

Paavo Räisänen

C# Windows ohjelmointi perusopas

www.ohjelmoimaan.net

Tätä opasta saa vapaasti kopioida, tulostaa ja levittää ei kaupallisissa tarkoituksissa.
Kuitenkaan omille nettisivuille opasta ei saa liittää.
Opetustarkoituksessa materiaali on vapaasti käytettävissä.
Verkko-opetuksessa oppaan saa julkaista oppilaille tarkoitetuilla sivuilla.

Sisällysluettelo

1: Alkusanat

2: Alkuun

3: Numeroiden tulostus ja lukeminen Windows ohjelmassa

4: Harjoitusohjelma Eurolaskuri

4: Valintaruutujen käyttö, ja tekstikentän vierittäminen

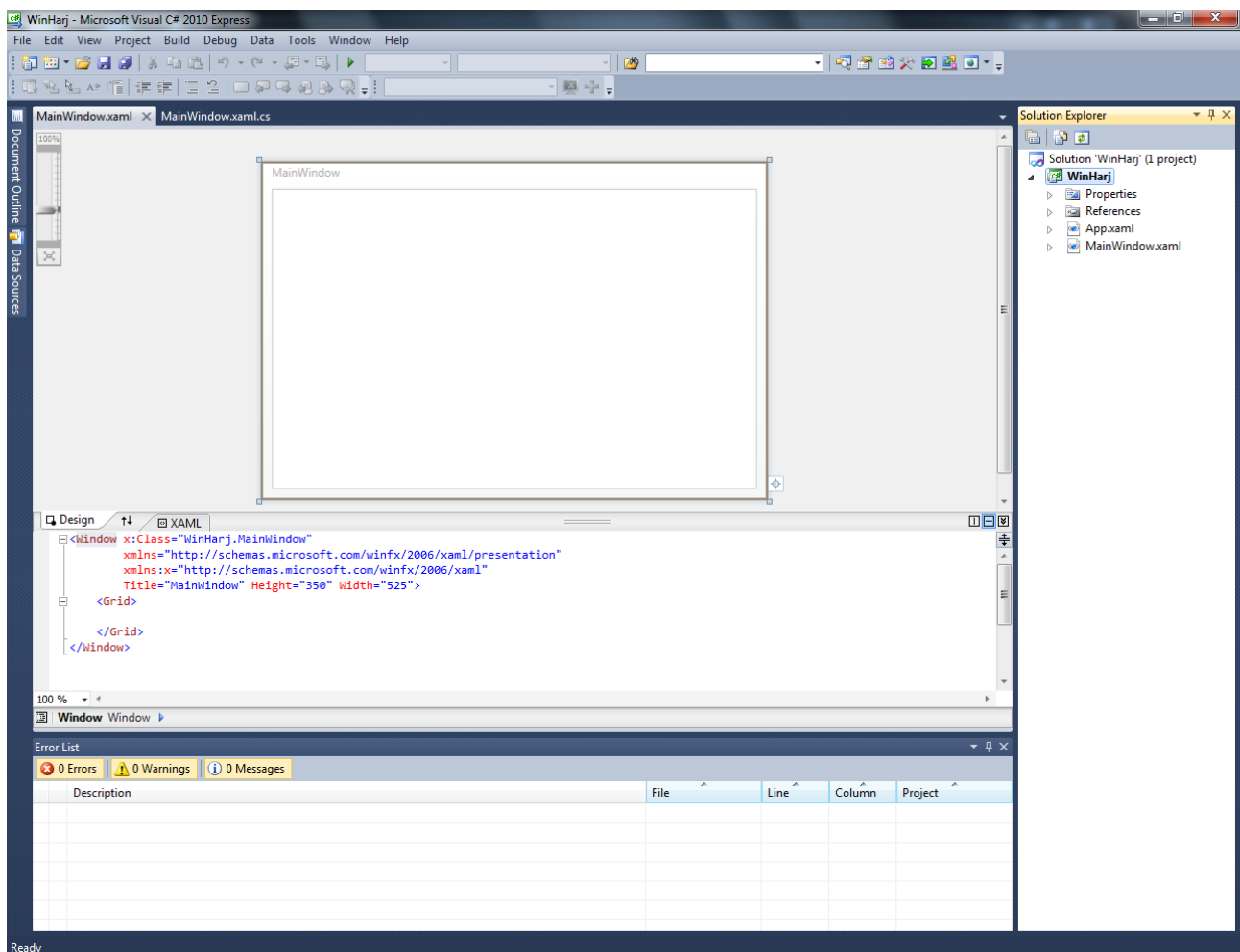
