Universidade Federal de Campina Grande – UFCG

Centro de Engenharia Elétrica e Informática – CEEI

Departamento de Sistemas e Computação – DSC

Disciplina: Laboratório de Programação 2
Período: 2011.1
Professores: Livia Campos (turma1)
Reinaldo Gomes (turma2)
Nazareno Andrade (turma 3)

Laboratório 09

Neste laboratório iremos praticar polimorfismo com herança.

Instruções:

- Data de entrega: 26/04/2011 até 8:00h (Turmas 1, 2 e 3)
- Crie o projeto lab09 no eclipse e programe todos os exercícios dentro do pacote lp2.lab09;
- **Salve o projeto** em um arquivo zip chamado lab09-<seuNome>.zip (ou .tgz) e o envie para juiz.lp2@gmail.com Use como *subject* do email lab09-<seuNome> (turma 1, 2 ou 3). **Nomeações incorretas dos arquivos a serem enviados serão penalizadas na nota do lab**;
- Certifique-se de que seus programas não têm erros de compilação;
- Antes da definição de cada classe escreva o seu nome em comentário (/*Aluno: <seu nome>*/);
- Não copie o programa do seu vizinho. Se tiver dúvida converse com o professor ou com um monitor.

1. Implemente uma hierarquia para as classes Veiculo, Moto e Carro. A superclasse deve ser abstrata e todos os seus métodos são abstratos:

- List<String> consertosPassados() //retorna lista de consertos pelos quais o veículo passou
- List<Verificacao> verificações(); //indica quais verificações o veículo precisa. Use enum para indicar esses tipos de verificações: pneus, oleo, motor, carroceria, eletrônica, suspensão e freios.
- void adicionaVerificacao(Verificacao v)
- void consertar(Verificacao v) //conserta o veículo e inclui o conserto na lista de consertos pelo qual o veículo passou
- void limpar() //lava o veículo
- int trocarPneus() //troca pneus avariados do veículo e retorna o número de pneus trocados

- boolean checaPneuAvariado(int i) // verifica se o pneu “i” está avariado

Estes métodos devem ser implementados nas subclasses da hierarquia. Você pode sugerir novos métodos para a superclasse e métodos específicos para as classes Carro e Moto. Implemente também a classe OficinaDeVeiculos. A oficina deve receber veículos para serem consertados, sejam carros ou motos. A oficina deve ter, pelo menos, os seguintes métodos:

- recebeVeiculo(Veiculo v), que recebe um veiculo do cliente e adiciona a este veiculo a lista de verificações a serem realizadas. Coloca o veículo recebido no fim da fila de veículos a serem tratados;
- proximo(), que retorna o próximo veiculo da fila a ser tratado na oficina;
- manutencao() trata o próximo veículo da fila, chamando os métodos definidos na classe veículo para realizar a manutenção do veículo de acordo com as verificações que o cliente deseja. Sempre depois de realizar a manutenção do veículo, este deve ser lavado na oficina. Quando o veículo “i” entra em manutenção, automaticamente, o próximo da fila será o veículo “i+1”;
- entrega(Veiculo v), após a lavagem a oficina deve entregar o veiculo. Isso significa que o veiculo deixa de estar na fila da oficina. Note que, um veículo só pode ser entregue se ele já tiver passado pela manutenção.

Escreva um método main na oficina para permitir que veículos sejam recebidos na oficina, sejam consertados, lavados e entregues.

Responda as questões abaixo no seu próprio código (como comentário)

1. Indique no código onde estão as chamadas polimórficas.
2. Você sugere uma outra forma onde haja maior reuso de código no seu sistema? Descreva sua sugestão se for o caso.

Observações:

1. Usem a criatividade para escreverem alguns dos métodos pedidos, ex. o método limpar() pode imprimir uma mensagem na tela indicando que o veículo está sendo limpo, ou como ele será limpo.
2. Inclua javadoc e testes de unidade para as classes implementadas.