

Fundamentos Java e Orientação a Objetos

Por
Thiago Faria

2.31. Exercício: operador ternário, decremento e estruturas de repetição

2. Fundamentos da linguagem

Uma loja de ferramentas está precisando da sua ajuda para guardar cada produto em uma gaveta numerada.

O dono da loja quer seguir uma lógica para classificar cada produto, pois assim ele poderá ser organizado melhor. Ele te chama para dizer que quer fazer a classificação pelo código do produto. E começa a te explicar como funcionará a classificação.

A loja possui dois corredores, cada um com 8 gavetas. No corredor da esquerda ficarão os produtos de código ímpar e, no corredor da direita os de código par.

Cada código que for divisível por 8 será armazenado na gaveta 8, os divisíveis por 7 na gaveta 7 e assim por diante.

Atenção: Você não pode armazenar um produto em mais de uma gaveta, então quando a encontrar pare de procurar. Outro detalhe é que a pesquisa deverá ser feita contando de trás pra frente senão, todos os produtos iriam ser armazenados na gaveta 1.

Você irá dizer a classificação para cada produto em um loop até encontrar o código 0 (zero).

1. Crie o arquivo "ClassificaProduto.java" e adicione o código abaixo. Perceba que a variável "codigoProduto" precisa ser inicializada antes de entrar na estrutura "do/while".

```
public static void main(String[] args) {
    Scanner entrada = new Scanner(System.in);

    int codigoProduto = 0;
    do {
        System.out.print("Digite o código do produto: ");
        codigoProduto = entrada.nextInt();

    } while (codigoProduto != 0);
}
```

2. Para descobrir qual corredor o produto ficará armazenado, vamos utilizar um operador ternário, mas antes, precisamos verificar que o usuário não digitou zero, caso contrário um erro poderia ocorrer.

```
...
if (codigoProduto != 0) {
    String corredor = (codigoProduto % 2 == 0) ? "\"direita\"" : "\"esquerda\"";
}
...
```

3. Para terminar nosso código, vamos adicionar dentro do "if" a estrutura que percorre as gavetas e, caso encontre a gaveta correta, pare de pesquisar.

```
...
for (int i = 8; i >= 1; i--) {
    if (codigoProduto % i == 0) {
        System.out.println("O produto de código: " + codigoProduto +
            " ficará no corredor da " + corredor + " e na gaveta " + i);

        break;
    }
}
...
```