1: Alkusanat

C# on käytetyimpiä Windows ohjelmointikieliä. Tässä oppaassa oletetaan, että sinulla on sellaiset C# ohjelmointitaidot, kuin saa lukemalla näiden sivujen C# Perusoppaan.

2: Alkuun

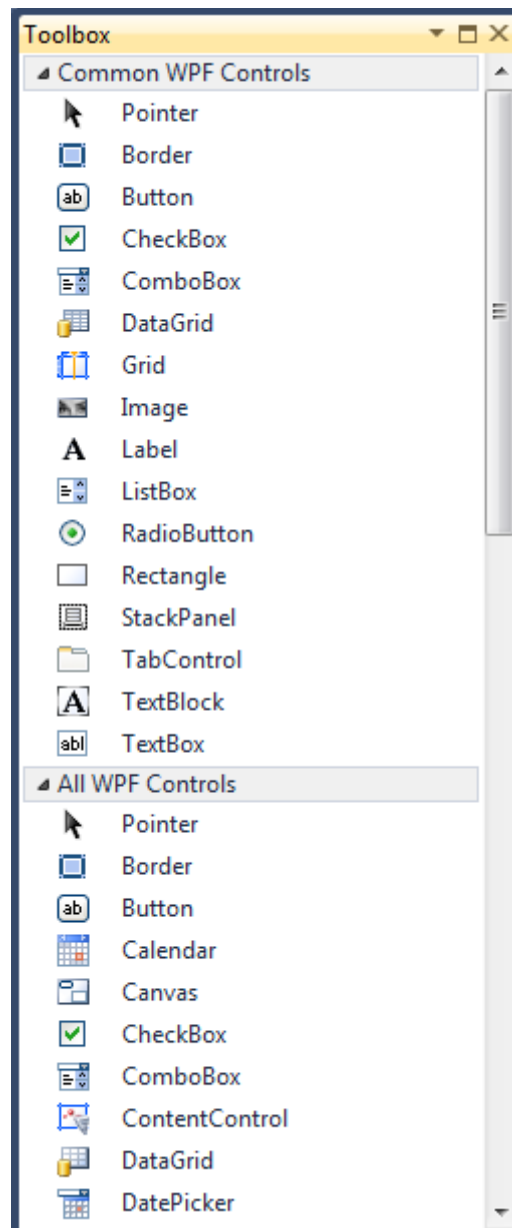
Käynnistä Visual Studio. Valitse ”new project”. Valitse projektimuodoksi ”WPF Application”. Ja anna nimeksi haluamasi nimi. Valitsen itse tässä nimeksi ”WinHarj”. Huomaa, että projektin nimeä on jälkepäin hyvin vaikea muuttaa, ja että sitä käytetään koko ajan ohjelmoinnissa.

Saat seuraavanlaisen näkymän:



