

---

## Programação Estruturada - Lista 11

---

Prof. Gilbert Azevedo da Silva

### I. Implementar classes em C# para representar:

#### 1. Um Círculo (forma geométrica)

A classe deve ter o atributo raio para armazenar a dimensão da figura e métodos para definir seu raio, calcular sua área e sua circunferência. O raio deve ser validado para que nenhum valor negativo seja armazenado.

Escrever um programa para testar a classe.

Círculo
- raio : double
+ SetRaio(raio : double) : void + CalcArea() : double + CalcCircunferencia() : double

#### 2. Uma Disciplina

A classe deve ter atributos para armazenar o nome e as notas da disciplina (dois bimestres e prova final) e métodos para definir os atributos e calcular a média final. As notas devem ser validadas para que apenas valores entre zero e cem sejam armazenados.

Escrever um programa para testar a classe.

Disciplina
- nome : string - nota1 : int - nota2 : int - notaF : int
+ SetNome(n : string) : void + SetNota1(n : int) : void + SetNota2(n : int) : void + SetNotaF(n : int) : void + CalcMedia() : int

#### 3. Um Abastecimento de Combustível de um Veículo

A classe deve ser utilizada para calcular o consumo de combustível de um veículo, tendo atributos para armazenar os dados de um abastecimento: valor do litro de combustível, valor pago no abastecimento e número de quilômetros rodados.

Os métodos "set" da classe devem ser usados para informar os valores dos atributos devendo os dados informados serem validados para que nenhum valor inválido seja armazenado.

O método MediaKmLitro calcula a média de consumo do veículo, medido em Km por litro, obtendo quantos quilômetros foram percorridos com um litro de combustível e o método MediaReaisKm, calcula o custo médio em reais para cada quilômetro percorrido.

Escrever um programa para testar a classe.

Abastecimento
- valorLitro : double - valorPago : double - kmRodados : double
+ SetValorLitro(valorLitro : double) : void + SetValorPago(valorPago : double) : void + SetKmRodados(kmRodados : double) : void + MediaKmLitro() : double + MediaReaisKm() : double

#### 4. Um Número Racional (fração)

A classe deve ter atributos para armazenar o numerador e o denominador de uma fração (números inteiros) e métodos “set” para definir esses valores.

A classe deve ter também métodos para realizar as operações de soma, subtração, multiplicação e divisão entre dois números racionais.

Escrever um programa para testar a classe.

Racional
- num : int - den : int
+ SetNum(x : int) : void + SetDen(x : int) : void + Somar(x : Racional) : Racional + Subtrair(x : Racional) : Racional + Multiplicar(x : Racional) : Racional + Dividir(x : Racional) : Racional + ToString() : string

#### 5. Uma Conta Bancária

A classe deve ter atributos para armazenar o nome do titular da conta, o número da conta e seu saldo e métodos para: definir o titular da conta, definir o número da conta, depositar um determinado valor (aumentando o saldo), sacar um determinado valor (diminuindo o saldo) e verificar o saldo.

Os valores informados nas operações de depositar e sacar devem ser positivos e o saldo mínimo da conta deve ser zero.

Escrever um programa para testar a classe e desenhar sua representação em UML.

#### 6. Uma Entrada de Cinema

A classe deve ter atributos para armazenar o dia e o horário de uma sessão de cinema e métodos para: definir o dia e horário da sessão, calcular o valor da entrada inteira e calcular o valor da meia-entrada.

O valor das entradas deve ser calculado com base nas seguintes regras:

- Segunda, terça e quinta, o valor base do ingresso é R\$ 16,00.
- Nas quartas todos pagam meia-entrada no valor de R\$ 8,00, em qualquer horário.
- Sexta, sábado e domingo, o valor base do ingresso é R\$ 20,00.
- Das 17h à meia-noite, há acréscimo de 50% no valor base do ingresso.

Escrever um programa para testar a classe e desenhar sua representação em UML.