

## XML Webszolgáltatás alapú osztott alkalmazás fejlesztése (Mandelbrot halmaz számítás)

### 1. Az alkalmazás feladata

Mandelbrot halmaz megjelenítése úgy, hogy az iterációs számításokat egy webszolgáltatást nyújtó kiszolgáló gép végzi el. A kliens gép feladata a felhasználói felület kezelése és az eredmény megjelenítése.

### 2. Röviden a Mandelbrot halmazokról

A Mandelbrot halmaz olyan komplex síkbeli pontok halmaza, amelyeket egy iteratív függvény segítségével számítanak ki. Az alkalmazott függvény egy lehetséges alakja az alábbi:

$$z_{k+1} = z_k^2 + c$$

ahol  $z_{k+1}$  a komplex szám  $(k+1)$ -ik iterációja,

$$z = a + bi \quad \text{és} \quad z_0 = 0, z_1 = c$$

$c$  egy komplex szám, ami meghatározza a pont pozícióját.

$$z_k^2 = a_k^2 + 2a_k b_k i + b_k^2 i^2 = a^2 - b^2 + 2a_k b_k i$$

$$z_{k+1, \text{real}} = z_{k, \text{real}}^2 - z_{k, \text{imag}}^2 + c_{\text{real}}$$

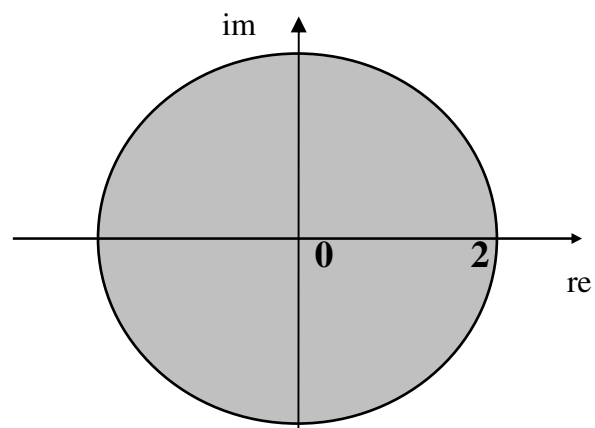
$$z_{k+1, \text{imag}} = 2z_{k, \text{real}} z_{k, \text{imag}} + c_{\text{imag}}$$

Az iteráció akkor áll le, ha a  $z$  nagysága

meghaladja a 2-es értéket  $\sqrt{z_{\text{real}}^2 + z_{\text{imag}}^2} > 2$

vagy az iterációs szám elérte a maximálisan megengedett iterációs számot. Ennek megfelelően  $c$ -t a 2 sugarú körből kell kiválasztani.

A grafikus megjelenítés során a fenti körből kiválasztunk egy téglalap alakú területet, aminek minden pontjára végrehajtjuk az iterációt. Minden pont esetén a szükséges iterációk száma határozza meg a pont színét.



### 3. A webszolgáltatást megvalósító szerveralkalmazás

A webszolgáltatást megvalósító szerveralkalmazás fejlesztése során az alábbi feladatokat kell megoldanunk:

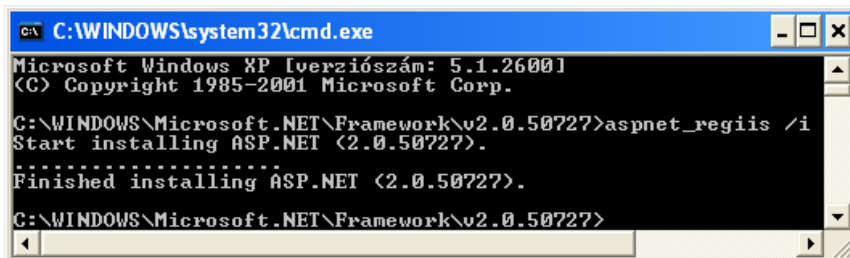
1. Az ASP.NET telepítése az IIS web szerveren
2. Projekt létrehozása
3. Egy képpont színét kiszámító metódus definiálása
4. Webszolgáltatást megvalósító metódus definiálása
5. Fordítás

### 3.1. ASP.NET az IIS web szerveren

Ha korábban nem történt meg, akkor elsőként telepíteni kell az ASP.NET-et az IIS web szerveren. Ehhez lépünk be

a C:\WINDOWS\Microsoft.NET\Framework\v2.0

.50727\könyvtárba. Ott adjuk ki az aspnet\_regiis /i parancsot.



