Universidade Federal do Espírito Santo – Departamento de Informática
Estruturas de Dados I (INF09292)
1º Trabalho Prático
Período: 2012/2
Profª Patrícia Dockhorn Costa
Email: pdcosta@inf.ufes.br

Data de Entrega: 14/03/2013 ***Trabalho Individual***

Este trabalho tem como objetivo praticar o uso de tipos abstratos de dados e estruturas do tipo Lista.

Regras Importantes
- Não é tolerado plágio. Trabalhos copiados serão penalizados com zero.
- A data de entrega é inadiável. Para cada dia de atraso, é retirado um ponto da nota do trabalho.

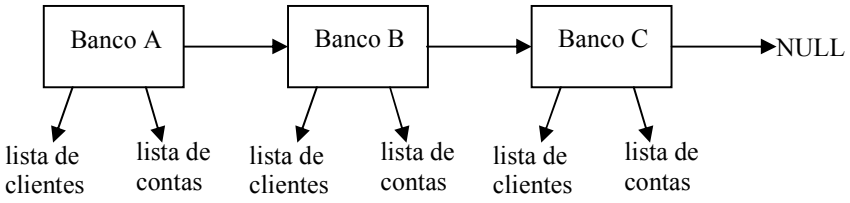
Material a entregar
- Impresso: Documentação do trabalho, que deve conter:
  - Introdução: descrição do problema a ser resolvido e visão geral sobre o funcionamento do programa (em termos de módulos, arquivos, etc.).
  - Implementação: descrição da implementação do programa. Devem ser detalhadas as estruturas de dados utilizadas (de preferência **com diagramas ilustrativos**), o funcionamento das principais funções utilizadas incluindo pré e pós condições, o formato de entrada e saída de dados, bem como decisões tomadas relativas aos casos e detalhes de especificação que porventura estejam omissos no enunciado. Modularize o seu programa usando a técnica de tipos abstrados de dados, como discutido em sala de aula.
  - Conclusão: comentários gerais sobre o trabalho e as principais dificuldades encontradas em sua implementação.
  - Bibliografia: bibliografia utilizada para o desenvolvimento do trabalho, incluindo sites da Internet se for o caso.

- Por email (pdcosta@inf.ufes.br):
  - O assunto da mensagem deve ser ed201202:trab1:<nome1>
    - Por exemplo: ed201202:trab1:<joaosilva>
  - Documentação do trabalho (em formato PDF).
  - Todos os arquivos .c e .h criados (exigido código muito bem documentado!).
  - O makefile.

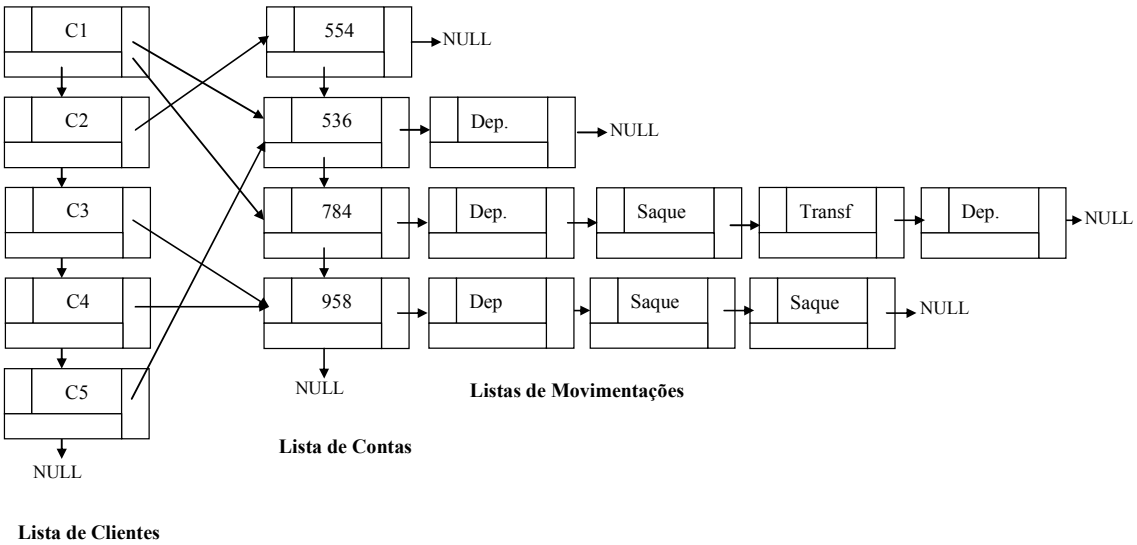
Sistema Bancário

Considere um sistema bancário simplificado no qual há bancos, clientes, contas e movimentações bancárias (saques, depósitos e transferências). A mesma pessoa pode ser cliente de vários bancos, ou seja, pode possuir contas em bancos diferentes. Em um determinado banco, clientes podem ter até 10 contas, e a mesma conta pode ser compartilhada por até 2 cliente (conta conjunta). Pode-se sacar quantias de uma determinada conta, bem como realizar depósitos, e também transferências entre contas. Todas as movimentações bancárias de uma determinada conta devem ser registradas.

