



Disciplina Técnicas Avançadas de Programação	Curso Sistemas de Informação	Período 5º
Professor Kleber Jacques F. de Souza (klebersouza@pucminas.br)		

## Lista de Exercícios - Reflexão / *Delegates* / Expressões *Lambda*

### Questões

1. Defina o conceito de Reflexão para linguagens Orientada a Objetos.
2. Exemplifique 3 situações em que poderíamos aplicar o conceito de Reflexão.
3. Implemente um programa usando reflexão que seja capaz de mapear e descrever todas as informações de um conjunto de classes que se encontram em um determinado diretório/pacote informado pelo usuário.
4. Defina o conceito de *Generics* utilizado no .NET e explique suas principais vantagens e desvantagens. Dê exemplos de sua utilização.
5. Defina o conceito de *Delegates* e *Delegates Multicast* utilizados no .NET e explique suas principais vantagens e desvantagens. Dê exemplos de sua utilização.
6. Defina o conceito de Expressões *Lambda* utilizado no .NET e explique suas principais vantagens e desvantagens. Dê exemplos de sua utilização.
7. Implemente um programa de Calculadora utilizando *delegates*. A calculadora deve implementar as funções de soma, subtração, multiplicação e divisão de números inteiros.
8. Implemente o exemplo abaixo, adaptando a classe MyClass para a classe aluno. O aluno deve ter um nome, um número de matrícula e uma data de nascimento. Use *delegates multicast* para imprimir os dados dos alunos.

```
1  class MyClass{
2      private string name;
3      private int number;
4      public MyClass( string name, int number){
5          this.name = name;
6          this.number = number;
7      }
8      public void ShowMyName(){
9          Console.WriteLine("My name is " + this.name);
10     }
11     public void ShowMyNumber(){
12         Console.WriteLine("My number is " + this.number);
13     }
14
15     delegate void DemoOp(); // declare delegate type
16
17     static void Main()      {
18         // cria algumas instâncias de MyClass
19         MyClass mc1 = new MyClass("First", 1);
20         MyClass mc2 = new MyClass("Second", 2);
21
22         DemoOp demo = null; // declara um delegate vazio
23
24         // adiciona métodos ao delegate
25         demo += new DemoOp(mc1.ShowMyName);
26         demo += new DemoOp(mc2.ShowMyNumber);
27
28         demo(); // invoca o delegate
29     }
30 }
```

9. Escreva uma expressão *lambda* que recebe uma tripla formada pelo nome, peso (em quilogramas) e altura (em metros) de uma pessoa e retorne o índice de massa corporal, dado pela equação  $IMC = \frac{massa}{altura * altura}$ . Crie um programa e realize testes na sua expressão.
10. Crie um programa para gerar a sequência (números) de Fibonacci. Esta sequência é constituída recursivamente da seguinte forma:

$$F_{n+2} = F_{n+1} + F_n, \quad n > 0, \quad F_0 = 1 \text{ e } F_1 = 1$$

Um exemplo da série de Fibonacci pode ser visto abaixo:

$$F = 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, \dots$$

Em seu programa você deve criar um função, usando *closure*, que a cada chamada gere o próximo número na sequência de Fibonacci.