



# **C# Avançado**

## **Exercícios Propostos**

**Delegates e Events**

## 1 Exercício

Crie um delegate chamado `Filter`, que pode referenciar métodos que recebem um `int` como parâmetro e retornam um `bool`.

Depois, crie um método chamado `FilterList()`, que recebe como parâmetro uma lista de números inteiros e um delegate `Filter`. Este método deve invocar o delegate `Filter` em cada um dos elementos e, no final, retornar uma nova lista. Esta nova lista conterá apenas os elementos cujo retorno do delegate for `true`.

Para testar o método `FilterList()`, crie uma lista de números de 0 a 10. Depois crie dois métodos que poderão ser referenciados pelo delegate. O primeiro método é o `FilterGreatedThan5()`, que deve retornar na nova lista apenas os números maiores do que 5. O segundo método é o `FilterOdd()`, que deve retornar na nova lista apenas os números ímpares.

**Dica:** Números ímpares são aqueles cujo resto da divisão por 2 (operador `"%"`) é diferente de 0.

## 2 Exercício

Crie uma classe chamada `Clock`, que irá representar um cronômetro. Esta classe possui o método `Start()`, que inicia um loop infinito. A cada 1 segundo passado, este método chama métodos de callback registrados informando quantos segundos se passaram desde o início da contagem (a contagem inicia quando `Start()` é chamado).

Para implementar este comportamento, você precisa de um delegate, que você chamará de `SecondsHandler`. Ele referencia métodos que recebem um parâmetro do tipo `long` (número de segundos) e retornam `void`.

Crie um método chamado `OnSecond()` na sua aplicação, que é chamado via delegate pelo objeto `Clock` a cada segundo. Imprima na tela o valor do segundo fornecido.

**Dica:** Utilize a chamada `Thread.Sleep(1000)` para fazer a aplicação pausar por 1 segundo. A classe `Thread` pertence ao namespace `System.Threading`.

## 3 Exercício

Implemente a mesma aplicação do Exercício 2, mas utilize um event para notificar os interessados a respeito da passagem do tempo.

Use o padrão de eventos definido pela Microsoft, onde o método chamado deve receber dois parâmetros: um `object`, que identifica o gerador do evento; e um objeto da classe `SecondElapsedEventArgs`, que será criada por você e conterá as informações do evento (neste caso, o número de segundos passados).