Este sistema bancário pode ser implementado com um conjunto de listas encadeadas: pode-se implementar uma lista de bancos, na qual cada banco possui uma lista de clientes e uma de contas, como mostrado na figura a seguir.



Cada conta de um banco pode manter uma lista de movimentações bancárias. A figura a seguir sugere, de maneira geral, a organização das listas de clientes, contas e movimentações.



Como pode ser visto na Figura, este banco possui 5 clientes, sendo que o cliente C1 é titular das contas "536" e "784", o cliente C2 é titular da conta "554", e assim por diante. Note que os clientes C3 e C4 compartilham a mesma conta. Uma conta conjunta se limita a dois clientes, equanto que um cliente pode ter no máximo 10 contas no mesmo banco. Cada conta possui uma lista de movimentações: a conta "784", por exemplo, registrou inicialmente um depósito, depois um saque, uma transferência e a última movimentação foi um depósito.

Nesse trabalho, você deverá implementar o Sistema Bancário. Faz parte do trabalho projetar os Tipos Abstratos de Dados necessários, bem como implementá-los. Como padrão, insira elementos ao final das listas.

O Programa Testador

O programa testador deverá ser capaz de ler as instruções do arquivo texto de entrada e realizar as devidas operações no Sistema Bancário. Para isto, o programa testador precisa manter uma lista de bancos.

O seu programa testador deverá ler os dados de entrada a partir de um arquivo, cujo nome é passado como parâmetro na linha de comando (faz parte do trabalho descobrir como manipular arquivos e strings em C). Exemplo de execução do programa a partir da linha de comando:

```
simulador entrada.txt
```

O arquivo de entrada é basicamente uma lista de comandos (um por linha) em formato texto. O último comando é a palavra FIM, que indica o final do arquivo. O formato a ser usado é exemplificado abaixo:

Exemplo de arquivo de entrada

```
CRIABANCO bancoA
CRIABANCO bancoB
CRIABANCO bancoC
INSERECLIENTE Cliente1 bancoA
INSERECLIENTE Cliente2 bancoA
INSERECLIENTE Cliente3 bancoA
INSERECLIENTE Cliente4 bancoA
INSERECLIENTE Cliente5 bancoB
INSERECLIENTE Cliente6 bancoB
INSERECLIENTE Cliente7 bancoB
INSERECLIENTE Cliente8 bancoB
INSERECLIENTE Cliente9 bancoC
INSERECLIENTE Cliente10 bancoC
INSERECLIENTE Cliente11 bancoC
INSERECLIENTE Cliente12 bancoC
INSERECLIENTE Cliente13 bancoA
INSERECLIENTE Cliente14 bancoB
INSERECONTA 4512 Cliente1 bancoA
INSERECONTA 4977 Cliente1 bancoA
INSERECONTA 5652 Cliente2 bancoA
INSERECONTA 8562 Cliente1 bancoB
INSERECONTA 7685 Cliente3 bancoA
INSERECONTA 7412 Cliente4 bancoA
INSERECONTA 1223 Cliente5 bancoB
INSERECONTA 9853 Cliente6 bancoB
INSERECONTA 1234 Cliente7 bancoB
INSERECONTA 1234 Cliente8 bancoB
INSERECONTA 1567 Cliente8 bancoB
INSERECONTA 1907 Cliente9 bancoC
INSERECONTA 2395 Cliente10 bancoC
INSERECONTA 1011 Cliente11 bancoC
INSERECONTA 1898 Cliente12 bancoC
INSERECONTACONJUNTA 1582 Cliente1 Cliente2 bancoA
INSERECONTACONJUNTA 4621 Cliente3 Cliente4 bancoA
INSERECONTACONJUNTA 6665 Cliente4 Cliente5 bancoA
INSERECONTACONJUNTA 1112 Cliente10 Cliente11 bancoC
IMPRIMETUDO
DEPOSITA bancoB 1234 500
DEPOSITA bancoA 4512 2
DEPOSITA bancoA 5652 74
DEPOSITA bancoB 1223 599
DEPOSITA bancoC 2395 1000
DEPOSITA bancoC 1011 200000
DEPOSITA bancoC 1011 300
DEPOSITA bancoC 9853 145
```

```
DEPOSITA bancoC 1011 200
DEPOSITA bancoC 1011 100
DEPOSITA bancoA 7412 300
DEPOSITA bancoA 1111 300
SACA bancoC 1011 350
SACA bancoC 2395 27
SACA bancoA 5652 88
SACA bancoA 5652 73
DEPOSITA bancoA 5652 100
SACA bancoB 1223 50
DEPOSITA bancoB 1567 200
SALDO bancoA 5652
SALDO bancoB 1223
EXTRATO bancoC 1011
EXTRATO bancoA 4977
TRANSFERE bancoB 1234 bancoA 7685 25
TRANSFERE bancoB 1567 bancoB 1234 10
EXTRATO bancoB 1234
EXTRATO bancoA 7685
EXCLUICONTA bancoA 5652
EXCLUICONTA bancoB 1567
EXCLUICONTA bancoC 1011
EXCLUICLIENTE bancoA Cliente1
EXCLUICLIENTE bancoA Cliente2
EXCLUICLIENTE bancoB Cliente5
EXCLUICLIENTE bancoC Cliente11
IMPRIMETUDO
EXCLUIBANCO bancoA
EXCLUIBANCO bancoB
EXCLUIBANCO bancoC
FIM
```

