

Aluno: Alaf do Nascimento Santos – Matrícula: 2017100781

Professor: Rodrigo F. Berriel

Programação Aplicada de Computadores (INF09324) – 2017/2

EXPLICAÇÃO DOS CÓDIGOS E EXECUÇÃO DO PROGRAMA

Entrada.h e Entrada.cpp

Uma biblioteca que possui uma função que serve para ler argumentos passados para main de um programa C++.

Erros.h

Uma classe utilizada para tratar exceções lançadas do banco para main.

CNPJ.h e CNPJ.cpp

A classe CNPJ serve para armazenar o documento de identificação de uma pessoa jurídica. Nessa classe existem dois atributos privados e constantes, que são duas formas de representar o CNPJ, um é string (com o formato padrão de um CNPJ) e o outro é um inteiro (sem os pontos e traço de um CNPJ).

Os principais métodos públicos são:

- **ConverteCNPJ:** converte o CNPJ de string (com pontos e traço) para um inteiro.
- **GetCNPJ:** retorna o inteiro cnpj, que servirá para identificar a Pessoa Jurídica em questão (cada pessoa jurídica possui um CNPJ).
- **GetId:** retorna a string cnpj, para que será impressa em algumas exceções, quando necessária.

CPF.h e CPF.cpp

A classe CPF serve para armazenar o documento de identificação de uma pessoa física. Nessa classe existem dois atributos privados e constantes, que são duas formas de representar o CPF, um é string (com o formato padrão de um CPF) e o outro é um inteiro (sem os pontos e traço de um CPF).

Os principais métodos públicos são:

- **ConverteCPF:** converte o CPF de string (com pontos e traço) para um inteiro.
- **GetCPF:** retorna o inteiro cpf, que servirá para identificar a Pessoa Física em questão (cada pessoa física possui um CPF).
- **GetId:** retorna a string cpf, para que será impressa em algumas exceções, quando necessária.

Pessoa.h, PessoaFisica.h, PessoaFisica.cpp, PessoaJuridica.h PessoaJuridica.cpp

Esses arquivos possuem códigos semelhantes, sendo que, Pessoa.h é uma classe abstrata utilizada na implementação de comportamentos polimórficos no programa e cada pessoa é composta de uma string nome, um double salario (constante), um CPF (pessoas físicas) ou CNPJ (pessoas jurídicas), um bool tipo (true para PF e false para PJ) e um bool contaAtiva (true se já tiver uma conta, false se não tiver uma conta), além disso, uma pessoa jurídica tem uma pessoa física, que é o seu dono.

Os métodos existentes são:

- **GetTipo:** retorna true, se for pessoa física, e false, se for pessoa jurídica.
- **GetContaAtiva:** retorn true, se já possuir uma conta, e false, se não tiver uma conta ativa.
- **SetContaAtiva:** muda o estado de conta ativa da pessoa, on/off, true/false.
- **GetSalario:** retorna o salario.
- **GetIdentificacao:** retorna um inteiro documento da pessoa, CPF para pessoa física e CNPJ para pessoa jurídica.
- **GetNome:** retorna o nome da pessoa.
- **GetId:** retorna uma string, que representa o documento de identificação da pessoa, isso é usado na impressão de exceções.

ContaBancaria.h, ContaCorrente.h, ContaCorrente.cpp, ContaPoupanca.h e ContaPoupanca.cpp

Esses arquivos possuem códigos semelhantes, sendo que, ContaBancaria.h é uma classe abstrata utilizada na implementação de comportamentos polimórficos no programa e cada conta possui um ponteiro de pessoa que irá apontar para o endereço de um objeto de pessoa física ou jurídica.

Os métodos existentes são:

- Deposito: soma um valor ao saldo da conta.
- Saque: diminui um valor do saldo da conta
- GetIdentificacao: retorna um documento que representa o cliente dono da conta
- GetSaldo: retorna o valor do saldo da conta.
- GetLimite: retorna o limite de credito da conta.
- GetTipo: um método bool que retorna 1 ou 0 para dizer se a conta é poupança ou corrente, respectivamente.
- GetTipoCliente: um método bool que retorna 1 ou 0 para dizer se o dono da conta é uma pessoa física ou jurídica, respectivamente.
- GetNomeCliente: retorna o nome do dono da conta.
- AumentaLimite: caso não exista dívida, esse método aumenta o limite de crédito de uma conta corrente no encerramento de mês.
- GetSalarioDono: retorna o valor do salário do dono.
- GetId: retorna o documento do dono CPF/CNPJ.

