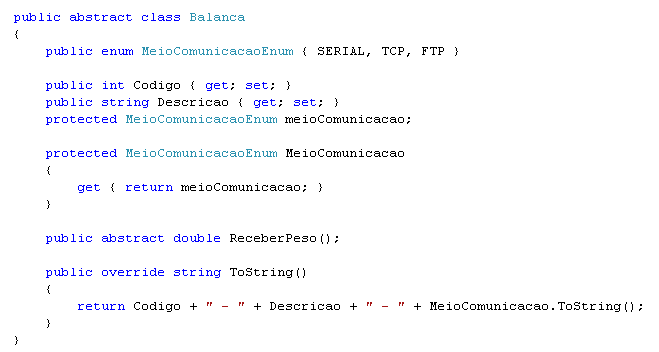
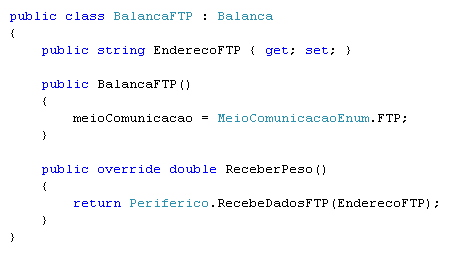
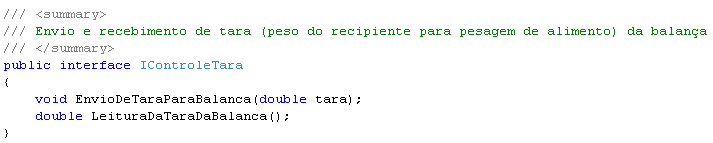
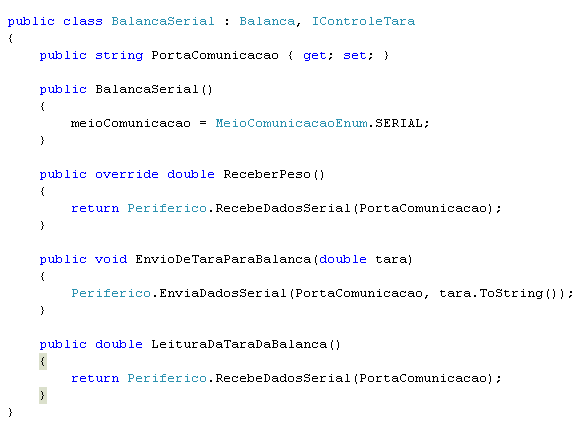
|  |  |  |
| --- | --- | --- |
| logo_CEFSA_H(grande) | **Código: <P319 >**  **Disciplina: < Programação Orientada a Objeto>** N1 | 2 bimestre| Curso: EC| Turma: 3 - 14/11/18 – 21h05 Prof.(ª): Eduardo R. Marcelino | Coord.: |Michele Bazana | **ftt** |
| Alunos(a):­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­ Nº: RA:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | Nota: |
| Orientações: Respostas a lápis não são passíveis de revisão. A interpretação faz parte da prova. SEM CONSULTA. Indique quando não resolver no local predeterminado. IDENTE SEU CÓDIGO. | |

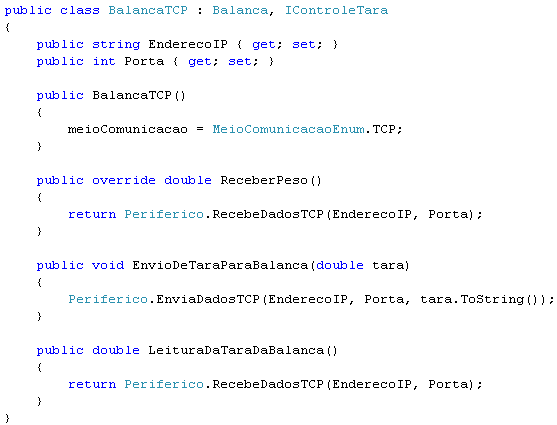
**1-) (Até 7,5 pontos)** Dada as classes e interfaces representadas no código em C# abaixo, responda as questões a seguir: (Obs: a unidade de peso da balança é medida em gramas (g). EX: 10kg = 10.000g)











Obs: Todas as saídas em vídeo nas questões abaixo podem ser realizadas utilizando-se o Console.Write e Console.WriteLine. A classe Periferico faz a interfaceamento com os hardwares, servindo apenas de apoio ao propósito destas questões. A lista utilizada pode ser da classe List do C#.

**a-)** **(1,0)** Sobrescreva o método ToString() da classe BalancaSerial para que ele também retorne os atributos públicos específicos desta classe.

public override string ToString()

{

return base.ToString() + “ Porta: “ + PortaComunicacao;

}

**b-) (1,5)** Qual o código em C# para criar uma lista que seja capaz de referenciar as balanças abaixo? Instancie em C# cada uma das balanças abaixo e armazene-as na referida lista.

