

# Windows Phone ohjelmointi perusteet

Paavo Räisänen

[www.ohjelmoimaan.com](http://www.ohjelmoimaan.com)

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.

## 1: Alkusanat

Microsoft tarjoaa Windows Phone ohjelmoijalle ilmaiset kehitystyökalut, eli Visual Studio IDE:n (koodieditori, käyttöliittymien suunnittelutyökalu ja debuggeri), Expression Blend:in (työkalu, jolla suunnitellaan Silverlight sovelluksia) ja Windows Phone emulaattorin. Ohjelmointi tapahtuu yleensä C# ohjelmointikielellä, ja tässä oppaassa oletetaan, että lukija osaa näillä sivuilla olevien oppaiden ”C# Perusopas” ja ”C# Windows ohjelmointi perusopas” sisältämät asiat, sekä ”C# olio-ohjelmointi perusopas” oppaan sisällöstä ainakin käsitteet olio, metodi ja luokka. Windows Phone (WP) ohjelmointi on .NET pohjaista Silverlight ja XNA (pelit) ohjelmointia.

Ohjelmien lataus puhelimeen tapahtuu Windows Phone Marketplace sovelluskaupan kautta. Marketplace maksaa 99 \$ vuodessa. Hintaan voi laittaa jakeluun rajattoman määrän maksullisia sovelluksia ja enintään sata kappaletta maksuttomia sovelluksia. Marketplacen kautta on helppo kaupallistaa oma ohjelma. Tuotosta saa itselle 70 % ja 30 % menee Microsoftille. Jos käyttää Microsoftin mainontapalveluja, on mahdollista saada myös mainostuloja.

## 2: Kehitysympäristön lataus

Lataa ensin Microsoft Visual Studio 2010 Express ohjelma osoitteesta <http://www.microsoft.com/express/Downloads/>. Windows Phone SDK, joka asentuu Visual Studioon on ladattavissa osoitteessa <http://dev.windowsphone.com/en-us/downloadsdk>. Huomioitavaa on, että Windows 8 SDK asettaa koneelle kovia vaatimuksia, joten tässä oppaassa käytetään vanhempaa versiota, jonka pitäisi Windows 7 koneissa toimia. Lataa ensin versio SDK 7.1 ja sitten päivitys SDK 7.1.1.

## 3: Emulaattori

Sovellusta voidaan ajaa PC:n emulaattorissa. Emulaattori emuloi myös kameraa, GPS:ää ja liiketunnistimia. Emulaattori saattaa olla paljon kännykkää tehokkaampi, joten ohjelmat täytyy ennen myyntiin laittoa testata Windows puhelimella. Emulaattorissa voi pause/break näppäimellä (tietokoneen mäppäimistöltä) valita, kirjoittaako emulaattoriin emulaattorin vai tietokoneen näppäimistöltä. Oletusarvoisesti ääkköset (ä,ö,å) eivät ole käytössä. Voit kyllä ohjelmallisesti tulostaa ääkkösiä, mutta emulaattorin näppäimistöllä niitä ei ole, etkä ohjelman ajon aikana pysty syöttämään ääkkösiä. Näin ainakin minun koneella.