Banco.h e Banco.cpp

Um banco composto de contas: basicamente o Banco possui um vetor de ponteiros ContaBancaria e cada conta bancária possui uma pessoa (física ou jurídica), além disso, o banco também possui um vetor de pessoas físicas e outro de pessoas jurídicas.

A principal característica do polimorfismo presente nesse programa se dá no banco, pois o vetor de ponteiros de ContaBancaria possui é utilizado para acessar funções de diferentes tipos de contas, ou seja, com uma mesma instrução, posso ter diferentes comportamentos, como por exemplo a função getTipo(), que pode retornar true ou false, dependendo do contexto.

As principais funções membro do banco são:

- Sacar: essa função verifica os possíveis casos de um saque e chama o método saque() através do vetor contas.
- Depositar: essa função verifica os possíveis casos de um depósito e chama o método deposito() através do vetor contas.
- Transferir: essa função verifica os possíveis casos de uma transferência e chama o método transferir() através do vetor contas.
- EncerrarMes: essa função aplica os juros, taxas, deposita os salários, etc. Basicamente esse método faz todos os processos necessários para se encerrar o mês e também armazena as variáveis “contadores” em seus respectivos vetores e depois zera elas para que possam voltar a contar os dados do mês. Esses vetores que guardam variáveis contadores são utilizados na hora de gerar o relatório.txt.
- EncerrarConta (operator-): essa é a sobrecarga do operador –, na qual um banco pode realizar uma operação de subtração com uma conta bancária, ou seja, “banco – conta” realiza a tentativa de encerramento daquela conta, podendo ou não obter sucesso, caso falhe, uma exceção é lançada para informar o motivo da falha.
- GerarRelatorio: essa função abre um arquivo relatório.txt e insere os dados de execução nele para mostrar informações sobre a execução do programa.

Main.cpp

O programa começa lendo os argumentos de entrada providos dos arquivos clientes.txt e transacoes.txt, isso é feito com o auxílio de uma função de leitura presente na biblioteca Entrada.h. Esses dados são armazenados e um vetor de transações é criado.

Com esse vetor de transações o programa verifica se deve abrir ou não o banco e se o saldo inicial do banco for menor ou igual a zero, o mesmo não é aberto e o programa termina instantaneamente após uma mensagem de erro.

Se o banco for aberto com sucesso, o programa usa uma função dele para “criar” os clientes no banco e depois disso um loop que vai durar enquanto houverem transações a serem feitas é iniciado.

Dentro desse loop há um bloco try que tenta realizar cada transação e caso exista algum erro ele é pego pelo catch e uma mensagem de erro ou aviso é impressa na tela e a próxima interação do loop é iniciada.

Quando as transações terminam o banco gera um relatório com dados sobre sua atividade ao longo do programa, esses dados são inscritos em um arquivo relatorio.txt., após isso o programa fecha.

EXPLICAÇÃO DAS ENTRADAS E SAIDAS GERADO.

➤ Pessoas Físicas:

- Pessoa1 Sobrenome1 que possui CPF 123.456.789-10 e salário R\$ 1.000,00.
- Pessoa2 Sobrenome2 que possui CPF 234.567.890-00 e salário R\$ 2.000,00.
- Pessoa3 Sobrenome3 que possui CPF 345.678.900-00 e salário R\$ 3.000,00.

➤ Pessoas Jurídicas:

- Empresa1 P1 que possui CNPJ 01.531.015/6969-35 e dono Pessoa1.
- Empresa2 P2 que possui CNPJ 02.101.024/7070-70 e dono Pessoa2.

PRIMEIRO MÊS

O banco iniciou com R\$ 100.000.000,00 de saldo.

O cliente Pessoa1 abriu uma conta corrente na qual foi definido um saldo inicial de R\$ 1.000,00 e limite de crédito R\$ 700,00 referente ao seu salário.

A empresa do cliente Pessoa1, chamada Empresa1 P1, tentou abrir uma conta poupança e não obteve sucesso, pois somente pessoas físicas podem abrir uma conta poupança.

