Programação Visual

Trabalho de Laboratório nº 3

Objetivo	Introdução aos componentes de C# com criação e utilização de propriedades e eventos.
Programa	Construir um programa para trabalhar com figuras geométricas. O programa deverá usar as classes implementadas no Lab02 e permitir visualizar as figuras num ecrã virtual.
Regras	Crie uma aplicação de consola. Implemente o código necessário e teste no fim de cada nível. Use as convenções de codificação adotadas para a linguagem C#. Não é necessário obter dados do utilizador. Forneça os dados no código.
Descrição **Nível 1**	1. Abra a solução "**LAB03_Materiais**" com a resolução do laboratório anterior. 2. Refaça o código (Refactoring) apagando todos os **Get** e **Set** associados a atributos e substituindo-os por propriedades. Classes a alterar: **Figura**, **Circulo, Quadrado, Reta** e **Desenho.** 3. Continuando o *refactoring* crie propriedades para a área das figuras substituindo os *Gets* existentes. Nestes casos são propriedades apenas de leitura. Substitua também o **GetArea** na classe **Desenho** por uma propriedade. 4. Imprima a string `\\NIVEL 1 **********` e teste a criação de objectos no main
Nível 2	1. Crie um evento **OnChanged** na classe **Desenho** que irá notificar da existência de alterações na lista de figuras do desenho. Baseie-se num ***delegate*** do template Action<String> O evento deve ser lançado sempre que ao desenho for adicionada ou removida uma figura. 2. Imprima a string `\\NIVEL 2 **********` e teste o evento criado na classe **Program** inscrevendo no evento uma lambda expression com a assinatura requerida pelo evento que escreva na consola "Desenho Alterado". Inscreva o método no evento **OnChanged** do objecto **desenho1** do **Main** e confirme que é chamado sempre que o desenho é alterado.
Nível 3	1. Crie na classe **Desenho** a propriedade **Titulo** (use propriedades implícitas) e altere o programa principal para que na criação do desenho **desesenho1** lhe seja atribuído o título "**DESENHO 1**" (use agora a sintaxe de inicialização de objetos do C#). 2. Crie um método de extensão **ToUpperFirstLetter** para a classe **string** que converta a *string* para minúsculas colocando só a primeira letra com Maiúsculas. (Ex: DESENHO 1 passa a Desenho 1). 3. Imprima a string `\\NIVEL 3 **********` e teste este método escrevendo para o ecrã o título do desenho no novo formato usando o método de extensão.
Nível 4	Na *framework* .NET os eventos têm normalmente o formato: `public delegate void EventHandler(Object sender, EventArgs e);` O argumento ***sender*** deste *delegate* é a referência para o objeto que está a lançar o evento, o argumento ***e*** é um objeto da classe **EventArgs** definida na *framework* .NET que aceita informação sobre o evento. **P**retende-se substituir o evento **OnChanged** por um evento com o formato da *framework* .NET. 1. Crie *um novo evento* ***On*ChangedDotNet** com a mesma função que o anterior (3) mas agora com o formato e funcionamento da *framework* .NET. Mantenha o **OnChanged** sem alterações. 2. Imprima a string `\\NIVEL 4 **********` e teste novo evento tal como fez para o evento OnChanged num novo objecto denominado desenho2. 3. Use agora uma Func e uma Action para o mesmo efeito. Teste.
Nível 5	1. Considerando que não faz sentido calcular a área para objectos da classe Reta proceda ao refactoring do programa eliminando a propriedade Area na classe Reta. Conselho: Recorra a uma super classe Forma. 2. Imprima a string `\\NIVEL 5 **********` e teste.
Nível 6	Crie uma List de objectos da classe forma, insira nele o circulo1 o quadrado 1 e a reta1 e faça a saída dos dados para o ecran. Substitua todos os arrays por Lists e refaça as iterações . Imprima a string \\NIVEL 6 ********** e teste.