



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

AD1 de Programação III

2º semestre de 2007.

Nome:

Matrícula:

Pólo:

Implemente um programa que permita a criação de objetos do tipo Conta Corrente, Conta Poupança, Cheque Especial e Conta Investimento. Esta implementação deve fazer uso intenso de mecanismos de reutilização de código, como a herança entre classes OO (Orientadas à Objeto). Dados comuns entre estas contas seria o número da conta, nome, CPF, data de abertura da conta, senha e saldo corrente. Defina construtores para a inicialização destes campos. Também devem estar disponíveis operações básicas em contas, como saque, depósito, saldo e transferência entre contas. Para alguns casos, como Conta Poupança, operações de saque e transferência só devem ser efetuadas em poupanças com saldo suficiente. Contas do tipo Cheque Especial deverão ter um campo a mais que indique o limite do saldo especial do cliente. Em contrapartida, Contas Investimento deverão ter um saldo mínimo mensal disponível em conta. Cada Conta Investimento deverá necessariamente estar associada a uma Conta Corrente. Com isso, operações de saque em Conta Investimento representarão resgates (transferências para a Conta Corrente associada), enquanto que transferências diretas para outras contas não são permitidas. Ainda comparando tipos de contas, as Corrente poderão ter seu saldo negativo 1 vez, devendo ser restabelecida numa transação seguinte, enquanto que as contas Poupança nunca poderão estar nesta situação. Movimentações de saque, depósito e transferência terão que sofrer desconto de CPMF. O valor dessa taxa é único entre as contas, mas que pode ser modificado. Resgates de Contas Investimento não terão desconto de CPMF, mas terão descontos fiscais de manutenção destas contas. Os valores destas taxas poderão ser fornecidos na própria classe, ou na inicialização dos objetos do tipo conta.

```

////////////////////////////////////
// Pacotes utilizados
import java.util.Date;

////////////////////////////////////
// Classe Conta com comportamento básico de contas
class Conta {
    int numero;
    String nome;
    String cpf;
    Date data_abertura;
    String senha;
    double saldo;

    static double cpmf = 0.01;

    public Conta(
        int num,
        String nom,
        String num_cpf,
        Date data,
        String pass,
        double valor) {
        numero = num;
        nome = nom;
        cpf = num_cpf;
        data_abertura = data;
        senha = pass;
        saldo = valor;
    }

    public void saque (double valor) {
        if (valor <= saldo)
            saldo = saldo - valor - valor*cpmf;
    }

    public void deposito (double valor) {
        if (valor > 0) {
            saldo = saldo + valor - valor*cpmf;
        }
    }

    public double saldo () {
        return saldo;
    }

    public void transferencia (Conta credito,

```

```

                                double valor) {
        if (valor <= saldo) {
            saldo = saldo - valor - valor*cpmf;
            credito.deposito(valor - valor*cpmf);
        }
    }
}

////////////////////////////////////
// Classe ContaCorrente com redefinição do método saque()
class ContaCorrente extends Conta {
    public ContaCorrente(int num,
                        String nom,
                        String num_cpf,
                        Date data,
                        String pass,
                        double valor) {
        super(num, nom, num_cpf, data, pass, valor);
    }

    public void saque (double valor) {
        if (saldo >= 0)
            saldo = saldo - valor - valor*cpmf;
    }
}

////////////////////////////////////
// Classe ContaPoupanca, desnecessária por não fazer
//nenhuma redefinição
class ContaPoupança extends Conta {
    public ContaPoupança(int num,
                        String nom,
                        String num_cpf,
                        Date data,
                        String pass,
                        double valor) {
        super(num, nom, num_cpf, data, pass, valor);
    }
}

////////////////////////////////////
// Classe ChequeEspecial com redefinição dos métodos
//saque() e transferencia() e definição do atributo
//saldo_especial
class ChequeEspecial extends Conta {
    double saldo_especial;

```

```

    public ChequeEspecial(int num,
                          String nom,
                          String num_cpf,
                          Date data,
                          String pass,
                          double valor,
                          double saldo) {
        super(num, nom, num_cpf, data, pass, valor);
        saldo_especial = saldo;
    }

    public void saque (double valor) {
        if (valor <= saldo + saldo_especial)
            saldo = saldo - valor - valor*cpmf;
    }

    public void transferencia (Conta credito,
                              double valor) {
        if (valor <= saldo + saldo_especial) {
            saldo = saldo - valor - valor*cpmf;
            credito.deposito(valor - valor*cpmf);
        }
    }
}

////////////////////////////////////
// Classe ContaInvestimento com redefinição dos métodos
//saque, transferencia (não faz nada e redefine o
//comportamento padrão, pois não faz sentido para conta
//investimento) e dos atributos minimo_mensal e da
//conta corrente associada
class ContaInvestimento extends Conta {
    double minimo_mensal;
    ContaCorrente cc;

    static double taxa_manut = 0.001;

    public ContaInvestimento(int num,
                             String nom,
                             String num_cpf,
                             Date data,
                             String pass,
                             double valor,
                             ContaCorrente conta,
                             double minimo) {
        super(num, nom, num_cpf, data, pass, valor);
        cc = conta;
    }
}

```

```

        minimo_mensal = minimo;
    }

    public void saque (double valor) {
        if (valor <= saldo && (saldo - valor) >=
minimo_mensal) {
            saldo = saldo - valor - valor*taxa_manut;
            cc.deposito(valor - valor*taxa_manut);
        }
    }

    public void transferencia (Conta credito,
                                double valor) {
    }
}

////////////////////////////////////
//Classe AD120072 principal, onde podemos criar instâncias
//para testar as classes definidas
public class AD120072 {
    public static void main(String[] args) {
    }
}

```