

Laboratório 10 – Eventos e Delegates

Este laboratório introduz a manipulação de tipos delegate e a declaração de eventos em C# diretamente em código.

1 Declarar eventos

1. Crie um novo projeto “Console Application” com nome “Laboratorio10”.
2. Adicione uma nova classe com o nome “Termometro” contendo código a seguir:

```
public class Termometro
{
    private double valor;

    public Termometro()
    {
        valor = 0.0;
    }

    public double Temperatura
    {
        get { return valor; }
        set { valor = value; }
    }

    virtual public void Aumentar()
    {
        valor += 0.1;
    }

    virtual public void Aumentar(double quantia)
    {
        valor += quantia;
    }

    virtual public void Diminuir()
    {
        valor -= 0.1;
    }

    virtual public void Diminuir(double quantia)
    {
        valor -= quantia;
    }

    override public string ToString()
    {
        return valor.ToString();
    }
}
```

3. Acrescente uma nova classe “TermometroLimite” como subclasse de “Termometro” e o seguinte código inicial:

```
public class TermometroLimite : Termometro
{
    private double limiteSuperior;
```

```

private bool disparadoEventoLimiteSuperior;

public TermometroLimite(double ls)
{
    limiteSuperior = ls;
    disparadoEventoLimiteSuperior = false;
}

public double LimiteSuperior
{
    get { return limiteSuperior; }
    set { limiteSuperior = value; }
}
}

```

4. Adicione a declaração de um delegate para os eventos da classe “TermometroLimite”:

```
public delegate void MeuDelegate(string msg);
```

5. Adicione um membro de evento na classe “TermometroLimite” através da seguinte declaração:

```
public event MeuDelegate LimiteSuperiorEvent;
```

6. Declare um método auxiliar para disparar o evento:

```

private void OnLimiteSuperiorEvent()
{
    // verifica se a temperatura ultrapassou o limite e
    // Verifica se o evento já foi disparado, para não disparar continuamente.
    if ((this.Temperatura > limiteSuperior) && (!disparadoEventoLimiteSuperior))
        // verifica se há tratador
        if (LimiteSuperiorEvent != null)
        {
            LimiteSuperiorEvent("Atenção: temperatura acima do limite!!!");
            disparadoEventoLimiteSuperior = true;
        }
}

```

7. Sobrescreva os métodos que aumentam a temperatura do termômetro para testarem se o valor ultrapassou o limite e dispararem o evento associado:

```

public override void Aumentar()
{
    base.Aumentar();
    OnLimiteSuperiorEvent();
}

public override void Aumentar(double quantia)
{
    base.Aumentar(quantia);
    OnLimiteSuperiorEvent();
}

```

8. Sobrescreva os métodos que diminuem a temperatura do termômetro para testarem se o valor baixou do limite superior e *desligarem* o aviso do evento:

```

public override void Diminuir()
{

```

```
        base.Diminuir();
        if ((this.Temperatura <= limiteSuperior) && (disparadoEventoLimiteSuperior))
        {
            disparadoEventoLimiteSuperior = false;
        }
    }

    public override void Diminuir(double quantia)
    {
        base.Diminuir(quantia);
        if ((this.Temperatura <= limiteSuperior) && (disparadoEventoLimiteSuperior))
        {
            disparadoEventoLimiteSuperior = false;
        }
    }
}
```

2 Desenvolvendo uma aplicação visual para testar

1. Altere o método Main de acordo com o seguinte código:

```
// inicializa o termometro e mostra seu valor
TermometroLimite term = new TermometroLimite(5);
Console.WriteLine(term.ToString());

// adiciona um tratador ao evento LimiteSuperiorEvent
term.LimiteSuperiorEvent += new TermometroLimite.MeuDelegate(TrataLimiteSuperior);

// aumentar a temperatura além do limite
term.Aumentar(6);
Console.WriteLine(term.ToString());
```

2. Adicione código para o tratamento de evento dentro da classe “Program”:

```
private static void TrataLimiteSuperior(string msg)
{
    Console.WriteLine(msg);
}
```

3. Teste a aplicação.

3 Exercícios

1. Altere a classe TermometroLimite para gerar um evento notificando que a temperatura voltou ao normal. Chame um método para diminuir a temperatura e utilize um tratador de evento adequado.

2. Altere o delegate para que seja fornecida, além de uma mensagem, o valor corrente da temperatura.