

Programação Centrada em Objetos

2021/2022

Série 1

Nos exercícios que se seguem, sempre que adequado, defina assinaturas de métodos, com potencial de reutilização, para abstrair as tarefas que são pedidas. Não esqueça a documentação para cada método que vai construir: um cabeçalho incluindo uma descrição sucinta e, se for caso disso, @param, @requires e @return.

Vá juntando os métodos que constrói numa classe UtilsPCO.

I – Tipos, operadores, variáveis, expressões, abstração procedimental

Revisões

1. O que é um tipo de dados?
2. Que tipos de dados conhece? Quais os valores e as operações de cada um desses tipos?
3. O que é uma variável? O que significa “declarar uma variável”? E “inicializar uma variável”?
4. Como se atribui um valor a uma variável? E como se “lê” o valor de uma variável?
5. O que é um literal?
6. O que é uma expressão?
7. O que é a precedência de operadores?
8. Em que situações se devem criar métodos (procedimentos ou funções)?
9. Qual o papel dos parâmetros na definição dos métodos?
10. Qual o âmbito dos parâmetros e das variáveis definidos num método?
11. Como se invocam os métodos?

Exercícios

12. Escreva um método que, dados dois inteiros positivos, calcule e devolva a percentagem que o primeiro representa relativamente ao segundo.

Construa um programa que invoque esse método e imprima no ecrã "`<valor1> representa <valor2> por cento de <valor3>`" para valores `valor1` e `valor3` escolhidos por si.
13. Escreva um método que converta euros em dólares e devolva o resultado, considerando que um euro vale 1.18 dólares.

Construa um programa que invoque esse método e imprima no ecrã "`<valor1> euros = <valor2> dolares`" para um valor `valor1` escolhido por si.
14. Generalize o método do exercício anterior de modo a que a taxa de câmbio possa variar.

Construa um programa que invoque esse método e imprima no ecrã "<valor1> euros = <valor2> dolares à taxa de cambio de <valor3>" para valores valor1 e valor3 escolhidos por si.

15. Escreva um método que calcule a soma dos algarismos de um número entre 10 e 99.

Construa um programa que invoque esse método e imprima no ecrã "Soma dos algarismos de <valor1> é <valor2>" para valor valor1 escolhido por si.

16. Que alterações faria ao método do exercício anterior para que calculasse antes a soma dos algarismos de um número entre 100 e 999?

17. Escreva um método que, dado um número entre 100 e 999, calcule a sua "imagem ao espelho" (Ex: se invocado com o argumento 123, o método deve devolver 321).

Construa um programa que invoque esse método e imprima no ecrã "Imagem de <valor1> ao espelho é <valor2>" para valor valor1 escolhido por si.

18. Escreva um método que dado um instante de tempo em horas, minutos e segundos, calcule o número de segundos equivalente.

Construa um programa que invoque esse método e imprima no ecrã "<valorH>H<valorM>M<valorS> = <valorSegs> segundos" para valores valorH, valorM e valorS escolhidos por si.

II – boolean e condicionais

Revisões

19. Caracterize o tipo `boolean`. Que operações conhece cujo resultado é deste tipo?
20. Que comandos aprendeu até agora que determinam o fluxo de controle de execução de um programa?
21. Dê exemplos que ilustrem a importância dos blocos (sequência de comandos delimitados por chavetas) na construção de comandos.
22. Que cuidados devemos ter quando testamos se dois números em vírgula flutuante são iguais? Porquê?

Exercícios

23. Assuma que `i` e `j` são duas variáveis de tipo `int` e `x` e `y` duas variáveis do tipo `double`. Escreva expressões java que expressem os seguintes valores:

- a) O resultado de testar a igualdade entre o dobro de `i` e a soma de `j` com `i`;
- b) O resultado de testar se a soma de `i` com metade de `x` é menor que 3.5;
- c) O resultado de testar se `i` é diferente do dobro de `j`;
- d) O resultado de testar se `i` e `j` estão ambos no intervalo `]-6,6[`;
- e) O resultado de testar se `x` é positivo ou está no intervalo `]-1,0[`;
- f) O resultado de testar se `x` é igual a `y`;

24. Considere os seguintes fragmentos de código. Que comentários escreveria em substituição dos ????

```
a) if (i != 3 && j <= 4) { // i é diferente de 3 e j menor ou igual a 4
    System.out.println("Olah");
} else { // ???
    System.out.println("Adeus");
}
```

```

b) if (i != 3 || j <= 4) { // i é diferente de 3 ou j menor ou igual a 4
    System.out.println("Bom dia");
} else { if (z > j) { // ???
    System.out.println("Boa tarde");
} else { // ???
    System.out.println("Boa noite");
}
}

```

25. Considere o fragmento de código abaixo. Qual destas instruções provocará um erro de compilação?

```

int i = 0;
System.out.println (i);
if (i == 0) {
    int j = 1;
    System.out.println (i + j);
}
System.out.println (j);

```

26. Escreva funções que, dados dois valores inteiros, calculam:

- a) o valor absoluto da sua diferença
- b) o seu máximo
- c) o seu mínimo

Construa um programa que invoque esses métodos com valores à sua escolha e imprima mensagens com os resultados.