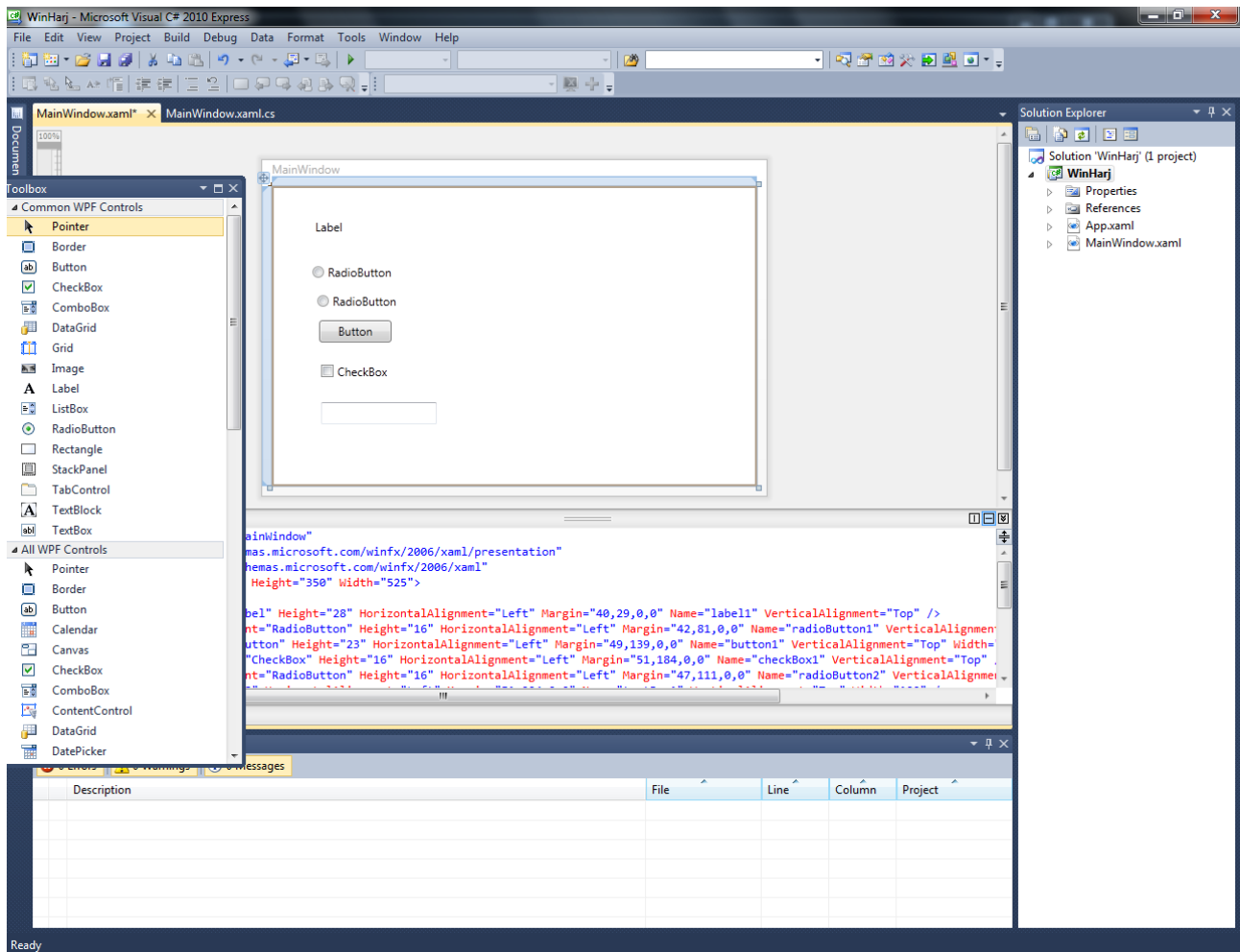
Jos työkalupalkki (Toolbox) ei ole vasemmalla näkyvillä, valitse oikealta ylhäältä kahden ristikkäisen vasaran kuva klikkaamalla.

Toolbox näyttää seuraavanlaiselta:



Tarkista, että sinulla on aktiivisena ”MainWindow.xaml”. Voit nyt klikata Toolboxissa eri valintoja, ja klikkamalla sen jälkeen jotain kohtaa MainWindow ikkunassa, elementti tulee siihen kohtaan.

