Curso: Sistemas de Informação

Disciplina: Linguagem de Programação Orientada a Objetos

Lista para APS

ESTA LISTA DEVE SER ENTREGUE ATÉ O DIA DA AP2, CONTENDO TODAS AS CLASSES DENSENVOLVIDAS COM C#.

Enviar OS CÓDIGOS DAS CLASSES em um arquivo texto (doc ou txt) para o e-mail: joao.neto@professor.unifametro.edu.br

O projeto da aplicação Desktop em C# (Windows Forms) deve contemplar as funcionalidades das questões 3 a 10 e o projeto deve ser compartilhado com o professor.

Este trabalho poderá ser selecionado para apresentação/arguição!

A EQUIPE PODE TER ATÉ 4 ALUNOS.

Nome do Aluno 1: Jéssica Rodrigues da Costa

Matrícula do Aluno 1: 1-2018111637

Nome do Aluno 2: João Pedro Souza Xavier

Matrícula do Aluno 2: 1-2018111607

Nome do Aluno 3: Macelino Rodrigues do Nascimento

Matrícula do Aluno 3: 1-2018111495

Nome do Aluno 4: Vitor Barese

Matrícula do Aluno 4: 1-2018110644

LISTA DE EXERCÍCIOS

Para todos os exercícios de programação abaixo deve, além de atender o enunciado do próprio exercício, utilizar a linguagem de programação C#.

1) Escreva um código em C# que apresente a classe Aluno, com atributos nomeCompleto, matrícula e cpf, além de um método imprimir para cada atributo. O método imprimir deve mostrar na tela o valor do atributo. **(0,5 ponto)**

```
=======================1 questão============================

class Aluno {

  private string nomeCompleto;
  private int matricula;
  private string cpf;

  public void imprimirNomeCompleto(){
    Console.WriteLine ("Nome Completo: "+nomeCompleto);
  }

  public void imprimirMatricula(){
    Console.WriteLine ("Matricula: "+matricula);
  }

  public void imprimirCpf(){
    Console.WriteLine ("CPF: "+cpf);
  }

}
```

2) Baseando-se no exercício anterior adicione dois construtores. O primeiro construtor deve permitir a definição dos atributos nomeCompleto e matricula e o segundo construtor deve permitir a definição dos atributos nomeCompleto e cpf no momento da instanciação do objeto. **(0,5 ponto)**

```
=======================2 questão============================


class Aluno {

  private string nomeCompleto;
  private int matricula;
  private string cpf;

  //construtores
  public Aluno (string nomeCompleto ,int matricula ){
    this.nomeCompleto = nomeCompleto;
    this.matricula = matricula;
  }

    public Aluno (string nomeCompleto ,string cpf ){
    this.nomeCompleto = nomeCompleto;
    this.cpf = cpf;
  }

  //metodos
  public void imprimirNomeCompleto(){
    Console.WriteLine ("Nome Completo: "+nomeCompleto);
  }

  public void imprimirMatricula(){
    Console.WriteLine ("Matricula: "+matricula);
  }

  public void imprimirCpf(){
```

```
    Console.WriteLine ("CPF: "+cpf);
   }

 }
```

3) Escreva um código em C# que apresente a classe Moto, com atributos marca, modelo, cor e marcha e, o método imprimir. O método imprimir deve mostrar na tela os valores de todos os atributos. O atributo marcha indica em que a marcha a Moto se encontra no momento, sendo representado de forma inteira, onde 0 - neutro, 1 – primeira, 2 – segunda, etc. **(0,5 ponto)**

```
 ======================3 questão============================


 class Moto {

   private string marca;
   private string modelo;
   private string cor;
   private int macha;


   public void imprimir(){
     Console.WriteLine("A marca e "+marca);
     Console.WriteLine("A modelo e "+modelo);
     Console.WriteLine("A cor e "+cor);
     Console.WriteLine("A macha e "+macha);
   }

 }
```

