Especificação do Trabalho Prático 1

Elaborar um programa em *C++* que gerencie contas correntes de clientes de um banco, oferecendo as seguintes funcionalidades:

- Gerenciamento de clientes: inclusão, alteração e exclusão de clientes. Para cada cliente, os seguintes dados são armazenados: nome, CPF, endereço, telefone, e e-mail. O programa não deve permitir CPFs repetidos para clientes.
- Gerenciamento de contas: inclusão, alteração e exclusão de contas. Para cada conta, os seguintes dados devem ser armazenados: número da conta, data de abertura, CPF do cliente, e saldo atual. O programa não deve permitir números repetidos de contas. O cliente da conta deve estar previamente cadastrado no sistema. Somente datas válidas devem ser permitidas.
- Lançamentos em conta corrente. Um lançamento deve especificar a data do lançamento, o número da conta corrente, o tipo (débito ou crédito), e o valor. Um lançamento de débito somente pode ocorrer se a conta tiver saldo suficiente para não ficar negativa. Após o lançamento ter sido realizado, o saldo atual da conta deve ser atualizado.
- Extrato de conta corrente. Para uma dada conta corrente, os respectivos lançamentos devem ser listados, apresentando-se o saldo final atualizado. O extrato deve considerar todo os lançamentos feitos para a conta em questão.

Considere as seguintes especificações para as classes a serem criadas:

- Para todas as classes, crie os métodos *setters* e *getters*.
- Para cada classe, crie um método que retorna uma *string* que representa os dados do objeto. Utilize comandos de formatação para criar uma *string* de fácil leitura.

```
string toString()
```

- O atributo `saldoAtual` da classe `ContaCorrente` deve ser do tipo `float`.
- O programa principal deve ter uma função que retorna o número de contas criadas no sistema.

```
int getQuantidadeDeContas()
```

- O programa principal deve ter uma função que retorna o número de clientes cadastrados no sistema.

```
int getQuantidadeDeClientes()
```

- O programa principal deve ter uma função que retorna o valor total armazenado no banco, considerando todas as contas cadastrada no sistema.

```
float getMontanteTotal()
```

Programa Principal e Interface

- O programa principal deve ser capaz de manipular conjuntos de elementos. Por exemplo, deve manipular os conjuntos de contas, conjuntos de clientes, e conjuntos de lançamentos em conta corrente. Sugere-se o uso de *arrays* para armazenar esses dados.

- A interface deve possibilitar a escolha da funcionalidade desejada pelo usuário. Deve permitir, *por exemplo*, a inclusão de um novo cliente, a consulta de clientes, a criação de uma nova conta, e o lançamento de débito ou crédito em uma conta.

Itens Obrigatórios e Desejáveis de Programação

- **Obrigatórios:** classes para representar as informações a serem manipuladas no sistema; documentação interna do código.

Dicas e Observações

- Utilize estruturas (*struct*) para representar dados compostos das classes criadas (por exemplo, data de abertura de conta, e endereço de cliente).

- Utilize uma (ou mais) função auxiliar para inicializar as coleções de objetos manipulados no sistema.

- Inicie pela implementação das classes mais simples. Pense na solução por partes, porém sempre considerando o sistema como um todo.

Critérios de avaliação

- (peso 5) Funcionalidades implementadas corretamente. As funcionalidade serão validadas em sala de aula, por meio de testes realizados com o professor. O professor irá definir e implementar um conjunto de testes que será executado no sistema criado.
- (peso 2) Qualidade da interface criada. A interface pode ser simples, porém com interação clara entre o usuário e a sistema.
- (peso 1) Qualidade da documentação interna do código.

Instruções Gerais

- O trabalho deverá ser realizado por grupos de 4 alunos. Exceções devem ser previamente autorizadas pelo professor.
- O trabalho deve ser entregue em arquivo único no Moodle. O arquivo deverá conter o programa fonte escrito em *C++*, compilável na IDE NetBeans. Quaisquer outros arquivos necessários deverão ser enviados junto com o programa. O mesmo programa enviado ao Moodle será utilizado para a avaliação junto ao professor.
- **A data de entrega é 08/10/2019 (horário limite: 15h55). Apenas 1 aluno do grupo precisa entregar. Os nomes de todos os alunos do grupo devem constar na entrega.**