte e-mail:

> *Olá chefe,*
>
> *Esqueci de algumas coisas importantes no diagrama que enviei para você anteriormente, por isso seguem algumas revisões:*
>
> 1. *Crie uma enumeração chamada SituacaoConta com as seguintes constantes: PENDENTE, PAGA e CANCELADA.*
> 2. *Inclua um atributo chamado "situacaoConta" do tipo SituacaoConta na classe ContaPagar.*
> 3. *O atributo "situacaoConta" deve possuir apenas o método getter. O setter não deve existir por questões de segurança. Ninguém pode pagar uma conta simplesmente mudando a situação dela, mas deve sempre passar pelo método pagar().*
> 4. *No construtor padrão (o que não recebe parâmetros) da classe ContaPagar, atribua a constante PENDENTE (da SituacaoConta) à variável "situacaoConta", assim, todas as contas a pagar instanciadas ficarão com o status PENDENTE por padrão.*
> 5. *No construtor que recebe os parâmetros, lembre de invocar o construtor padrão usando a instrução this(), pois também neste caso é importante que a "situacaoConta" seja definida com a constante PENDENTE.*
> 6. *O método pagar() deve verificar a situação da conta antes de efetivar o pagamento. Se a situação for CANCELADA ou PAGA, uma mensagem de erro deve ser exibida ao usuário. Uma conta só deve ser paga se a situação atual for PENDENTE.*
> 7. *No caso de uma conta ser paga através do método pagar(), não esquecer de atribuir a constante PAGA à variável "situacaoConta".*
> 8. *Crie um método cancelar() na classe ContaPagar que muda a situação da conta para CANCELADA e exibe uma mensagem para o usuário. A regra neste caso é a seguinte: não se pode cancelar uma conta que já foi cancelada ou paga.*
> 9. *As classes ContaPagar e Fornecedor e a enumeração SituacaoConta devem ficar no pacote "com.algaworks.cursojava.financeiro.modelo".*
>
> *Se precisar de ajuda, não me ligue.*
> *Brincadeira... estou à disposição.*

Ok, seu analista lhe deu muito trabalho, mas pense como isso será útil para você aprender sobre as enumerações, pacotes e, claro, praticar Java. :)

Quando terminar todas as solicitações do analista, modifique a classe Principal do desafio anterior para o código-fonte abaixo, compile tudo e execute.

```
package com.algaworks.cursojava.financeiro;

import com.algaworks.cursojava.financeiro.modelo.ContaPagar;
import com.algaworks.cursojava.financeiro.modelo.Fornecedor;

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");

		// pagamento de conta pendente (ok, deve funcionar)
		conta1.pagar();

		// tentativa de pagar uma conta cancelada (não deve aceitar pagamento)
		conta2.cancelar();
		conta2.pagar();

		// tentativa de pagar uma conta duas vezes (não deve aceitar na segunda vez)
		conta3.pagar();
		conta3.pagar();
	}

}
```

Preste bastante atenção se todas as regras estão sendo executadas corretamente. Só para lembrar:

- Uma conta que já foi paga não pode ser paga novamente e nem cancelada.
- Uma conta que já foi cancelada não pode ser cancelada novamente e nem paga.

Boa sorte!

Acesse o código-fonte desta aula

Comentários sobre esta aula

Nenhum comentário para esta aula. Efetue login para enviar uma mensagem.

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 7m 0s GRÁTIS
3.14. Exercício: composição de objetos e chamada de métodos GRÁTIS

4. Wrappers, boxing e arrays

4.1. Wrappers do java.lang 3m 31s GRÁTIS
4.2. Boxing 6m 47s GRÁTIS
4.3. Desafio: wrappers e boxing GRÁTIS
4.4. Trabalhando com arrays 16m 37s GRÁTIS
4.5. Exercício: arrays GRÁTIS

5. Orientação a Objetos - parte 2

5.1. Introdução à UML e diagrama de classes 7m 31s GRÁTIS
5.2. Desafio: diagrama de classes GRÁTIS
5.3. O objeto this 8m 18s GRÁTIS
5.4. Construtores 11m 43s GRÁTIS
5.5. Encapsulamento e modificadores de acesso public e private 11m 7s GRÁTIS
5.6. Criando JavaBeans 8m 40s GRÁTIS
5.7. Desafio: objeto this, construtores e
5.8. Organizando os projetos em pacotes 11m
5.9. Modificador de acesso default 6m 55s

6. Tópicos avançados