Valitse esim. Label, kaksi RadioButtonia, Button ja CheckBox. Sen jälkeen sinulla on seuraava näyttö:



Nämä ovat eniten käytettyjä toimintoja. Seuraavassa esittely:

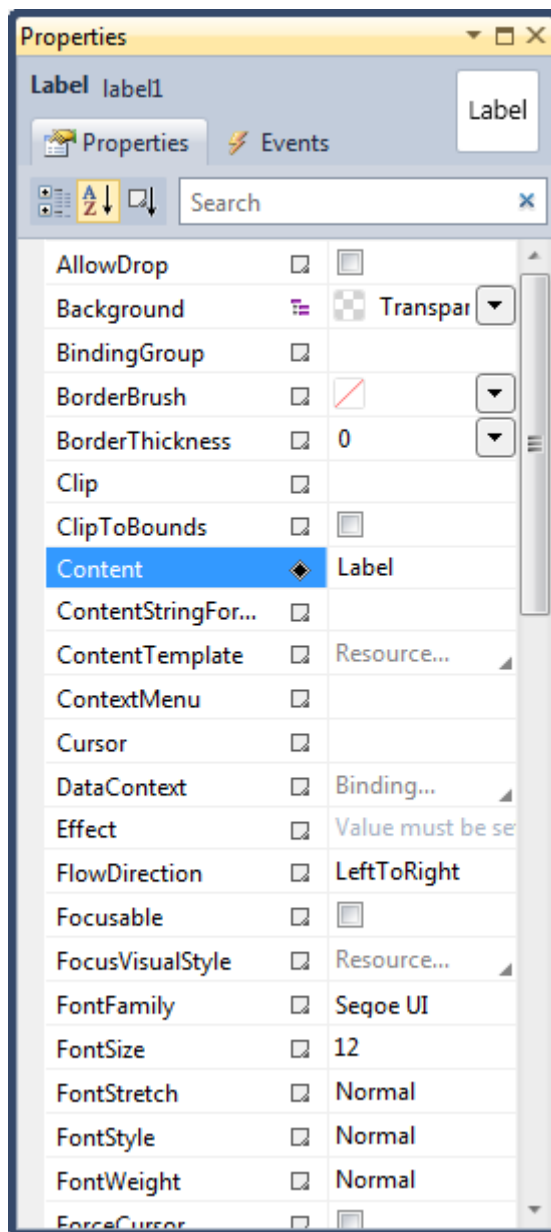
-Label on tekstikenttä, mihin voit kirjoittaa valmiiksi tekstiä, jota ei lueta ohjelman ajon aikana, vaan se on vain näkyvillä.

-RadioButton on valintanappi, jossa vain yksi valinnoista voi kerrallaan olla valittuna.

-CheckBox on valintaruutu. Jos on useita valintaruutuja, niistä voi olla valittuna joko yksi tai useampi valinta, tai ei yhtään.

-Button on toimintonäppäin, jonka painalluksesta suoritetaan koodi, eli esim. lasketaan tai tulostetaan valintojen perusteella jotain.

Klikkaa nyt oikealta ylhäältä ”Properties Window” pikakuvaketta. Klikkaa sen jälkeen näytölle lisäämääsi Label elementtiä. Saat seuraavan näkymän:



Muuta nyt Properties ikkunassa ”Content” kohtaa. Enterin painalluksen jälkeen kirjoittamasi teksti tulee näytölle. Saman voit tehdä Radio Buttoneille, Buttonille ja CheckBoxille. TextBoxilla ei ole tätä toimintoa, eikä siihen yleensä kirjoiteta oletukseksi mitään, vaan jätetään tyhjäksi, mutta voit tietenkin sen ”Text” kohtaan Properties ikkunassa kirjoittaa valmiiksi oletustekstin. Voit siirellä elementtejä MainWindow ikkunassa hiirellä siirtämällä. Voit myös klikata MainWindowsin nimeä, ja muuttaa ”Title” kohdasta nimeksi, joka näkyy ylälaidassa esim. ”Testiohjelma.”

3: Numeroiden tulostus ja lukeminen Windows ohjelmassa

Yleensä ohjelmassa syötetään pyydetty luvut TextBoxissa, ja samoiten tulokset tulostetaan TextBoxiin. TextBox ei kuitenkaan osaa käsitellä numeroita, vaan ainoastaan merkkijonoja. Siksi luettaessa numeroita, tai tulostettaessa niitä on tehtävä tyyppimuunnos.