ERRO: clientes PJ não podem abrir CP

A empresa do cliente Pessoa1 (Empresa1 P1) abriu uma conta corrente com um saldo inicial de R\$ 5.000,00 e limite de créditos igual a R\$ 1.500,00. Ela sacou esse dinheiro e logo após, encerrou a conta.

O cliente Pessoa2 abriu uma conta poupança e teve seu salário depositado, ou seja, um saldo inicial de R\$ 2.000,00.

Pessoa1 tenta sacar R\$ 10.000,00 e não consegue, pois, seu saldo é insuficiente.

ERRO: o cliente 123.456.789-10 não possui saldo suficiente

Pessoa1 saca R\$ 1.500,00 com sucesso, utilizando seu limite de crédito.

AVISO: esse saque usou o limite de crédito do cliente 123.456.789-10

Saldo_Pessoa1: - R\$ 500,00

Saldo_Pessoa2: + R\$ 2.000,00

Pessoa2 tenta abrir mais uma conta poupança e não consegue pois só é permitido uma conta bancária por pessoa.

ERRO: o cliente 234.567.890-00 não pode abrir mais contas

Depois disso Pessoa2 transfere com sucesso R\$ 1.000,00 para Pessoa1 e esse valor é utilizado parcialmente para pagar a dívida que Pessoa1 possuía com o banco.

AVISO: esse valor foi usado (total ou parcialmente) para pagar dívida de créditos rotativo

Saldo_Pessoa1: + R\$ 500,00

Saldo_Pessoa2: + R\$ 990,00

Pessoa2 saca R\$ 990,00 e Pessoa1 saca R\$ 500,00 e assim termina o mês. Todos os clientes receberam seus salários e as devidas taxas e juros foram aplicadas (1% de rendimento para poupanças, 10% de juros para crédito PF, 15% de juros para crédito PJ e 1% de rendimento para o banco) e todos os clientes receberam seus salários.

Saldo_Banco: +R\$ 100.999.000,10

Saldo_Pessoa1: +R\$ 1.000,00

Limite_credito_Pessoa1: +R\$ 714,00

Saldo_Pessoa2: +R\$ 2.000,00

SEGUNDO MÊS

O cliente Pessoa3 abriu uma conta poupança, com saldo inicial de R\$ 3.000,00, e depositou R\$ 7.000,00 deixando seu saldo igual a R\$ 10.000, depois disso ele tentou encerrar sua conta, porém isso só é possível quando o cliente não possui dívida ou saldo na conta.

ERRO: não é permitido encerrar conta com saldo ou dívida

Pessoa1 efetuou um saque no valor de R\$ 500,00 (ficando assim com R\$ 500,00 na conta) e Pessoa2 depositou o mesmo valor (ficando assim com R\$ 2.500,00 na conta).

Pessoa1 tentou transferir R\$ 1.000,00 para Pessoa2, mas não possuía saldo suficiente para pagar a taxa de transferência.

ERRO: o cliente 123.456.789-10 não possui saldo suficiente

Pessoa1 efetuou um saque no valor de R\$ 500,00 (ficando assim com R\$ 0,00 na conta) e Pessoa2 depositou R\$ 500,00 (ficando assim com R\$ 3.000,00 na conta).

Pessoa3 transferiu R\$ 5.000,00 para Pessoa1 e R\$ 4.900,00 para Pessoa2, depois tentou fechar sua conta, mas ainda restava R\$ 1,00 lá.

ERRO: não é permitido encerrar conta com saldo ou dívida

Pessoa2 sacou R\$ 7.900,00, zerando seu saldo.

O mês é encerrado e todos os clientes receberam seus salários e as devidas taxas e juros foram aplicadas (1% de rendimento para poupanças, 10% de juros para crédito PF, 15% de juros para crédito PJ e 1% de rendimento para o banco) e todos os clientes receberam seus salários.

Saldo_Banco: +R\$ 102.009.090,09

Saldo_Pessoa1: + R\$ 6.000,00

Limite_Pessoa1: + R\$ 728,28

Saldo_Pessoa2: + R\$ 2.000,00

Saldo_Pessoa3: + R\$ 3.001,01

TERCEIRO MÊS

Pessoa1 sacou R\$ 1.000,00.

Pessoa2 sacou R\$ 1.000,00.

Pessoa3 sacou R\$ 1.000,00.

Empresa2 P2 (da Pessoa2) abriu uma conta corrente com saldo inicial de R\$ 10.000,00 e limite de crédito R\$ 3.000,00.