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [verziószám: 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727>aspnet_regiis /i
Start installing ASP.NET (2.0.50727).
Finished installing ASP.NET (2.0.50727).
C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727>
```

### 3.2. Projekt létrehozása

Hozunk létre egy új C# projektet, melynek típusa (Templates) ASP.NET Web Service és helye: http://localhost/MandelbrotServer. A gyakorlat során úgy a szerver, mint a kliens alkalmazás a helyi gépen fog futni, és http protokollon keresztül kommunikálnak egymással.

A projektet az IIS webszerver

weblapok

tárolására

szolgáló

könyvtárban

helyezzük el.

Ennek érdekében

a Location

legördülő listában

a http értéket

választjuk, majd a

konkrét helynek a

http://localhost/M

andelbrotServer

címet adjuk meg.

A fájlok fizikai

helye

a

C:\inetpub\wwwroot\MandelbrotServer\

lesz. Ettől függetlenül a projektet leíró

MandelbrotServer.sln és MandelbrotServer.suo állományok a Visual Studio projektek

alapértelmezett könyvtárba kerülnek. Ez pl. a C:\Documents and

Settings\Felhasználónév\Dokumentumok\Visual Studio 2005\Projects\ lehet. A Location:http

választásnak az a célja, hogy más gépekről is elérhető legyen a szolgáltatás.

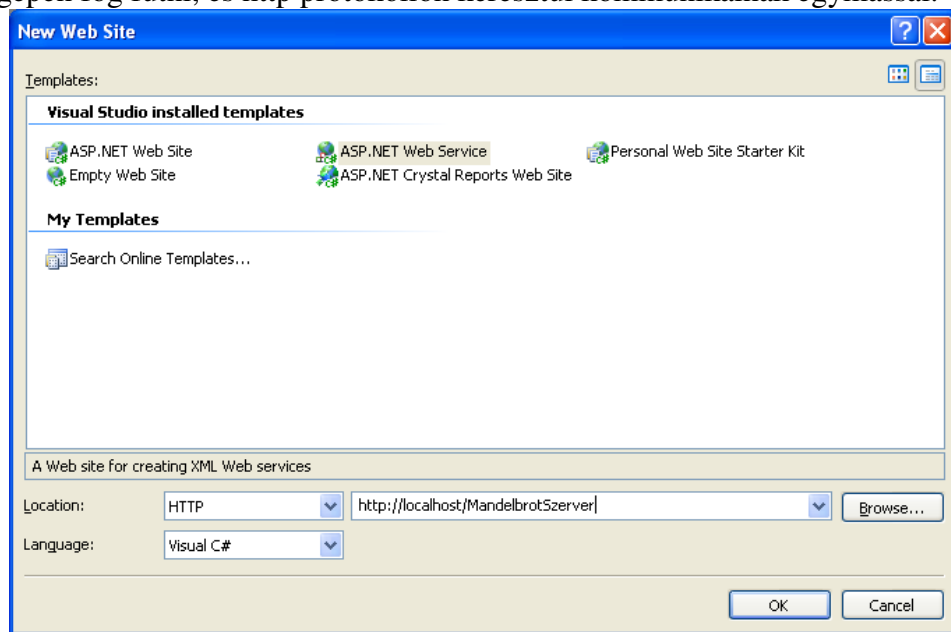
Ha csak a helyi gépről akarnánk elérni a MandelbrotServer-t vagy nem lenne telepítve az IIS

webszerver, akkor a Location:File System-et választanánk. A Visual Studio rendelkezik egy

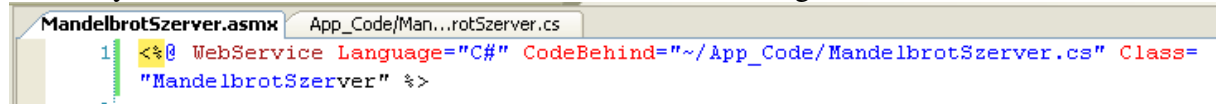
beépített egyszerű webszerverrel, aminek az a célja, hogy segítségével kipróbáljuk a

fejlesztett alkalmazásokat. Ha a projekt helyének File System-et adunk meg, akkor az általunk

választott könyvtárba menthető a projekt, és a kipróbálás a beépített webszerverrel történik.



Solution Explorerben nevezzük át a Service.asmx állományt MandelbrotSzerver.asmx-re, majd nyissuk meg kódnézetben. Solution Explorerben nyissuk meg a App\_Code mappát. A benne található Service.cs állományt nevezzük át MandelbrotSzerver.cs-re. A szolgáltatás osztályát (Service) nevezzük át MandelbrotSzerver-re. A MandelbrotSzerver.asmx állományban a hivatkozásokat is módosítjuk a fentieknek megfelelően.



A MandelBrotSzerver osztály definíciója előtti sorban írjuk át az alábbiak szerint az első attribútumot:

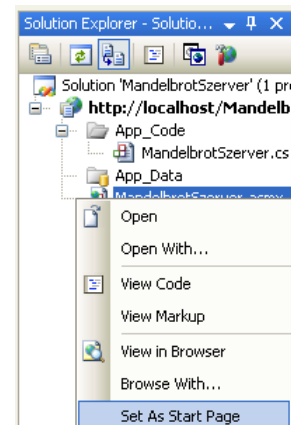
```
[WebService(Namespace="http://VezeteknévKeresztnév/MandelbrotSzerver/")]  
public class MandelbrotSzerver : System.Web.Services.WebService  
{ ... }
```

Itt egy egyedi névtér-azonosítót adunk a szolgáltatásunknak. Ez egy ál URI, konvencionálisan használjuk az URI formátumot. A **VezeteknévKeresztnév** részben mindenki a saját nevét adja meg.

### 3.3. Egy képpont színét kiszámító metódus definiálása

Készítsünk egy metódust a MandelbrotSzerver osztályban, ami paraméterként megkapja a komplex sík egy pontjának két koordinátáját (valós és képzetes rész), valamint az iterációk maximális megengedett számát, és ezekből meghatározza az adott ponthoz tartozó képpont színét.

