

# Primeiro Exercício-Programa

Norton Trevisan Roman

Fábio Nakano

1 de abril de 2011

## 1 Caixa Eletrônico

Os caixas eletrônicos das Ilhas Weblands operam com todos os tipos de notas disponíveis, mantendo um estoque de cédulas para cada valor (B\$ 50,00, B\$10,00, B\$5,00 e B\$1,00). Os clientes do banco utilizam os caixas eletrônicos para efetuar retiradas de um certo número inteiro de Bits.

Sua tarefa é escrever um método que, dado o valor de Bits desejado pelo cliente (parâmetro), determine o número de cada uma das notas necessário para totalizar esse valor, de modo a minimizar a quantidade de cédulas entregues. Por exemplo, se o cliente deseja retirar B\$50,00, basta entregar uma única nota de cinquenta Bits. Se o cliente deseja retirar B\$72,00, é necessário entregar uma nota de B\$50,00, duas de B\$10,00 e duas de B\$1,00.

Seu método deve necessariamente ter a seguinte assinatura:

```
static void fazRetirada(int valor)
```

Para esse exercício, não é necessário controlar o estoque de notas. Ou seja, você pode supor que o caixa sempre tem notas suficientes para efetuar o pagamento ótimo.

### 1.1 Entrada

A entrada é composta pelo parâmetro do método, que recebe um número inteiro positivo  $V \geq 0$ , indicando o valor solicitado pelo cliente.

### 1.2 Saída

Como saída, seu método deve abastecer quatro inteiros I, J, K e L (atributos do programa), que representam o resultado encontrado pelo seu programa: I indica o número de cédulas de B\$50,00, J indica o número de cédulas de B\$10,00, K indica o número de cédulas de B\$5,00 e L indica o número de cédulas de B\$1,00.

Por exemplo, se a retirada for de B\$25, o caixa deve dispensar 2 notas de B\$10 e uma de B\$5. Para que você teste o funcionamento do método fazRetirada, poderia fazer o método main imprimir o valor e a quantidade de notas dispensadas, como ilustrado abaixo (isso apenas para que você teste):

25

I J K L

0 2 1 0

Em caso de erro ( $V < 0$ ), cada uma das variáveis I, J, K e L deve ser abastecida com -1, como ilustrado abaixo:

-25

I J K L

-1 -1 -1 -1

### 1.3 Material a Ser Entregue

Um arquivo, denominado *CaixaEletronico.java*, contendo o método *fazRetirada* e qualquer outro método que ache necessário.

**Atenção!** Para avaliação, apenas o método *fazRetirada* será invocado diretamente. Em especial, qualquer código dentro do *main* será ignorado. Então tenha certeza de que o problema é resolvido chamando-se diretamente somente esse método.

## 2 Cálculo da Raiz Quadrada

Nesse exercício prático você não usará o método `Math.sqrt()`. Em vez disso, você criará um método que calcula  $\sqrt{x}$  usando o método de Newton. Segundo esse método, uma aproximação da raiz quadrada de um número  $a > 0$  pode ser calculada conforme descrito pelas seguintes equações:

$$x_0 \rightarrow \frac{a}{2}$$

$$x_{i+1} \rightarrow \frac{1}{2}\left(x_i + \frac{a}{x_i}\right)$$

para  $i = 0, 1, 2, \dots$ . Ou seja, obtemos  $x_1$  fazendo  $\frac{1}{2}(x_0 + \frac{a}{x_0})$ , e assim por diante. Note que quanto mais alto o valor de  $i$ , melhor a aproximação de  $\sqrt{a}$  dada por  $x_{i+1}$ .

Esse processo deve ser repetido enquanto  $|x_{n+1} - x_n| \geq \epsilon$ , onde  $\epsilon$  é um número positivo que representa a precisão do cálculo da raiz. Assim, a aproximação de  $\sqrt{x}$  será o primeiro valor  $x_{n+1}$  para o qual  $|x_{n+1} - x_n| \geq \epsilon$ .