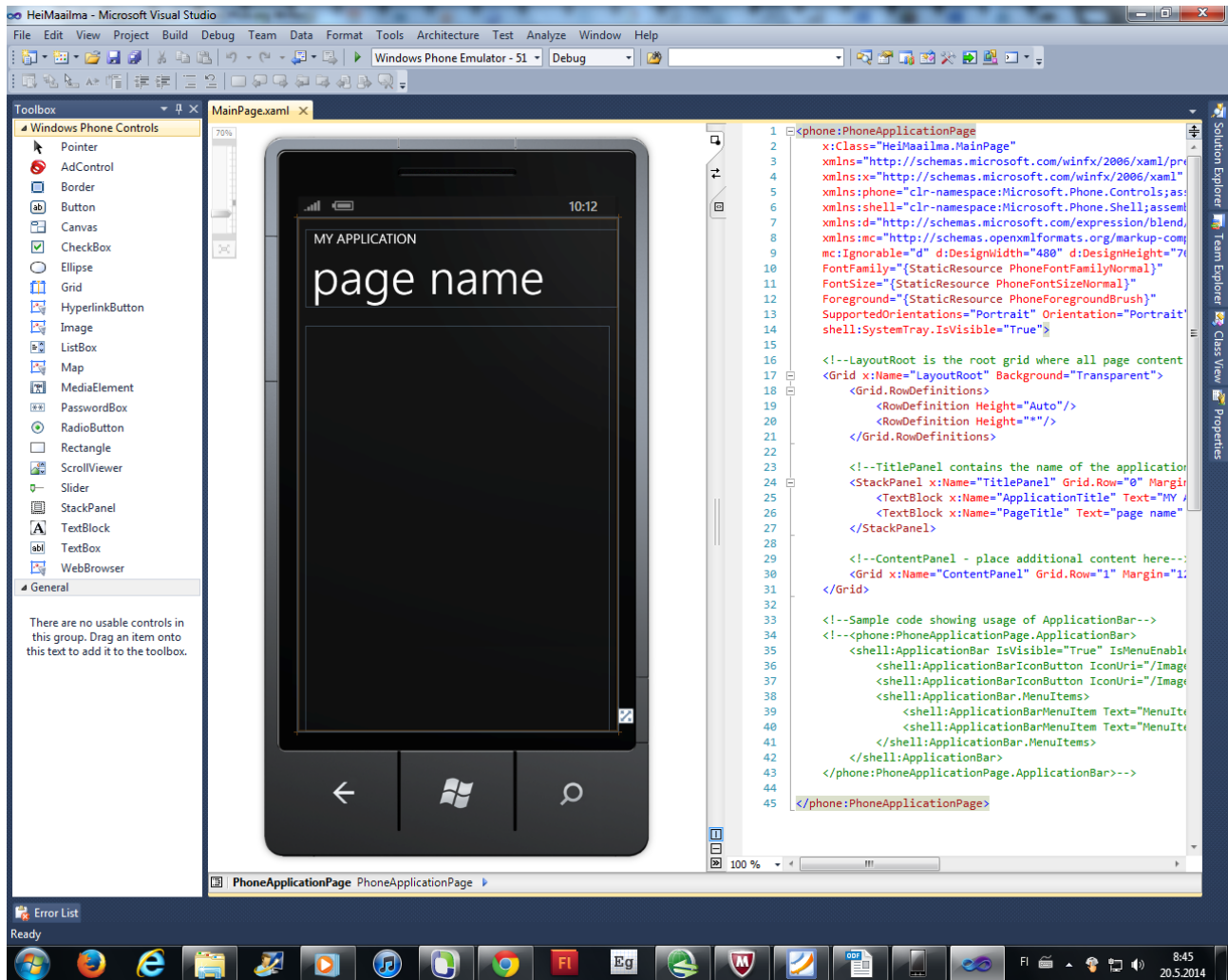
Emulaattorissa ”Additional Tools” työkalu aukeaa kun viet hiiren Emulaattorin oikean yläkulman viereen ”>>” toiminnolla. Siinä voi ”Screenshot” toiminnossa ottaa kuvakaappauksen, jota voi käyttää vaikka ohjelman markkinoinnissa Marketplaceissa. Kuva otetaan klikkaamalla ”Additional Tools” työkalussa ensin ”Capture” ja sitten sitten kuva tallennetaan tietokoneeseen ”Save” toiminnolla.

Kun viet hiiren Emulaattorin oikean yläkulman viereen, aukeaa myös muita toimintoja. Ylimmärtä toiminnosta voi kääntää emulaattorin vaaka-asentoon. Toiseksi ylin toiminto palauttaa emulaattorin koon oletusasentoon, ja sen alla olevasta toiminnosta voi säätää emulaattorin koon haluamukseen.

## 4: Ensimmäinen ohjelma ”Hello World”

Käynnistä Visual Studio. Klikkaa ”Start page” ja ”New Project”. Valitse ”Visual C#” valikosta ”Silverlight for Windows Phone”. Klikkaa kerran ”Windows Phone Application” kohtaa ja anna projektille nimi ”HeiMaaailma”. Sitten ”Ok”. Avautuvasta ikkunasta valitaan ”Windows Phone OS 7.1” ja ”Ok”.

Aukeaa seuraava näkymä:



Mikäli ”Toolbox” ei ole väkyvillä vasemmalla, klikkaa ”View/ Toolbox” ja kiinnitä Toolbox näkymään aina Toolboxin oikealta yläkulmasta ”Auto hide” toiminnolla (keskimmäinen ikoni).

Valitse Toolboxista ”Button” elementti, ja raahaa se emulaattori näytölle, tai klikkaa elementtiä, ja sen jälkeen emulaattorin kohtaa, mihin sen haluat. Siirrä myös ”TextBlock” elementti.

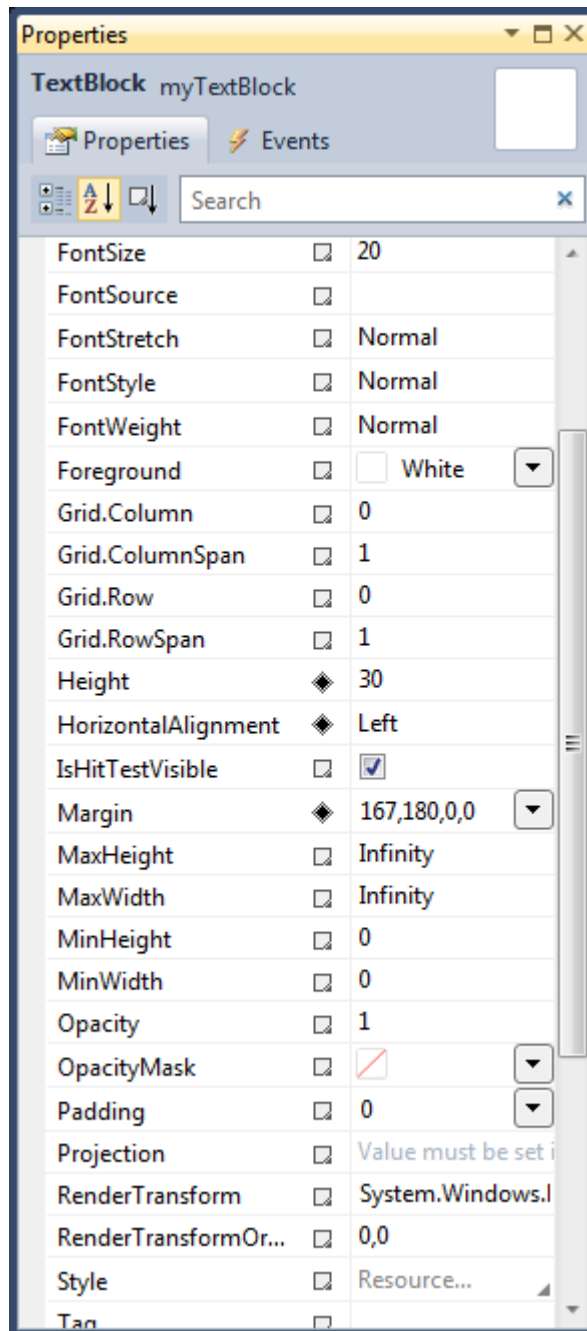
Näyttö on seuraava:

