Curso de Análise e Desenvolvimento de Sistemas

Disciplina de Lógica de Programação Algoritmos II

Profa. Edhelmira Lima M.

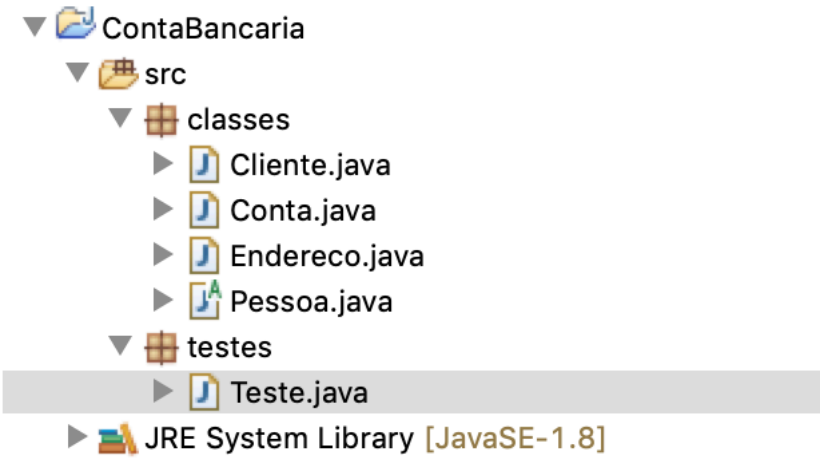
A1F - Semestre 2020-II

Nome:_____

Observações:

- O IDE recomendado para a realização do trabalho é o eclipse (disponível para download no site http://www.eclipse.org/downloads/).
- Deve-se criar um projeto Java para cada questão, um projeto Java no eclipse está composto por um diretório (do mesmo nome do projeto) que por sua vez contém o subdiretório **src** que armazena o código fonte das classes (**com extensão .java**) e o subdiretório **bin** (que possui as classes compiladas com **extensão .class**). O aluno deverá enviar o diretório correspondente ao projeto Java (compactado com extensão .zip).
- O código-fonte deve estar devidamente comentado.
- O trabalho é individual, caso seja constatado que as respostas de alunos distintos são cópias umas das outras, independentemente de qualquer motivo, a todas será atribuída a nota ZERO.
- Data de entrega: até **27/09/2020**

Questão 1 (Total: 3,0 pontos)
Crie um projeto Java chamado de **ContaBancaria** com dois pacotes (classes e testes) como mostrado na Figura a seguir.



No pacote **classes** crie as classes Endereco, Pessoa, Cliente e Conta conforme o código mostrado em anexo. Crie a classe Teste no pacote testes, compile e execute o projeto e analise seu comportamento.
Considerando o projeto implemente o seguinte:

a) **(1 pontos)** Modifique a classe **Conta** de modo que vire uma classe abstrata e crie uma classe chamada **ContaCorrente** derivada de Conta.

b) **(1 pontos)** Crie a classe **ContaPoupanca** derivada da classe conta. Esta classe possui uma constante que representa um índice de reajuste mensal. Adicione os **construtores** e **métodos** getter e setter necessários. Depois acrescente um método chamado reajustar que permita atualizar o saldo da conta somando o percentual determinado pelo índice de reajuste.

c) **(1 pontos)** Modifique todas as classes que possuam métodos críticos de modo a realizar um **tratamento de erros** adequado. Por exemplo, na classe Conta ao realizar um saque que exceda o saldo disponível.
Finalmente Modifique a classe Teste de modo a provar todas as funcionalidades implementadas.

ANEXO

==========➔Classe Pessoa

```java
package classes;

public abstract class Pessoa {
    protected String cpf;
    protected String nome;
    protected Endereco endereco;

    public Pessoa(String cpf, String nome, Endereco end) {
        this.cpf = cpf;
        this.nome = nome;
        this.endereco = end;
    }

    public String getCpf() {
        return cpf;
    }
    public void setCpf(String cpf) {
        this.cpf = cpf;
    }
    public String getNome() {
        return nome;
    }
    public void setNome(String nome) {
        this.nome = nome;
    }
    public Endereco getEndereco() {
        return endereco;
    }
    public void setEndereco(Endereco end) {
        this.endereco = end;
    }

    @Override
    public String toString() {
        return "[cpf:" + cpf + ", nome:" + nome + ", endereço:" + endereco + "]";
    }
}
```

==========➔Classe Cliente

```java
package classes;

public class Cliente extends Pessoa{
    private String tipoEmprego;
    private double renda;


    public Cliente(String cpf, String nome, Endereco end, String tipoEmprego, double renda) {
        super(cpf, nome, end);
        this.tipoEmprego = tipoEmprego;
        this.renda = renda;
    }

    @Override
    public String toString() {
        return "Cliente:"+super.toString()+" [tipoEmprego=" + tipoEmprego + ", renda=" + renda + "]";
    }
}
```

==========➔Classe Endereço

```java
package classes;

public class Endereco {
    private String logradouro;
    private String nome;
    private int numero;
    private String complemento;

    public Endereco(String logradouro, String nome, int numero, String complemento) {
        this.logradouro = logradouro;
        this.nome = nome;
        this.numero = numero;
        this.complemento = complemento;
    }

    @Override
    public String toString() {
        return "[" + logradouro + " " + nome + " #" + numero + ", complemento:"
                + complemento + "]";
    }
}
```

==========➔Classe Conta

```java
package classes;

public class Conta {
    protected Cliente titular;
    protected double saldo;

    public Conta(Cliente titular) {
        this.titular = titular;
        this.saldo = 0;
    }

    public Cliente getTitular() {
        return titular;
    }

    public void setTitular(Cliente titular) {
        this.titular = titular;
    }

    public double getSaldo() {
        return saldo;
    }

    // Outros métodos
    public void depositar(double valor){
        this.saldo += valor;
    }

    public void sacar(double valor) {
        if(valor <= this.saldo)
            this.saldo -= valor;
        else
            System.out.println("saldo insuficiente");
    }

    @Override
    public String toString() {
        return "Conta:\n\t [titular=" + titular + ", \n\t saldo=" + saldo + "]";
    }
}
```

==========➔Classe Teste

```java
package testes;

import classes.*;

public class Teste {

    public static void main(String[] args) {
        Cliente c1 = new Cliente("123", "João", new Endereco("av.","visconde",666,"apt 1"), "empresario",5000);
        System.out.println(c1);

        System.out.println("=======CONTA NORMAL =====");
        Conta conta1 = new Conta(c1);
        conta1.depositar(1000);
        System.out.println(conta1);
        conta1.sacar(200);
        System.out.println(conta1);
    }
}
```