

LISTA - LINGUAGEM DE PROGRAMAÇÃO 2 (4,0PTS) CRIANDO FUNÇÕES

Nesta lista você irá implementar uma série de funções. Para isso, é pedido que, para cada exercício, você crie uma classe com o nome solicitado e, em seguida, implemente as funções sugeridas.

Para criar uma classe, clique em Projeto->Adicionar Classe e dê o nome a sua classe. Por exemplo, no primeiro exercício, chame a sua classe de MatemáticaBásica. Feito isso, você verá que um arquivo denominado MatemáticaBásica.cs será incluído no Gerenciador de Soluções. Clique duas vezes neste arquivo e insira as funções que são pedidas dentro do escopo da classe.

EXERCÍCIO 01 - MATEMÁTICA

Crie uma classe denominada MatemáticaBásica e implemente as funções pedidas.

- a) Crie uma função cuja assinatura é:

```
public static double Potência(double a, double b)
```

Essa função recebe dois números reais a e b e retorna um valor real que armazena a potência de a por b, ou seja, a^b .

- b) Crie uma função cuja assinatura é:

```
public static bool EhPar(int x)
```

Essa função recebe um número inteiro x e retorna um valor booleano que indica se x é par (true) ou não (false).

- c) Crie uma função cuja assinatura é:

```
public static bool EhPrimo(int x)
```

Essa função recebe um número inteiro x e retorna um valor booleano que indica se x é primo (true) ou não (false).

- d) Crie uma função cuja assinatura é:

```
public static int QuantidadePrimos(int a, int b)
```

Essa função recebe dois números inteiros a e b que representam um intervalo de números fechado $[a, b]$. Sua função deve retornar a quantidade de números primos que existem entre a e b.

Exemplo: se a função recebe $a = 3$ e $b = 10$, então sabemos que há no intervalo $[3, 10]$ um total de 3 números primos. Sua função então deve retornar o inteiro 3.

Dica: você pode usar a função que você criou no exercício c dentro do escopo da função do exercício atual.

Agora que você terminou de implementar todas as funções, vamos testá-las na Main, ou seja, no fluxo primário de execução do seu código.

Copie e cole o código do arquivo Main.cs no escopo da sua função Main referente ao Exercício 01.

EXERCÍCIO 02 - MU

Crie uma classe denominada UM e implemente as funções pedidas.

- a) Crie uma função cuja assinatura é:

```
public static double VelocidadeMedia(double ds, double dt)
```

Essa função recebe dois números reais ds (variação no espaço) e dt (variação no tempo) de um elemento em movimento uniforme. Sua função deve retornar um valor real que indica a velocidade média desse movimento. Lembre-se que a velocidade média pode ser calculada por

$$V = \frac{\Delta s}{\Delta t}$$

b) Crie uma função cuja assinatura é:

```
public static
double VelocidadeMedia(double so, double sf,
double to, double tf)
```

Essa função recebe quatro parâmetros:

sf: posição final

so: posição inicial

tf: tempo final

to: tempo inicial

que indicam informações a respeito de um movimento uniforme percorrido por um elemento. Sua função deve retornar a velocidade média desse movimento. Lembre-se que:

$$V = \frac{s_f - s_0}{t_f - t_0}$$

c) Crie uma função cuja assinatura é:

```
public static
int TipoDeMovimento(double sf, double so)
```

Essa função retorna 1 caso o movimento seja progressivo, -1 caso seja regressivo ou 0 caso não haja movimento.

Lembre-se que: um movimento uniforme é dito progressivo quando $s_f > s_0$ e regressivo quando $s_f < s_0$.

d) Crie uma função cuja assinatura é:

```
public static
double PosicaoFinal(double so, double v,
double t)
```

Essa função recebe três parâmetros:

v: velocidade

s₀: posição inicial

t: tempo final

e retorna a posição final do elemento no movimento uniforme.

Lembre-se que a posição final s é dada por

$$s = s_0 + v \cdot t$$

Agora que você terminou de implementar todas as funções, vamos testá-las na Main, ou seja, no fluxo primário de execução do seu código.

Copie e cole o código do arquivo Main.cs no escopo da sua função Main referente ao Exercício 02.

EXERCÍCIO 03 - MUV

Crie uma classe denominada MUV e implemente as funções pedidas.

a) Crie uma função cuja assinatura é:

```
public static
double Aceleracao(double dv, double dt)
```

Essa função recebe dois números reais dv (variação da velocidade) e dt (variação no tempo) de um elemento em movimento uniformemente variado. Sua função deve retornar um valor real que indica a aceleração média desse movimento. Lembre-se que

$$A = \frac{\Delta v}{\Delta t}$$

b) Crie uma função cuja assinatura é:

```
public static
double Aceleracao(double vo, double vf,
double to, double tf)
```

Essa função recebe quatro parâmetros:

vf: velocidade final

vo: velocidade inicial

tf: tempo final

to: tempo inicial

que indicam informações a respeito de um movimento uniformemente variado percorrido por um elemento. Sua função deve retornar a aceleração média desse movimento. Lembre-se que:

$$A = \frac{v_f - v_0}{t_f - t_0}$$

c) Crie uma função cuja assinatura é:

```
public static int TipoDeMovimento(double a)
```

Essa função retorna 1 caso o movimento seja acelerado ($a > 0$), -1 caso seja retardado ($a < 0$) ou 0 caso o movimento não seja uniformemente variado.

d) Crie uma função cuja assinatura é:

```
public static
double VelocidadeFinal(double vo,
                        double a, double t)
```

Essa função recebe três parâmetros:

a : aceleração

v_0 : velocidade inicial

t : tempo final

e retorna a velocidade final do movimento uniformemente variado.

Lembre-se que a velocidade final v é dada por

$$v = v_0 + a \cdot t$$

e) Crie uma função cuja assinatura é:

```
public static
double VelocidadeFinal(double vo, double a,
                        double sf, double so)
```

Essa função recebe quatro parâmetros:

a : aceleração

v_0 : velocidade inicial

s_f : posição final

s_0 : posição inicial

e retorna a velocidade final do movimento uniformemente variado.

Lembre-se que a velocidade final v pode ser dada pela Equação de Torricelli

$$v^2 = v_0^2 + 2 \cdot a \cdot \Delta s$$

Agora que você terminou de implementar todas as funções, vamos testá-las na Main, ou seja, no fluxo primário de execução do seu código.

Crie um código na função Main que chame algumas de suas funções para testá-las.