4) Baseando-se no exercício anterior adicione um método construtor que permita a definição de todos os atributos no momento da instanciação do objeto. **(0,5 ponto)**

```
=======================4 questão============================


class Moto {

  private string marca;
  private string modelo;
  private string cor;
  private int macha;

//construtor
  public Moto(string marca, string modelo, string cor, int
macha){

    this.marca = marca;
    this.modelo = modelo;
    this.cor = cor;
    this.macha = macha;
  }
  //metodos
  public void imprimir(){
    Console.WriteLine("A marca e "+marca);
    Console.WriteLine("A modelo e "+modelo);
    Console.WriteLine("A cor e "+cor);
```

```
    Console.WriteLine("A macha e "+macha);
  }

}
```

5) Baseando-se no exercício anterior adicione os métodos marchaAcima e marchaAbaixo que deverão efetuar a troca de marchas, onde o método marchaAcima deverá subir uma marcha, ou seja, se a moto estiver em primeira marcha, deverá ser trocada para segunda marcha e assim por diante. O método marchaAbaixo deverá realizar o oposto, ou seja, descer a marcha. O método imprimir deve ser modificado de forma a mostrar na tela os valores de todos os atributos. **(0,5 ponto)**

```
======================5 questão============================

class Moto {

  private string marca;
  private string modelo;
  private string cor;
  private int macha;

//construtor
  public Moto(string marca, string modelo, string cor, int
macha){

    this.marca = marca;
    this.modelo = modelo;
    this.cor = cor;
    this.macha = macha;
  }
  //metodos
  public void imprimir(){
    Console.WriteLine("A marca e "+marca);
    Console.WriteLine("A modelo e "+modelo);
    Console.WriteLine("A cor e "+cor);
    Console.WriteLine("A macha e "+macha);
  }

  public void subirMacha(){
      macha++;
  }
  public void descerMacha(){
      macha--;
  }

}
```

6) Baseando-se no exercício anterior adicione os atributos menorMarcha e maiorMarcha, onde o atributo menorMarcha indica qual será a menor marcha possível para a moto e o atributo maiorMarcha indica qual será a maior marcha possível. Desta forma os métodos marchaAcima e marchaAbaixo devem ser reescritos de forma a não permitirem a troca de marchas para valores abaixo da menorMarcha e acima da maiorMarcha. O método imprimir deve ser modificado de forma a mostrar na tela os valores de todos os atributos. **(0,5 ponto)**

```
======================6 questão============================
```

```
class Moto {

  private string marca;
  private string modelo;
  private string cor;
  private int macha;
  private int menorMarcha=0;
  private int maiorMarcha=5;

//construtor
  public Moto(string marca, string modelo, string cor, int
macha){

    this.marca = marca;
    this.modelo = modelo;
    this.cor = cor;
    this.macha = macha;
  }
  //metodos
  public void imprimir(){
    Console.WriteLine("A marca e "+marca);
    Console.WriteLine("A modelo e "+modelo);
    Console.WriteLine("A cor e "+cor);
    Console.WriteLine("A macha e "+macha);
    Console.WriteLine("A menor Marcha e "+menorMarcha);
    Console.WriteLine("A maior Marcha e "+maiorMarcha);
  }
  public void subirMacha(){
    if (macha<maiorMarcha){
       macha++;
    }
  }
  public void descerMacha(){
    if(macha>menorMarcha){
      macha--;
    }
  }

}
```

7) Baseando-se no exercício anterior adicione um método construtor que permita a definição de todos os atributos no momento da instanciação do objeto. **(0,5 ponto)**

```
======================7 questão============================


class Moto {

  private string marca;
  private string modelo;
  private string cor;
  private int macha;
  private int menorMarcha;
  private int maiorMarcha;

//construtor
  public Moto(string marca, string modelo, string cor, int
macha,int menorMarcha, int maiorMarcha){

    this.marca = marca;
    this.modelo = modelo;
    this.cor = cor;
```

```
    this.macha = macha;
    this.menorMarcha = menorMarcha;
    this.maiorMarcha = maiorMarcha;
  }
  //metodos
  public void imprimir(){
    Console.WriteLine("A marca e "+marca);
    Console.WriteLine("A modelo e "+modelo);
    Console.WriteLine("A cor e "+cor);
    Console.WriteLine("A macha e "+macha);
    Console.WriteLine("A menor Marcha e "+menorMarcha);
    Console.WriteLine("A maior Marcha e "+maiorMarcha);
  }
  public void subirMacha(){
    if (macha<maiorMarcha){
       macha++;
    }
  }
  public void descerMacha(){
    if(macha>menorMarcha){
      macha--;
    }
  }

}
```

