

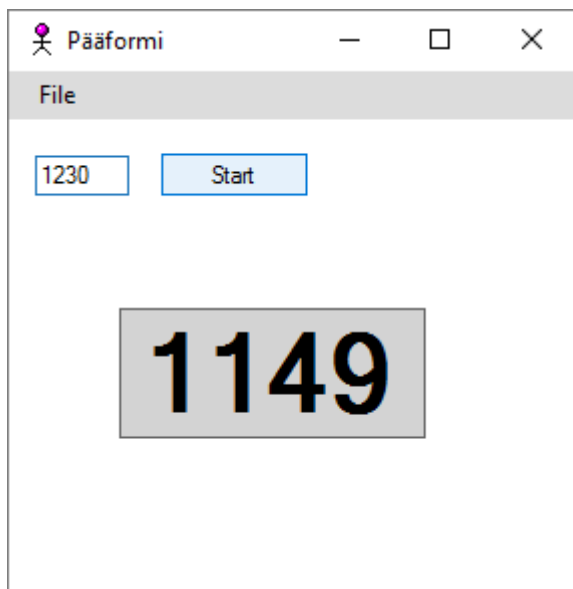
## 05 For

Vaihe 1.

Kirjoita Windows Forms -ohjema, jonka formissa (Kuva 1-1) on TextBox-komponentti kokonaisluvun syöttämistä varten, Start-nappi ja Label-komponentti numeroiden tulostamista varten.

Kun nappia klikataan hiirellä, niin napin tapahtumankäsittelijässä luetaan TextBox:ssa oleva numero ja for-loopissa tulostetaan numerot 1:tä alkaen kyseiseen numeroon asti esimerkiksi 100 ms välein.

For-luuppi on kätevä silloin, kun etukäteen tiedetään luupin kierrosten lukumäärä.



Kuva 1-1. Pääformi

Vaihe 2.

Kierroksien suoritussnopeutta voidaan säätää [Thread.Sleep](#) metodilla. Numerot ei automaattisesti muutu tekstielementissä, vaan ne täytyy päivittää jokaisen muutoksen jälkeen `update()` tai `refresh()` kutsuilla.

Vaihe 3 (Vapaaehtoinen. Tutustu, jos kiinnostaa), hieman säikeistä ja moniajasta.

Pääikkuna "jäätty" silmukan suorituksen ajaksi. Tämä johtuu siitä että ohjelma toimii vain yhdessä *säikeessä* (*Thread*). Ts. ohjelman suoritus tapahtuu yhdessä kohdassa kerrallaan. Kun laskuria päivitetään on käyttöliittymä jäätynyt koska ohjelman suoritus ei palaa tarkistamaan esim. käyttäjän tekemisiä ennen kuin laskuri on laskettu loppuun.

Tilanteen korjaamiseksi täytyy ohjelmaan lisätä toinen suorituslinja/säie ja laskuri siirretään sinne. Nyt ohjelma pystyy seuraamaan samaan aikaan dialogia, ja päivittämään laskuria. Samalla ohjelma muuttuu moniajo-ohjelmaksi.

Enemmän teoriaa englanniksi: [ohje](#).

Voit kopioida alla olevan koodin ohjelmaasi hyödyntääksesi moniajoa.

```
//Lisätään alkuun uusi nimiavaruus

using System.Threading;

private void StartButton_Click(object sender, EventArgs e)

{

    Thread OmaSaie;

    // Luodaan toissijainen säie ja käynnistetään se Start()-metodilla.

    OmaSaie = new Thread(new ThreadStart(this.SaieProceduuri));

    OmaSaie.Start();

}

private void SaieProceduuri()

{

    // Muutetaan tekstinä oleva numero kokonaisluvuksi.

    Int32 Kierrokset = Convert.ToInt32(NumberTextBox.Text);

    for (Int32 i = 0; i < Kierrokset; i++)

    {

        string uusiNumero = (i + 1).ToString();

        this.AsetaTeksti(uusiNumero);

    }

}
```

```

        // Viive
        Thread.Sleep(10);
    }
}

delegate void StringArgReturningVoidDelegate(string texti);
private void AsetaTeksti(string texti)
{
    // InvokeRequired required compares the thread ID of the
    // calling thread to the thread ID of the creating thread.
    // If these threads are different, it returns true.

    if (this.DisplayLabel.InvokeRequired)
    {
        //Herätetään alkuperäinen säie
        StringArgReturningVoidDelegate d = new
StringArgReturningVoidDelegate(AsetaTeksti);

        this.Invoke(d, new object[] { texti });
    }
    else
    {
        //Jos on alkupeäinen säie laitetaan se ´muuttamaan teksti
        this.DisplayLabel.Text = texti;
    }
}
}

```