The screenshot displays the Microsoft Visual Studio IDE with a Windows Phone emulator running. The emulator shows a mobile application interface with a title bar, a status bar at the top displaying '10:12', and a main content area. The content area contains a title 'MY APPLICATION', a subtitle 'page name', a 'Button', and a 'TextBlock'. The status bar at the bottom of the emulator shows navigation icons: back, Windows logo, and search.

The XAML code in the Solution Explorer is as follows:

```
1 <phone:PhoneApplicationPage
2   x:Class="HeiMaailma.MainPage"
3   xmlns="http://schemas.microsoft.com/winfx/2006/xaml/pr
4   xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
5   xmlns:phone="clr-namespace:Microsoft.Phone.Controls;as
6   xmlns:shell="clr-namespace:Microsoft.Phone.Shell;assem
7   xmlns:d="http://schemas.microsoft.com/expression/blend
8   xmlns:mc="http://schemas.openxmlformats.org/markup-com
9   mc:Ignorable="d" d:DesignWidth="480" d:DesignHeight="720"
10  FontFamily="{StaticResource PhoneFontFamilyNormal}"
11  FontSize="{StaticResource PhoneFontSizeNormal}"
12  Foreground="{StaticResource PhoneForegroundBrush}"
13  SupportedOrientations="Portrait" Orientation="Portrait"
14  shell:SystemTray.IsVisible="True" Name="MyTextBlock">
15
16  <!--LayoutRoot is the root grid where all page content
17  <Grid x:Name="LayoutRoot" Background="Transparent">
18    <Grid.RowDefinitions>
19      <RowDefinition Height="Auto"/>
20      <RowDefinition Height="*"/>
21    </Grid.RowDefinitions>
22
23    <!--TitlePanel contains the name of the application
24    <StackPanel x:Name="TitlePanel" Grid.Row="0" Margi
25      <TextBlock x:Name="ApplicationTitle" Text="MY A
26      <TextBlock x:Name="PageTitle" Text="page name"
27    </StackPanel>
28
29    <!--ContentPanel - place additional content here--
30    <Grid x:Name="ContentPanel" Grid.Row="1" Margin="1
31      <Button Content="Button" Height="72" Horizontal
32      <TextBlock Height="30" HorizontalAlignment="Le
33    </Grid>
34  </Grid>
35
36  <!--Sample code showing usage of ApplicationBar-->
37  <!--<phone:PhoneApplicationPage.ApplicationBar>
38    <shell:ApplicationBar IsVisible="True" IsMenuEnabl
39    <shell:ApplicationBarIconButton IconUri="/Imag
40    <shell:ApplicationBarIconButton IconUri="/Imag
41    <shell:ApplicationBar.MenuItems>
42      <shell:ApplicationBarMenuItem Text="MenuIt
43      <shell:ApplicationBarMenuItem Text="MenuIt
44    </shell:ApplicationBar.MenuItems>
45  </shell:ApplicationBar>
46  </phone:PhoneApplicationPage.ApplicationBar-->
47
48 </phone:PhoneApplicationPage>
```

Mikäli "Properties" ikkuna ei ole näkyvässä, klikkaa "View/Properties Window". Muuta TextBlockin nimeksi "myTextBlock". Katso, että ikkunassa on "Properties" välilehti valittuna. Properties näyttö on seuraava. Ihan ylhäällä "TextBlock" sanan vieressä on nimi, jota usein, kuten nyt, muutetaan.



Vieritä "Properties" ikkunassa "Text" näkyviin (jos ei jo ole) ja tyhjennä "Text" kentän sisältö. Teksti "TextBlock" näviää emulaattorista. Klikkaa nyt kerran "Button" painiketta, ja anna nimeksi "myButton". Kirjoita sitten "Content" kohtaan "Click Me".