Por exemplo,  $\sqrt{63}$ , com  $\epsilon = 0.001$  fica:

| $n$ | $x_n$              | $x_{n+1}$          | $ x_{n+1} - x_n $        |
|-----|--------------------|--------------------|--------------------------|
| 0   | 31.5               | 16.75              | 14.75                    |
| 1   | 16.75              | 10.255597014925373 | 6.494402985074627        |
| 2   | 10.255597014925373 | 8.19929204939465   | 2.056304965530723        |
| 3   | 8.19929204939465   | 7.94144112240038   | 0.25785092699427015      |
| 4   | 7.94144112240038   | 7.937255037058497  | 0.004186085341882695     |
| 5   | 7.937255037058497  | 7.9372539331938485 | 0.0000011038646485772574 |

Escreva um método que receba como parâmetro um número  $a$ , e o valor de  $\epsilon$ , retornando a  $\sqrt{a}$  com precisão  $\epsilon$ . Seu método deve necessariamente ter a seguinte assinatura:

```
static double raizQuadrada(double a, double epsilon)
```

Para o cálculo do valor absoluto  $|x_{n+1} - x_n|$ , você não poderá usar o método `Math.abs()`, devendo portanto solucionar o problema de outra forma.

## 2.1 Entrada

A entrada é composta pelos parâmetros  $a \geq 0$ , indicando o número cuja raiz se quer calcular, e  $0 < \epsilon < 1$ , indicando a precisão do cálculo.

## 2.2 Saída

Como saída, o método retorna o valor de  $\sqrt{a}$  com precisão  $\epsilon$ , ou  $-1$ , caso  $a < 0$ ,  $\epsilon \leq 0$  ou  $\epsilon \geq 1$ .

## 2.3 Material a Ser Entregue

Um arquivo, denominado *Raiz.java*, contendo o método *raizQuadrada* e qualquer outro método que ache necessário.

**Atenção!** Para avaliação, apenas o método *raizQuadrada* será invocado diretamente. Em especial, qualquer código dentro do *main* será ignorado. Então tenha certeza de que o problema é resolvido chamando-se diretamente somente esse método.

## 3 Entrega

Deverão ser entregues dois arquivos .java: CaixaEletronico.java e Raiz.java. No início de cada arquivo, acrescente um cabeçalho informativo, como o seguinte:

```

/*****
/**  ACH2001 - Introdução à Ciência da Computação I          **/
/**  EACH-USP - Primeiro Semestre de 2011                    **/
/**  <turma> - <nome do professor>                             **/
/**                                                              **/

```

```

/**    Primeiro Exercício-Programa                                **/
/**    Arquivo: <nome do arquivo>                                **/
/**                                                                 **/
/**    <nome do(a) aluno(a)>                                     <número USP> **/
/**                                                                 **/
/**    <data de entrega>                                         **/
/*****/

```

A entrega será feita unica e exclusivamente via col, até a data marcada para entrega. Deverá ser postado no col um zip com os arquivos .java correspondentes a cada um dos exercícios. Os arquivos devem ser compactados em um único arquivo, tendo seu número USP como nome, ou seja:

número\_usp.zip

Somente este arquivo zip deve ser postado no col. A responsabilidade de postagem nele é exclusiva do aluno. Por isso, problema referentes ao uso do sistema devem ser resolvidos com antecedência.

## 4 Avaliação

Para avaliação, serão observados os seguintes quesitos:

1. Documentação: se há comentários explicando o que se faz nos passos mais importantes e para que serve o programa (Tanto o método quanto o programa em que está inserido)
2. Apresentação visual: se o código está legível, endentado etc
3. Corretude: se o programa funciona

Além disso, algumas observações pertinentes ao trabalho, que influem em sua nota, são:

- Este exercício-programa deve ser elaborado individualmente.
- Não será tolerado plágio, em hipótese alguma.
- Exercícios com erro de sintaxe (ou seja, erros de compilação), receberão nota ZERO