
POO - Lista 03 - Construtores e ToString

Prof. Gilbert Azevedo da Silva

I. Implementar classes em C# para representar:

1. Um Retângulo

Escrever a classe Retangulo de acordo com o diagrama UML apresentado.

A classe deve ter como atributos: a base (b) e a altura (h) do retângulo. O construtor da classe recebe os valores iniciais da base e altura do retângulo, que devem ser positivos. Os métodos SetBase e SetAltura podem modificar os valores da base e da altura, respectivamente; os métodos GetBase e GetAltura retornam os valores armazenados nos atributos; CalcArea e CalcDiagonal calculam a área e a diagonal do retângulo. O método ToString deve retornar um texto com os atributos do objeto.

Retangulo
- b : double - h : double
+ Retangulo(b : double, h : double) + SetBase(b : double) : void + SetAltura(h : double) : void + GetBase() : double + GetAltura() : double + CalcArea() : double + CalcDiagonal() : double + ToString() : string

2. Um Frete

Escrever a classe Frete de acordo com o diagrama UML apresentado.

A classe deve utilizar os atributos distância e peso para calcular o frete cobrado por uma transportadora para transportar uma carga de uma origem até um destino. O construtor da classe recebe os valores iniciais da distância e do peso, que devem ser positivos. Os métodos de acesso podem alterar e recuperar esses valores. O peso (massa) da carga é dado em Kg e a distância percorrida até o destino em Km. O valor do frete calculado pelo método CalcFrete deve ser de um centavo de real para cada quilo transportado por quilometro. O método ToString deve retornar um texto com os atributos do objeto.

Frete
- distancia : double - peso : double
+ Frete(d : double, p : double) + SetDistancia(d : double) : void + SetPeso(p : double) : void + GetDistancia() : double + GetPeso() : double + CalcFrete() : double + ToString() : string

3. Um Conversor Decimal-Binário

Escrever a classe Conversor acordo com o diagrama UML apresentado.

A classe deve permitir a conversão de um valor decimal inteiro e positivo, armazenado no atributo num, para o formato binário. O construtor da classe recebe o valor inicial do número na base dez. Os métodos de acesso podem alterar e recuperar esse valor.

O método Binario deve realizar a conversão do número decimal e retornar a sua representação em binário. O método ToString deve retornar um texto com o número nas bases decimal e binária.

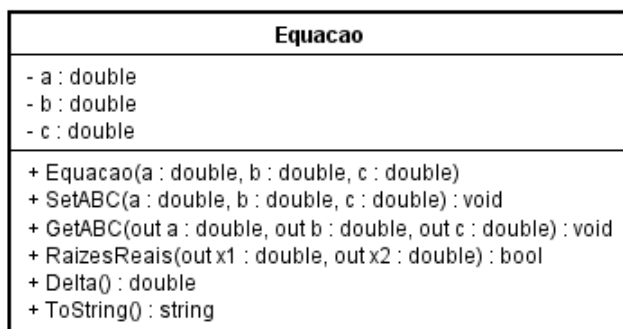
Conversor
- num : int
+ Conversor(num : int) + SetNum(num : int) : void + GetNum() : int + Binario() : string + ToString() : string

4. Uma Equação do II Grau

Escrever a classe Equacao de acordo com o diagrama UML apresentado abaixo.

A classe deve calcular o delta e as raízes de uma equação do segundo grau, com base nos coeficientes a, b e c de uma equação $f(x) = ax^2 + bx + c$. O construtor da classe recebe os valores iniciais dos atributos. Os métodos de acesso podem alterar e recuperar esses valores.

O método Delta retorna o valor do delta usado no cálculo das raízes. Os métodos RaizesReais retorna um booleano informando se a equação tem ou não raízes reais; caso afirmativo, retorna as raízes nos parâmetros de saída X1 e X2. O método ToString deve retornar um texto com os atributos do objeto.



5. Uma Data

Escrever a classe Data de acordo com o diagrama UML apresentado abaixo.

A classe deve ter como atributos o dia, mês e ano de uma data.

Um construtor da classe recebe os valores iniciais de dia, mês e ano que devem representar uma data válida. O outro construtor recebe uma data no formato “dd/mm/aaaa”;

Os métodos GetDia, GetMes, GetAno e SetData são usados para definir e recuperar os valores dos atributos, que devem também ser validados. O método ToString retorna a data no formato “dd/mm/aaaa”.