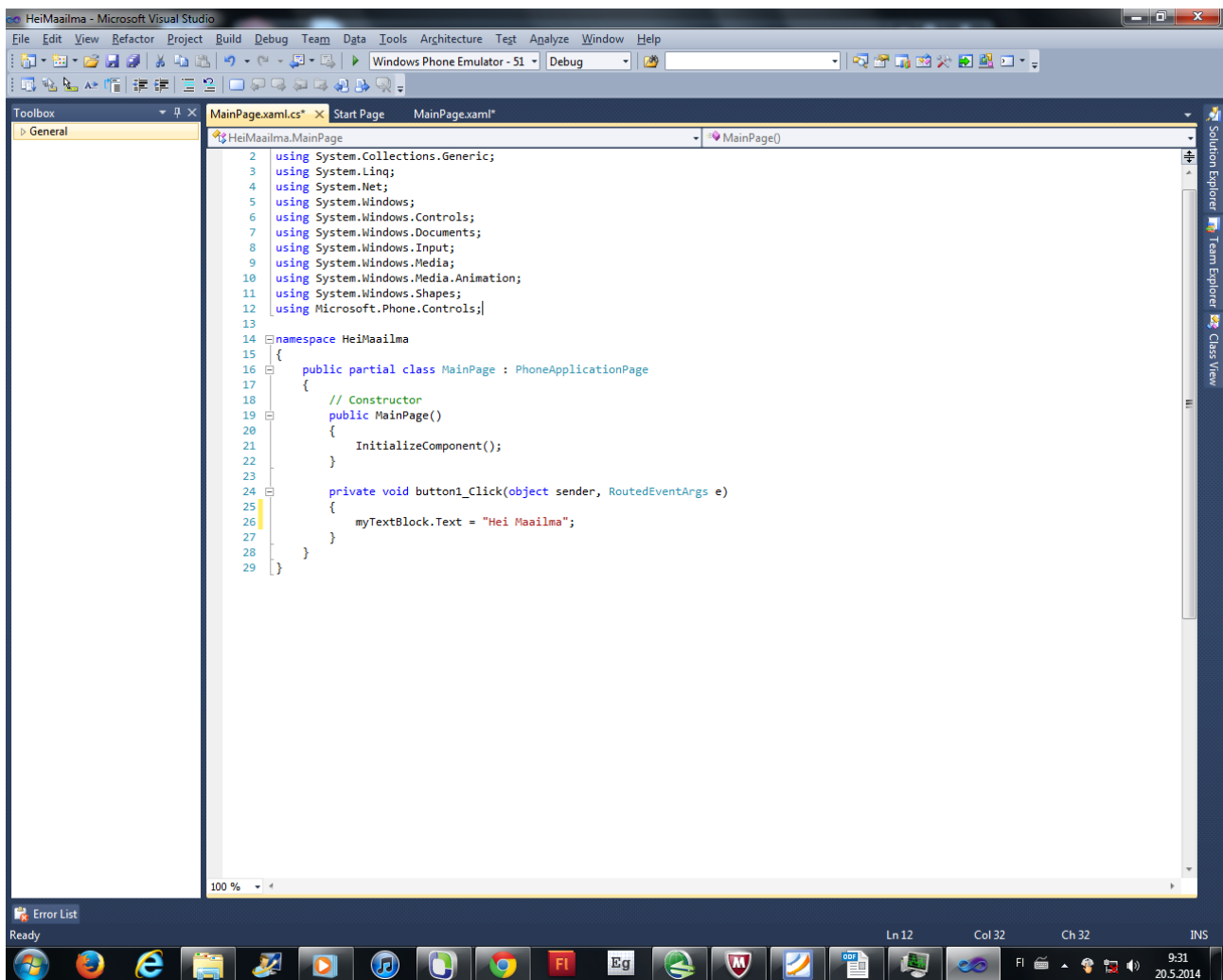
Klikkaa nyt emulaattorissa ”Click Me” painiketta. Avautuu seuraava C# koodi, jonne voi

```
private void button1_Click(object sender, RoutedEventArgs e)
{
}
```

kohtaan kirjoittaa, mitä painiketta painettaessa tapahtuu. Kirjoita aaltosulkujen väliin

```
private void button1_Click(object sender, RoutedEventArgs e)
{
    myTextBlock.Text = "Hei Maailma";
}
```

Koko C# koodi on nyt seuraava:



```
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Net;
5 using System.Windows;
6 using System.Windows.Controls;
7 using System.Windows.Documents;
8 using System.Windows.Input;
9 using System.Windows.Media;
10 using System.Windows.Media.Animation;
11 using System.Windows.Shapes;
12 using Microsoft.Phone.Controls;
13
14 namespace HeiMaailma
15 {
16     public partial class MainPage : PhoneApplicationPage
17     {
18         // Constructor
19         public MainPage()
20         {
21             InitializeComponent();
22         }
23
24         private void button1_Click(object sender, RoutedEventArgs e)
25         {
26             myTextBlock.Text = "Hei Maailma";
27         }
28     }
29 }
```

Huomaathan automaattisen tietojen syötön (IntelliSense). Se auttaa valitsemaan, mitä seuraavaksi kirjoitetaan. IntelliSense nopeuttaa kirjoittamista ja vähentää virheitä.

