

LPOO – Lista de Exercícios

Classes e Objetos

1. Modele um funcionário. Ele deve ter o nome do funcionário, o departamento onde trabalha, seu salário (*double*), a data de entrada no banco (*String*) e seu RG (*String*).

Você deve criar alguns métodos de acordo com sua necessidade. Além deles, crie um método *recebeAumento* que aumenta o salario do funcionário de acordo com o parâmetro passado como argumento. Crie também um método *calculaGanhoAnual*, que não recebe parâmetro algum, devolvendo o valor do salário multiplicado por 12.

A ideia aqui é apenas modelar, isto é, só identifique que informações são importantes e o que um funcionário faz. Desenhe no papel tudo o que um Funcionario tem e tudo que ele faz.

Dica: use o software Violet UML Editor (violet.sourceforge.net) para fazer a modelagem

2. Transforme o modelo acima em uma classe Java. Teste-a, usando uma outra classe que tenha *main*. Você deve criar a classe do funcionário com o nome *Funcionario*, mas pode nomear como quiser a classe de testes, contudo, ela deve possuir pelo menos o método *main*.

Você pode (e deve) compilar seu arquivo java sem que você ainda tenha terminado sua classe *Funcionario*. Isso evitará que você receba dezenas de erros de compilação de uma vez só. Crie a classe *Funcionario*, coloque seus atributos e, antes de colocar qualquer método, compile o arquivo java. O arquivo *Funcionario.class* será gerado, mas não podemos "executá-lo" já que essa classe não tem um *main*. De qualquer forma, a vantagem é que assim verificamos que nossa classe *Funcionario* já está tomando forma e está escrita em sintaxe correta. Esse é um processo incremental. Procure desenvolver assim seus exercícios, para não descobrir só no fim do caminho que algo estava muito errado.

3. Crie um método *mostra()*, que não recebe nem devolve parâmetro algum e simplesmente imprime todos os atributos do nosso funcionário. Dessa maneira, você não precisa ficar copiando e colando um monte de *System.out.println()* para cada mudança e teste que fizer com cada um de seus funcionários, você simplesmente vai fazer:

```
Funcionario f1 = new Funcionario();  
// ...  
f1.mostra();
```

4. Construa dois funcionários com *new* e compare-os utilizando o operador *==*. Faça com que esses funcionários tenham os mesmos atributos. Eles foram considerados iguais? Explique o motivo.
5. Crie duas referências para o mesmo funcionário e compare-os utilizando o operador *==*. Eles foram considerados iguais? Explique o motivo.
6. Ao invés de utilizar uma *String* para representar a data, crie uma outra classe, chamada *Data*. Ela possui 3 campos *int*, para dia, mês e ano. Faça com que seu funcionário passe a usá-la. Em alguma classe que possua *Main*, crie um *Funcionario*. Faça o desenho do estado da memória quando um *Funcionario* é criado.
7. O objetivo dos exercícios a seguir é fixar o conceito de classes e objetos, métodos e atributos. Dada a estrutura de uma classe, basta traduzi-la para a linguagem Java e fazer uso de um objeto da mesma em um programa simples.

(a) **Classe: Pessoa**

Atributos: nome, idade.

Método: void fazAniversario()

Crie uma pessoa, coloque seu nome e idade iniciais, faça alguns aniversários (aumentando a idade) e imprima seu nome e sua idade.

(b) Classe: Porta

Atributos: aberta, cor, dimensaoX, dimensaoY, dimensaoZ

Métodos: void abre()

void fecha()

void pinta(String s)

boolean estaAberta()

Crie uma porta, abra e feche a mesma, pinte-a de diversas cores, altere suas dimensões e use o método *estaAberta* para verificar se ela está aberta.

(c) Classe: Casa

Atributos: cor, porta1, porta2, porta3

Método: void pinta(String s),

int quantasPortasEstaoAbertas()

Crie uma casa e pinte-a. Crie três portas e coloque-as na casa; abra e feche as mesmas como desejar. Utilize o método *quantasPortasEstaoAbertas* para imprimir o número de portas abertas.