```
/// <summary>  
/// Kiszámítja a pont színét RGB rendszerben.  
/// </summary>  
/// <param name="kpont_re">A pont valós koordinátája  
/// (c_re).</param>  
/// <param name="kpont_im">A pont képzetes  
/// koordinátája (c_im).</param>  
/// <param name="max_iter">Az iterációk maximális  
/// megengedett száma.</param>  
/// <returns>A pont színe RGB rendszerben.</returns>  
private int SzínSzámít(double kpont_re, double kpont_im, int max_iter)  
{ // Hányadik iterációnál tartunk  
  int iter_szam = 0;  
  // Az iteráció kezdetén a komplex szám azonos a kezdőponttal  
  double pont_re = kpont_re;  
  double pont_im = kpont_im;  
  // A komplex szám abszolút értékének a négyzete  
  double pont_abszn;  
  do  
  { iter_szam++;  
    pont_re = pont_re * pont_re - pont_im * pont_im + kpont_re;  
    pont_im = 2 * pont_re * pont_im + kpont_im;  
    pont_abszn = pont_re * pont_re + pont_im * pont_im;  
  } while (iter_szam < max_iter && pont_abszn < 4);  
  // Az iteráció leállításának két feltétele van "vagy" kapcsolattal:  
  // - a komplex szám abszolút értéke meghaladja a 2-t  
  // - az iterációk száma eléri a megszabott korlátot  
  // A végrehajtott iterációk száma határozza meg a színt.
```



```
// RGB-ben a legnagyobb szín 0xFFFFFFFF a legkisebb 0x0
// Ezt az intervallumot felosztjuk annyi részre, amennyi a legnagyobb
// megengedett iterációs szám. Ezzel a szín "kvantummal" fogjuk
// megszorozni a tényleges iterációs számot, és a megkapott érték
// lesz a megjelenítendő szín int-ben RGB rendszerben megadva.
int szin_kvant = 0xFFFFFFFF / max_iter;
return szin_kvant * iter_szam;
}
```

### 3.4. Webszolgáltatást megvalósító metódus definiálása

Készítsünk egy metódust a MandelbrotSzerver osztályban, ami nyilvánosan hozzáférhető webszolgáltatásként elvégzi a Mandelbrot halmaz számítást. Paraméterként megkapja, hogy a kép megjelenítéskor milyen széles és magas lesz. A számítást kép\_szél\*kép\_mag számú pontban végzi el a [min\_re,max\_re] és [min\_im,max\_im] intervallumok által meghatározott téglalap belsejében. További paraméterként megkapja az iterációk maximálisan megengedett számát. A metódus nyilvános kell legyen, és definíciója előtt el kell helyezzük a [WebMethod] attribútumot. Ezek biztosítják a webszolgáltatásként történő közzétételt.