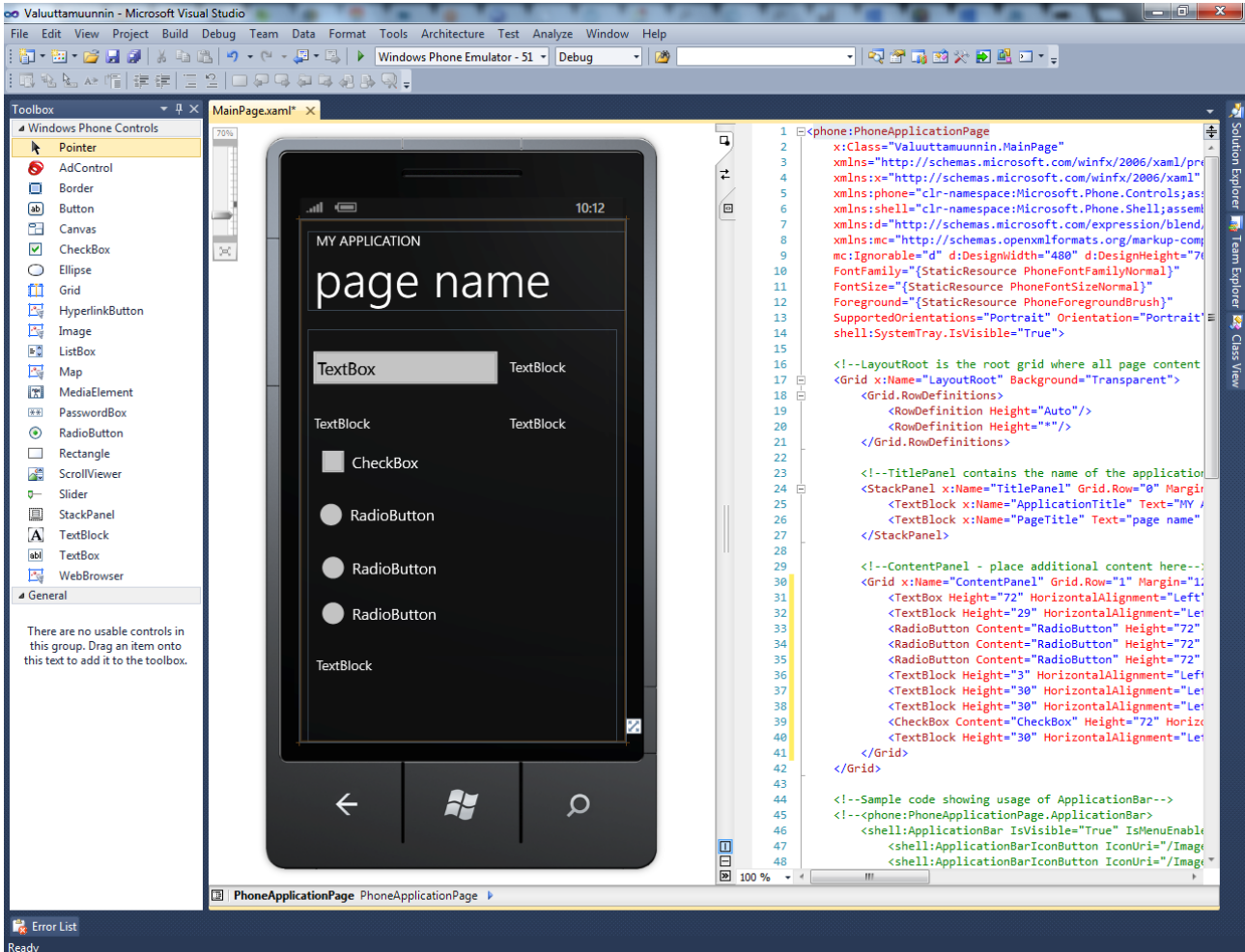
Ohjelma on valmis. Talleta kaikki ”File/Save all” toiminnolla ja aja se ”Debug/Start debugging” toiminolla. Voit myös ajaa ohjelman vihreästä nuolesta, joka on yksi pikavalinta ylhäällä olevissa pikatoiminnoissa.

Ajettaessa näytölle tulee ”Click me” painike, jota painettaessa ilmestyy teksti ”Hei Maailma”. Ohjelman ajo emulaattorissa pysähtyy ”Debug/ Stop debugging” toiminnolla tai ajon voi lopettaa emulaattorin ”Back” näppäimellä (nuoli vasemmalla alhaalla).

#### 4: Toinen ohjelma, Valuuttamuunnin

Kun olet tallentanut edellisen työn, klikkaa ”File/Close Solution” tai sulje ikkuna, ja avaa ohjelma uudestaan. Tee uusi projekti nimellä ”Valuuttamuunnin”.

Raahaa näytölle seuraavan kuvan mukaiset elementit, eli yksi TextBox, neljä TextBlockia, yksi CheckBox ja kolme RadioButtonia:



