
JOÃO LUIZ LAGÔAS

2º ANO DO ENSINO MÉDIO –
LINGUAGEM DE
PROGRAMAÇÃO II

PROFESSOR: JOÃO LUIZ
LAGÔAS

COLÉGIO PEDRO II,
CAMPUS SÃO CRISTÓVÃO III

Curso Técnico de Ciência da
Computação

CONTEÚDO

Caros alunos,

O objetivo desta lista é exercitar os conceitos de Classes e Objetos. Seu propósito será construir classes de acordo com o que é pedido e depois testá-las através da instanciação de alguns objetos.

Foque no que é um atributo e método dentro de uma classe e exercite a criação de objetos na memória.

Esta lista é composta por exercícios de diferentes níveis. O cristal colorido que precede cada EXERCÍCIO indica o seu nível de dificuldade de acordo com sua cor. Observe a tabela abaixo:



BÁSICO



SIMPLES



MÉDIO



DIFÍCIL



INTENSO

Busque realizar todos os exercícios na sequencia!



EXERCÍCIO 01

Crie uma classe chamada Retângulo. Esta classe deverá conter dois atributos inteiros:

```
public int Altura
public int Largura
```

Essa classe deverá conter dois métodos de acordo com as assinaturas:

```
public int CalcularPerímetro()
public int CalcularArea()
```

O primeiro método deverá retornar um inteiro correspondente ao perímetro do objeto Retângulo.

O segundo método deverá retornar um inteiro correspondente à área do objeto Retângulo.

Após implementar a classe, teste seus métodos instanciando um objeto, inserindo valores nos seus atributos e chamando os métodos dentro da Main(). Observe o código abaixo para realizar o seu teste e garantir que a Classe que você criou funciona adequadamente:

```
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Entre com a altura e largura de um retangulo: ");

        Retangulo ret = new Retangulo();

        ret.Altura = Convert.ToInt32(Console.ReadLine());
        ret.Largura = Convert.ToInt32(Console.ReadLine());

        int perimetro = ret.CalcularPerímetro();
        int area = ret.CalcularÁrea();

        Console.WriteLine("O perímetro do retângulo vale {0}", perimetro);
        Console.WriteLine("A área do retângulo vale {0}", area);
    }
}
```



EXERCÍCIO 02

Crie uma classe chamada Círculo. Esta classe deverá conter um atributo inteiro:

```
public int Raio
```

Essa classe deverá conter dois métodos de acordo com as assinaturas:

```
public double CalcularPerímetro()
public double CalcularArea()
```

O primeiro método deverá retornar um double correspondente ao perímetro do objeto Círculo.

O segundo método deverá retornar um double correspondente à área do objeto Círculo.

Após implementar a classe, teste seus métodos instanciando um objeto, inserindo valores nos seus atributos e chamando os métodos dentro da Main()



EXERCÍCIO 03

Escreva uma classe C# chamada de Livro com as seguintes características:

- Atributos:
 - **Título** para o título do livro com tipo String;
 - **Autor** para o nome do autor com tipo String;
 - **Preço** para o preço do livro com o tipo double;
- Métodos:
 - `public string Descrever()`: este método retorna como parâmetro uma string contendo a seguinte informação: "O livro {nome_do_livro} foi escrito por {nome_do_autor} e custa {preço_do_livro} reais".

No método Main(), instancie um objeto da classe Livro com os seguintes parâmetros:

Nome: Harry Potter e o Cálice de Fogo

Autor: J. K. Rowling

Preço: 50.00;

Após instanciar esse objeto, chame o método Descrever() para obter a string de resposta. Depois disso, imprima essa string no Console ainda no método Main().

Instancie também outros objetos da classe Livro e note que você está reutilizando todo o código implementado na classe!



EXERCÍCIO 04

Escreva uma classe C# chamada de Data com as seguintes características:

- Atributos:
 - **Dia** representando um número inteiro.
 - **Mês** representando um número inteiro.
 - **Ano** representando um número inteiro.
- Métodos:
 - `public string Descrever()`: este método retorna como parâmetro uma string contendo a seguinte informação: "Este objeto data refere-se a data de {dia} do mês de {NOME DO MÊS} do ano de {ano}".
 - `public bool EhBissexto()`: este método retorna true caso o ano do objeto Data seja bissexto ou false caso contrário. Um ano é bissexto se ele for divisível por 4 e não por 100 ou se ele for divisível por 400.