```
/// <summary>
/// Mandelbrot halmaz számítást végez. A számítást kép_szél*kép_mag
/// számú
/// pontban végzi el a [min_re,max_re] és [min_im,max_im] intervallumok
/// által
/// meghatározott téglalap belsejében.
/// </summary>
/// <param name="kép_szél">Hány pontos lesz a képernyőn vízszintesen
/// a kép.</param>
/// <param name="kép_mag">Hány pontos lesz a képernyőn függőlegesen
/// a kép.</param>
/// <param name="min_re">Téglalap bal alsó sarkának vízszintes
/// koordinátája.</param>
/// <param name="min_im">Téglalap bal alsó sarkának függőleges
/// koordinátája.</param>
/// <param name="max_re">Téglalap jobb felső sarkának vízszintes
/// koordinátája.</param>
/// <param name="max_im">Téglalap jobb felső sarkának függőleges
/// koordinátája.</param>
/// <param name="Szinek">Kimenő paraméter. Egydimenziós tömb, ami a
/// képpontok színét tartalmazza RGB rendszerben.</param>
/// <param name="max_iter">Megadja, hogy legfeljebb hány iteráció
/// hajtható
/// végre egy pont esetén</param>
[WebMethod]
public void MandelSzámít(int kép_szél, int kép_mag,
    double min_re, double min_im, double max_re, double max_im,
    out int[] Szinek, int max_iter)
{ // Létrehozunk egy tömböt a képpontszínek tárolására.
  Szinek = new int[kép_szél * kép_mag];
  // Meghatározunk függőleges és vízszintes irányban egy-egy arány
  // értéket, amelyek lehetővé teszik, hogy minden képponthez
  // kiszámítsunk egy pontot a komplex síkon belül kiválasztott
  //téglalapban.
  double arany_re = (max_re - min_re) / (kép_szél - 1);
  double arany_im = (max_im - min_im) / (kép_mag - 1);
```

# Johanyák Zsolt Csaba: XML Webszolgáltatás alapú osztott alkalmazás fejlesztése (Mandelbrot halmaz számítás)– oktatási segédlet

http://www.johanyak.hu

e-mail: johanyak.csaba@gamf.kefo.hu

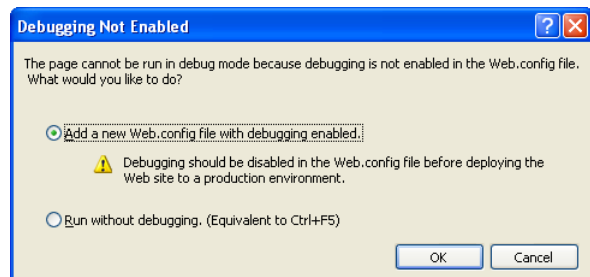
Copyright © 2008 Johanyák Zsolt Csaba

---