Seuraavalla lauseella voi muuttaa double tyyppin muuttujan ”luku1” TextBoxiin tulostettavaan muotoon merkkijonoksi:

```
double luku1;  
luku1.ToString();
```

Taasen luettaessa TextBoxista numeroita, ne ovat oletusarvoisesti merkkijonoja, eli ennen laskutoimituksia ne on muutettava numeroiksi. Seuraavassa muutetaan merkkijonomuuttuja luku2 tyyppimuuksella double luvuksi luk2;

```
double luk2;  
luk2 = Double.Parse(luku2.Text);
```

4: Harjoitusohjelma Eurolaskuri

Tehdään seuraavaksi ensimmäinen ihan kunnan Windows ohjelma Eurolaskuri. Avaa uusi WPF Application sovellus nimellä ”Eurolaskuri” (new project).

Anna ”MainWindow” ikkunalle Properties ikkunan Title kohdassa nimeksi ”Eurolaskuri”.

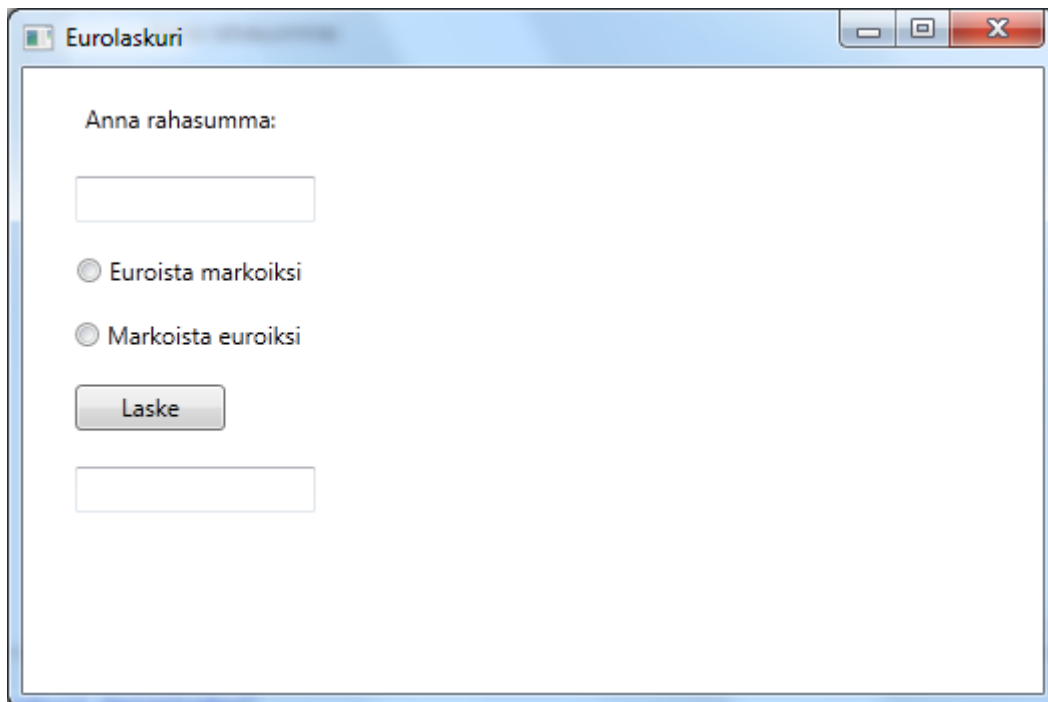
Laita ikkunan ylälaitaan Label elementti ja muuta Content kohtaan arvoksi ”Anna rahasumma:”.

Lisää ikkunaan TextBox elementti. Anna Properties ikkunassa ylhäältä ”TextBox” sanan jälkeen kenttään, joka aktivoituu klikkaamalla kentän nimeä, nimeksi ”luku1”. Lisää kaksi Radio Buttonia, nimeä ne kuten TextBox:kin nimillä ensimmäinen ”euro” ja toinen ”markka”, ja anna niille Properties ikkunan Content kohdassa tekstiksi ”Euroista markoiksi” (euro) ja ”Markoista euroiksi” (markka).