4. Compile e execute o programa. Faça vários testes e tente fazer de formas diferentes.

[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 escape5m 14sGRÁTIS

2.7. Trabalhando com variáveis6m 18sGRÁTIS

2.10. Exercício: variáveis e operadores aritméticosGRÁTIS

2.13. Conversão de tipos primitivos12m 39sGRÁTIS

2.16. Trabalhando com strings7m 5sGRÁTIS

2.19. Estruturas de controle if, else if e else12m 23sGRÁTIS

2.22. Operadores lógicos15m 13sGRÁTIS

2.25. Operador ternário6m 49sGRÁTIS

2.28. Estrutura de controle do-while3m 47sGRÁTIS

2.31. Exercício: operador ternário, decremento e estruturas de repetiçãoGRÁTIS

2.34. Exercício: instalando o Eclipse IDEGRÁTIS
- 2.5. Palavras reservadas3m 32sGRÁTIS

2.8. Nomeando variáveis5m 42sGRÁTIS

2.11. Tipos primitivos12m 0sGRÁTIS

2.14. Promoção aritmética6m 25sGRÁTIS

2.17. Recebendo entrada de dados7m 41sGRÁTIS

2.20. Exercício: Strings, entrada de dados, operadores de comparação e if elseGRÁTIS

2.23. Exercício: operadores lógicosGRÁTIS

2.26. Operadores de incremento e decremento8m 11sGRÁTIS

2.29. Estrutura de controle for4m 15sGRÁTIS

2.32. Introdução e instalação do Eclipse IDE13m 40sGRÁTIS
- 2.6. Convenções de código2m 28sGRÁTIS

2.9. Operadores aritméticos9m 36sGRÁTIS

2.12. Outros operadores de atribuição4m 43sGRÁTIS

2.15. Exercício: tipos primitivos e outros operadores de atribuiçãoGRÁTIS

2.18. Operadores de comparação e igualdade6m 40sGRÁTIS

2.21. Escopo de variáveis6m 3sGRÁTIS

2.24. Estrutura de controle switch7m 10sGRÁTIS

2.27. Estrutura de controle while5m 45sGRÁTIS

2.30. Cláusulas break e continue7m 2sGRÁTIS

2.33. Depurando códigos com o Eclipse8m 43sGRÁTIS

3. Orientação a Objetos - parte 1

- 3.1. O que é POO?2m 57sGRÁTIS

3.4. Instanciando objetos7m 59sGRÁTIS

3.7. Composição de objetos9m 28sGRÁTIS

3.10. Criando, nomeando e chamando métodos8m 2sGRÁTIS

3.13. Argumentos por valor ou referência7m 0sGRÁTIS
- 3.2. Classes e objetos5m 16sGRÁTIS

3.5. Acessando atributos de objetos8m 32sGRÁTIS

3.8. Valores padrão5m 59sGRÁTIS

3.11. Métodos com retorno11m 13sGRÁTIS

3.14. Exercício: composição de objetos e chamada de métodosGRÁTIS
- 3.3. Criando uma classe com atributos2m 48sGRÁTIS

3.6. Exercício: instanciando e acessando atributos do objetoGRÁTIS

3.9. Variáveis referenciam objetos9m 22sGRÁTIS

3.12. Passando argumentos para métodos5m 25sGRÁTIS

4. Wrappers, boxing e arrays

- 4.1. Wrappers do java.lang3m 31sGRÁTIS

4.4. Trabalhando com arrays16m 37sGRÁTIS
- 4.2. Boxing6m 47sGRÁTIS

4.5. Exercício: arraysGRÁTIS
- 4.3. Desafio: wrappers e boxingGRÁTIS

5. Orientação a Objetos - parte 2

- 5.1. Introdução à UML e diagrama de classes7m 31sGRÁTIS

5.4. Construtores11m 43sGRÁTIS

5.7. Desafio: objeto this, construtores e JavaBeansGRÁTIS

5.10. Modificadores static e final12m 40sGRÁTIS

5.13. Desafio: pacotes e enumeraçõesGRÁTIS

5.16. Sobreposição7m 48sGRÁTIS

5.19. Exercício: sobrecargaGRÁTIS

5.22. Desafio: polimorfismo e classes abstratasGRÁTIS
- 5.2. Desafio: diagrama de classesGRÁTIS

5.5. Encapsulamento e modificadores de acesso public e private11m 7sGRÁTIS

5.8. Organizando os projetos em pacotes11m 51sGRÁTIS

5.11. Desafio: static e finalGRÁTIS

5.14. Herança e modificador protected10m 42sGRÁTIS

5.17. Desafio: herança e sobreposiçãoGRÁTIS

5.20. Polimorfismo, casting de objetos e instanceof18m 49sGRÁTIS

5.23. Interfaces11m 49sGRÁTIS
- 5.3. O objeto this8m 18sGRÁTIS

5.6. Criando JavaBeans8m 40sGRÁTIS

5.9. Modificador de acesso default6m 55sGRÁTIS

5.12. Enumerações17m 26sGRÁTIS

5.15. Classe java.lang.Object4m 13sGRÁTIS

5.18. Sobrecarga7m 48sGRÁTIS

5.21. Classes abstratas9m 49sGRÁTIS

5.24. Exercício: interfaces e polimorfismoGRÁTIS

6. Tópicos avançados

- 6.1. Coleta de lixo8m 40sGRÁTIS

6.4. Tratando e lançando exceções29m 12sGRÁTIS
- 6.2. Classe java.lang.Math16m 6sGRÁTIS

6.5. Desafio: exceçõesGRÁTIS
- 6.3. Desafio: classe java.lang.MathGRÁTIS

6.6. Classes String, StringBuffer e StringBuilder8m 26sGRÁTIS

- 6.7. Trabalhando com datas

19m 28s

GRÁTIS
- 6.8. Desafio: datas

GRÁTIS
- 6.9. Trabalhando com números

9m 12s

GRÁTIS
- 6.10. Desafio: números

GRÁTIS
- 6.11. Collections Framework

22m 25s

GRÁTIS
- 6.12. Desafio: collections

GRÁTIS
- 6.13. Arquivos JAR

6m 19s

GRÁTIS
- 6.14. Exercício: arquivos JAR

GRÁTIS
- 6.15. Documentação javadoc

9m 55s

GRÁTIS
- 6.16. Desafio: javadoc

GRÁTIS
- 6.17. Próximos passos

4m 8s

GRÁTIS
- 6.18. Conclusão

2m 6s

GRÁTIS

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