**UNIVERSIDADE PRESBITERIANA MACKENZIE  
CIÊNCIA DA COMPUTAÇÃO (2023.2)  
ESTRUTURA DE DADOS II – 04D**

**POO: Herança e Polimorfismo**

|  |  |
| --- | --- |
| **NOME** | **TIA** |
| Jônatas Garcia de Oliveira | 42181232 |
| Pedro Henrique Araujo Farias | 32246994 |

**RELATÓRIO**

**EXERCÍCIO 01**

**Código-fonte: *ContaBancaria.java***

public class ContaBancaria {

    private String cliente;

    private int nroConta;

    private float saldo;

    public ContaBancaria() {

        this("\0", 0, 0.00f);

    }

    public ContaBancaria(String clienteArg, int nroContaArg, float saldoArg) {

        this.cliente = clienteArg;

        this.nroConta = nroContaArg;

        this.saldo = saldoArg;

    }

    public String getCliente() {

        return cliente;

    }

    public void setCliente(String clienteArg) {

        cliente = clienteArg;

    }

    public int getNroConta() {

        return nroConta;

    }

    public void setNroConta(int nroContaArg) {

        nroConta = nroContaArg;

    }

    public float getSaldo() {

        return saldo;

    }

    public void setSaldo(float saldoArg) {

        saldo = saldoArg;

    }

    public void sacar(float valor) {

        if(saldo < 0) {

            System.out.println("Não é possível realizar a operação\nO saldo atual da conta é insuficiente");

        } else {

            if(saldo - valor < 0) {

                System.out.println("Não é possível realizar a operação\nO saldo da conta após a operação é negativo");

            } else {

                saldo -= valor;

                System.out.println("Foram sacados R$" + valor + "\nSaldo atual: R$" + saldo);

            }

        }

    }

    public void depositar(float valor) {

        saldo += valor;

        System.out.println("Foram depositados R$" + valor + "\nSaldo atual: R$" + saldo);

    }

    @Override

    public String toString() {

        return "Nome: " + cliente + " | Número: " + nroConta +

        " | Saldo: R$" + saldo;

    }

}

**Código-fonte: *ContaEspecial.java***

public class ContaEspecial extends ContaBancaria {

    private float limite;

    public ContaEspecial() {

        this(0.00f);

    }

    public ContaEspecial(float limiteArg) {

        super("\0", 0, 0.00f);

        this.limite = limiteArg;

    }

    public ContaEspecial(String clienteArg, int nroContaArg, float saldoArg, float limiteArg) {

        super(clienteArg, nroContaArg, saldoArg);

        this.limite = limiteArg;

    }

    public float getLimite() {

        return limite;

    }

    public void setLimite(float limiteArg) {

        this.limite = limiteArg;

    }

    @Override

    public void sacar(float valor) {

        if(valor > super.getSaldo() + limite) {

            System.out.println("Não é possível realizar a operação\nO saldo limite é insuficiente");

        } else {

            limite = super.getSaldo() + limite - valor;

            super.setSaldo(0);

            System.out.println("Foram sacados R$" + valor + "\nSaldo atual: R$" + super.getSaldo());

        }

    }

    @Override

    public String toString() {

        return super.toString() + " | Limite: R$" + limite;

    }

}

**Código-fonte: *ContaPoupanca.java***

public class ContaPoupanca extends ContaBancaria {

    private int diaDeVencimento;

    public ContaPoupanca() {

        this(0);

    }

    public ContaPoupanca(int diaDeVencimentoArg) {

        super("\0", 0, 0.00f);

        this.diaDeVencimento = diaDeVencimentoArg;

    }

    public ContaPoupanca(String clienteArg, int nroContaArg, float saldoArg, int diaDeVencimentoArg) {

        super(clienteArg, nroContaArg, saldoArg);

        this.diaDeVencimento = diaDeVencimentoArg;

    }

    public int getDiaDeVencimento() {

        return diaDeVencimento;

    }

    public void setDiaDeVencimento(int diaDeVencimentoArg) {

        this.diaDeVencimento = diaDeVencimentoArg;

    }

    public void calcularNovoSaldo(float taxaRendimento) {

        float novoSaldo = super.getSaldo() + super.getSaldo() \* (taxaRendimento/100);

        super.setSaldo(novoSaldo);

        System.out.println("Novo saldo: R$" + novoSaldo);

    }

    @Override

    public String toString() {

        return super.toString() + " | Dia de Vencimento: " + diaDeVencimento;

    }

}

**Código-fonte: *Main.java***

public class Contas {

    public static void main(String[] args) {

        // Item A

        ContaBancaria contaSeuMadruga = new ContaBancaria("Seu Madrugada", 72, 200.0f);

        ContaPoupanca contaDonaClotilde = new ContaPoupanca("Dona Clotilde", 71, 1000.0f, 13);

        ContaEspecial contaDonaFlorinda = new ContaEspecial("Dona Florinda", 14, 2000.0f, 499.99f);

        // Item B

        System.out.println("|--- Sacando R$250,00 da conta de " + contaSeuMadruga.getCliente() + "---|");

        System.out.println("Saldo de " + contaSeuMadruga.getCliente() + ": R$" + contaSeuMadruga.getSaldo());

        contaSeuMadruga.sacar(250.0f);

        System.out.println("\n|--- Sacando R$750,00 da conta de " + contaDonaClotilde.getCliente() + "---|");

        System.out.println("Saldo de " + contaDonaClotilde.getCliente() + ": R$" + contaDonaClotilde.getSaldo());

        contaDonaClotilde.sacar(750.0f);