Huomaat, että samalla viereiseen XAML koodieditoriin on tullut lisää rivejä, eli:

```
<!--ContentPanel - place additional content here-->
<Grid x:Name="ContentPanel" Grid.Row="1" Margin="12,0,12,0">
  <TextBox Height="72" HorizontalAlignment="Left" Margin="-4,21,0,0"
Name="textBox1" Text="TextBox" VerticalAlignment="Top" Width="295" />
  <TextBlock Height="29" HorizontalAlignment="Left" Margin="9,124,0,0"
Name="textBlock1" Text="TextBlock" VerticalAlignment="Top" Width="260" />
  <RadioButton Content="RadioButton" Height="72" HorizontalAlignment="Left"
Margin="6,237,0,0" Name="radioButton1" VerticalAlignment="Top" />
  <RadioButton Content="RadioButton" Height="72" HorizontalAlignment="Left"
Margin="9,315,0,0" Name="radioButton2" VerticalAlignment="Top" />
  <RadioButton Content="RadioButton" Height="72" HorizontalAlignment="Left"
Margin="9,382,0,0" Name="radioButton3" VerticalAlignment="Top" />
  <TextBlock Height="3" HorizontalAlignment="Left" Margin="29,451,0,0"
Name="textBlock2" Text="TextBlock" VerticalAlignment="Top" Width="262" />
  <TextBlock Height="30" HorizontalAlignment="Left" Margin="297,42,0,0"
Name="textBlock3" Text="TextBlock" VerticalAlignment="Top" />
  <TextBlock Height="30" HorizontalAlignment="Left" Margin="297,124,0,0"
Name="textBlock4" Text="TextBlock" VerticalAlignment="Top" />
```

```

        <CheckBox Content="CheckBox" Height="72" HorizontalAlignment="Left"
Margin="9,159,0,0" Name="checkBox1" VerticalAlignment="Top" />
        <TextBlock Height="30" HorizontalAlignment="Left" Margin="12,480,0,0"
Name="textBlock5" Text="TextBlock" VerticalAlignment="Top" />
    </Grid>
</Grid>

```

Sitä mukaa kuin muuttelet graafista näkymää, XAML koodi muuttuu. Periaatteessa voit jopa tehdä näkymän XAML koodin avulla. Jos otat copy/paste toiminnolla ylläolevan koodin, ja laitat sen XAML ikkunaan, näkyviin tulee juuri ylläolevat napit ja tekstikentät. Yleensä vain ohjelma tehdään kopioimatta mistään, ja niinpä painikkeiden siiro graafisesti on helpoin tapa.

#### 4.1: XAML koodista

Margin arvo kertoo marginaalin koon elementin ympärillä (x,y,x2,y2). Niitä ei aina tarvitse muuttaa, vaan ne asettuvat lisättäessä elementtejä graafisesti.

Name on sinun komponentille antama nimi. Jolla C# koodi voi ohjelmallisesti käsitellä elementtiä ja Text on sinun antama nimi, joka näkyy käyttäjälle puhelimen (tai emulaattorin) näytöllä.

HorizontalAlignment ja VerticalAlignment toimiolla voidaan muuttaa kontrollia ryhmittymistä vaaka- ja pystysuunnassa.

Height komennolla pystyy muuttamaan elementi korkeutta ja Width komennolla elementin leveyttä. Näitä muutetaan kuitenkin useinmiten graafisesti ”vetämällä” elementin vasemmasta alareunasta.

#### 4.2: Tekstin vieritys

Joskus tarvitaan TextBlockin tai TextBoxin tekstiä vierittävä vieritinpalkki. Sen saa ilmaantumaan Properties valikosta ”TextWrapping” komennon avulla. Komennon ollessa ”NoWrapping” (oletus), kyseisen kentän tekstiä ei vieritetä ja komennon ollessa ”Wrap” vieritinpalkki tarvittaessa ilmaantuu.

#### 4.3: Näytön muokkausta

Ota ylimmän TextBoxin teksti pois Properties ikkunassa, ja anna TextBoxille nimi ”euroTextBox”. Muuta laatikon päässä olevan TextBlockin teksti tekstiksi ”Eurot”. Ota teksti pois TextBoxin alla olevasta TextBlockista ja anna sille nimi ”markkoinaTextBlock”. Muuta sen perässä olevan TextBlockin nimeksi ”mk”.

Muuta sitten CheckBoxin nimeksi ”myCheckBox1” ja kirjoita sille teksti Content kohtaan ”Ilmoita myös markkoina”.

Kirjoita sitten radioButtononeihin ylhäältä alas teksteiksi ”Dollareina”, ”Puntina” ja ”Ruotsin Kruunuina”. Anna radioButtononeille vastaavasti nimet ”dollariRadioButton”, ”puntaRadioButton” ja ”Ruotsin Kruunuina”.

Huomaathan, että TextBlockeista tulee näytölle ”näkyttömiä”. Jos haluat muuttaa niiden tietoja, klikkaa vastaavaa kohtaa XAML koodissa, niin pääset muuttamaan sitä myös Properties ikkunassa. Voit myös siirtämisen helpottamiseksi XAML koodissa kirjoittaa TextBlockiin jotain (esim. ”nnnnnnnn”).

Lisää nyt alimman TextBlockin ylle yksi ”Button” elementti. Lisäyksen helpottamiseksi siirrä alinta TextBlockia hieman alaspäin. Anna Buttonille nimeksi ”myButton” ja Content arvoksi ”Laske”.

”Button” elementti on toki mahdollista lisätä myös samalla, kun lisää muutkin elementit ihan alussa.

Poista nyt teksti alimmasta TextBlockista ja anna nimi ”tulosTextBlock”.