8) Baseando-se no exercício anterior adicione o atributo ligada que terá a função de indicar se a moto está ligada ou não. Este atributo deverá ser do tipo booleano. O método imprimir deve ser modificado de forma a mostrar na tela os valores de todos os atributos. **(0,5 ponto)**

```
======================8 questão============================

class Moto {

  private string marca;
  private string modelo;
  private string cor;
  private int macha;
  private int menorMarcha;
  private int maiorMarcha;
  private bool ligada=true;

//construtor
  public Moto(string marca, string modelo, string cor, int
macha,int menorMarcha, int maiorMarcha){

    this.marca = marca;
    this.modelo = modelo;
    this.cor = cor;
    this.macha = macha;
    this.menorMarcha = menorMarcha;
    this.maiorMarcha = maiorMarcha;
  }
  //metodos
  public void imprimir(){
    Console.WriteLine("A marca e "+marca);
    Console.WriteLine("A modelo e "+modelo);
    Console.WriteLine("A cor e "+cor);
    Console.WriteLine("A macha e "+macha);
    Console.WriteLine("A menor Marcha e "+menorMarcha);
```

```
    Console.WriteLine("A maior Marcha e "+maiorMarcha);
    Console.WriteLine("Esta ligada "+ligada);
  }
  public void subirMacha(){
    if (macha<maiorMarcha){
       macha++;
    }
  }
  public void descerMacha(){
    if(macha>menorMarcha){
      macha--;
    }
  }

}
```

9) Baseando-se no exercício anterior adicione um método construtor que permita a definição de todos os atributos no momento da instanciação do objeto. **(0,5 ponto)**

```
=======================9 questão============================


class Moto {

  private string marca;
  private string modelo;
  private string cor;
  private int macha;
  private int menorMarcha;
  private int maiorMarcha;
  private bool ligada;

//construtor
  public Moto(string marca, string modelo, string cor, int
macha,int menorMarcha, int maiorMarcha, bool ligada){

    this.marca = marca;
    this.modelo = modelo;
    this.cor = cor;
    this.macha = macha;
    this.menorMarcha = menorMarcha;
    this.maiorMarcha = maiorMarcha;
    this.ligada = ligada;
  }
  //metodos
  public void imprimir(){
    Console.WriteLine("A marca e "+marca);
    Console.WriteLine("A modelo e "+modelo);
    Console.WriteLine("A cor e "+cor);
    Console.WriteLine("A macha e "+macha);
    Console.WriteLine("A menor Marcha e "+menorMarcha);
    Console.WriteLine("A maior Marcha e "+maiorMarcha);
    Console.WriteLine("Esta ligada "+ligada);
  }
  public void subirMacha(){
    if (macha<maiorMarcha){
       macha++;
    }
  }
  public void descerMacha(){
    if(macha>menorMarcha){
```

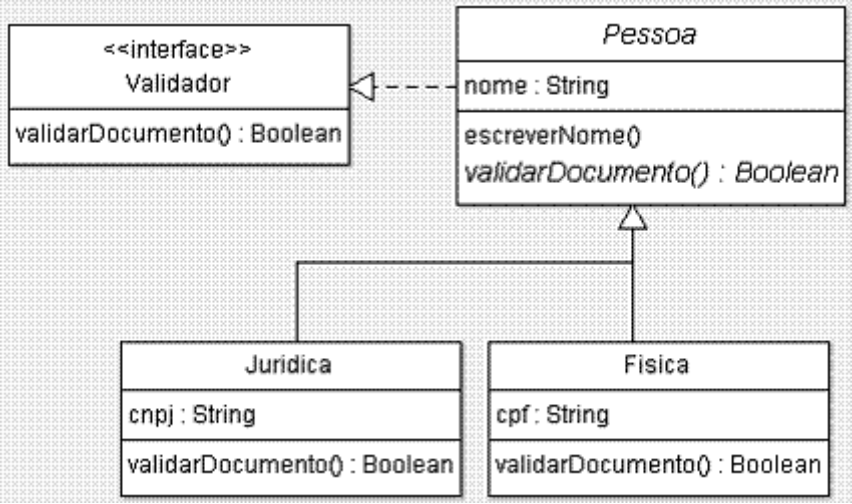
```
        macha--;
      }
    }

 }
```