Lisää sitten yksi Button. Sitä ei yleensä nimetä erikseen, vaan käytetään ohjelmassa ”button1” nimellä. Anna kuitenkin Content kohdassa tekstiksi ”Laske”.

Lisää sitten vielä alle yksi TextBox, ja anna sille nimeksi ylhäältä Properties ikkunassa ”Tulos”.

Voit nyt ajaa ohjelman ”Debug” valikosta komennolla ”Start Debugging” (tai F5). Sinulle aukeaa seuraava ikkuna.



Ohjelma toimii ja pyörii jo, mutta ei tee mitään, koska siihen ei ole vielä lisätty koodia. Lisätää seuraavaksi koodi.

Lopeta juuri avatun ohjelman ajo ruksista oikealta ylhäältä, ja siirry MainWindow.xaml ikkunaan. Tuplaklikkaa ”Laske” painiketta. Ohjelma siirtyy MainWindow.xaml.cs ikkunaan, jossa voi kirjoittaa koodia.

Koodi, joka aukeaa, näyttää seuraavalta:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace Eurolaskuri
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, RoutedEventArgs e)
        {

        }
    }
}
```

Lisää nyt koodia button1_Click lohkokon (metodiin). button1_Click muokattuna näyttää seuraavalta:

```
private void button1_Click(object sender, RoutedEventArgs e)
{
    double luku_uusi1, muunnos = 0; //huomaa, että joudut antamaan muunnos
    //muuttujalle oletusarvon
    luku_uusi1 = Double.Parse(luku1.Text); //muutetaan teksti double numeroksi

    //Tarkistetaan kumpi Radio Button on valittuna, ja suoritetaan lasku
    if ((bool)euro.IsChecked) muunnos = luku_uusi1 * 5.94573;
    if ((bool)markka.IsChecked) muunnos = luku_uusi1 / 5.94573;

    Tulos.Text = muunnos.ToString(); //tulostetaan vastaus
}
```

Aja ohjelma, ja kokeile syöttää eri lukuja. Ohjelma toimii muuten, mutta kaatuu, jos syötät numeron sijaan kirjaimen. Ohjelmaan on syytä tehdä aina kaikki mahdolliset tarkistukset. Kun suuria ohjelmia tehdään, niille tehdään vaikka mitä tarkistuksia, kuten esim. katkaistaan tietokoneesta virta kesken ohjelman ajon jne. Kun laskutoimitukset laitetaan try catch lohko väliin, niissä tapahtuvat virheet eivät kaada ohjelmaa.

Muokataan `button1_Click` metodi seuraavaksi:

```
private void button1_Click(object sender, RoutedEventArgs e)
{
    double luku_uusi1, muunnos=0; //huomaa, että joudut antamaan muunnos
    //muuttujalle oletusarvon

    try
    {
        luku_uusi1 = Double.Parse(luku1.Text); //muutetaan merkkijono tekstiksi