#### 4.4: Ylävalikon muokkaus

Ylävalikossa on XAML koodi:

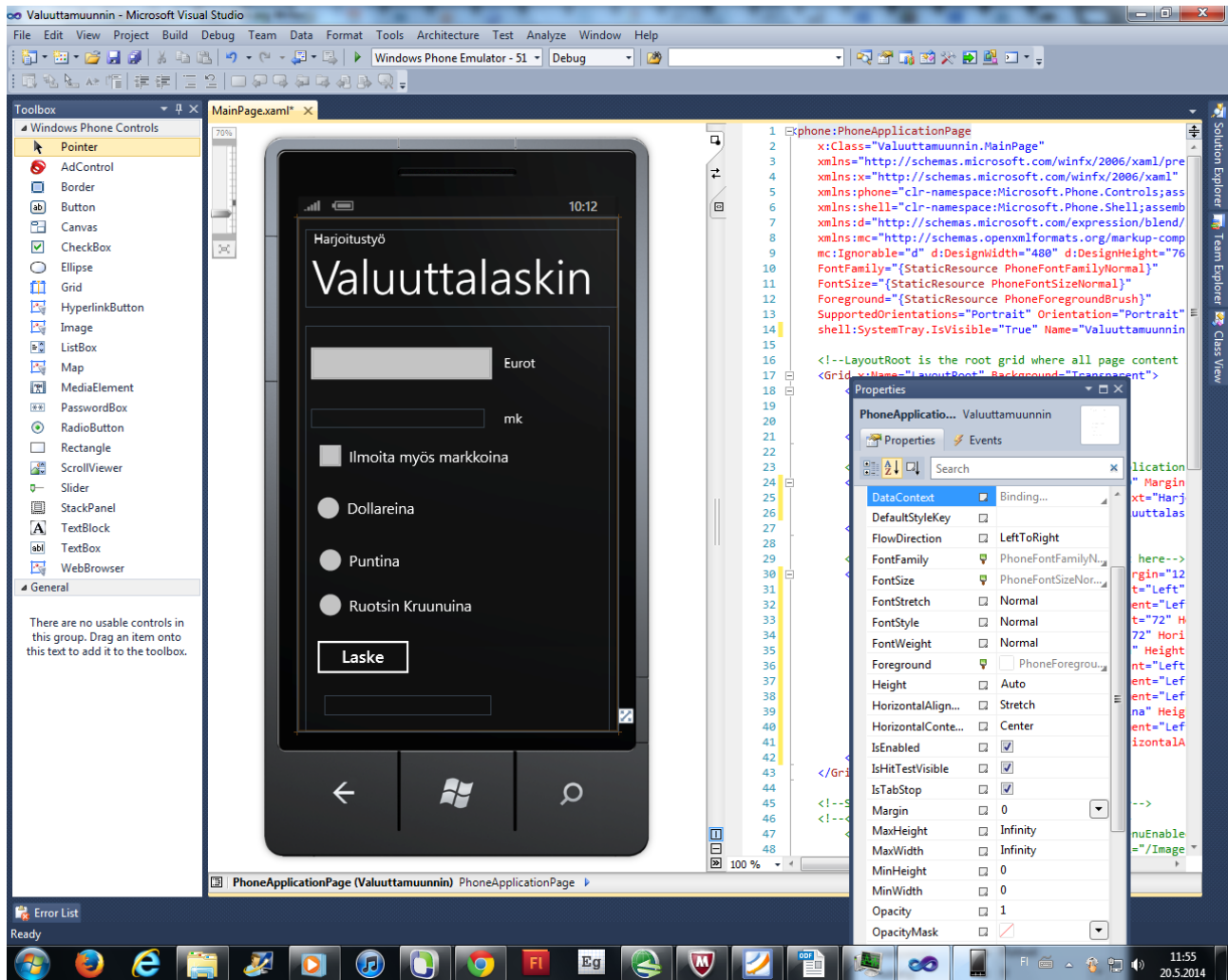
```
<StackPanel x:Name="TitlePanel" Grid.Row="0" Margin="12,17,0,28">
    <TextBlock x:Name="ApplicationTitle" Text="MY APPLICATION"
Style="{StaticResource PhoneTextNormalStyle}"/>
    <TextBlock x:Name="PageTitle" Text="page name" Margin="9,-7,0,0"
Style="{StaticResource PhoneTextTitle1Style}"/>
</StackPanel>
```

Muuta se seuraavanlaiseksi kirjoittamalla XAML koodiin:

```
    <StackPanel x:Name="TitlePanel" Grid.Row="0" Margin="12,17,0,28">
        <TextBlock x:Name="ApplicationTitle" Text="Harjoitustyö" Style="{StaticResource
PhoneTextNormalStyle}"/>
        <TextBlock x:Name="PageTitle" Text="Valuuttalaskin" Margin="9,-7,0,0"
Style="{StaticResource PhoneTextTitle1Style}"/>
    </StackPanel>
```

Eli ihan ylös tekstiksi ”Harjoitustyö” ja sen alle ylös tekstiksi ”Valuuttalaskin.

Sinulla pitäisi olla seuraava näkymä.



