UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE COMPUTAÇÃO
DEPARTAMENTO DE CIÊNCIAS DE COMPUTAÇÃO
SCC0504/SCC0204 – Programação Orientada a Objetos
Prof. José Fernando Rodrigues Júnior

Aula 04 – Herança e polimorfismo

1. Crie uma estrutura hierárquica que contenha as classes Veiculo, Bicicleta e Automóvel.
Os métodos da classe Veiculo são todos simples que apenas imprimem uma mensagem de teste:

- listarVerificacoes( )
- ajustar( )
- limpar( )

Estes métodos devem ser redefinidos nas subclasses Automóvel e Bicicleta.
Acrescentar na classe Automóvel o método trocarOleo( ).

Para desenvolver a classe Teste que é apresentada a seguir é necessário criar também a classe Oficina que terá dois métodos:
• proximo() que retorna o próximo objeto do tipo Veiculo (Bicicleta ou Automóvel)
• manutencao(Veiculo v) que recebe como parâmetro um objeto do tipo Veiculo e chama os métodos definidos na classe Veiculo: listarVerificacoes( ), ajustar( ), limpar( ), e SE o veiculo for Automóvel ele deve também chamar o método trocarOleo( ) -- pesquise e use o recurso instanceOf para saber qual é a classe de um objeto.

```
class Teste{
      public static void main(String args[])
      {
            Oficina o = new Oficina();
            Veiculo v;
            for(int i=0;i<4;++i){
                  v=o.proximo();
                  o.manutencao(v);
            }
      }
}
```

2. Crie uma classe “Produto” que possua os atributos “nomeloja” e “preco”, crie os métodos sets e gets para estes atributos. Crie também o atributo “descrição” e seu método chamado “getDescrição” que retorna uma string com o simples conteúdo “Produto de informática”.

Crie duas classes filhas de “Produto”, que serão “Mouse” com o atributo “tipo” e “Livro” com o atributo “autor”, no método construtor de cada uma dessas classes passe como argumento a descrição desse produto, por exemplo, Mouse(“Mouse óptico, Saída USB, 1.600 dpi”). Crie o método “getDescrição” que retorna a descrição que foi passada no argumento do construtor concatenada com o atributo que a classe tiver, “autor” no caso de livro e “tipo” no caso de mouse, esse método deve ter a mesma assinatura do método “getDescrição” da classe pai “Produto”.

Crie uma classe “Main” que irá simular a compra de um cliente de vários mouses e livros, deve haver apenas um vector/arraylist na classe “Main” para armazenamento de todos os livros e mouses. Esse vector/arraylist deve se chamar “carrinho” que simula o carrinho de compras de produtos variados de um cliente em um e-commerce. Insira nesse “carrinho” vários mouses e livros e depois chame o método “getDescrição” de todos os objetos presentes no vector/arraylist. Para o usuário do carrinho saber as informações dos produtos em seu carrinho.

3. Implemente uma hierarquia de classes para praticar três conceitos fundamentais da herança: reuso de código, modificação de comportamento, e extensão de funcionalidades com super.

a) Reuso de código: Crie a classe Funcionario com atributos nome e salarioBase e um método calcularSalario() que retorna salarioBase. Crie a classe Estagiario que herda de Funcionario e apenas herda o método calcularSalario() sem modificá-lo.

b) Modificação de comportamento: Crie a classe Gerente que herda de Funcionario e sobrescreve calcularSalario() para retornar salarioBase + 2000.

c) Extensão de funcionalidade: Crie a classe Vendedor que herda de Funcionario, adiciona os atributos vendas e percentualComissao, e sobrescreve calcularSalario() chamando super.calcularSalario() e somando (vendas * percentualComissao / 100).

d) Escreva um método main para exemplificar todas as classes e o comportamento polimórfico definido.

4. Herança

a) Defina a classe Produto
- Os atributos de um produto são: código, preco unitário, descrição e quantidade no estoque – todos com visibilidade protegida;
- O construtor deve receber todos os atributos como parâmetros;
- Deve oferecer métodos “get” e “set” para os atributos preco e descrição;
- Deve oferecer métodos “get” para código e quantidade;
- Deve oferecer um método onde se informa certa quantidade a ser retirada do estoque e outra onde se informa uma certa quantidade a ser acrescentada ao estoque;
- O método onde se informa uma quantidade a ser retirada do estoque deve retornar a quantidade que efetivamente foi retirada (para os casos em que havia menos produtos do que o solicitado);
- Deve oferecer um método que imprime a descrição básica do produto incluindo dados de todos os atributos, por exemplo: “Produto 3, arroz, custo de R$ 5, quantidade 100”.

b) Defina a classe ProdutoPerecivel
- Esta deve ser derivada de Produto;
- Possui um atributo extra que guarda a data de validade do produto;
- Deve possuir métodos get/set para a data de validade;
- O método de retirada deve receber também por parâmetro a data do dia corrente; se a data de validade do produto for expirar dentro de 30 dias o método deve zerar o estoque e devolver 0, pois produtos vencidos são jogados fora ➔ pesquise aritmética com o tipo Date do Java;
- O método de acréscimo no estoque só deve acrescentar os novos produtos caso o estoque esteja zerado, de maneira a evitar misturar produtos com prazos de validade diferentes. O método retorna true ou false caso tenha sido possível, ou não, alterar a quantidade.

c) Defina a classe ProdutoPerecivelEspecial
- Esta é derivada de ProdutoPerecivel;
- Oferece um método de impressão de dados capaz de imprimir uma nota de controle onde consta o código, a descrição, a quantidade em estoque e a data de validade do produto.

d) Defina a classe ProdutoNaoPerecivel
- Esta é derivada de Produto;
- Deve possuir campos para armazenar o tempo de garantia (em número de anos) do produto;
- A classe deve oferecer métodos para permitir obter o tempo de garantia, mas não para alterar.

e) Defina a classe Estoque
- Esta mantém uma lista com os produtos em estoque (do tipo Produto) – usar array simples ou a classe ArrayList do Java;
- A classe deve ter métodos para cadastrar produtos, consultar produtos pelo código (caso não encontre, retorna null), e retirar produtos do estoque, bem como para informar o custo total do estoque armazenado, e imprimir a descrição básica de todos os produtos.

f) Escreva um pequeno programa que cria um produto de cata tipo especializado (perecível, perecível especial, e não perecível). Acrescente estes produtos a um estoque. Teste a sua busca por um produto. Imprima o total do estoque, e liste todos os produtos.

➔ Para todos os itens, defina o construtor da classe com os atributos necessários para sua inicialização considerando as necessidades da super classe.