

4.15 Identifique e corrija os erros em cada um dos seguintes fragmentos de código. [*Observação:* pode haver mais de um erro em cada trecho de código.]

```
a) if (age >= 65);  
    System.out.println("Age is greater than or equal to 65");  
    else  
        System.out.println("Age is less than 65");
```

if (age >= 65) // ; fechando if e age não declarado

System.out.println("Age is greater than or equal to 65");

else

System.out.println("Age is less than 65"); // " fecha aspas após o) fecha parênteses

```
b) int x = 1, total;  
    while (x <= 10)  
    {  
        total += x;  
        ++x;  
    }
```

int x = 1, total = 0;

while(x <= 10){

total += x; //total não inicializado

++x;

}

```
c) while (x <= 100)  
    total += x;  
    ++x;
```

int total = 0;

int x = 0;

while(x <= 100){ //faltando o abre chaves

total += x; //total não declarado

++x; //X não declarado

} // faltando o fecha chaves

```
d) while (y > 0)  
    {  
        System.out.println(y);  
        ++y;
```

int y = 0;

while (y > 0)//critério de parada infinito

{

```
System.out.println(y);

++y; // y não declarado

} //faltando o fecha chaves
```

4.16 O que o seguinte programa imprime?

```
1 // Exercício 4.16: Mystery.java
2 public class Mystery
3 {
4     public static void main(String[] args)
5     {
6         int x = 1;
7         int total = 0;
8
9         while (x <= 10)
10        {
11            int y = x * x;
12            System.out.println(y);
13            total += y;
14            ++x;
15        }
16
17        System.out.printf("Total is %d\n", total);
18    }
19 } // fim da classe Mystery
```

O programa imprime a variável y que contém o dobro de x a cada incremento e no final imprime a soma de y no total.

4.20 (*Calculador de salários*) Desenvolva um aplicativo Java que determina o salário bruto de cada um de três empregados. A empresa paga as horas normais pelas primeiras 40 horas trabalhadas por cada funcionário e 50% a mais por todas as horas trabalhadas além das 40 horas. Você recebe uma lista de empregados, o número de horas trabalhadas por eles na semana passada e o salário-hora de cada um. Seu programa deve aceitar a entrada dessas informações para cada empregado e, então, determinar e exibir o salário bruto do empregado. Utilize a classe Scanner para inserir os dados.

Pasta Exe4_20.

4.25 O que o seguinte programa imprime?

```
1 // Exercício 4.25: Mystery2.java
2 public class Mystery2
3 {
4     public static void main(String[] args)
5     {
6         int count = 1;
7
8         while (count <= 10)
9         {
10            System.out.println(count % 2 == 1 ? "****" : "++++++");
11            ++count;
12        }
13    }
14 } // fim da classe Mystery2
```

Imprime **** se Count for numero ímpar ou ++++++ se for um número par.

4.26 O que o seguinte programa imprime?

```
1 // Exercício 4.26: Mystery3.java
2 public class Mystery3
3 {
4     public static void main(String[] args)
5     {
```

continua

116 Java: como programar

```
6     int row = 10;
7
8     while (row >= 1)
9     {
10        int column = 1;
11
12        while (column <= 10)
13        {
14            System.out.print(row % 2 == 1 ? "<" : ">");
15            ++column;
16        }
17
18        --row;
19        System.out.println();
20    }
21 }
22 } // fim da classe Mystery3
```

continuação

Imprime "<" se o número de row for ímpar e ">" se o número for par.

4.27 (Problema do *else* oscilante) Determine a saída para cada um dos conjuntos dados de código quando x é 9 e y é 11 e quando x é 11 e y é 9. O compilador ignora o recuo em um programa Java. Da mesma forma, o compilador Java sempre associa um *else* com o *if* imediatamente precedente a menos que instruído a fazer de outro modo pela colocação de chaves (`{}`). À primeira vista, o programador pode não ter certeza de qual *if* um *else* particular corresponde — essa situação é conhecida como "problema do *else* oscilante". Eliminamos o recuo do seguinte código para tornar o problema mais desafiador. [Dica: aplique as convenções de recuo que você aprendeu.]

```
a) if (x < 10)
    if (y > 10)
        System.out.println("*****");
    else
        System.out.println("#####");
    System.out.println("$$$$$");

b) if (x < 10)
    {
        if (y > 10)
            System.out.println("*****");
        }
    else
    {
        System.out.println("#####");
        System.out.println("$$$$$");
    }
```

a) Para $x = 9$ e $y = 11$:

