PROGRAMAÇÃO FUNCIONAL DENTRO DO C#

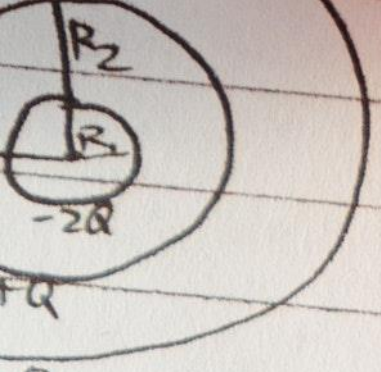
2022 – MATHEUS DA FONSECA DUMMER

O QUE SIGNIFICA?

Programação funcional pega a ideia da matemática para lidar com grandes abstrações através de pequenos blocos de "fórmulas".

Criando assim pequenas "fábricas" abstratas que se comportam de maneira semelhante mas que podem receber valores diferentes de processamento.

- Eficiência e legibilidade do código
- Escalabilidade
- Reutilização
- Portabilidade



...e R_1 is isolated

...to get

...ge on small shell

to cancel out $-2Q$ charge from outside

of R_1 inside of R_2 must be $+2Q$

To get total of $+Q$ on R_2 outside

of R_2 must be $-Q$

...f R_3 must cancel out $-Q$ so it is

...f $R_3 = +Q$

...s inside

...al

...y

COMO INICIAR NA PROGRAMAÇÃO FUNCIONAL?

- Primeiro precisamos saber com o que estamos lidando a nível de problema
- Funções podem retornar um tipo ou não retornar nada, e isso é muito importante saber!
- Funções trabalham com os mesmos tipos que as variáveis trabalham, ou seja, podem ser INT, BOOL, DOUBLE, STRING, CHAR, FLOAT ou ainda do tipo VOID.

- VOID é usado quando não se deseja retornar nenhum tipo na função, como por exemplo, apenas fazer um Console.WriteLine. VOID pode ser traduzido como VAZIO.

CONSTRUINDO UMA FUNÇÃO BÁSICA

Para exemplificarmos, vamos montar uma função que quando é chamada, vai exibir uma mensagem de boas vindas, semelhante a um hello world.

```
static void Saudacao()
    {
        Console.Write("Olá, Humano!");
    }


    public static void Main()
    {
        Saudacao();
    }
```

OUTRO EXEMPLO

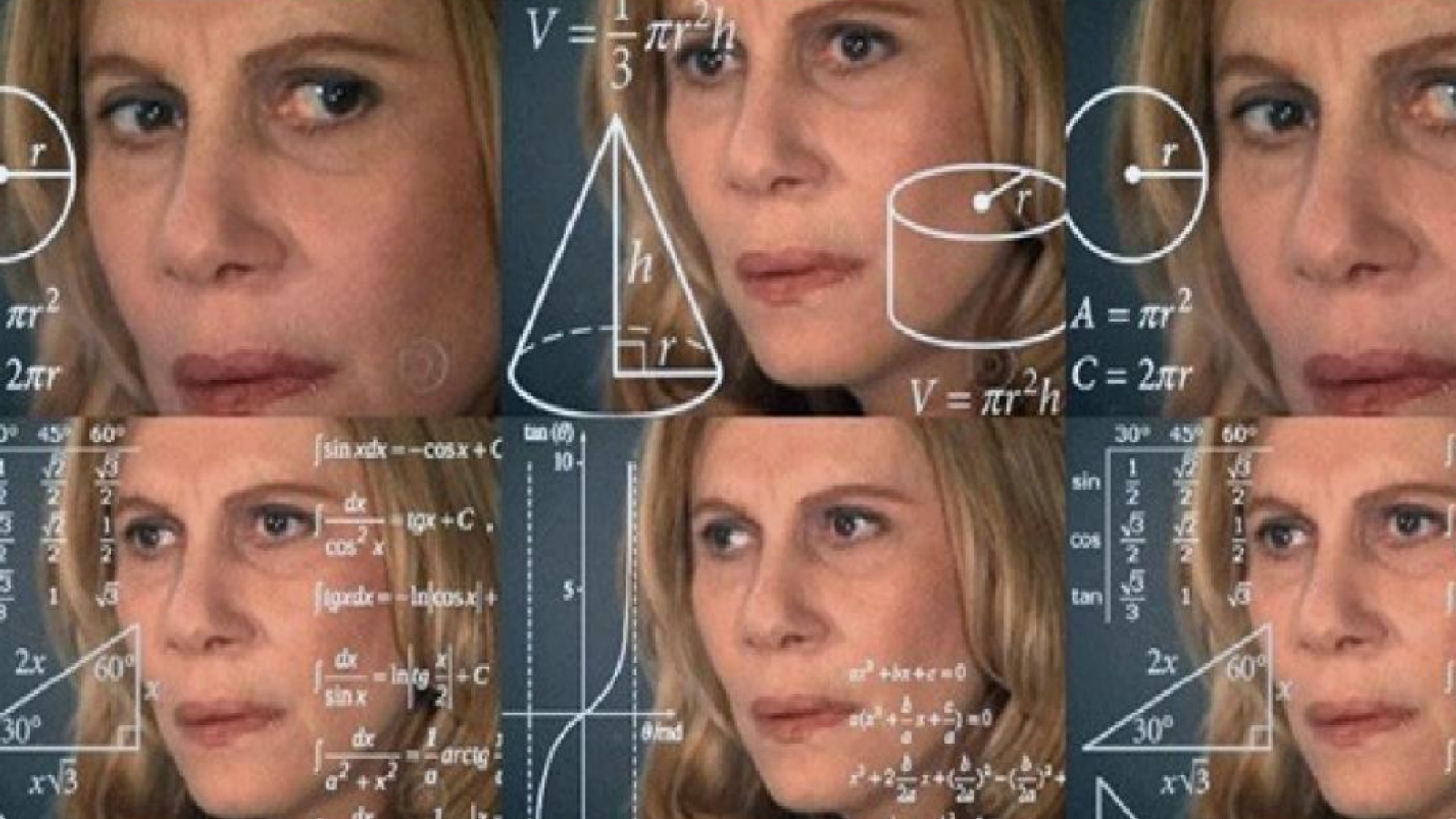
- Neste caso, a função recebe dois valores de tipo int (x,y).
- Dentro da função, esses valores são somados e salvos dentro de outra variável (z).
- Por fim, a função retorna esta variável (z).

```
static int Soma(int x, int y)
{
    int z;
    z = x + y;
    return z;
}

public static void Main()
{
    int resultado;
    resultado = Soma(1,3);
    Console.Write(resultado);
}
```