Balança TCP:

Código: 1

Descrição: Balança do açougue

IP: 192.168.0.10

Porta: 9000

Balança Serial:

Código: 2

Descrição: Balança da Frutaria

Porta : COM1

Balança FTP:

Código: 3

Descrição: Balança da Padaria

IP: 192.168.0.12

List<Balanca> lista = new List<Balanca>();

BalancaTCP b = new BalancaTCP();

b.Codigo = 1;

b.Descricao = "Balança do açougue";

b.EnderecoIP = "192.168.0.10";

b.Porta = 9000;

lista.Add(b);

BalancaSerial bs = new BalancaSerial();

bs.Codigo = 2;

bs.Descricao = "Balança da Frutaria";

bs.PortaComunicacao = "COM1";

lista.Add(bs);

BalancaFTP bf = new BalancaFTP();

bf.Codigo = 3;

bf.Descricao = "Balanca da Padaria";

bf.EnderecoFTP = "192.168.0.12";

lista.Add(bf);

**c-)** **(1,0)** Dada uma lista que seja capaz de referenciar todos os tipos de balança, como a definida na questão anterior, faça um algoritmo para exibir todos os atributos públicos das balanças desta lista.

Foreach(Balanca b in Lista)

If (b is BalancaTCP)

(b as BalancaTCP).ToString();

Console.WriteLine (b.ToString() );

for (int i = 0; i < lista.Count(); i++)

{

Console.WriteLine(lista[i].ToString());

if(lista[i] is BalancaTCP)

{

Console.WriteLine(lista[i].IEnderecoIP +

(lista[i] as BalancaTCP).Porta);

}

else

{

Console.WriteLine((lista[i] as BalancaFTP).EnderecoFTP);

}

}

**d-)** **(1,0)** Escreva o construtor da classe Balanca para que o mesmo receba por parâmetro o código e a descrição da balança.

public Balanca(int codigo, string descricao)

{

Codigo = codigo;

Descricao = descricao;

}

**e-) (1,5)** Escreva o construtor da Classe BalancaSerial para ele receba os seus atributos específicos, além dos já recebidos pela classe base.

public BalancaSerial(int codigo, string descricao, string portaComunicacao) : base(codigo, descricao)

{

PortaComunicacao = portaComunicacao;

}

**f-)** **(1,5)** Dada uma lista que seja capaz de referenciar todos os tipos de balança, envie uma tara de 10 kg (10.000g) para todas as balanças dessa lista através do método EnvioDeTaraParaBalanca. Utilize para isso a interface IControleTara.

Foreach(Balanca b in Lista)

If (b is IControleTara)

(b as IControleTara).EnvioDeTaraParaBalanca(10000);

for (int i = 0; i < lista.Count(); i++)

{

if (lista[i] is BalancaSerial)

{

(lista[i] as BalancaSerial).EnvioDeTaraParaBalanca(10000);

}

else if(lista[i] is BalancaTCP)

{

(lista[i] as BalancaTCP).EnvioDeTaraParaBalanca(10000);

}

}

**2-)** Dada a hierarquia de classes abaixo, responda:

⚫ Atributo público

Classe Object

◆ Método público

♣ Protegido

Classe Funcionario

⚫Codigo

⚫ Nome

⚫Salario

◆CalcularSalario()

Classe Gerente

⚫ DeptoGerenciado

⚫ QtdeFuncionarios

♣ UsaCarroCedido

Classe Vendedor

⚫Comissao

⚫TotalVendasNoMes

◆CalcularSalario()

**a-) (até 0,5 ponto)** Pode-se afirmar que a classe Gerente também possui os atributos Codigo, Nome e Salario? Por quê?

**Sim, pois a classe gerente herda todos os atributos da classe Funcionário**

**b-)** **(até 0,5 ponto)** Variáveis de qual(ais) classe(s) poderiam fazer referência a objetos instanciados da classe Gerente?

R: **Funcionario e Object e GERENTE**

**c-)(até 0,5 ponto)** Dado que a assinatura do método não possui qualquer tipo de problema, verifique se o restante do código abaixo possui algum problema. Possui problema? R: **Sim.** Caso sim, corrija-o.

|  |  |
| --- | --- |
| void ImprimeVetor( Funcionario[] vetor)  {  for (int n=0; n< vetor.Length; n++)  {  Console.WriteLine( vetor[n].Comissao );  }  } | Espaço para correção:  void ImprimeVetor( Funcionario[] vetor)  {  for (int n=0; n< vetor.Length; n++)  {  If(vetor[n] is Vendedor)  {  Console.WriteLine( (vetor[n] as Vendedor).Comissao );  }  }  } |

**d-)(até 0,5 ponto)** Dado que a assinatura do método não possui qualquer tipo de problema, verifique se o restante do código abaixo possui algum problema. Possui problema? R: **Não**.\_ Caso sim, corrija-o.

|  |  |
| --- | --- |
| void ImprimeVetor( Funcionario[] vetor)  {  for (int n=0; n< vetor.Length; n++)  {  Console.WriteLine( vetor[n]. CalcularSalario() );  }  } | Espaço para correção: |

**e-)(até 0,5 ponto)** A classe Funcionario possui acesso a todos os atributos dos seus descendentes, incluindo os protegidos como por exemplo do atributo UsaCarroCedido da classe Gerente? S/N \_\_

**Não, a classe base não tem acesso as classes derivadas, apenas as derivadas é que herdam da classe pai.**