Ohjelman XAML koodi näiltä osin on seuraava:

```
<Grid x:Name="ContentPanel" Grid.Row="1" Margin="12,0,12,0">
    <TextBox Height="72" HorizontalAlignment="Left" Margin="-4,21,0,0"
Name="eurotTextBox" Text="" VerticalAlignment="Top" Width="295" TextWrapping="Wrap" />
    <TextBlock Height="29" HorizontalAlignment="Left" Margin="9,124,0,0"
Name="markkoinaTextBlock" Text="" VerticalAlignment="Top" Width="260" />
    <RadioButton Content="Dollareina" Height="72" HorizontalAlignment="Left"
Margin="6,237,0,0" Name="dollariRadioButton" VerticalAlignment="Top" />
    <RadioButton Content="Puntina" Height="72" HorizontalAlignment="Left"
Margin="9,315,0,0" Name="puntaRadioButton" VerticalAlignment="Top" />
    <RadioButton Content="Ruotsin Kruunuina" Height="72" HorizontalAlignment="Left"
Margin="9,382,0,0" Name="kruunuRadioButton" VerticalAlignment="Top" />
    <TextBlock Height="3" HorizontalAlignment="Left" Margin="29,451,0,0"
Name="textBlock2" Text="TextBlock" VerticalAlignment="Top" Width="262" />
    <TextBlock Height="30" HorizontalAlignment="Left" Margin="297,42,0,0"
Name="textBlock3" Text="Eurot" VerticalAlignment="Top" />
    <TextBlock Height="30" HorizontalAlignment="Left" Margin="297,124,0,0"
Name="textBlock4" Text="mk" VerticalAlignment="Top" />
    <CheckBox Content="Ilmoita myös markkoina" Height="72"
HorizontalAlignment="Left" Margin="9,159,0,0" Name="myCheckBox1" VerticalAlignment="Top" />
    <TextBlock Height="30" HorizontalAlignment="Left" Margin="29,552,0,0"
Name="tulosTextBlock" Text="" VerticalAlignment="Top" Width="249" />
    <Button Content="Laske" Height="72" HorizontalAlignment="Left"
```

```
Margin="6,460,0,0" Name="myButton" VerticalAlignment="Top" Width="160" />
</Grid>
```

#### 4.5: Oletusvalinnat

Lisätään oletusvalinnoiksi, että ”Ilmoita myös markkoina” valintaruutu on valittuna ja Radiobuttonista ”dollariRadioButton” on valmiiksi valittuna.

Muuta checkBoxin isChecked arvo arvoon ”true” Properties ikkunassa tai XAML koodissa seuraavasti:

```
<CheckBox Content="Ilmoita myös markkoina" Height="72" HorizontalAlignment="Left"
Margin="9,159,0,0" Name="myCheckBox1" VerticalAlignment="Top" IsChecked="True" />
```

Muuta ylimmän Radiobuttonin, eli ”dollariRadioButton”:in isChecked arvo arvoon ”true” Properties ikkunassa tai tee XAML koodissa seuraavasti:

```
<RadioButton Content="Dollareina" Height="72" HorizontalAlignment="Left" Margin="6,237,0,0"
Name="dollariRadioButton" VerticalAlignment="Top" IsChecked="True" />
```

#### 4.6: C# (CS) koodi

Tuplaklikkaa ”Laske” Buttonia. Aukeaa CS ikkuna. Muokkaa siellä myButton\_Click metodia seuraavaksi:

```
private void myButton_Click(object sender, RoutedEventArgs e)
{
    double summa;
    string valuutta;
    try
    {
        double euroMaara = double.Parse(eurotTextBlock.Text);
        if (dollariRadioButton.IsChecked == true)
        {
            summa = euroMaara * 1.37;
            valuutta = "dollaria";
        }
        else if (puntaRadioButton.IsChecked == true)
        {
            summa = euroMaara * 0.82;
            valuutta = "puntaa";
        }
        else
        {
            summa = euroMaara * 9.04;
            valuutta = "Ruotsin kruunua";
        }

        tulosTextBlock.Text = summa.ToString() + " " + valuutta;

        if (myCheckBox1.IsChecked == true) markkoinaTextBlock.Text =
(double.Parse(eurotTextBlock.Text) * 5.95).ToString();
        else markkoinaTextBlock.Text = "";
    }
    catch
    {
        MessageBox.Show("Virhe. Anna eurot numerona!");
    }
}
```

TextBox:eista luetaan aina tekstityypin (.Text) muuttuja. Se on ennen laskuja muutettava numeroksi parsettamalla (esim. int.Parse(muunnettava numero) tai tässä double.Parse(muunnettava numero)). Tapahtuu virhe, jos yritetään muuttaa numeroksi kirjainta. Tässä on käytetty Try/Catch poikkeuskäsittelyä, joka tulostaa MessageBox:iin virheilmoituksen, jos euroiksi oli syötetty kirjaimia tms. merkkejä, joita ei voi muuttaa numeroksi.

Kun numero tulostetaan näytölle, se on tulostettava tekstinä. Ennen tulostusta merkki muutetaan numeroksi .ToString() komennolla. Markkoiksi muuttamisen jälkeen on lisätty + operaattorilla merkkijonoon tekstiä, eli "mk" ja muissa valuutoissa "valuutta" muuttujaan taitetaan valuutta nimeltä, ja tulostetaan yhdessä rahamäärän kanssa. Huomaa valuttoja tulostettaessa tyhjät " " merkit, jolla saadaan välilyönti summan ja yksikön väliin. Tässä tapauksessa olisi voinut tehdä myös niin, että välilyönnin laittaa syötettävän valuutan eteen, eli " dollari". Jos checkBox ei ollut valittuna, tyhjennetään "markkoinaTextBlock" kohta.