Programação Funcional

PARADIGMAS DE PROGRAMAÇÃO

O que é programação funcional?

- Não existe uma definição unânime sobre o que é programação funcional, mas podemos dizer que ela é um paradigma que se concentra mais nos resultados computacionais do que nas ações que serão executadas para chegar a esse resultado. Geralmente as linguagens funcionais são muito sucintas e expressivas, sempre procurando chegar no resultado usando o menor número de recursos.

O que é programação funcional?

- As linguagens declarativas (funcionais) possuem uma forma diferente de estender o vocabulário que usamos para criar um programa. Não estamos limitados a criar novos comandos, podemos criar novas estruturas de fluxo e compor a forma como o programa se comporta.

O que é programação funcional?

- Na programação funcional, desenvolvemos funções lidando com os parâmetros de entrada e gerando resultados, sem alterarmos estados de coisas preexistentes no programa. Com isso, podemos notar alguns benefícios, como:
  - Maior previsibilidade do comportamento;
  - Facilidade de extensão;
  - Melhora de testabilidade.

Como iremos programar funcional?

- Existem diferentes alternativas para a programação funcional, como foi dito em aula, algumas linguagens não estão presas a somente a um paradigma e tem suporte a programação funcional além de paradigmas imperativos. Mas para aliciar a experimentação de uma linguagem nova, com algo que dê suporte total ao paradigma, utilizaremos a solução .NET para isso: F#.
- O F# é uma linguagem bastante concisa, poucos parênteses são utilizados, blocos de código são separados por indentação (não usa chaves), e a quebra de linha já é um delimitador.

Começando...

- Criem um projeto novo no Visual Studio, em F# e Console Application, alterem levemente o código original para parecer com isso:

```
[<EntryPoint>]
let main argv =
    printf "Olá Mundo"
    0
```

F#

- Vamos criar nossa primeira função:

```
let OlaMundo() =
    printfn "Olá Mundo"


[<EntryPoint>]
let main argv =
    OlaMundo()
    0
```

Primeiro Exemplo

- Vamos pensar em um programa que faça a seguinte operação: Retorne a soma dos quadrados dos valores de 1 a 10.
- Primeiro vamos pensar em C++...
- Podemos criar uma função que retorne o número elevado ao quadrado e em seguida fazer uma função que faça a soma desses números.

Agora em F#

```fsharp
let elevaNumeroAoQuadrado numero = numero * numero

let SomaQuadrados =
    [1..10] |> List.map elevaNumeroAoQuadrado |> List.sum

[<EntryPoint>]
let main argv =
    printfn "%d" SomaQuadrados
    0
```

Paradigma Funcional

- Normalmente em linguagens imperativas, gerenciamos o estado de nossos programas através de propriedades, armazenadas em nossos programas e objetos. Na programação funcional fazemos esse gerenciamento através de expressões.

Paradigma Funcional

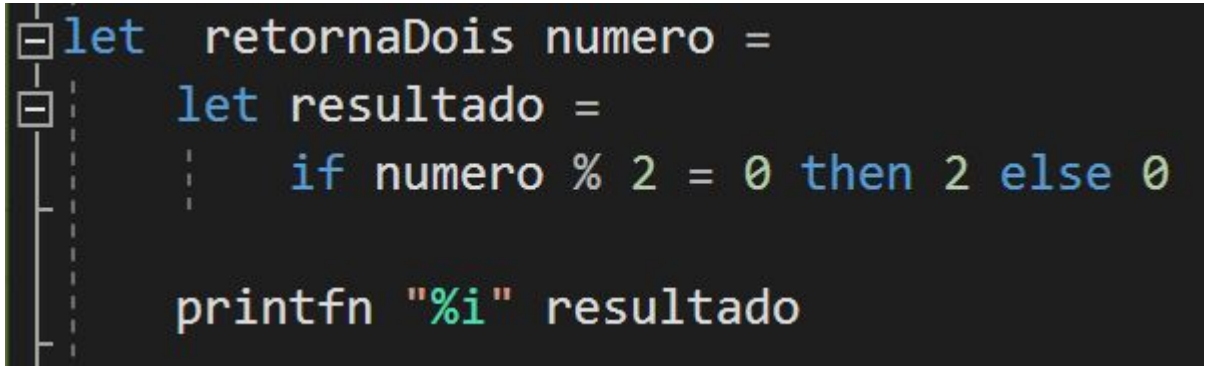
- No paradigma funcional, enfatizamos muito mais o uso de expressões em detrimento ao uso de declarações, pois normalmente as expressões geram blocos mais seguros e com menos problemas.
- Expressão indica uma combinação entre valores e funções para gerar um resultado, já uma declaração é uma de unidade de execução que indica uma ação que não retorna nenhum tipo de valor e sempre vai gerar um efeito colateral no código.
- Em linguagens puramente funcionais (o que não é o caso do F#) não existe a possibilidade de fazer uma declaração.

