



---

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,

Nesta lista iremos intensificar nossa prática a respeito de Classes e Objetos e focar principalmente nos métodos Construtores e no conceito de Encapsulamento.



## EXERCÍCIO 01

Escreva uma classe `Data` que inclui três informações: um mês (int), um dia (int) e um ano (int). Sua classe deve ter um construtor para inicializar essas informações de modo válido. **Quais são os valores válidos que esses atributos poderiam assumir?**

Para isso, considere a implementação de métodos `Setters` e `Getters` que evitam a alteração inadequada desses atributos.

Escreva também um método `ExibirData` que não recebe parâmetros mas retorna uma string contendo a `Data` no formato "01/08/1990".

Na classe `Programa`, teste sua classe `Data` instanciando um objeto, chamando seu método e mostrando o resultado no `Console`.

## EXERCÍCIO 02

Crie uma classe chamada `Funcionario` que inclui três informações como atributos: o primeiro nome, o sobrenome e o salário mensal de um funcionário (valor decimal maior ou igual que 0 e menor que 20000).

Sua classe deverá ter um construtor que inicializa esses três valores. Para valores `default`, considere o nome e o sobrenome como textos vazios e o salário como 0.

Escreva métodos `Getters` e `Setters` para as variáveis que precisam ser protegidas pelo encapsulamento. Por exemplo, se um salário tentar ser alterado de modo inválido de acordo com a regra do negócio, o atributo deve ser mantido inalterado.

Na classe `Programa`, teste sua classe `Funcionario`. Crie duas instâncias de `Funcionario` e exiba no `Console` seus nomes e salários. Depois disso, aplique um aumento de 20% em seus salários e novamente faça a exibição no `Console`.

## EXERCÍCIO 03

Crie uma classe `Produto` que uma loja de materiais vai utilizar para representar um item vendido na loja. Um `Produto` deve possuir 4 informações: um número de identificação (inteiro positivo que varia de 1 até 9999), a quantidade de itens (inteiro não negativo), o nome do produto (texto) e o preço por item (um valor decimal não negativo).

Sua classe deve ter um construtor que inicializa os quatro valores de modo íntegro. Sendo assim, para as informações que achar necessário, crie métodos `Getters` e `Setters`. Para valores `default`, considere que um número de identificação padrão é 1, a quantidade é 0, o preço é 0 e o nome é um texto vazio.

## EXERCÍCIO 04

O imperador Romano, Júlio César, usava a Cifra de César para enviar ordens secretas aos seus generais. A ideia era trocar cada letra de uma mensagem pela terceira letra seguinte. Ou seja, a letra 'a' era substituída por 'd', a letra 'b' era substituída por 'e', e assim por diante.

Escreva uma classe chamada `Cifra`. Essa classe não possui atributos mas possui um método denominado `Criptografar` que recebe uma string como parâmetro e retorna outra string criptografada assim como a Cifra de César. Essa classe também implementa o método `Descriptografar` que recebe como parâmetro uma string cifrada e retorna a mensagem original.

Teste na classe `Programa` dentro do método `Main` a utilização da sua classe.

## QUESTÃO 5 – BASE PARA ENTENDIMENTO

Uma Função Polinomial do 2º Grau ou Função Quadrática é a função real definida por:

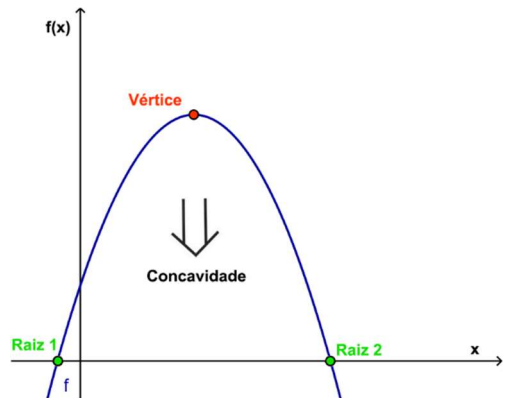
$$f(x) = ax^2 + bx + c$$

onde  $a$ ,  $b$  e  $c$  são coeficientes reais, sendo o primeiro coeficiente necessariamente não nulo ( $a \neq 0$ ). O gráfico obtido a partir de uma Função Quadrática no plano cartesiano é uma parábola como mostra a figura ao lado.

Várias são as características que podem ser extraídas de Funções Quadráticas. Vejamos:

### I. Concavidade da parábola

A parábola pode ter a concavidade voltada para cima ou para baixo dependendo do sinal envolvido no coeficiente  $a$ . A parábola tem a concavidade voltada para cima quando  $a > 0$  enquanto tem a concavidade voltada para baixo quando  $a < 0$ .



### II. Raízes

A parábola intercepta o eixo  $x$  (eixo das abscissas) no ponto  $(x, 0)$ , ou seja, sempre que  $f(x) = 0$ . Os valores de  $x$  que anulam a função são chamados de raízes. Uma Função Quadrática pode ter até duas raízes e seus valores são calculados através das seguintes expressões:

$$x_1 = \frac{-b + \sqrt{b^2 - 4 \cdot a \cdot c}}{2a} \quad x_2 = \frac{-b - \sqrt{b^2 - 4 \cdot a \cdot c}}{2a}$$

### III. Vértice

O vértice da parábola determina o ponto de mínimo ou de máximo da função. Tal vértice será o par ordenado  $(x_v, y_v)$ . O  $x$  do vértice é calculado como  $x_v = -b/2a$  e o  $y$  do vértice é calculado como  $y_v = -(b^2 - 4 \cdot a \cdot c)/4a$ .

