

Lista de Exercícios 02

Programando em Java

Classes e Objetos: Uma Visão mais Detalhada

Departamento de Computação
Universidade Federal de Ouro Preto
Prof. José Romildo Malaquias
2015–2

Instruções

- Todos os exercícios que envolvem programas devem ser resolvidos usando a linguagem Java.
- Na solução dos exercícios devem ser utilizados os conceitos listados no cabeçalho desta lista.
- Eventuais dúvidas podem ser sanadas com o professor.

1. Uma aplicação do método `Math.floor` é arredondar um valor para o inteiro mais próximo. Por exemplo

```
y = Math.floor(x+0.5);
```

arredonda `x` para o inteiro mais próximo e atribui o resultado a `y`. Pode-se ainda utilizá-lo para arredondar um valor para uma determinada quantidade de casas decimais. Por exemplo

```
y = Math.floor(x+0.5)/10;
```

arredonda `x` para dezenas. Crie um programa que defina 4 métodos:

- (a) `arredondaInteiro(numero)`
- (b) `arredondaDezenas(numero)`
- (c) `arredondaCentenas(numero)`
- (d) `arredondaMilhares(numero)`

Utilize a importação estática (*static import*) para que o método `floor` possa ser utilizado como se fosse declarado na classe criada por você.

2. Crie uma classe com um método que implemente o algoritmo de Euclides para cálculo do MDC.
3. Crie uma classe com um método estático `max3()`, que recebe três valores inteiros como argumentos e retorna o valor do maior. Adicione a esta classe uma sobrecarga do método anterior, que faz a mesma operação com valores do tipo `double`.
4. Crie uma classe com um método estático `maioria()`, que recebe três argumentos do tipo `boolean` e retorna `true` se pelo menos dois argumentos tiverem o valor `true`, ou retorna `false` caso contrário. Não utilize a instrução `if`.
5. Crie uma classe com um método estático `converte()`, que recebe um argumento do tipo `float` e retorna o mesmo como `double` utilizando apenas a promoção de argumentos.
6. Crie uma classe que possui atributos e métodos adequados para calcular a nota de um aluno no trabalho prático de BCC221. Crie um método `calculaNota()`, que deve possuir 3 versões (utilizar sobrecarga):
 - (a) Uma versão possui apenas um parâmetro inteiro, que indica a quantidade de problemas resolvidos.

- (b) Uma versão possui dois parâmetros, a porcentagem (número real) de problemas resolvidos e um booleano que indica se foi entregue o relatório em LaTeX (1 ponto extra, se a porcentagem for $\geq 60\%$).
 - (c) A última versão recebe apenas uma string, com o endereço do site de onde o aluno supostamente copiou o trabalho. Neste caso a nota é zero.
7. Crie uma classe `Data` que inclua três atributos: mês (inteiro), dia (inteiro) e ano (inteiro). Crie métodos para:
- (a) Funcionar como *getter* e *setter*, validando os dados para garantir que os valores são reais.
 - (b) Funcionar como um construtor (com parâmetros padronizados) com três parâmetros para inicialização dos três atributos.
 - (c) Imprimir a data, com os campos separados por `/`.
 - (d) Calcular a quantidade de dias do ano até aquele mês, recebendo como parâmetro o número do mês.
 - (e) Sobrecarregue o método anterior para realizar o mesmo cálculo, porém, recebendo como parâmetro o nome do mês, em letras minúsculas.
8. Crie uma classe `Tempo` com três atributos: horas, minutos e segundos. Crie dois construtores: um para inicializar os atributos com valor zero e outro para inicializar os atributos com valores passados como argumentos. Crie métodos para:
- (a) Funcionar como *getter* e *setter*.
 - (b) Imprimir os atributos no formato `hh:mm:ss`.
 - (c) Subtrair dois objetos e colocar o resultado no objeto que o chamou.
 - (d) Somar dois objetos e colocar o resultado no objeto que o chamou.
 - (e) Sobrecarregue este último método para que retorne um objeto com o resultado da operação.
9. Crie uma classe `Estacionamento` para armazenar dados de um estacionamento. Os atributos devem representar a placa e modelo do carro além da hora de entrada e saída do estacionamento. Utilize dois objetos da classe `Tempo` criada no exercício anterior. Crie métodos para:
- (a) Funcionar como *getter* e *setter*.
 - (b) Inicializar os dados com vazio ou zero.
 - (c) Imprimir os dados de um carro estacionado.
 - (d) Calcular e retornar o valor a ser pago pelo carro estacionado. Considere o preço de R\$1,50 por hora. Utilize o método da classe `Tempo`.
10. Crie uma enumeração `Semaforo`, cujas constantes (`VERMELHO`, `AMARELO`, `VERDE`) possuem um parâmetro: a duração de cada luz. Teste a enumeração, imprimindo a constante e a duração correspondente.