



Trabalho Prático 1

Data de divulgação: 01/09/10
Prazo para entrega: 29/09/10

Descrição do Problema

Um pequeno banco decidiu implantar um sistema para gerenciar sua carteira limitada de clientes. O banco oferece basicamente os serviços de depósito, saque e transferência, e cada cliente possui apenas uma conta corrente. Deseja-se que o novo sistema automatize a realização dessas atividades e gerencie as contas dos clientes. A seguir, são listadas algumas informações que devem ser mantidas pelo cadastro, bem como os requisitos funcionais do sistema.

Dados

Cliente

Código do cliente	Valor numérico (chave para inserção, eliminação e busca)
Nome do cliente	String com até 50 caracteres
Telefone	String com até 20 caracteres
Conta corrente	Conta corrente vinculada ao cliente

Conta Corrente

Código	String (Número da conta corrente)
Saldo	Valor numérico
Transações	Lista de transações financeiras relacionadas à conta

Transações

Código da transação	Valor numérico (chave para inserção e buscas)
Tipo da transação	Depósito, Saque ou Transferência (D, S ou T)
Origem	Valor numérico indicando a conta de onde se saca
Destino	Valor numérico indicando a conta em que se deposita
Valor	Valor da transação



Requisitos Funcionais

Quanto aos Dados:

- O código do cliente deve ser único, ou seja, não é possível que dois ou mais clientes tenham o mesmo código;
- O código da transação deve ser único, ou seja, não é possível que duas ou mais transações tenham o mesmo código.

Quanto às Operações:

Depósito

Pré-condições	<ul style="list-style-type: none">• O cliente beneficiário existe;• O valor a ser depositado deve ser válido.
Pós-condições	<ul style="list-style-type: none">• O valor é creditado na conta do beneficiário (Saldo aumenta);• A transação é inserida na lista de transações.
Ações	<ol style="list-style-type: none">1. Cria-se uma nova instância de transação;<ol style="list-style-type: none">a. Atribui as informações de tipo e valor;b. Deve ser atribuída uma constante inválida à conta de origem;2. Credita o valor da transação na conta do cliente beneficiário;3. Insere a transação na lista de transações.

Saque

Pré-condições	<ul style="list-style-type: none">• O cliente sacado existe;• O valor a ser debitado deve ser válido;• O saldo do cliente sacado deve ser maior ou igual ao valor do saque.
Pós-condições	<ul style="list-style-type: none">• O valor é debitado na conta do cliente sacado (Saldo diminui);• A transação é inserida na lista de transações.
Ações	<ol style="list-style-type: none">1. Cria-se uma nova instância de transação;<ol style="list-style-type: none">a. Atribui as informações de tipo e valor;b. Deve ser atribuída uma constante inválida à conta de destino;2. Debita o valor da transação na conta do cliente sacado;3. Insere a transação na lista de transações.

Transferência



Pré-condições	<ul style="list-style-type: none">• Ambos os clientes, sacado e beneficiário, existem;• O valor a ser transferido é válido;• O saldo do cliente sacado deve ser maior ou igual ao valor do saque.
Pós-condições	<ul style="list-style-type: none">• O valor é debitado na conta do sacado e creditado na conta do beneficiário;• A transação é inserida na lista de transações.
Ações	<ol style="list-style-type: none">1. Cria-se uma nova instância de transação;<ol style="list-style-type: none">a. Atribui as informações de tipo e valor;b. Atribui os números das contas dos clientes;2. Debita o valor da transação na conta do cliente sacado;3. Credita o valor da transação na conta do cliente beneficiário;4. Insere a transação na lista de transações.

Enunciado do Problema

Em equipe de até 3 alunos, desenvolver um sistema para gerenciar o cadastro de clientes e transações financeiras do banco. O banco consegue atender a no máximo **50** clientes e o cadastro de clientes deve ser mantido utilizando uma **lista sequencial ordenada** pelo código do cliente. A ordenação será consequência da inserção ordenada dos clientes num cadastro inicialmente vazio.

O sistema deve permitir a inclusão, exclusão e atualização de clientes. Ao incluir um novo cliente, devem ser preenchidas as informações cadastrais. O cliente também deve efetuar um depósito inicial maior ou igual a R\$50,00 em sua própria conta. A operação de atualização poderá somente alterar as informações cadastrais do cliente, ou seja, *nome* e *telefone*. A operação de exclusão deve remover o cliente da lista de clientes.

O cadastro de transações de cada cliente deve ser mantido em uma **pilha sequencial**. As funções de inserção e remoção devem atender aos requisitos funcionais mencionados anteriormente.

Atenção: o sistema deve utilizar Tipos Abstratos de Dados (TAD) adequados à aplicação, com as funções de manutenção necessárias. Desta forma, as estruturas utilizadas deverão estar coerentes, pois serão avaliadas.

