

- 1) O programa deve controlar os saques e os depósitos simultâneos realizados por dois clientes de um Banco que possuem uma conta em conjunto. Para isso, implemente uma thread que permite gerenciar as operações de saque e de depósito. Um cliente realiza essas operações mediante o valor informado. Enquanto a operação está sendo realizada o outro cliente (que possui conta conjunta) não poderá visualizar o valor do saldo e nem fazer uma operação de saque ou depósito, sendo a operação liberada somente após a anterior ter sido concluída.
- 2) Implemente a interface gráfica da Figura 1

A interface gráfica, intitulada "Operação bancária - saque ou depósito", é organizada em seções distintas. No topo, sob o título "Dados da conta", há campos para "Número da conta" e "Saldo inicial". Abaixo, a seção "Dados do cliente 1" contém campos para "Nome" e "Valor", seguidos por botões de opção para "Saque" e "Depósito". Uma seção idêntica, "Dados do cliente 2", segue abaixo. O espaço restante da janela é ocupado por uma grande área vazia, provavelmente para mensagens ou logs. No rodapé, um botão "Realizar operação" está centralizado.

Para criar a borda da Figura 1, é necessário criar um título e inserir na borda conforme o código-fonte 1.

Código-fonte 1 – Criar título e borda

```
TitledBorder dadosConta = BorderFactory.createTitledBorder("Dados da conta");  
dadosConta.setTitleJustification(TitledBorder.CENTER);  
pnContaCli.setBorder(dadosConta);
```

- 2) Implemente a classe Conta tendo como atributos o número e o saldo.
- 3) Implemente a classe Cliente tendo como atributos o nome e a conta.
- 4) Implemente a superclasse “Operacao” tendo como atributos o valor da operação, o cliente e a conta, e outros atributos que julgar necessários. Essa classe deve ser abstrata e implementar a interface Runnable.
- 5) Implemente a subclasse “Saque” (herda da classe Operacao) com o método sacar(). Esse método deve ser sincronizado para realizar o saque.
- 6) Implemente a subclasse “Deposito” (herda da classe Operacao) com o método depositar(). Esse método deve ser sincronizado para realizar o depósito.
- 7) Implemente a classe principal para que quando o usuário informar os dados e o tipo de operação do cliente1 e do cliente2 e clicar no botão “Realizar operação”, seja criada duas threads para realizar as operações de saque e/ou depósito de cada cliente. No resultado apresentado na Figura 2 foram realizadas 3 operações simultâneas para cada cliente.