Diferença no uso de declarações e expressões

- Vamos fazer primeiro em C++.
- Uma função que receba como parâmetro um número, e caso o número seja par, ela imprime o número 2.

- Agora vamos ver em F#...

Diferença no uso de declarações e expressões



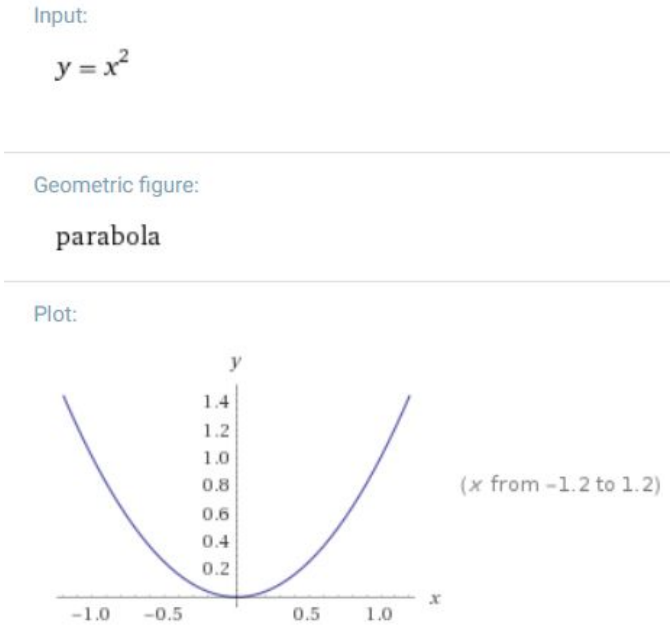
Funções?

- Aqui temos uma diferença do que julgávamos como funções em paradigmas imperativos, onde determina o comportamento do programa, ou de um objeto (no caso da linguagem orientada a objetos). Na programação funcional, as funções são coisas isoladas, elas não precisam estar associadas a outra coisa. Elas são membros de primeira ordem.

- Elas podem ser passadas como parâmetro para outras funções e podem ser retornadas por uma função e até atribuídas para uma variável.

Funções?

- Elas geram um resultado (retorno) a partir de uma operação, que pode envolver os parâmetros recebidos. É exatamente como um função matemática:



Funções e Valores

- O retorno das funções seriam o que chamamos de variáveis, mas em programação funcional trabalhos com algo chamado **imutabilidade**, parece estranho, mas é interessante que nossas variáveis não se alterem ao longo do programa, que elas tenham um valor único e consistente.
- Chamamos então as variáveis de valores, e utilizamos a palavra reservada **let** para declararmos um novo valor, podemos atribuir um valor básico, como um tipo primitivo, mas também uma função, que nada mais é que um valor que precisa ser avaliado, como a função matemática.

Exercícios

Faça uma função em F# que dado um valor escreve o seu módulo (valor sem sinal)

Faça uma função em F# que receba 2 valores e calcule a média aritmética entre eles

Faça uma função que recebe três valores e retorna o menor

Faça uma função que receba 3 números e retorne se eles formam um triângulo equilátero, isósceles ou escaleno