27. Escreva uma função que, dado um valor representando um ano, verifique se esse ano é ou não bissexto (um ano é bissexto se for divisível por 4, exceto no caso em que também é divisível por 100 mas não por 400).

Construa um programa que invoque esse método com um valor à sua escolha e imprima uma mensagem adequada.

28. Escreva uma função que, dados dois valores representando um ano e um mês, calcule o número de dias desse mês.

29. Escreva funções que, dados três valores inteiros, calculam:

- a) o seu máximo
- b) quantos têm um valor positivo
- c) quantos têm um valor igual

Escreva um programa que inicialize três variáveis com valores à sua escolha e imprima mensagens com os resultados de invocar cada um dos três métodos.

30. Escreva um programa que declara uma variável inteira *grade*, inicializa-a com um valor entre 0 e 20 e imprime a conversão do seu valor numa avaliação entre F e A. A conversão é efetuada do seguinte modo: A = [18,20], B = [15,17], C = [11,14], D = [7,10], F = [0,6]. (NOTA: construa um método que recebe um inteiro e devolve a *String* correspondente à conversão desse inteiro como descrito atrás).

31. O Miguel consegue nadar um dado número de metros em ambiente de piscina. Sabendo que uma dada piscina só tem escadas na extremidade de início de prova e na extremidade oposta, queremos saber em que sítio da piscina ele irá parar, para percebermos se temos que o ir salvar ou não. Consideramos que ele consegue desenhencilhar-se sozinho se ficar a 1 metro ou menos de uma das extremidades da piscina.

Construa um programa que declare e inicialize duas variáveis representando o tamanho da piscina em metros e o número de metros que o Miguel consegue nadar, e imprima uma mensagem que esclareça se é necessário ir salvar o Miguel ou não; em caso afirmativo, deve também indicar a que distância ele está do início da piscina.

III – Ciclos

Revisões

32. Quais as partes constituintes de um ciclo?
33. O que é um ciclo infinito? O que pode causar um?
34. Quando devemos usar cada um dos três comandos de repetição que aprendeu?
35. Qual a diferença fundamental entre os comandos `while` e `do-while`?
36. Relembre algumas regras de estilo na escrita de comandos.

Exercícios

Não esquecer: para cada método definido nos exercícios que se seguem, deve escrever um cabeçalho incluindo uma descrição sucinta e geral do método e, se for caso disso, @param, @requires e @return.

37. O que imprimem os seguintes fragmentos de código?

- a)

```
int x = 10;
for (int i = 0 ; i < 5 ; i++) {
    System.out.print ("K");
    System.out.println ( i * x + i );
}
```
- b)

```
int x = 10;
for (int i = 1 ; i <= 5 ; i++) {
    System.out.print ("K");
    System.out.println ( i * x + i );
}
```
- c)

```
int x = 10;
for (int i = 5 ; i >= 1 ; i--) {
    System.out.print ("K");
    System.out.println ( i * x + i );
}
```
- d)

```
for (int i = 5 ; i <= 1 ; i++) {
    System.out.println ("K");
}
```
- e)

```
for (int i = 5 ; i <= 12 ; i--) {
    System.out.println ("K");
}
```
- f)

```
for (int i = 5 ; i != 12 ; i += 2) {
    System.out.println ("K");
}
```

38. Escreva métodos para:

- a) dado um inteiro positivo n , calcular a soma dos números positivos menores que n
- b) dado um inteiro positivo n , calcular a soma dos números positivos pares menores que n
- c) dados dois inteiros positivos n e m , calcular a soma dos números positivos maiores que n e menores que m
- d) dado um inteiro positivo n , calcular o fatorial de n ($n!$)
- e) dado um inteiro positivo n , calcular o fatorial duplo de n ($n!!$); o fatorial duplo de n é $1 \times 3 \times 5 \times \dots \times n$ se n for ímpar e $2 \times 4 \times 6 \times \dots \times n$ se n for par

39. Escreva métodos para:

- a) dado um inteiro positivo n , calcular o número de divisores de n
- b) dado um inteiro positivo n , decidir se n é primo.
- c) dado um inteiro positivo n , calcular o menor divisor de n que é diferente de 1.
- d) dados dois inteiros positivos n e m , onde $m < n$, calcular o menor divisor de n que é maior que m .
- e) dados dois inteiros positivos k e m , onde $k < m$, os múltiplos positivos de k que são menores que m (incluindo o k).