Empresa2 depositou R\$ 100.000,00 e depois transferiu R\$ 5.000,00 para Pessoa1, Pessoa2 e Pessoa3.

Saldo_Pessoa1: +R\$ 10.000,00

Saldo_Pessoa2: +R\$ 6.000,00

Saldo_Pessoa3: +R\$ 7.001,01

Saldo_Empresa2: +R\$ 94.850,00

Pessoa1 sacou R\$ 1.000,00.

Pessoa2 sacou R\$ 1.000,00.

Pessoa3 sacou R\$ 1.000,00.

Pessoa1 sacou R\$ 1.000,00.

Pessoa2 sacou R\$ 1.000,00.

Pessoa3 sacou R\$ 1.000,00.

Pessoa1 sacou R\$ 1.000,00.

Pessoa2 sacou R\$ 1.000,00.

Pessoa3 sacou R\$ 1.000,00.

Pessoa1 sacou R\$ 1.000,00.

Pessoa2 sacou R\$ 1.000,00.

Pessoa3 sacou R\$ 1.000,00.

Empresa2 depositou R\$ 100.000,00 e depois transferiu R\$ 5.000,00 para Pessoa1, Pessoa2 e Pessoa3, respectivamente.

Saldo_Pessoa1: +R\$ 11.000,00

Saldo_Pessoa2: +R\$ 7.000,00

Saldo_Pessoa3: +R\$ 8.001,01

Saldo_Empresa2: +R\$ 179.700,00

Empresa2 sacou R\$ 179.690,00 e tentou encerrar sua conta, mas ainda havia saldo lá.

ERRO: não é permitido encerrar conta com saldo ou dívida

Pessoa2 depositou R\$ 7.000,00.

O mês é encerrado e todos os clientes receberam seus salários e as devidas taxas e juros foram aplicadas (1% de rendimento para poupanças, 10% de juros para crédito PF, 15% de juros para crédito PJ e 1% de rendimento para o banco) e todos os clientes receberam seus salários.

O programa encerrou junto ao encerramento do mês e os valores finais foram:

- Saldo do Banco: R\$103029483,99
- Pessoa1: R\$ 12.000,00
- Limite de crédito Pessoa1: R\$ 742,84
- Pessoa2: R\$ 16.140,00
- Pessoa3: R\$ 11.081,02
- Empresa2: R\$ 10,00
- Limite de crédito Empresa2: R\$ 3.060,00

RELATÓRIO GERADO:

Quantidade de contas ativas: 4 (2 CP e 2 CC)

Saldo do banco: R\$ 103029483.99 (+ R\$ 3029483.99)

Dívida dos clientes: R\$ 0.00 (0 PF e 0 PJ)

Total disponibilizado em limite de crédito: R\$ 3802.00

Erros: 7

MÊS 1

Transações: 9 efetuadas e 3 com erro

Total de contas ativas: 2 (3 novas e 1 encerradas)

Rendimento pago a CP: R\$ 0.00

Rendimento recebido: R\$ 999990.10

Juros e taxas recebidas: R\$ 490.00

Dívida dos clientes: R\$ 0.00

Saldo dos clientes: R\$ 0.00

Cliente mais rico: Pessoa1 Sobrenome1

Cliente com maior dívida: N/A

Saldo do banco: R\$ 100999000.10

MÊS 2

Transações: 11 efetuadas e 3 com erro

Total de contas ativas: 3 (1 novas e 0 encerradas)

Rendimento pago a CP: R\$ 0.01

Rendimento recebido: R\$ 1009991.00

Juros e taxas recebidas: R\$ 99.00

Dívida dos clientes: R\$ 0.00

Saldo dos clientes: R\$ 5001.01

Cliente mais rico: Pessoa1 Sobrenome1

Cliente com maior dívida: N/A

Saldo do banco: R\$ 102009090.09

MÊS 3

Transações: 32 efetuadas e 1 com erro

Total de contas ativas: 4 (1 novas e 0 encerradas)

Rendimento pago a CP: R\$ 220.01

Rendimento recebido: R\$ 1020313.91

Juros e taxas recebidas: R\$ 300.00

Dívida dos clientes: R\$ 0.00

Saldo dos clientes: R\$ 33231.02

Cliente mais rico: Pessoa3 Sobrenome3

Cliente com maior dívida: N/A

Saldo do banco: R\$ 103029483.99