\$\$\$\$\$

Para $x = 11$ e $y = 9$:

\$\$\$\$\$

b) Para $x = 9$ e $y = 11$:

Para $x = 11$ e $y = 9$:

#####

\$\$\$\$\$

4.28 (Outro problema do *else* oscilante) Modifique o código dado para produzir a saída mostrada em cada parte do problema. Utilize técnicas de recuo adequadas. Não faça nenhuma alteração além de inserir chaves e alterar o recuo do código. O compilador ignora recuo em um programa Java. Eliminamos o recuo do código fornecido para tornar o problema mais desafiador. [Observação: é possível que não seja necessária nenhuma modificação para algumas das partes.]

```
if (y == 8)
if (x == 5)
System.out.println("@@@@");
else
System.out.println("#####");
System.out.println("$$$$$");
System.out.println("&&&&");
```

a) Supondo que $x = 5$ e $y = 8$, a seguinte saída é produzida:

```
@@@@@
$$$$$
&&&&
```

```
int x = 5;

int y = 8;

if (y == 8)

if (x == 5)

System.out.println("@@@@");

else

System.out.println("#####");

System.out.println("$$$$$");

System.out.println("&&&&");
```

b) Supondo que $x = 5$ e $y = 8$, a seguinte saída é produzida:

```
@@@@@
```

```
int x = 5;

int y = 8;

if (y == 8)

if (x == 5)

System.out.println("@@@@");

else{

    System.out.println("#####");

    System.out.println("$$$$$");

    System.out.println("&&&&");

}
```

c) Supondo que $x = 5$ e $y = 8$, a seguinte saída é produzida:

```
@@@@@
```

```
int x = 5;
int y = 8;
if (y == 8)
if (x == 5)
System.out.println("@@@@@");
else{
    System.out.println("#####");
    System.out.println("$$$$$");
    System.out.println("&&&&&");
}
```

d) Supondo que $x = 5$ e $y = 7$, a seguinte saída é produzida. [Observação: todas as três últimas instruções de saída depois do else são partes de um bloco].

```
#####
$$$$$
&&&&&
```

```
int x = 5;
int y = 7;
if (y == 8){
    if (x == 5)
        System.out.println("@@@@@");
}
else{
    System.out.println("#####");
    System.out.println("$$$$$");
    System.out.println("&&&&&");
}
```

4.29 (Quadrado de asteriscos) Escreva um aplicativo que solicite ao usuário que insira o tamanho do lado de um quadrado e, então, exibe um quadrado vazio desse tamanho com asteriscos. Seu programa deve trabalhar com quadrados de todos os comprimentos de lado possíveis entre 1 e 20.

4.37 (Fatorial) O fatorial de um inteiro não negativo n é escrito como $n!$ (pronuncia-se “ n fatorial”) e é definido como segue:

$$n! = n \cdot (n-1) \cdot (n-2) \cdot \dots \cdot 1 \quad (\text{para valores de } n \text{ maiores ou iguais a } 1)$$

e

$$n! = 1 \quad (\text{para } n = 0)$$

Por exemplo, $5! = 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1$, o que dá 120.

a) Escreva um aplicativo que lê um inteiro não negativo, calcula e imprime seu fatorial.

Pasta Exe4_37 arquivo QuestA.java

b) Escreva um aplicativo que estima o valor da constante matemática e utilizando a fórmula a seguir. Permita ao usuário inserir o número de termos a calcular.

$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \dots$$

Pasta Exe4_37 arquivo QuestB.java

c) Escreva um aplicativo que computa o valor de e^x utilizando a fórmula a seguir. Permita ao usuário inserir o número de termos a calcular.

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$$

Pasta Exe4_37 arquivo QuestC.java