40. Escreva métodos para:

- a) dado um inteiro positivo n , imprimir no ecrã uma linha com n asteriscos
- b) dados dois inteiros positivos n e m , imprimir no ecrã um retângulo com $n \times m$ asteriscos
- c) dados dois inteiros positivos n e m , imprimir no ecrã um retângulo com n linhas cada uma com os números de 1 a m
- d) dados dois inteiros positivos n e m , imprimir no ecrã um retângulo com n linhas e m colunas, com os números de 1 a $(n \times m)$, ordenados da esquerda para a direita e de cima para baixo
- e) dados dois inteiros positivos n e m , imprimir no ecrã um retângulo com n linhas oco, ou seja, em que a 1ª e a última linhas têm m asteriscos seguidos e as restantes têm um asterisco na primeira posição e um asterisco na m -ésima posição.
- f) imprimir no ecrã os triângulos correspondentes às alíneas c) a f)

41. Qual o valor da variável `teste` quando cada ciclo termina?

- a)

```
int teste = 10;
while (teste > 5) {
    teste -= 2;
}
```
- b)

```
int teste = 10;
do {
    teste -= 2;
} while (teste > 5);
```

E se o valor de `teste` for 4, inicialmente?

42. Escreva métodos para:

- a) dado um inteiro positivo n devolver a soma dos algarismos de n .
- b) dado um inteiro positivo n devolver o inteiro resultante de inverter a ordem dos algarismos de n .

43. Construa os seguintes métodos:

- a) dado um inteiro positivo n , determinar se n é perfeito (um número é perfeito se e só se for igual à soma dos seus divisores próprios; os divisores próprios de um número são todos os seus divisores exceto ele próprio; Ex: 6 – a soma de 1, 2 e 3 dá 6)
- b) dado um inteiro positivo n , calcular o número de números perfeitos menores que n
- c) dado um inteiro positivo n , calcular a soma dos dois números formados pela metade esquerda e pela metade direita dos algarismos de n . Se n tiver um número ímpar de algarismos, então será a metade esquerda que contará com mais um algarismo. *Exemplo:* a função `somaPartes` aplicada ao número 1234 terá 46 como resultado; quando aplicada ao número 8 terá 8 como resultado e quando aplicada ao número 123 terá 15 como resultado.

44. O "check digit" ou dígito de controlo é uma forma de testar se determinados números utilizados para identificação (número de Cartão de Cidadão, número de Contribuinte, número de Conta Bancária, etc) estão ou não corretos. Este processo é muito comum em sistemas informáticos, que

obrigam à introdução desses números manualmente e por isso necessitam de testar se não houve erro na sua digitação.

Normalmente, o último dígito ou os últimos dois dígitos do número são os dígitos de controlo. Para testar a validade do número é efetuada uma operação matemática com os restantes dígitos. Se, no final, o resultado for igual ao dígito de controlo, isso significa que o número é válido.

O algoritmo de controlo é público e pode ser facilmente encontrado com uma pesquisa na *internet*. Como curiosidade, fica aqui a forma de testar a validade do número de contribuinte, cujo último dígito é o "check digit". Tomemos como exemplo o número de contribuinte: **214560732**

- (**d1** significa primeiro dígito, **d2** significa segundo dígito e assim por diante)
- Primeiro passo: calcular $9 \times \mathbf{d1} + 8 \times \mathbf{d2} + 7 \times \mathbf{d3} + 6 \times \mathbf{d4} + 5 \times \mathbf{d5} + 4 \times \mathbf{d6} + 3 \times \mathbf{d7} + 2 \times \mathbf{d8}$
Exemplo: $9 \times 2 + 8 \times 1 + 7 \times 4 + 6 \times 5 + 5 \times 6 + 4 \times 0 + 3 \times 7 + 2 \times 3 = 141$
- Segundo passo: obter o resto da divisão por 11
Exemplo: $141 / 11 = 12$ e dá resto 9
- Terceiro passo: se o resto for inferior a 2, o resto é o "check digit", caso contrário o "check digit" é obtido fazendo: $11 - \text{resto}$
Exemplo: $11 - 9 = 2$

Assim sendo, o número de contribuinte **214560732** é válido, já que o "check digit" do número é igual ao último dígito: **2**.

Construa um método que verifique se um dado número de contribuinte (entre 100 000 000 e 999 999 999) é válido.