O seu programa deve checar consistência de dados do arquivo de entrada, como por exemplo: não incluir uma conta se o cliente não existir em um determinado banco; não incluir uma conta, caso já exista uma conta no banco com o dado número; não realizar movimentações em conta inexistente; não realizar saque se o saldo for insuficiente; não realizar transferencia se o saldo da conta origem for insuficiente, etc. Para qualquer comando do arquivo de entrada, o teste de consistência deve ser realizado.

Os comandos de impressão de dados (SALDO, EXTRATO E IMPRIMETUDO), imprimem os dados com um formato exemplificado no arquivo de saída.

Os comandos SACA e DEPOSITA realizam, respectivamente, operações de retirada e depósito nas contas especificadas. O comando "TRANSFERE orgiem destino" realiza uma operação de retirada na conta origem e de depósito na conta destino (de mesma quantia):

SACA banco conta quantia
DEPOSITA banco conta quantia
TRANSFERE banco_origem conta_origem banco_destino conta_destino quantia

Considerando o arquivo de entrada acima, espera-se o seguinte no arquivo de saída:

Arquivo de saída para o arquivo entrada.txt

```
ERRO: Cliente Cliente1 não existe no bancoB
ERRO: Conta 1234 já existe no bancoB
ERRO: Cliente Cliente5 não existe no bancoA
BANCO bancoA
CLIENTE Cliente1
  CONTA 4512
  CONTA 4977
  CONTA 7412
CLIENTE Cliente2
  CONTA 5652
  CONTA 1582
CLIENTE Cliente3
  CONTA 7685
  CONTA 4621
CLIENTE Cliente4
  CONTA 7421
  CONTA 4621
CLIENTE Cliente13

BANCO bancoB
CLIENTE Cliente5
  CONTA 1223
CLIENTE Cliente6
  CONTA 9853
CLIENTE Cliente7
  CONTA 1234
CLIENTE Cliente8
  CONTA 1567
CLIENTE Cliente14

BANCO bancoC
CLIENTE Cliente9
  CONTA 1907
CLIENTE Cliente10
  CONTA 2395
  CONTA 1112
CLIENTE Cliente11
  CONTA 1011
  CONTA 1112
CLIENTE Cliente12
  CONTA 1898

ERRO: não existe conta 9853 no bancoC
ERRO: não existe conta 1111 no bancoA
ERRO: não foi possível sacar 88 da conta 5652 - saldo insuficiente
SALDO bancoA - conta 5652: 101
SALDO bancoB - conta 1223: 549
EXTRATO bancoC - conta 1011:
   DEPOSITO 200000
   DEPOSITO 300
```

```
   DEPOSITO 200
   DEPOSITO 100
   RETIRADA 350
EXTRATO bancoA - conta 4977
   Nenhuma movimentacao
EXTRATO bancoB - conta 1234
   DEPOSITO 500
   RETIRADA 25
   DEPOSITO 10
EXTRATO bancoA - conta 7685
   DEPOSITO 25

BANCO bancoA
CLIENTE Cliente3
  CONTA 7685
  CONTA 4621
CLIENTE Cliente4
  CONTA 4621
CLIENTE Cliente13

BANCO bancoB
CLIENTE Cliente6
  CONTA 9853
CLIENTE Cliente7
  CONTA 1234
CLIENTE Cliente8
CLIENTE Cliente14

BANCO bancoC
CLIENTE Cliente9
  CONTA 1907
CLIENTE Cliente10
  CONTA 2395
  CONTA 1112
CLIENTE Cliente12
  CONTA 1898
```

BOM TRABALHO!