

Agradecemos o preenchimento de [CC-02AN-DOO-Atividade-08 - Revisão Herança - 2024-2](#)

Veja as respostas enviadas.

CC-02AN-DOO-Atividade-08 - Revisão Herança - 2024-2

Utilizando os conceitos vistos em sala de aula referente a Herança, implemente os códigos abaixo:

Seu e-mail (pedro.costa4@uscsonline.com.br) foi registrado quando você enviou este formulário.

1. Construir uma classe para representar uma **Pessoa** com os seguintes dados: nome, endereço, **CPF** e **RG**. Considerando **encapsulamento** de dados, defina os construtores (padrão **com** passagem de parâmetros e **sem** passagem de parâmetros). Defina uma função chamada **ImprimePessoa()** que imprima os dados da pessoa.

Observações:

a)
Considerar que todas as classes estão em um único package chamado **uscs**.

b)
Considerar
os conceitos de herança.

c)
Considerar
os conceitos de Atributos e Métodos de Classe e Instância

```

package uscs;
public class Pessoa {
    private String nome;
    private String endereco;
    private String cpf;
    private String rg;

    public Pessoa(){
    }

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

    public void ImprimePessoa(){
        System.out.println("Dados da Pessoa:");
        System.out.println("Nome: " + nome);
        System.out.println("Endereco: " + endereco);
        System.out.println("CPF: " + cpf);
        System.out.println("RG: " + rg);
    }

    public String getNome(){
        return nome;
    }

    public void setNome(String nome){
        this.nome = nome;
    }

    public String getEndereco(){
        return endereco;
    }

    public void setEndereco(String endereco){
        this.endereco = endereco;
    }

    public String getCpf(){
        return cpf;
    }

    public void setCpf(String cpf){
        this.cpf = cpf;
    }

    public String getRg(){
        return rg;
    }

    public void setRg(String rg){
        this.rg = rg;
    }
}

```

2. Definir uma classe para representar um **Aluno** que obviamente é uma **pessoa**. Considere que o aluno tenha código de matrícula, nota **P1**, nota **P2** e uma certa quantidade de faltas. Considerando **encapsulamento** de dados, defina os **construtores** (padrão com passagem de parâmetros e sem passagem de parâmetros). Considere também que o aluno com mais de 10 faltas está automaticamente **reprovado** (definir o limite de **faltas = 10**). A média de

aprovação do aluno é a média aritmética das provas **P1** e **P2**. Um aluno com média maior ou igual a 7.0 está **aprovado**

(definir

media_de_aprovacao = 7.0). Média abaixo de **3.0** está **reprovado** (definir

media_de_reprovacao = 3.0) e média entre 3.0 e 7.0 está de **reavaliação**

(**exame**). Defina uma função chamada **ImprimeAluno()** para imprimir todos os dados do aluno

(inclusive os da classe **Pessoa**), uma

função **ImprimeFaltas()** para

imprimir o total de faltas e outra função **Resultado()**

para imprimir se aluno está **aprovado**,

reprovado ou em reavaliação (**exame**).

Observações:

a) Considerar que todas as classes estão em um único package chamado **uscs**.

b) **Considerar os conceitos de herança.**

c) **Considerar os conceitos de Atributos e Métodos de Classe e Instância**

*

```
package uscs;
public class Aluno extends Pessoa{
    private int matricula;
    private double p1;
    private double p2;
    private double qtdeFaltas;

    public static int limiteFaltas = 10;
    public static double mediaAprovacao = 7.0;
    public static double mediaReprovacao = 3.0;

    public Aluno(){
    }

    public Aluno(String nome, String endereco, String cpf, String rg, int matricula, double p1,
double p2, int qtdeFaltas){
        super(nome, endereco, cpf, rg);
        this.matricula = matricula;
        this.p1= p1;
        this.p2 = p2;
        this.qtdeFaltas = qtdeFaltas;
    }

    public void ImprimeAluno(){
        System.out.println("Dados do Aluno:");
        System.out.println("Nome: " + getNome());
        System.out.println("Endereco: " + getEndereco());
        System.out.println("CPF: " + getCpf());
        System.out.println("RG: " + getRg());
        System.out.println("Matricula: " + matricula);
        System.out.println("Nota P1: " + p1);
        System.out.println("Nota P2: " + p2);
        System.out.println("Quantidade de Faltas: " + qtdeFaltas);
    }

    public void ImprimeFaltas(){
        System.out.println("Quantidade de Faltas: " + qtdeFaltas);
    }

    public void Resultado(){
        if (qtdeFaltas > limiteFaltas){
            System.out.println("Aluno REPROVADO!");
        }
        else{
            if( (p1+p2)/2 >= mediaAprovacao ){
                System.out.println("Aluno APROVADO!");
            }
        }
    }
}
```

aprovado ou em exame).

Observações:

- c) Considerar os conceitos de Atributos e Métodos de Classe e Instância

*