        System.out.println("\n|--- Sacando R$2200,99 da conta de " + contaDonaFlorinda.getCliente() + "---|" );

        System.out.println("Saldo de " + contaDonaFlorinda.getCliente() + ": R$" + contaDonaFlorinda.getSaldo());

        contaDonaFlorinda.sacar(2200.99f);

        // Item C

        System.out.println("\n|--- Depositando R$1000,00 na conta de " + contaSeuMadruga.getCliente() + "---|");

        contaSeuMadruga.depositar(1000.0f);

        System.out.println("\n|--- Depositando R$750,00 na conta de " + contaDonaClotilde.getCliente() + "---|");

        contaDonaClotilde.depositar(750.0f);

        System.out.println("\n|--- Depositando R$500,00 na conta de " + contaDonaFlorinda.getCliente() + "---|");

        contaDonaFlorinda.depositar(500.0f);

        // Item D

        System.out.println("\n|--- Saldo de " + contaDonaClotilde.getCliente() + "; Rendimento: 0,80% ---|");

        contaDonaClotilde.calcularNovoSaldo(0.80f);

        // Item E

        ContaBancaria[] arrayContas = {contaSeuMadruga, contaDonaClotilde, contaDonaFlorinda};

        System.out.println("\nInformações das contas registradas:");

        for(int i = 0; i < arrayContas.length; i++) {

            System.out.println(arrayContas[i]);

        }

    }

}

**Execução do método *main()* em *Contas.java***

**Texto

Descrição gerada automaticamente**

**EXERCÍCIO 02**

**Código-fonte: *Caminhao.java***

public class Caminhao extends VeiculoTerrestre {

    private int totalDeEixos;

    public Caminhao(int ano, String cor, int totalDeEixos) {

        super(ano, cor);

        this.totalDeEixos = totalDeEixos;

    }

    public int getTotalDeEixos() {

        return totalDeEixos;

    }

    public void setTotalDeEixos(int totalDeEixos) {

        this.totalDeEixos = totalDeEixos;

    }

    @Override

    public String toString() {

        return "Caminhao: " + super.toString() + " | Total de Eixos: " + totalDeEixos;

    }

}

**Código-fonte: *CarroDePasseio.java***

public class CarroDePasseio extends VeiculoTerrestre {

    private int totalDePassageiros;

    public CarroDePasseio() {

        super(0, "");

        this.totalDePassageiros = 0;

    }

    public CarroDePasseio(int ano, String cor, int totalDePassageiros) {

        super(ano, cor);

        this.totalDePassageiros = totalDePassageiros;

    }

    public int getTotalDePassageiros() {

        return totalDePassageiros;

    }

    public void setTotalDePassageiros(int totalDePassageiros) {

        this.totalDePassageiros = totalDePassageiros;

    }

    @Override

    public String toString() {

        return "Carro de passeio: " + super.toString() + " | Total de Passageiros: " + totalDePassageiros;

    }

}

**Código-fonte: *VeiculoTerrestre.java***

public class VeiculoTerrestre {

    private int ano;

    private String cor;

    protected VeiculoTerrestre() { // Acesso permitido apenas a esta classe e suas subclasses

        this(0, "");

    }

    public VeiculoTerrestre(int ano, String cor) {

        this.ano = ano;

        this.cor = cor;

    }

    public int getAno() {

        return ano;

    }

    public void setAno(int ano) {

        this.ano = ano;

    }

    public String getCor() {

        return cor;

    }

    public void setCor(String cor) {

        this.cor = cor;

    }

    @Override

    public String toString() {

        return "Ano: " + ano + " | Cor: " + cor;

    }

}

**Código-fonte: *Main.java***

public class Main {

    public static void main(String[] args) {

        VeiculoTerrestre[] exemplo = new VeiculoTerrestre[8]; // Criando um vetor de veículos terrestres

        exemplo[0] = new CarroDePasseio(1990, "Azul", 5);

        exemplo[1] = new CarroDePasseio(2004, "Cinza", 4);

        exemplo[2] = new Caminhao(1999, "Verde", 4);

        exemplo[3] = new CarroDePasseio(2010, "Preta", 5);

        exemplo[4] = new CarroDePasseio(2009, "Preta", 7);

        exemplo[5] = new Caminhao(2011, "Verde", 4);

        exemplo[6] = new Caminhao(2000, "Azul", 4);

        exemplo[7] = new Caminhao(2005, "Preto", 3);

        System.out.println("\t\t\tLISTA DE VEÍCULOS\n");

        for (VeiculoTerrestre veiculo : exemplo) {

            System.out.println(veiculo);

        }

        System.out.println("\n\n--- Carros de passeio com 5 passageiros e ano menor que 2010 ---\n");

        for (VeiculoTerrestre veiculo : exemplo) {

            if (veiculo.getAno() < 2010 && veiculo instanceof CarroDePasseio) {

                CarroDePasseio carro = (CarroDePasseio) veiculo;

                if (carro.getTotalDePassageiros() >= 5) {

                    System.out.println(carro);

                }

            }

        }

        System.out.print("\n\nTotal de Veículos do tipo caminhão que são verdes e têm 4 eixos: ");

        int total = 0;

        for (VeiculoTerrestre veiculo : exemplo) {

            if (veiculo instanceof Caminhao && veiculo.getCor().equals("Verde")) {

                Caminhao c = (Caminhao) veiculo; // precisa fazer um casting caso queria procurar um atributo da

                                                 // minha subclasse

                if (c.getTotalDeEixos() == 4) {

                    ++total;

                }

            }

        }

        System.out.println(total);

    }

}

**Execução do método *main()* em *Main.java***

Texto

Descrição gerada automaticamente