

Aufgabe Pi Berechnung

Verwenden Sie die folgende Methode für einen asynchronen Aufruf und erwarten Sie das Ergebnis von `ComputePi()`.

```
double ComputePi()
{
    double sum = 0.0;
    const double step = 1e-9;

    for(int i = 0; i < 1_000_000_000; i++)
    {
        double x = (i + 0.5) * step;
        sum += 4.0 / (1.0 + x * x);
    }

    return sum * step;
}
```

Starten Sie die Berechnung durch einen Buttonklick. Der Click-Ereignishandler soll anschließend den Button deaktivieren. Warten Sie auf das Ergebnis der Berechnung und setzen Sie das Ergebnis der Berechnung in ein Label ein. Aktivieren Sie den Button anschließend wieder.

Um aus einem Task auf die Elemente einer Windows-Forms-Anwendung threadsicher zugreifen zu können, verwenden Sie die Methode `BeginInvoke` der Klasse `Control`.

Ein Beispiel (Code im Task):

```
BeginInvoke((MethodInvoker)delegate
{
    textBox1.Text += "Text";
});
```

Mögliche Zusätze zur Aufgabe:

- Zeigen Sie während der Berechnung die Sanduhr an.
 - Mittels `this.Cursor = Cursors.WaitCursor` wird eine Sanduhr angezeigt. `Cursors.Default` ist der Standard Mauszeiger.
- Fügen Sie einen Button zum Abbruch der Berechnung ein.
- Verwenden Sie ein `Progressbar`-Control sowie die Klasse `Progress` (siehe MSDN) zum Anzeigen des Fortschritts der Berechnung.