10) Baseando-se no exercício anterior adicione os métodos ligar e desligar que deverão mudar o conteúdo do atributo ligada conforme o caso. **(0,5 ponto)**

```
======================10 questão============================

class Moto {

  private string marca;
  private string modelo;
  private string cor;
  private int macha;
  private int menorMarcha;
  private int maiorMarcha;
  private bool ligada;

//construtor
  public Moto(string marca, string modelo, string cor, int
macha,int menorMarcha, int maiorMarcha, bool ligada){

    this.marca = marca;
    this.modelo = modelo;
    this.cor = cor;
    this.macha = macha;
    this.menorMarcha = menorMarcha;
    this.maiorMarcha = maiorMarcha;
    this.ligada = ligada;
  }
  //metodos
  public void imprimir(){
    Console.WriteLine("A marca e "+marca);
    Console.WriteLine("A modelo e "+modelo);
    Console.WriteLine("A cor e "+cor);
    Console.WriteLine("A macha e "+macha);
    Console.WriteLine("A menor Marcha e "+menorMarcha);
    Console.WriteLine("A maior Marcha e "+maiorMarcha);
    Console.WriteLine("Esta ligada "+ligada);
  }
  public void subirMacha(){
    if (macha<maiorMarcha){
       macha++;
    }
  }
  public void descerMacha(){
    if(macha>menorMarcha){
      macha--;
    }
  }
  public void ligar(){
        ligada=true;
  }
  public void desligar(){
        ligada=false;
  }

}
```

11) Implemente, utilizando a linguagem C#, a interface e as classes conforme apresentadas no diagrama abaixo permitindo o polimorfismo: **(1,0 ponto)**

* Utilize modificador de acesso público para todos os atributos e métodos.




```
======================11 questão============================
====================Interface Validador=====================

interface IValidador{
  bool validarDocumento();
}

========================Class Pessoa==========================

public class Pessoa:IValidador{
  protected string nome;

  public void escreverNome(string nome){
    this.nome=nome;
  }

  public virtual bool validarDocumento(){
    throw new NotImplementedException();
  }

  public virtual void imprimir(){
    Console.WriteLine ("Nome: "+nome);
  }
}

========================Class Fisica==========================

class Fisica:Pessoa{
  private string cpf;

  public void escreverCpf(string cpf){
    this.cpf=cpf;
  }

  public override bool validarDocumento(){
```

```
   if (this.cpf !=""){
      return true;
   }else{
      return false;
   }
  }

  public override void imprimir(){
   Console.WriteLine ("Nome: "+nome);
   Console.WriteLine ("cpf: "+cpf);
   Console.WriteLine ("Documento Validado: "+validarDocumento());
  }

}

========================Class Juridica==============-===========

class Juridica:Pessoa{
private string cnpj;

  public void escreverCnpj(string cnpj){
   this.cnpj=cnpj;
  }

  public override bool validarDocumento(){
   if (this.cnpj !=""){
      return true;
   }else{
      return false;
   }
  }

  public override void imprimir(){
   Console.WriteLine ("Nome: "+nome);
   Console.WriteLine ("cnpj: "+cnpj);
   Console.WriteLine ("Documento Validado: "+validarDocumento());
  }

}
```

QUESTÃO PROJETO

12) Baseando-se nas questões de 3 a 10, criar uma aplicação Desktop com C# (Windows Forms) contendo campos para receber todos os dados da Moto, botões para mostrar os dados da moto em um exato momento, botões para aumentar e diminuir a marcha, botões para ligar e desligar a moto **(atenção para as observações abaixo)**. **(4,0 pontos)**

a. Observações:
   i. As características da interface e usabilidade ficam a critério dos desenvolvedores;
   ii. As funcionalidades a serem acionadas pelos botões devem utilizar as classes, com seus atributos e métodos, desenvolvidas dentre as questões 3 a 10.
   iii. Links para referência e ajuda:
      1. https://www.caelum.com.br/apostila-csharp-orientacao-objetos/introducao-ao-visual-studio-com-windows-form/
      2. https://msdn.microsoft.com/pt-br/library/jj153219.aspx