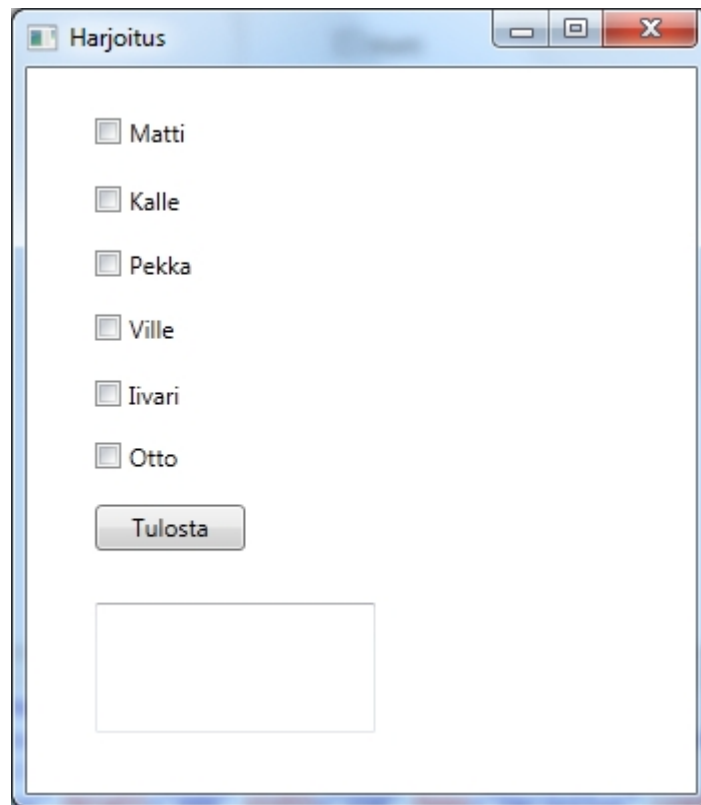
        //Tarkistetaan kumpi Radio Button on valittuna, ja suoritetaan lasku
        if ((bool)euro.IsChecked) muunnos = luku_uusi1 * 5.94573;
        if ((bool)markka.IsChecked) muunnos = luku_uusi1 / 5.94573;
    }
    catch {}

    Tulos.Text= muunnos.ToString(); //tulostetaan vastaus
}
```

Nyt kun ajat ohjelman, sille voi syöttää mitä haluaa, ja ohjelman ei pitäisi kaatua.

5: Valintaruutujen käyttö, ja tekstikentän vierittäminen

Avaa uusi projekti. Lisää ikkunaan kuusi CheckBox:ia Toolbox valikosta. Lisää Button ja TextBox. Anna CheckBoxeille arvot Content osiossa, kuten alla. Jätä silti Properties valikon yläreunassa nimet muuttamatta. Muuta ikkunan koko Height kohtaan 400 ja Width kohtaan 350 Properties ikkunasta ikkunassa, jossa voi myös nimetä ikkunan (Window), eli joka aktivoituu aivan MainWindow paneelin yläreunasta klikkaamalla. Nimeä ikkuna ”Title” kohdasta nimellä ”Harjoitus”.



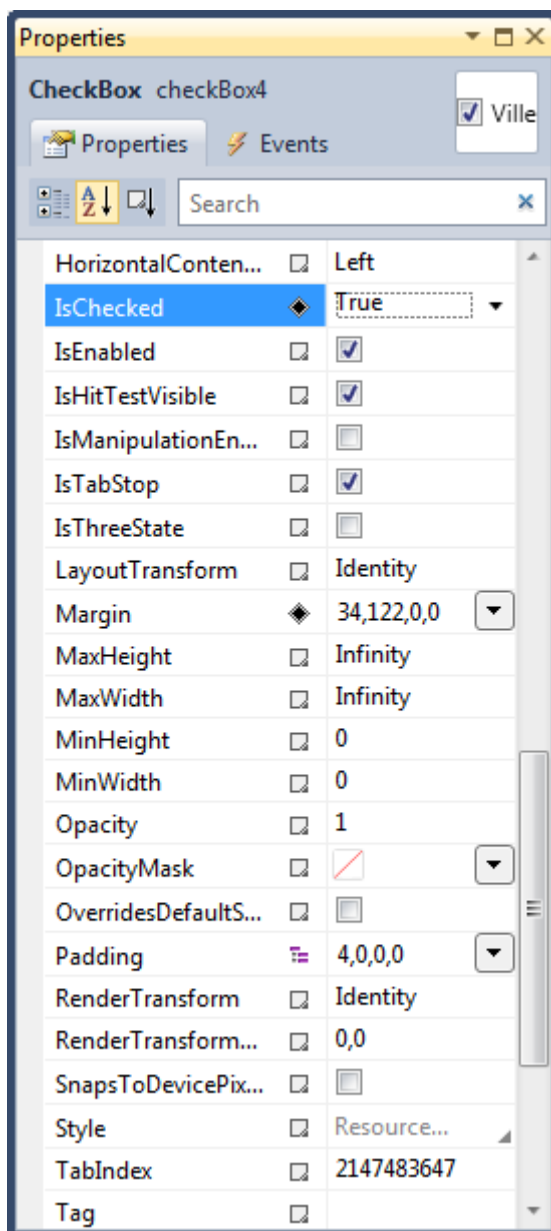
Muuta TextBox:in Height ominaisuudeksi 65 ja Width ominaisuudeksi 140. Aktivoi VerticalScrollBarVisibility kohta valitsemalla toiminto ”Auto”. Nyt käyttöön tulee tarvittaessa pystysuuntainen vieritysikkuna. Vastaavasti saat vaakasuuntaisen HorizontalScrollBarVisibility kohdasta, mutta sitä ei tarvita tässä ohjelmassa.