```
// Végighaladunk minden képponton.
for (int y = 0; y < kép_mag; y++)
    for (int x = 0; x < kép_szél; x++)
    { // Meghatározzuk, hogy az adott képponthez milyen pont
      // tartozik a komplex síkban.
      double pont_re = min_re + ((double)x * arany_re);
      double pont_im = min_im + ((double)y * arany_im);
      // Meghatározzuk a képpont színét RGB rendszerben, és tároljuk
      // egy egydimenziós tömbben.
      Szinek[y * kép_szél + x] = SzínSzámít(pont_re, pont_im,
        max_iter);
    }
}
```

## 3.5. Fordítás

Az esetleges gépelési hibák felderítése érdekében fordítsuk le az alkalmazást. A megjelenő párbeszédablakban válasszuk az első lehetőséget. Ezzel egy olyan Web.config fájlt hozunk létre, amiben engedélyezzük a hibakeresést és a lépésenkénti végrehajtást. Egy végleges alkalmazásnál ezt nem ajánlott engedélyezni.



# Johanyák Zsolt Csaba: XML Webszolgáltatás alapú osztott alkalmazás fejlesztése (Mandelbrot halmaz számítás)– oktatási segédlet

http://www.johanyak.hu

e-mail: johanyak.csaba@gamf.kefo.hu

Copyright © 2008 Johanyák Zsolt Csaba

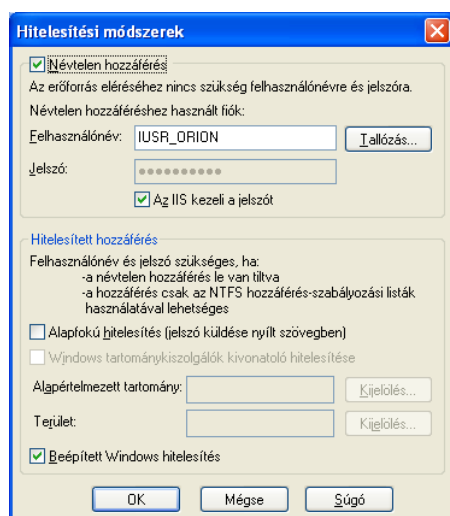
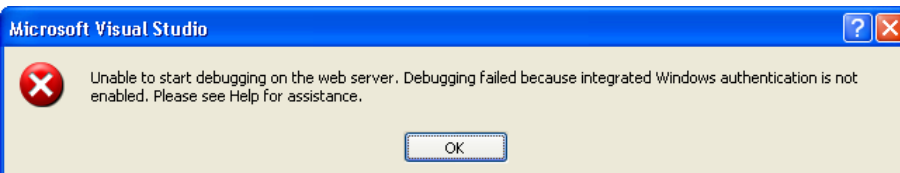
A megjelenő weblapon a MandelSzámít hivatkozást választva leírást kapunk a webszolgáltatásról és a használata során használatos SOAP üzenetekről.

Ha Debug/Start Debugging F5-tel indítjuk az alkalmazást

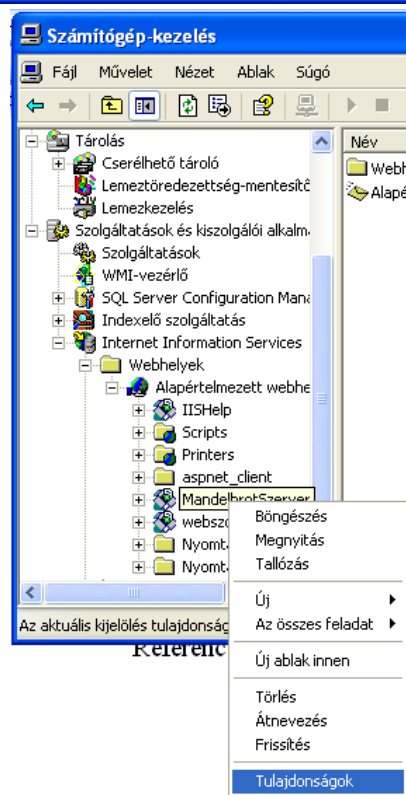
## MandelbrotSzerver

The following operations are supported. For a formal definition, please review the [Service Description](#).

- [HelloWorld](#)
- [MandelSzámít](#)



előfordulhat, hogy kapunk egy hibaüzenetet, miszerint nincs engedélyezve a Windows hitelesítés, és emiatt a fejlesztőrendszer nem tudja indítani a hibakeresést. Az egyik lehetséges megoldás ilyenkor az, hogy az Asztalon jobb egérgombbal kattintunk a Sajátgépen, majd a Kezelés menüpontot

