ELE078 - PROGRAMAÇÃO ORIENTADA A OBJETOS

TRABALHO PRÁTICO 3

I – Observações:

1) O trabalho pode ser feito em grupo de no máximo 5 pessoas. Em nenhuma hipótese poderá haver mais de 5 alunos por grupo. É permitido discutir os problemas e estratégias de solução com outros grupos, mas quando se tratar de escrever ou implementar computacionalmente as soluções, isto deve ser feito apenas com os componentes de seu grupo. Utilizar o trabalho dos outros, como se fosse seu, é plágio. Desonestidade acadêmica será punida com severidade.

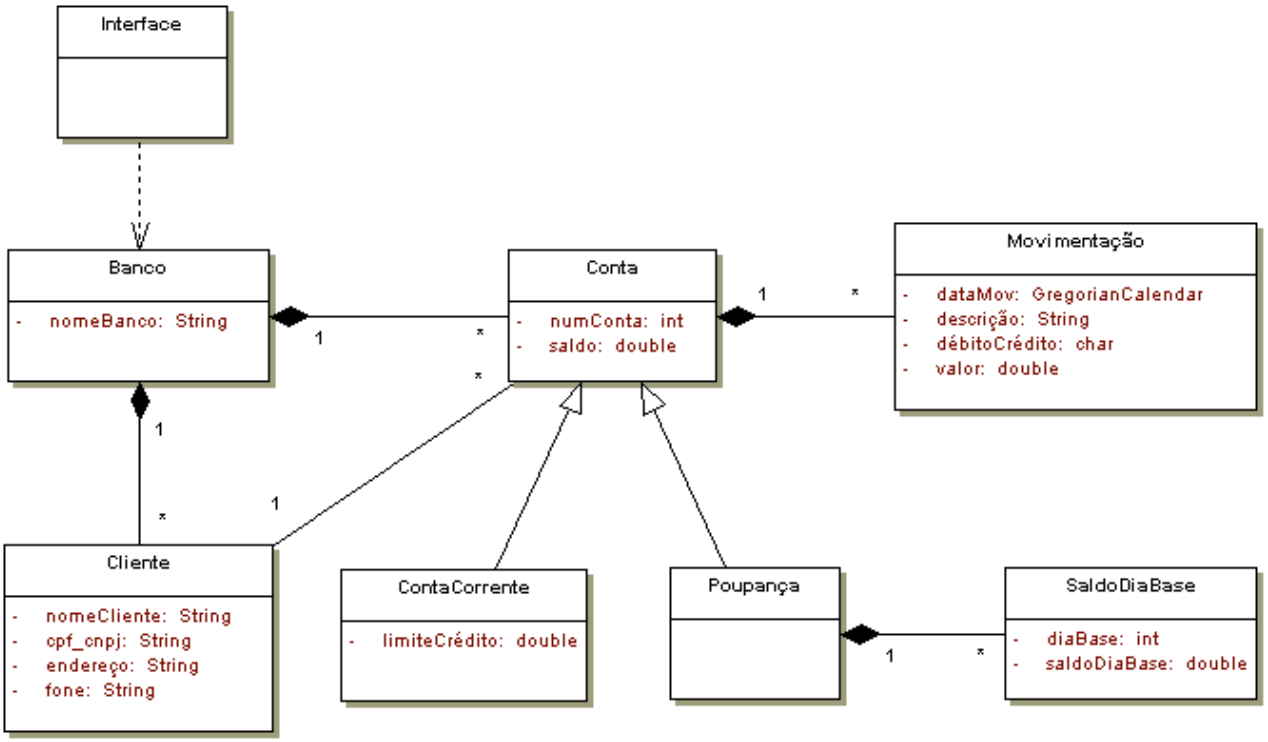
2) Forma de entrega: O trabalho deve ser entregue em formato digital por meio do Moodle. Utilizar a opção "Entrega do Trabalho Pratico 1". Anexe um único arquivo .zip contendo todos os arquivos do trabalho (códigos-fonte, executáveis, documentação, etc.). O nome do arquivo .zip a ser enviado deve ser formado pelos nomes de todos os integrantes do grupo. Assim, se o grupo é formado pelos alunos: Cristiano Leite Castro, Danilo Melges e André Freire, o nome do arquivo deverá ser:

CristianoCL_DaniloM_AndreF.zip.

