

Exercício 01

Implemente as seguintes classes:

| Funcionario |
|---|
| ~ nome : String ~ matricula : long ~ profissao : Profissao ~ salario : double |
| ~ Funcionario() ~ Funcionario(matricula : long) ~ Funcionario(matricula : long, nome : String) ~ Funcionario(matricula : long, nome : String, profissao : Profissao) ~ exhibirDadosFuncionario() : void |

| Profissao |
|-----------------|
| ~ nome : String |
| |

Exercício 02

Construa uma classe chamada Veiculo com:

A. Os atributos: Placa, Marca, Modelo e Ano

B. Os seguintes métodos:

a. Construtor sem argumentos

b. Construtor todos os argumentos para definir as propriedades

c. Regra de validação: $\text{Ano} \geq 1900$ e $\text{Ano} \leq 2017$

Exercício 03

Escreva uma classe chamada Data que contenha três atributos do tipo int chamados dia, mes e ano

- a) Crie um método construtor que inicialize os dados com zero e outro método construtor que inicialize os dados com os valores recebidos como parâmetros.
- b) Crie um método para retornar a data como string no formato: dd/mm/aaaa
- c) Crie a validação do dia. Ele deve estar entre 1 e 31. Caso contrário inicializar o atributo com o valor 1.
- d) Crie a validação do mês. Ele deve estar entre 1 e 12. Caso contrário inicializar o atributo com 1.
- e) Crie a validação do ano. Ele não deve ser negativo. Se o ano for inválido, defina o valor do atributo ano com o ano corrente.
- f) Crie um método que retorne o nome do mês de acordo com o número que está armazenado no atributo mes.

Exercício 04

Crie uma classe chamada **Empregado**

a) Crie as propriedades **Nome**, **Funcao** e **Salario**

b) Crie um construtor que receba como parâmetros o nome, função e salário do empregado e depois inicialize as respectivas propriedades.

c) Crie um método chamado **AumentarSalario**, que receberá como parâmetro o valor do aumento. Este método deverá atualizar o salário do empregado.

d) No arquivo Program.cs:

a. Crie um objeto da classe Empregado que tenha como nome “João”, salário inicial de R\$5.400,00 e função “Analista de Sistemas”. Utilize o construtor criado no **item b**.

b. Aumente o salário deste funcionário em R\$600,00, através da utilização do método criado no **item c**.

c. Exiba na tela valor dos propriedades **Nome**, **Funcao** e **Salario**.

Exercício 05

Crie uma classe chamada **TrianguloRetangulo**

Crie as propriedades **Cateto1** e **Cateto2**

b) Declare o construtor padrão para a classe **TrianguloRetangulo**.

c) Crie um construtor que receberá como parâmetros os catetos(cateto1 e cateto2) e atribuirá esses valores aos respectivas propriedades.

d) Crie um método chamado **CalcularHipotenusa**, sem parâmetros de entrada para realizar o cálculo da hipotenusa e retornar o seu tamanho (ver dica no próximo slide).

e) No arquivo Program.cs

- Crie 2 objetos: o primeiro utilizando o construtor padrão, com catetos de tamanho 5 e 10 e o segundo utilizando o construtor de classe criado no item c, com catetos de tamanho 4 e 5.
- Exiba na tela os valores das hipotenusas.

Exercício 05

Nota: Para efetuar o cálculo da hipotenusa, utilize a fórmula:

$$\text{hipotenusa}^2 = \text{cateto1}^2 + \text{cateto2}^2$$

Para realizar a potenciação, utilize o método Pow da classe **Math**.

Este método recebe dois parâmetros do tipo double e retorna o valor obtido da operação.

Por exemplo, para x^y : `double resultado = Math.Pow(x, y);`

Para realizar a raiz quadrada, utilize o método Sqrt da classe **Math**.

Este método retorna a raiz quadrada do número informado como parâmetro, o parâmetro de entrada e o retorno são do tipo double.

Por exemplo, para raiz quadrada de 16: `double resultado = Math.Sqrt(16.0);`

Exercício 06

Crie uma classe chamada Hora

a) Crie as propriedades **Hora**, **Minuto** e **Segundo**

b) Crie um construtor que receba 3 parâmetros: a hora, o minuto e o segundo e em seguida atribua esses valores as respectivas propriedades da classe.

c) Crie um método chamado **ObterHora**, para retornar a hora, no formato: hh:mm:ss

d) No arquivo Program.cs: Instanciar um objeto da classe Hora e definir seu estado de modo que ao ser invocado o método **ObterHora** seja exibido no console a hora 12:40:15

Exercício 07

Crie uma classe chamada Lampada

1. Crie as propriedades **Cor** do tipo String e **Ligada** do tipo boolean
2. Crie um construtor sem parâmetro.
3. Crie um construtor que receba 2 parâmetros: a cor e o estado da lâmpada (true ou false) e atribua as respectivas propriedades.
4. Crie um construtor que receba apenas um parâmetro: a cor da lâmpada. Neste caso, atribua a cor a respectiva propriedade e por padrão, atribua a propriedade **Ligada** o valor true.
5. No arquivo Program.cs, instancie 3 lâmpadas.
 - a. A primeira deverá ser instanciada através do construtor sem parâmetro
 - b. A segunda, através do construtor com um parâmetro
 - c. A terceira, através do construtor com dois parâmetros
6. Imprima as cores e o estados das lâmpadas instanciadas

Exercício 08

Crie uma classe chamada Ponto, que armazene as coordenadas (x,y) de um ponto num plano. A coordenada default(padrão) para qualquer ponto deve ser (0.0 , 0.0).

- a) Além do construtor sem parâmetro, esta classe deve conter um construtor que receba como parâmetros os valores de x e y.
- b) Crie um método para calcular a distância de 2 pontos (x1,y1) e (x2,y2). Este método receberá como parâmetro um objeto da classe Ponto e retornará a distância entre os pontos.
- c) Crie uma classe de Teste e calcule a distância de 2 pontos e exiba o resultado na tela.
- d) Use o construtor sem parâmetro para instanciar o primeiro ponto e o construtor com parâmetros para instanciar o segundo ponto.

Nota: Para calcular a distância de 2 pontos utilize a fórmula:

dx = x2 - x1; **dy** = y2 - y1; **distancia** = Math.Sqrt(dx*dx + dy*dy);

Exercício 09

Crie uma classe chamada Cor

1. Crie as propriedades do tipo inteiro **Red**, **Green** e **Blue**
2. Forneça os construtores apropriados para possibilitar as seguintes instâncias:
 - a. `Cor cor1 = new Cor(10, 100, 20); //red,green,blue`
 - b. `Cor cor2 = new Cor(125, 30); //red,green`
 - c. `Cor cor3 = new Cor(50); //red`
 - d. `Cor cor4 = new Cor();`
3. Crie um método na classe Cor chamada ObterCor que retornará uma string no formato:
"(<red>,<green>,<blue>)"
4. No arquivo Program.cs, realizar as instâncias definidas nos itens I, II, III e IV e exibir as cores.

Exercício 10

Responda:

- 1) Defina sobrecarga de métodos e em seguida implemente uma classe onde é expresso um exemplo.
- 2) O que é um construtor?
- 3) O que é um construtor padrão? Quais as suas características?