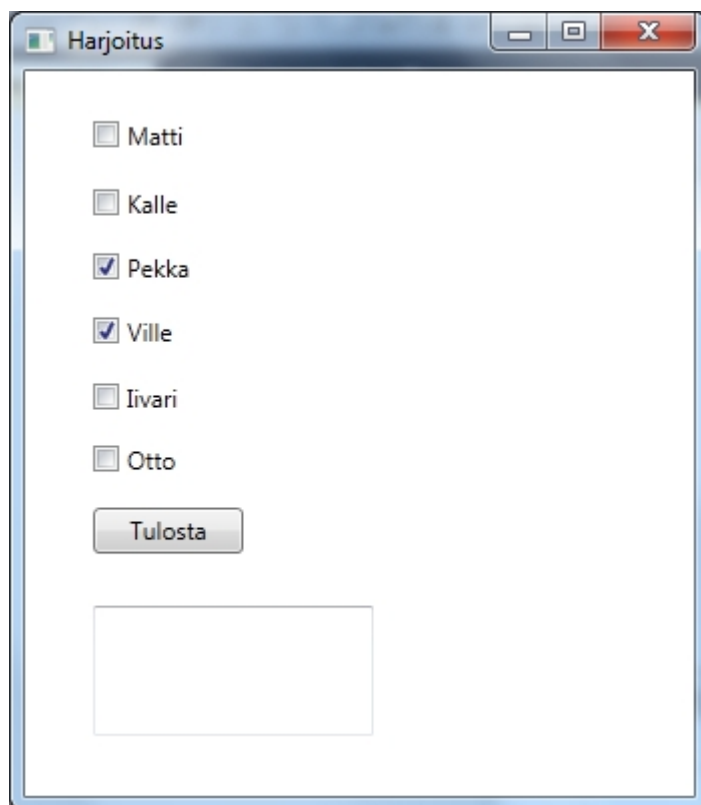
Muuta nyt button1_Click metodi seuraavaksi:

```
private void button1_Click(object sender, RoutedEventArgs e)
{
    string tulos=""; //huomaa, että joudutaan antamaan alkuarvo
    if ((bool)checkBox1.IsChecked) tulos = "Matti\n";
    //huomaa perässä rivinvaihto \n
    if ((bool)checkBox2.IsChecked) tulos = tulos + "Kalle\n";
    //+ operaattorilla voi lisätä merkkijonoon
    if ((bool)checkBox3.IsChecked) tulos = tulos + "Pekka\n";
    if ((bool)checkBox4.IsChecked) tulos = tulos + "Ville\n";
    if ((bool)checkBox5.IsChecked) tulos = tulos + "Iivari\n";
    if ((bool)checkBox6.IsChecked) tulos = tulos + "Otto";

    Tulostus.Text = tulos; //tulostetaan nimet
}
```

Properties ikkunassa IsChecked kohdasta voi valita oletusarvona ”True”, jos haluaa, että CheckBox on jo valmiiksi oletusarvoisesti valittu.





Tässä on valittu oletusarvoiksi ”Ville” ja ”Pekka”, jotka tulevat näyttöön heti ohjelmaa käynnistettäessä. Samoin Radio Buttonin voi asettaa oletusarvoiseksi.