NESTE CASO...

- Como a função retorna um valor INT, ela não tem a capacidade de se auto exibir, apenas fazer uma operação e armazenar/retornar o valor para processamento posterior.
- Sendo necessário exibirmos ou processarmos o resultado manualmente.


r
πr^2
$2\pi r$
$V = \frac{1}{3}\pi r^2 h$
h
r
r
$V = \pi r^2 h$
r
$A = \pi r^2$
$C = 2\pi r$
30° 45° 60°
sin
cos
tan
$\int \sin x dx = -\cos x + C$
$\int \frac{dx}{\cos^2 x} = \operatorname{tg} x + C$
$\int \operatorname{tg} x dx = -\ln|\cos x| +$
$\int \frac{dx}{\sin x} = \ln\left|\operatorname{tg}\frac{x}{2}\right| + C$
$\int \frac{dx}{a^2 + x^2} = \frac{1}{a}\operatorname{arctg}$
2x
60°
30°
x
$x\sqrt{3}$
tan (θ)
10
5
θ rad
$ax^2 + bx + c = 0$
$a(x^2 + \frac{b}{a}x + \frac{c}{a}) = 0$
$x^2 + 2\frac{b}{2a}x + (\frac{b}{2a})^2 - (\frac{b}{2a})^2 +$

PODEMOS DEFINIR FUNÇÕES COMO:

- Pequenas fabricas, feitas para lidar com problemas específicos e recorrentes no nosso programa.
- Estas fabricas agem como blocos que reutilizamos um número infinito de vezes e com um número também infinito de dados que podemos alimentar a função.
- Dentro do C# chamamos também funções de MÉTODOS (mas normalmente o nome método está associado ao paradigma orientado a objetos)

ANATOMIA DE UMA FUNÇÃO

<VISIBILIDADE> <TIPO DE RETORNO> NOME DA FUNÇÃO (<PARAMETROS>)

```
{
    // Código
   // ...
}
```

VISIBILIDADE

Escopo	Public	Protected	Private	Internal	Private protected	Protected internal
Todo o programa	sim	não	não	não	não	não
Classe pertencente	sim	sim	sim	sim	sim	sim
Projeto atual	sim	não	não	sim	não	sim

TIPO DE RETORNO

- INT retorna apenas tipos inteiros
- BOOL retorna apenas verdadeiro ou falso
- STRING retorna apenas strings
- DOUBLE retorna apenas doubles
- FLOAT retorna apenas floats
- VOID não retorna nenhum valor especifico (TIPO VAZIO)

PARÂMETROS

- É onde especificamos os dados para a nossa função processar

- É de extrema importância sempre inserir os parâmetros de tipo certo, caso contrário pode ser difícil de processar os dados da maneira ideal ou até mesmo ter de realizar conversões complexas posteriormente!

BOAS PRÁTICAS

- É considerado boa pratica sempre definir o nome de uma função com letra maiúscula, para diferenciarmos ela no código e sabermos logo de cara que se trata de uma função/método.
- Normalmente sempre queremos retornar algum valor na função, então é importante lidarmos com a tipagem correta.

EXERCÍCIOS

- Faça um programa que tenha uma função para concatenar os nomes dos usuários de um sistema, onde a função receba NOME e SOBRENOME .
- Faça um programa que tenha uma função para retornar sempre o MENOR NÚMERO entre dois números inseridos nos parâmetros.
- Faça um programa que tenha uma função para retornar a RAIZ QUADRADA de um determinado número.

REFERENCIAS

- Microsoft. C# Reference Guide, 2022. Disponível em: https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/

- MANZANO, José Augusto N. G. Microsoft C# Community 2015. 1. Ed. São Paulo: Erica, 2016