Abstrações e Funções de Alta Ordem - Expressões Lambda

Kleber Jacques F. de Souza

Expressões Lambda

- **Expressões lambda** são “delegates inline”
- É uma maneira de se representar **ponteiros para métodos sem** a necessidade de se **criar toda a estrutura** que vimos anteriormente para representação de um *delegate*.

Expressões Lambda

- Os métodos que são passados através de uma expressão lambda são **métodos anônimos.**
- Não possuem nome e nem um corpo declarado no decorrer da classe onde são utilizados.

Expressões Lambda

- Uma expressão lambda possui a seguinte sintaxe:

(parâmetros) => expressão-ou-declaração-de-bloco

Expressões Lambda

```
public delegate int CalculoMatematico(int a, int b);

// Delegate sendo utilizado com uma expressão lambda
CalculoMatematico soma = (a, b) => a + b;

Console.WriteLine(soma(1, 2)); // Irá escrever "3"
```

Expressões Lambda

- Por comodidade, é possível omitir os parênteses se, e somente se, houver exatamente um parâmetro na expressão.

```
public delegate int CalculoPotencia(int a);
CalculoPotencia c = x => x * x * x;
Console.WriteLine(c(2)); //Irá escrever "8"
```

Expressões Lambda

- Expressões lambda são interessantes porque auxiliam na produção de um código mais "**direto**" e mais **limpo**.
- Com expressões lambda podemos **eliminar** a necessidade de se criar um método com toda a **burocracia** necessária para se executar uma pequena ação em cima de um *delegate*.

Expressões Lambda

- Podemos ter dois tipos de expressões lambda:
  - lambdas de **expressão**
  - lambdas de **instrução**.

Expressões Lambda

- Lambdas de expressão **retornam** algum tipo de **dado**, definido previamente pelo *delegate* para o qual ela aponta.
- Geralmente, são constituídas por uma expressão do lado direito do sinal de lambda (=>).

Expressões Lambda

```
n => n < 5;

(x, y) => x <= y;

(int a, int b) => a* b;
```

Expressões Lambda

- Já lambdas de instrução possuem um corpo de método, geralmente executando ações.
- São identificadas pelo fato de a expressão do lado direito do sinal de lambda (=>) estar entre chaves.

Expressões Lambda

```
(s) => {Console.Write(string.Format("Foi digitado {0}", s));};

(a, b) => {

        int resultado = (a + b) * 10;
        Console.WriteLine(string.Format("O resultado é {0}",
        resultado));
};
```

Expressões Lambda

- A **tipagem** dos parâmetros em uma expressão lambda **é feita por inferência** do compilador, **de acordo com o *delegate*** para o qual ela está apontando.
- Não é necessário definir explicitamente o tipo dos parâmetros em uma expressão lambda.

Expressões Lambda

```
Func<int, int, double> potenciacao =
    (int b, int p) => { return Math.Pow(b, p); };
Console.WriteLine(potenciacao(2, 3)); // Irá imprimir 8
```

```
Func<int, int, double> potenciacao =
(b, p) => Math.Pow(b, p);
Console.WriteLine(potenciacao(2, 3)); // Irá imprimir 8
```

Referências Bibliográficas

Microsoft 2017. Expressões Lambda. Disponível em: https://docs.microsoft.com/pt-br/dotnet/csharp/programming-guide/statements-expressions-operators/lambda-expressions