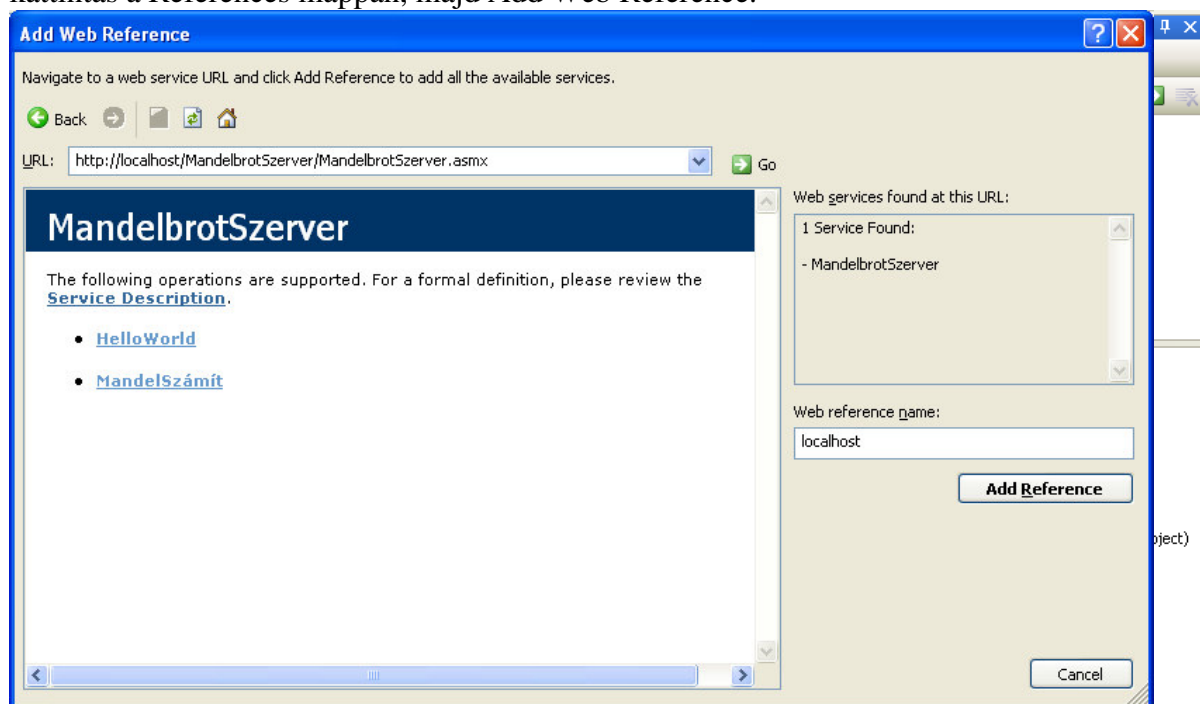
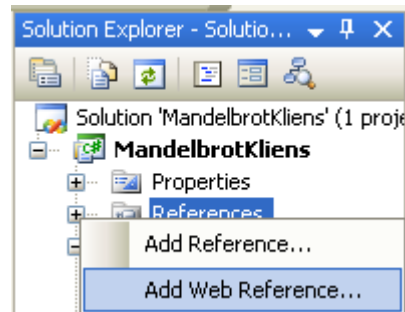


választjuk. A megjelenő ablakban a bal oldali listában kinyitjuk az Internet Information Services mappát, majd a Webhelyeket, majd az alapértelmezett webhelyet, és itt jobb egérgombbal kattintunk a MandelbrotSzerveren. A gyorsmenüben a Tulajdonságok menüpontot választjuk. A MandelbrotSzerver tulajdonságai ablakban a Könyvtárbiztonság fület válasszuk, majd kattintsunk a Névtelen és hitelesített hozzáférés beállításai csoportban a Szerkesztés gombra. A megjelenő Hitelesítési módszerek ablakban jelöljük be a Beépített Windows hitelesítést, majd kattintsunk az OK gombon. A MandelbrotSzerver tulajdonságai ablakban kattintsunk az OK gombon.

## 4. A kliens alkalmazás

Indítsuk el a Visual Studio egy új példányát, és hozzunk létre benne egy új Windows Application típusú projektet MandelbrotKliens néven. A form osztályát nevezzük át frmMandelbrotKliens-re, és az őt tartalmazó állomány neve legyen: frmMandelbrotKliens.cs. Az ablak felirata legyen „Mandelbrot halmaz megjelenítő”.

Adjunk a projekthez egy hivatkozást a MandelbrotSzerver web szolgáltatásra. A Solution Explorerben jobb egérgomb kattintás a References mappán, majd Add Web Reference.



A párbeszédablakban kattintsunk a Web services on the local machine linken, majd a megjelenő Services listában kattintsunk a MandelbrotSzerver linken. Ha több MandelbrotSzerver link jelenik meg, akkor a http://localhost/MandelbrotSzerver/MandelbrotSzerver.asmx-nek megfelelőt válasszuk. Ezután kattintsunk az Add reference nyomógombon. Ezután a fejlesztőrendszer a MandelbrotKliens.localhost névtérben a projekten belül létrehoz egy proxy osztályt MandelbrotSzerver néven, amin keresztül a továbbiakban elérhetjük a közzétett webszolgáltatást.

Az frmMandelbrotKliens osztályban hozzuk létre az alábbi adattagokat:

