

## Descrição do Exercício

### Classes, Objetos, Atributos e Métodos

**Objetivo:** Este exercício tem como objetivo a familiarização com a criação de classes, atributos e métodos, através de uma aplicação bancária simplificada.

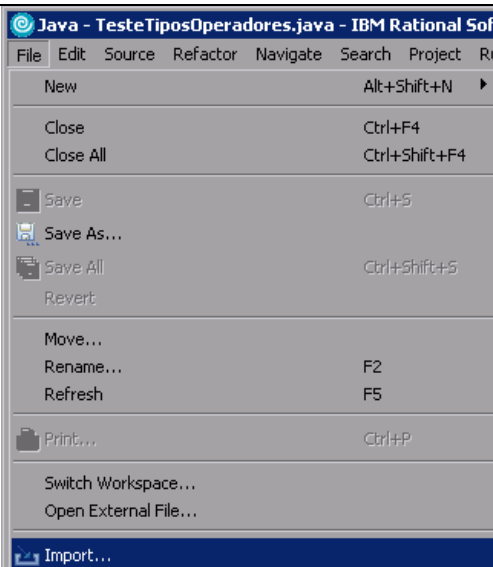
**Tempo Estimado:** 2h 45min

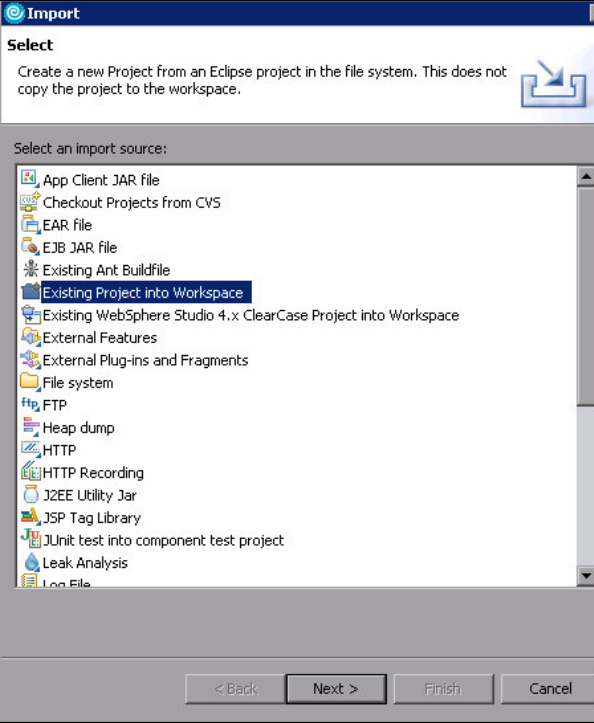
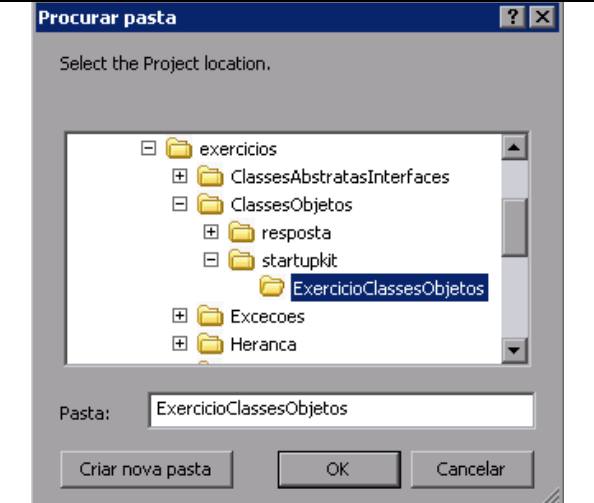
#### Passos para execução do exercício

##### Parte I

1. Utilizar a IDE para importar o projeto intitulado `ExercicioClassesObjetos`. Para tanto, realizar os passos a seguir:

a. Clicar em *File* → *import*



<p>b. Escolher “<i>Existing Project into Workspace</i>” e clicar em <i>next</i></p>	
<p>c. Clicar em <i>Browse</i>, selecionar a pasta <code>ExercicioClassesObjetos</code>, dentro da pasta <code>exercicios\ClassesObjetos\startupkit\</code> do CD e clicar em <i>Finish</i>.</p>	

2. Criar dentro deste projeto uma nova classe, chamada `Cliente`. (para instruções sobre como criar uma classe, olhar exercício passado: Sintaxe Básica).
3. Acrescentar à classe `Cliente` atributos que correspondam ao **nome** e **cpf** do cliente. Qual o tipo de dados adequado para estes atributos (`int`, `boolean`, `String`,...)? Quais os modificadores adequados para estes atributos (`public`, `private`, ...)?
4. Adicionar um Construtor à classe `Cliente`. Este construtor deve receber todos os valores necessários para inicializar os atributos de `Cliente`. Você pode também adicionar um construtor *default*.
5. Acrescentar à classe `Cliente` métodos para acessar cada um de seus atributos, isto é, métodos para obter (`get`) e modificar (`set`) atributos. Qual a importância desses métodos?
6. Ainda no projeto `Banco` criar uma nova classe intitulada `Conta`.

7. Adicionar os seguintes atributos para a classe `Conta`: seu **número**, seu **saldo** e seu dono (um **cliente**). Quais devem ser os tipos de cada atributo?
8. Adicionar dois construtores à classe `Conta`: um recebendo somente o **numero** e o **cliente** da conta para inicialização e outro recebendo todos valores necessários para a inicialização de **todos** os atributos de `Conta`. Você também pode definir o construtor *default*, se desejar.
9. Criar métodos de acesso (*get* e *set*) para cada um dos atributos de `Conta`.
10. Implementar o método `creditar` na classe `Conta`, que incrementa o valor do saldo da conta. Que parâmetros este método deve receber?
11. Implementar o método `debitar`, que decrementa o valor do saldo da conta. Que parâmetros este método deve receber?
12. Ainda no mesmo projeto, criar uma classe de teste com o método `main`. Neste método, declare dois objetos do tipo `Conta`, e atribua a cada um valores de saldo 100 e 0 (Dica: utilize método `setSaldo` em cada objeto para setar o valor do saldo correspondente).
13. Ainda no método `main`, imprimir na tela o saldo das contas criadas. Em seguida, testar a execução dos métodos `creditar` e `debitar` e imprimir novamente o valor dos saldos das contas.

## Parte II

1. Voltar à classe `Conta` e implementar o método `transferir`, que transfere dinheiro de uma conta para outra. É necessário que este método receba alguma `Conta` como parâmetro? Uma? Duas? Nenhuma?
2. Crie uma outra classe de teste com um método `main` ou utilize a mesma já criada na **Parte I**, para testar a execução do método `transferir`. No método `main`, você deve realizar uma transferência entre as contas e imprimir seus saldos antes e depois da transferência.

## Parte III

1. Voltar novamente à classe `Conta` e modificar o método `debitar` de modo a evitar que o débito seja realizado caso o saldo seja insuficiente. Imprimir uma mensagem na tela informando se a operação foi ou não realizada com sucesso. Reflita: por que a impressão de uma mensagem na tela não é uma abordagem interessante para lidar com a falha da operação?
2. Tentar realizar uma transferência cuja conta de origem possui saldo insuficiente. Observe a mensagem de erro. Verifique se a transferência cancela com sucesso a operação caso o saldo da conta a ser debitada seja insuficiente. O que seria necessário para que isso acontecesse?
3. Modifique o método `transferir` de modo a imprimir uma mensagem na tela para sinalizar que o saldo da conta de origem é insuficiente para o valor a ser transferido.