No método Main(), instancie um objeto da classe Data e o construa com os parâmetros que julgar necessário. Após instanciar esse objeto, chame o método Descrever() para obter a string de resposta. Depois disso, imprima essa string no Console ainda no método Main().

Teste também o funcionamento do método EhBissexto(). Para isso, chame-o e verifique se ele retornará true ou false de acordo com o valor do atributo ano do objeto.

Instancie por fim outros objetos da classe Data e note que você está reutilizando todo o código implementado na classe!



EXERCÍCIO 05

Ainda no Exercício 05, implemente agora um método que irá comparar a data do objeto que o chama com a data de um outro objeto `Data` que é passado como parâmetro.

O método deverá retornar 1 caso a data do parâmetro seja maior que a data do objeto chamador ou 0 caso contrário. A assinatura do método deverá ser:

```
public int CompararDatas(Data data)
```

Nota: para facilitar a implementação, inicie ignorando o mês.

Observe que um objeto também pode ser passado como parâmetro! Afinal de contas, um objeto é uma variável como outra qualquer.

Após terminar de construir o método, instancie no método `Main()` dois objetos de `Data` e realize o teste chamando o método `CompararDatas`.



EXERCÍCIO 06

Uma função linear pode ser escrita como $y = ax + b$, onde a determina o seu coeficiente angular e b o seu coeficiente linear. Conhecendo a e b , podemos determinar qualquer ponto da função dado um valor de x .

Crie uma classe chamada `FuncaoLinear`. Esta classe deverá conter dois atributos `double` que correspondem ao coeficiente angular e ao coeficiente linear da função.

```
public double CoefAng  
public double CoefLin
```

Essa classe deverá conter dois métodos de acordo com as assinaturas:

```
public double CalcularY(double x)  
public double CalcularRaiz()
```

O primeiro método deverá retornar um `double` correspondente ao valor de y da função linear dado como parâmetro um valor qualquer x .

O segundo método deverá retornar um `double` correspondente ao valor da raiz da função que fora definida.

Após implementar a classe, teste seus métodos instanciando um objeto e chamando os métodos dentro da `Main()`.

Por exemplo:

Para um objeto `FuncaoLinear` com atributo $a = 1$ e $b = 2$, temos que a chamada do método `CalcularY(2)` deve retornar 4 e a chamada do método `CalcularRaiz()` deve retornar -2.

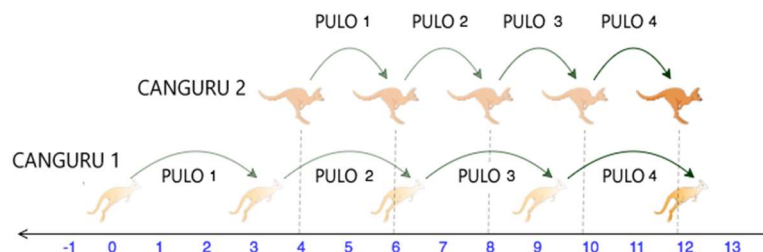


EXERCÍCIO 07

Um adestrador de animais é contratado para realizar uma coreografia de animais para um circo. Para a primeira apresentação, dois cangurus estarão postos numa mesma linha preparados para pular na direção positiva (no eixo x para frente).

- O primeiro canguru começa na posição x_1 e se move a uma velocidade constante de v_1 metros por pulo.
- O segundo canguru começa na posição x_2 e se move a uma velocidade constante de v_2 metros por pulo.
- É entendido também que as posições e as velocidades não podem assumir valores negativos.

Você tem que ajudar o adestrador a descobrir uma maneira onde os dois cangurus estejam numa mesma posição durante a apresentação. Observe a imagem que se segue:



Para isso, faça o que é pedido.

- Implemente uma classe `Canguru` que contenha dois atributos reais, `Posição` e `Velocidade`.
- Implemente o método `Pular` que é utilizado para atualizar a posição do Canguru sempre que é chamado, isto é, através de sua velocidade, determinar uma nova posição.
- Escreva uma classe `Programa` contendo o método `Main` que recebe como entrada as duas posições iniciais dos cangurus (através do console) assim como suas duas velocidades. Além disso, este método deve utilizar a classe `canguru` criada nos exercícios anteriores (através da instanciação de 2 objetos) e sinalizar para o adestrador se ele vai conseguir uma posição onde ambos estarão juntos ou não.

Entrada:

Posição Inicial do Canguru 1: 0

Posição Inicial do Canguru 2: 4

Velocidade do Canguru 1: 3

Velocidade do Canguru 2: 2

Saída:

Sim, os cangurus se encontrarão na posição 12.