```
/// <summary>  
/// A vizsgált valós tartomány alsó határa.  
/// </summary>  
private double min_re;  
/// <summary>  
/// A vizsgált képzetes tartomány alsó határa.  
/// </summary>  
private double min_im;
```



## Johanyák Zsolt Csaba: XML Webszolgáltatás alapú osztott alkalmazás fejlesztése (Mandelbrot halmaz számítás)– oktatási segédlet

http://www.johanyak.hu

e-mail: johanyak.csaba@gamf.kefo.hu

Copyright © 2008 Johanyák Zsolt Csaba

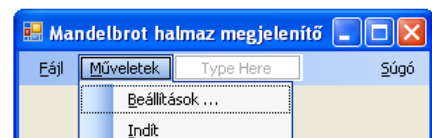
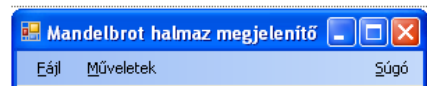
---

```
/// <summary>
/// A vizsgált valós tartomány felső határa.
/// </summary>
private double max_re;
/// <summary>
/// A vizsgált képzetes tartomány felső határa.
/// </summary>
private double max_im;
/// <summary>
/// Az iterációk maximálisan megengedett száma.
/// </summary>
private int max_iter;
/// <summary>
/// A rajzterület szélessége.
/// </summary>
private int kép_szél;
/// <summary>
/// A rajzterület magassága.
/// </summary>
private int kép_mag;
/// <summary>
/// Az egyes képpontokhoz tartozó színek RGB rendszerben
/// egy egydimenziós
/// tömbben.
/// </summary>
private int[] iSzínek;
/// <summary>
/// A képet leíró bitkép objektum.
/// </summary>
private Bitmap bmKép;
/// <summary>
/// A szerveret elérhetővé tevő proxy osztály objektuma.
/// </summary>
private localhost.MandelbrotSzerver Szerver;
```

Az frmMandelbrotKliens osztály konstruktorában adjunk kezdőértéket a vizsgált téglalapot meghatározó adattagoknak, és adjuk meg az iterációk megengedett maximális értékét is.

```
public frmMandelbrotKliens()
{ InitializeComponent();
  // A vizsgált téglalap alakú terület bal felső sarka.
  min_re = min_im = 0;
  // A vizsgált téglalap alakú terület jobb alsó sarka.
  max_re = max_im = 1.5;
  // Az iterációk maximálisan megengedett száma.
  max_iter = 10;
}
```

Helyezzünk el egy főmenü (MenuStrip) komponenst msFőmenü néven a formon, és helyezzük el benne a következő menüpontokat: Fájl (tsmiFájl), Műveletek (tsmiMűveletek), Súgó (tsmiSúgó). Ez utóbbi legyen jobbra igazítva (Align=Right). A Fájl menüben helyezzük el a Kilépés (tsmiKilépés) menüpontot. A Műveletek menüben helyezzük el a Beállítások ... (tsmiBeállítások) és





## Johanyák Zsolt Csaba: XML Webszolgáltatás alapú osztott alkalmazás fejlesztése (Mandelbrot halmaz számítás)– oktatási segédlet

<http://www.johanyak.hu>

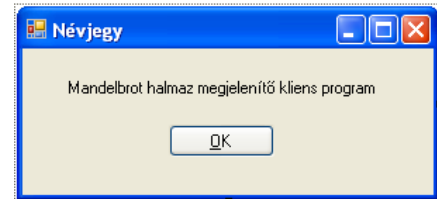
e-mail: [johanyak.csaba@gamf.kefo.hu](mailto:johanyak.csaba@gamf.kefo.hu)

Copyright © 2008 Johanyák Zsolt Csaba

Indít (tsmiIndít) menüpontokat. A Súly menüben helyezzük el a Névjegy (tsmiNévjegy) menüpontot.

Hozunk létre egy új formot az frmNévjegy.cs állományban, és nevét állítsuk át frmNévjegy-re. Felületét alakítsuk ki a mellékelt ábrának megfelelően. Az OK gomb lenyomása az ablak bezárását idézi elő.

Készítsünk egy eseménykezelőt a főablak Névjegy menüpontjához, melyben megjelenítjük a névjegy párbeszédablakot.

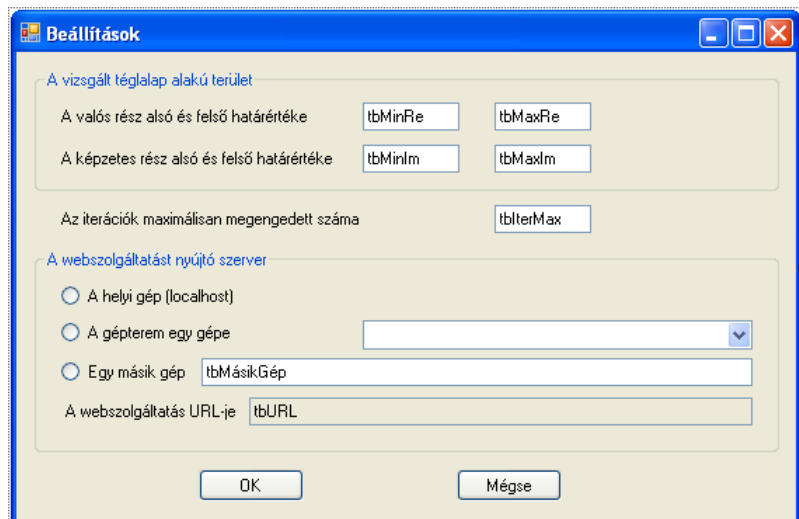


