MC302 – Programação Orientada a Objetos 09/04/2013
Instituto de Computação – UNICAMP
Profa.: Cecília Mary Fischer Rubira

LAB05: Polimorfismo e Acoplamento Dinâmico

Nota: Os exercícios devem ser executados na ordem apresentada, pois o nível de dificuldade é crescente.

Pacote *br.unicamp.ic.mc302.contaCor*:

1. Abra os arquivos `ContaCor.java` e `ExemploCoercao.java` e compile-os. O compilador acusa alguns erros. Mude o tipo de todas as variáveis da operação main( ) da classe ExemploCoercao, de modo que o método seja executado com a seqüência int, byte, byte e recompile. O que aconteceu? Modifique os tipos das variáveis para float. Aconteceu o esperado?

2. a) Defina um construtor adicional na classe ContaCor que recebe como argumentos apenas o nome do cliente, o número da conta e a senha; lembrando que o estado da conta deve ser ajustado para 1. Para evitar repetição de código, é possível usar this( ) do mesmo modo que se usa super( ). A diferença é que, ao invés de chamar o construtor de uma superclasse, this( ) chama um outro construtor na própria classe. Modifique o construtor de ContaCor que recebe como argumentos o nome do cliente, o saldo inicial, o número da conta e a senha para que use o construtor que você acabou de definir. Implemente na classe ContaCor operações para obter os valores de titular, numConta e senha.

b) Crie a classe `ExemploDoisConstrutores` que contenha um método `main()`. Esse método instancia dois objetos do tipo ContaCor passando os mesmos valores para titular, numConta e senha. O primeiro objeto deve ser criado usando o construtor com quatro parâmetros, passando 0 (zero) como valor do saldoAtual. O segundo deve ser criado usando o construtor definido no Item (a). Inclua código no programa principal para imprimir as informações dos dois objetos criados. Compile e execute ExemploDoisConstrutores. Pelos resultados obtidos, que conclusões você pode tomar?

Pacote *br.unicamp.ic.mc302.documento*:

3. Abra os arquivos `Documento.java`, `Carta.java`, `Telegrama.java` e `ExemploPolimorfismoSemRedefinicao.java`. Tente compilá-los. Por que a compilação falha? Usando o operador instanceof definido por Java e *casting* de objetos, modifique a classe `ExemploPolimorfismoSemRedefinicao` para que o compilador pare de acusar erros na classe. O operador instanceof diz se um objeto é ou não de um determinado tipo. Por exemplo:

```
ContaCor c = new ContaEsp();
if(c instanceof ContaEsp) {  // true! Entra no if.
    ContaEsp cEsp = (ContaEsp)c; // casting }
if(c instanceof ContaCor) { // também true. Entra no if. }
if(c instanceof String) { // falso! c não é do tipo String. }
```

Por que o problema foi resolvido?

Pacote *br.unicamp.ic.mc302.contaCor*:

4. Abra os arquivos `ContaCor.Java`, `ContaEsp.java` e `ExemploPolimorfismoComRedefinicao.java`. Compile-os. Por que a compilação não funcionou? Coloque em comentários as linhas que têm erro no arquivo `ExemploPolimorfismoComRedefinicao.Java` (preceda cada linha com o símbolo "//", como no trecho de código acima). Tente compilar o arquivo novamente. Execute a classe ExemploPolimorfismoComRedefinicao e observe o resultado produzido. Inclua nas operações debitarValor( ) de ContaCor e ContaEsp uma linha que imprime o nome da classe na qual o método está definido. Compile os arquivos e execute novamente a classe ExemploPolimorfismoComRedefinicao. O que você pode dizer sobre o funcionamento do exemplo, com relação à execução da operação debitaValor( )?

5. Adicione o modificador final à operação debitaValor( ) da classe ContaCor. Tente compilar a classe ContaEsp. Por que não compila? Desfaça a modificação e adicione o modificador final à operação debitaValor( ) da classe ContaEsp. Tente compilar a classe ContaEsp novamente. Explique o aconteceu.

6. Crie uma classe ContaEspPoup, que herda de ContaEsp e redefine o método debitaValor(). O que acontece? Por quê?

Pacote *br.unicamp.ic.mc302.veiculos*:

7. Abra os arquivos `FilaVeiculo.java`, `Veiculo.java`, `Carro.java`, `Caminhao.java` e `Inicial.java`. Compile e execute a classe Inicial. Leia o código da operação mostraFila( ) da classe FilaVeiculo e da operação mostra( ) nas classes Carro, Caminhao e Veiculo. Modifique o programa para imprimir o estado completo dos objetos do tipo `Carro` e `Caminhao`. Modifique a visibilidade da operação mostra( ) da classe Carro para private. Tente recompilá-la. Que conclusões você tira desse resultado?

8. Em uma oficina mecânica trabalham 10 funcionários. A oficina executa 3 tipos de serviços diferentes, sendo que Tipo1 custa R$ 15, Tipo2 custa R$ 50,00 e Tipo3 custa R$ 40,00. Cada funcionário recebe 10% por cada serviço realizado. Entretanto, se o funcionário é um gerente, ele recebe uma comissão de 15%, além de ter um salário mensal mais alto do que um funcionário comum. Considere que a oficina tenha 1 gerente.

Faça um programa Java que recebe o número do funcionário (entre 1 e 10), a informação se ele é ou não gerente e quantas vezes ele executou cada um dos 3 serviços. Ao final, calcule a comissão de cada funcionário, bem como o seu pagamento do final do mês. Use o conceito de operações polimórficas na sua solução.