**Laboratório 5**



5.7.1 Objetivo:

Além dos objetivos definidos nos laboratórios anteriores temos também a implementação do conceito de herança e override de métodos.

5.7.2 Definição:

Fazer uma classe Java chamada Lab05Sistema no pacote view e copiar o conteúdo da classe Lab04Sistema. Alterar onde aparece Lab04Sistema para Lab05Sistema.

Fazer uma classe Java chamada Lab05ContaCorrenteEspecial no pacote model que herde todas as características da classe Lab03ContaCorrente. Para a classe Lab05ContaCorrenteEspecial adicionar o atributo limite de crédito (privado e real).

O atributo limite de crédito deve ser gravado em um arquivo a parte, com o mesmo nome do arquivo de conta corrente e com a extensão .esp.

Deve-se sobrecarregar (override) o método sacar () da classe Lab03ContaCorrente na classe Lab05ContaCorrenteEspecial para que considere o valor do limite na realização do saque. Pode-se deixar o saldo negativo até o valor do limite de crédito cadastrado.

Deve-se sobrecarregar (override) o método removerArquivo () da classe Lab03ContaCorrente na classe Lab05ContaCorrenteEspecial para que considere a remoção do arquivo .esp quando a conta corrente for removida. Implementá-lo com o seguinte conteúdo:

public boolean removerArquivo ()

{

super.removerArquivo ();

File tArq1;

tArq1 = new File (getNum\_age() + "." + getNum\_conta() + ".esp");

tArq1.delete();

return true;

}

Na classe Lab05ContaCorrenteEspecial criar também um construtor com todos os parâmetros da classe Lab03ContaCorrente (agencia, conta, nome e saldo) mais o limite. Em seguida executar nesse novo construtor, o construtor da classe Lab03ContaCorrente com os devidos parâmetros (Usando o comando super). Também devemos criar na classe Lab05ContaCorrenteEspecial um construtor que receba a agência e conta. Este na sua primeira linha deverá executar o construtor da super classe passando estes dois parâmetros. Na classe Lab05ContaCorrenteEspecial alterar o método imprimir () para passar a imprimir o limite de crédito também. Este método deverá executar o método imprimir () da superclasse e em seguida imprimir o limite de crédito.

Deve-se também alterar os atributos da classe Lab03ContaCorrente para o tipo protected permitindo assim o acesso destes atributos na classe Lab05ContaCorrenteEspecial. Podemos também mantê-las como private e realizar o acesso através dos métodos de acesso. Como boa prática manter como privado e realizar o acesso através dos métodos de acesso

Alterar o método execCadastramento () na classe Lab05Sistema para considerar como conta especial as contas com a agência maior ou igual a 5000. Caso seja indicado que a conta seja especial deve-se pedir o limite e instanciar um objeto da classe Lab05ContaCorrenteEspecial em vez de Lab03ContaCorrente.

Alterar os métodos execSaque (), execConsulta () e execExtrato () para considerar que contas com a agência maior ou igual á 5000 sejam contas especiais. Caso a conta seja especial deve-se instanciar a Lab05ContaCorrenteEspecial e chamar o método de saque, depósito ou impressão correspondente.

5.7.2.1 Sugestões:

Dentro da classe Lab05Sistema alterar os métodos execSaque (), execDeposito (), execConsulta () e execExtrato () de modo que a instanciação de um objeto da classe Lab03ContaCorrente ou Lab05ContaCorrenteEspecial seja feita somente depois de se ter lido os dados para a realização da transação. Baseado no valor da agência se instancia uma ou outra classe e chama-se os métodos de uma ou outra classe.

**OBS.: Sobre a chamada de métodos sobrescritos dentro de construtores:** Se for usado na classe Lab05ContaCorrenteEspecial os métodos recuperar () e gravar () com a mesma assinatura da superclasse, chamar explicitamente dentro de cada um deles o respectivo método da superclasse . A seguir apresentamos algumas dicas para evitar este problema.

Para o compilador quando um objeto da classe Lab05ContaCorrenteEspecial executa o seu construtor e este executa um método com a mesma assinatura da subclasse ao invés de ser executado o método da superclasse será executado o método da subclasse, pois a referência do objeto esta apontando para a subclasse independente se o método esta sendo chamado dentro do construtor da superclasse. Uma outra alternativa para evitar o problema apresentado seria manter a chamada dos métodos recuperar () e gravar () nos construtores da superclasse, porém alterar o nome dos métodos para gravaContaEspecial () e recuperaContaespecial (). Nesta última alternativa não temos mais o conceito de override, logo não podemos aplicar o conceito de polimorfismo