## QUESTÃO 5 - PERGUNTAS

De acordo com o que foi dito, implemente uma classe que representa uma Função Quadrática seguindo as instruções abaixo:

- Implemente uma classe `FuncaoQuadratica`. Como sabemos que uma Função Quadrática é definida pelos seus três coeficientes, adicione nesta classe três atributos reais denominados por A, B e C.
- Determine se há necessidade de aplicar o encapsulamento em algum ou todos os atributos da classe de acordo com a descrição dada de Função Quadrática. Se houver a necessidade, crie os métodos Getters e Setters para o(s) atributo(s) e justifique com um comentário no código.
- Implemente um construtor para esta classe. Esse construtor irá receber três valores reais (a, b e c) como parâmetros e irá atribuí-los aos atributos da classe.
- Implemente os métodos `Raiz1()` e `Raiz2()`. Esses métodos deverão retornar o valor de cada raiz de um objeto `FuncaoQuadratica`. Você pode utilizar o método `Math.Sqrt()` da classe `Math`.
- Implemente os métodos `XVertice()` e `YVertice()`. Esses métodos deverão retornar o valor do x do vértice e y do vértice, respectivamente, de um objeto `FuncaoQuadratica`.
- Implemente o método `Concavidade()` que retorna um texto indicando “para cima” ou “para baixo” de acordo com o coeficiente *a*.
- Implemente o método `Descricao()` que retorna um texto descrevendo a função. Exemplo: se os parâmetros A, B e C valem 3, -4 e 1, então o método deverá retornar a string “ $y=3x^2-4x+1$ ”.
- Na classe `Program`, dentro do método `Main`, escreva o código abaixo para testar se sua classe funciona adequadamente. Entre com os valores 3, -4 e 1.

```
static void Main(string[] args)
{
    Console.WriteLine("Bem-vindo à calculadora de funções quadráticas!");
    Console.WriteLine("Favor entre com os valores de a, b e c:");
    // Captura dos dados de entrada
    double a = Convert.ToDouble(Console.ReadLine());
    double b = Convert.ToDouble(Console.ReadLine());
    double c = Convert.ToDouble(Console.ReadLine());

    //Instanciação de um objeto FuncaoQuadratica
    FuncaoQuadratica objFunc = new FuncaoQuadratica(a, b, c);

    Console.WriteLine("O objeto função criado é {0}", objFunc.Descricao());

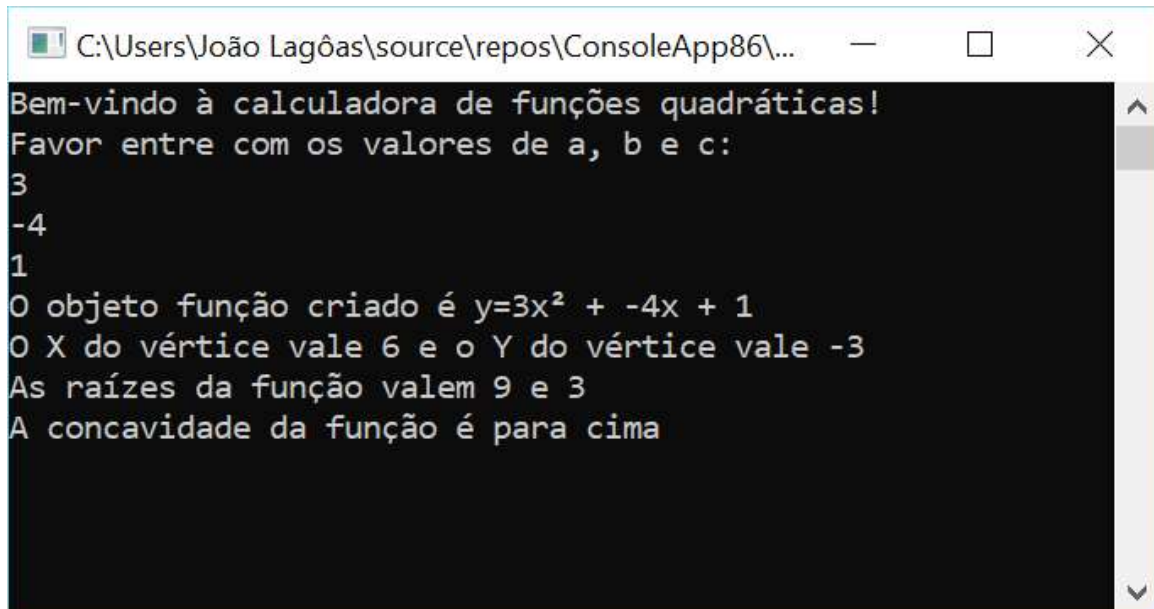
    //Agora que o objeto foi criado, vamos chamar seus métodos
    Console.WriteLine("O X do vértice vale {0} e o Y do vértice vale {1}",
        objFunc.XVertice(), objFunc.YVertice());

    Console.WriteLine("As raízes da função valem {0} e {1}", objFunc.Raiz1(),
        objFunc.Raiz2());

    Console.WriteLine("A concavidade da função é {0}", objFunc.Concavidade());

    Console.ReadLine();
}
```

Se tudo correu bem, você irá receber como resposta a seguinte saída:



```
C:\Users\João Lagôas\source\repos\ConsoleApp86\...
Bem-vindo à calculadora de funções quadráticas!
Favor entre com os valores de a, b e c:
3
-4
1
O objeto função criado é y=3x² + -4x + 1
O X do vértice vale 6 e o Y do vértice vale -3
As raízes da função valem 9 e 3
A concavidade da função é para cima
```