Interface

Ao ser executado, o sistema deve exibir a identificação do grupo que o desenvolveu (número do grupo, nome dos participantes) e apresentar um Menu principal com as seguintes opções:

1. Clientes;
2. Transações;



3. Sair.

Ao acessar o menu **clientes**, deverá ser exibido um sub-menu com as seguintes opções:

1. Inserir novo cliente;
2. Consultar dados de um cliente; *(esta opção deverá imprimir as informações cadastrais do cliente e as transações relacionadas a sua conta).*
3. Atualizar cadastro de um cliente;
4. Remover um cliente;
5. Imprimir lista de clientes;
6. Voltar.

Ao acessar o menu **transações**, deverá ser exibido um sub-menu com as seguintes opções:

1. Depósito;
2. Saque;
3. Transferência;
4. Imprimir lista de transações;
5. Voltar.

Ferramentas

A implementação do trabalho será em linguagem **C**, utilizando o compilador GCC.

Diversos ambientes de programação utilizam o GCC como compilador padrão, com é o caso do Dev-C++ (<http://dev-c.softonic.com.br/>) e o Code Block (<http://www.codeblocks.org/>), inclusive esse compilador é o padrão da maioria das distribuição Linux.

Caso deseje instalar esse compilador para ser usado em linha de comando nos sistemas Windows, baixar o MinGW (<http://www.mingw.org/>, baixar em <http://sourceforge.net/projects/mingw/>).



Entrega do Trabalho

Envie, com antecedência, para o monitor, os nomes dos alunos de seu grupo. Ele receberá um número e este número deverá constar do arquivo a ser enviado ao monitor por ocasião da entrega do trabalho.

Deve ser entregue, até a data determinada, um arquivo zipado chamado GrupoXX.zip, onde XX corresponde ao número do seu grupo, contendo os arquivos abaixo:

- **Clientes.c** (TAD da implementação da lista de clientes);
- **Clientes.h** (*Header* do TAD da implementação da lista de clientes);
- **Transacoes.c** (TAD da implementação da lista de transações);
- **Transacoes.h** (*Header* do TAD da implementação da lista de transações);
- **Banco.c** (Programa principal com a interface do sistema);

A entrega do arquivo deve ser pelo email arquivos.maziero@gmail.com. Serão considerados os email enviados até às 23:59 do dia 29/09/2010. Trabalhos fora do prazo serão penalizados (como descrito abaixo).

Critérios de Avaliação e Penalidade

A nota do trabalho 1 (NT1) será constituída da seguinte maneira:

$$NT1 = 0.4*NP1 + 0.4*NP2 + 0.2*NP3, \text{ onde:}$$

NP1 é a nota do TAD Clientes (implementação usando lista sequencial);

NP2 é a nota do TAD Transações (implementação usando lista sequencial);

NP3 é a nota do programa principal.

As notas dos programas **NPx** terão os seguintes critérios, com os respectivos pesos:

$$NPx = 0.5*NC + 0.2*NE + 0.1*NI + 0.2*NCF, \text{ onde:}$$

NC é a nota do critério Correção;

NE é a nota do critério Eficiência;

NI é a nota do critério Interface;

NCF é a nota do critério Código Fonte.



- **Correção (NC):** o programa faz o que foi solicitado? Faz tudo o que foi solicitado?

Utiliza encapsulamento de informação? serão considerados nos critérios de avaliação:

- O uso correto dos TAD e operações de manutenção relacionadas;
 - Cumprimento dos requisitos funcionais, ou seja, realiza as transações da maneira correta;
- **Eficiência (NE):** as operações são executadas da maneira mais eficiente para cada estrutura de dados? Evita código duplicado/redundante/não atingível?
- **Interface (NI):** é simples de usar, prático, tolera os erros mais óbvios? O trabalho foi entregue dentro das especificações (zipado, com os nomes de arquivo solicitados)?
 - Interface do programa;
 - Interface de entrega do trabalho;
- **Código fonte (NCF):** é claro e organizado? Nomes de variáveis são sugestivos? Está bem documentado?
 - Clareza;
 - Nomes de variáveis;
 - Documentação/comentários no código.

A nota do trabalho sofrerá penalidade se entregue depois do prazo estipulado, conforme os critérios abaixo:

NT1 = $k \cdot \text{NT1}$, onde k é um fator multiplicador dado por:

$k = 1.0$, se o trabalho for entregue dentro do prazo determinado (não há penalidade);

$k = 0.7$, se o trabalho for entregue até 24hs após o prazo determinado;

$k = 0.5$, se o trabalho for entregue entre 24hs e 48hs após o prazo estipulado;

$k = 0.0$, se o trabalho for entregue mais de 48hs após o prazo estipulado (será considerado desistente).

Para cada trabalho recebido por e-mail será enviada uma resposta de confirmação!