



Abstrações e Funções de Alta Ordem - Closures

Kleber Jacques Ferreira de Souza

Closures

- **Closure** é formada quando **o corpo de uma função** se refere a uma ou mais **variáveis livres**.
 - Variáveis que **não são passadas como argumentos ou definidas localmente**, mas sim **definidas no escopo fechado onde a função é definida**.

Closures

- Em C# podemos aplicar este conceito utilizando Funções de Alta Ordem com *delegates* e expressões *lambdas*.
- Variáveis lambda têm acesso tanto aos parâmetros que são passados a ela quanto a **variáveis no escopo** do método e da classe onde estão inseridos.

Contexto de Variáveis em uma Expressões lambda

```
private const int FATOR_MULTIPLICACAO = 5;

static void Main(string[] args)
{
    int x = 2, y = 3;
    int fatorSoma = x + y; 2 + 3 = 5

    Func<int, int, double> operacao = (b, p) =>
        (Math.Pow(b, p) + fatorSoma) * FATOR_MULTIPLICACAO;

    Console.WriteLine("resultado: " + operacao(x, y));
    (23 + 5)*5 = (8+5)*5 = 13*5 = 65
}
```

Qual o resultado do código abaixo?

```
public static Action CreateAction(){  
  
    int contador = 0;  
  
    Action action = () => {  
        contador++;  
        Console.Write(contador);  
    };  
  
    return action;  
}
```

```
static void Main(){  
  
    var action = CreateAction();  
  
    action();           1  
    action();           2  
    action();           3  
    action();           4  
    action();           5  
  
}
```

Referências Bibliográficas

Wampler, Dean. **Programação Funcional**. Novatec, São Paulo, 2014.

Microsoft 2017. **Delegates**. Disponível em: <https://docs.microsoft.com/pt-br/dotnet/csharp/programming-guide/delegates/>

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