3) O link para envio do TP2 no Moodle terá uma tolerância de 24 horas para recebimentos dos trabalhos. Após esse período, os trabalhos não serão recebidos.

Sistema de Controle de Bancário (Parte II)

Implemente em C++ um conjunto de classes para um sistema de controle bancário. Este trabalho consiste em uma extensão (alteração) do Trabalho Prático I, com as seguintes considerações:



- Contas devem ser somente do tipo ContaCorrente ou Poupança, não existirão objetos do tipo Conta. Para isso, a classe Conta passa a ser uma classe abstrata;
- O atributo limiteCrédito de ContaCorrente indica um valor limite que a conta pode ficar com saldo negativo. Por exemplo, se uma conta tem um limite de crédito de R$800,00 e um saldo de R$1.000,00, significa que pode haver um débito (saque, transferência etc.) de até R$1.800,00;
- Uma Poupança possui um controle de saldo por dia base. O dia base é o dia do mês em que são creditados rendimentos de juros para a conta. Uma poupança pode ter vários dias base e o saldo total da conta será a soma do saldo de todos os dias base. Ao efetuar um depósito em uma poupança, se já existir um objeto para o dia do depósito, o valor do saldo para este dia deve ser incrementado e, se não houver, um novo objeto deve ser criado. Créditos nos dias 29, 30 e 31 de um mês devem ser devem ser criados como se tivessem sido efetuados no dia 28. Ao efetuar um débito em uma poupança, o saldo do dia base correspondente ao dia do débito deve ser decrementado. Se não houver saldo suficiente neste dia base, o saldo do dia base anterior mais próximo deve ser decrementado. Esse ciclo deve ser efetuado sucessivamente até decrementar o valor total do débito ou cancelar a operação se não houver saldo suficiente;
- Todas as situações de exceção devem ser tratadas. Por exemplo, débitos com saldo insuficiente e exclusão de cliente com conta não devem ser permitidos. Crie classes de exceção e faça todo o tratamento necessário. Impressão de mensagens de erros devem ser apresentadas somente na classe de interface;
- Alguns métodos devem ser alterados e outros acrescentados:
  - Os métodos para criar uma nova conta devem ser alterados para permitir que o usuário informe o tipo de conta: corrente ou poupança. No caso de conta corrente, o limite de crédito deve ser informado;
  - Os métodos de débito e crédito da classe Conta devem ser ajustados de acordo com as considerações acima;
  - Devem ser adicionados métodos para obter os saldos por dia base de poupanças. Ao exibir um extrato também devem ser exibidas informações de saldo detalhadas por dia base;
  - Devem ser adicionados métodos para creditar rendimentos de juros de poupança. Ao ser acionada a opção de creditar juros, os saldos para o dia base correspondente ao dia da data corrente, para todas as contas de poupança, devem receber um crédito correspondente a 1% do valor do saldo do dia. Por simplicidade do trabalho, os juros serão fixos em 1% e a opção de rendimento pode ser acionada a qualquer momento, inclusive várias vezes no mesmo dia.
- **Use polimorfismo sempre que possível**.

Observações a serem seguidas:

- As classes, atributos e métodos descritos constituem o projeto básico do sistema. Pode-se adicionar outros métodos e atributos necessários para completar o sistema.
- Leitura de dados do usuário e impressão na tela devem ser feitos somente na classe Interface;

Pontos Extras

- 3,0 pontos extras para trabalho com interface gráfica ou web.

- 2,0 pontos extras para trabalho com armazenamento de dados em um banco de dados.
- Somente receberão os pontos extras os alunos do grupo que souberem explicar detalhadamente o funcionamento do seu trabalho.