```
/// <summary>
/// Megjeleníti modálisan a Névjegy párbeszédablakot.
/// </summary>
private void tsmiNévjegy_Click(object sender, EventArgs e)
{
    frmNévjegy nf=new frmNévjegy();
    nf.ShowDialog();
}
```

Készítsünk egy eseménykezelőt a főablak Kilépés menüpontjához, melyben kilépünk a programból.

```
/// <summary>
/// Kilépés a programból.
/// </summary>
private void tsmiKilépés_Click(object sender, EventArgs e)
{
    Application.Exit();
}
```

Hozunk létre egy új formot a frmBeallitasok.cs állományban. A form nevét írjuk át frmBeállítások-ra. A form felületét a mellékelt ábra szerint alakítsuk ki. A szerkesztőmezők nevei az ábrán a szerkesztőmezők területén olvashatóak. A választógombok nevei rbHelyigép, rbGépterem és rbMásikGép legyen. A legördülő lista egy kombinált választóablak legyen



(ComboBox) cbGépterem néven. A Mégse gombon (Name=btMégse) történő kattintás az ablak Cancel típusú bezárását idézze elő (DialogResult=Cancel). Az OK gombon (Name=btOK) történő kattintás az ablak OK típusú bezárását idézze elő (DialogResult=OK). A cbGépterem és tbMásikGép vezérlők kezdetben nem engedélyezettek (Enabled=False). A tbURL csak olvasható (ReadOnly=True). Az rbHelyiGép van kezdetben kiválasztva (Checked=True). A cbGépterem legördülő lista típusú (DropDownStyle=DropDownList). Írjuk át a form feliratát Text=Beállítások.

## Johanyák Zsolt Csaba: XML Webszolgáltatás alapú osztott alkalmazás fejlesztése (Mandelbrot halmaz számítás)– oktatási segédlet

<http://www.johanyak.hu>

e-mail: [johanyak.csaba@gamf.kefo.hu](mailto:johanyak.csaba@gamf.kefo.hu)

Copyright © 2008 Johanyák Zsolt Csaba

---

Az frmBeállítások osztályban hozzunk létre egy adattagot, ami képes tárolni a géptermi gépek listáját. Ennek tartalma fog megjelenni a cbGépterem legördülő listaablakban.

```
/// <summary>
/// A géptermi gépek legördülő listaablakában megjelenő gépnevek
/// listája.
/// </summary>
private string[] GéptermiGépek;
```

A konstruktorban állítsuk elő a géptermi gépek neveit, majd helyezzük el őket a legördülő listaablakban. Az alábbi példa a 7-es gépteremhez készült. Amennyiben a gyakorlatot egy másik gépteremben hajtják végre, akkor értelemszerűen a "7-" szövegrészt módosítani kell.

```
public frmBeállítások()
{ InitializeComponent();
  // Tömb létrehozása a nevek számára.
  GéptermiGépek = new string[15];
  // Nevek előállítása abból kiindulva, hogy a gépek 1-től 15-ig
  // sorszámozottak.
  for (int i = 1; i <= 15; i++)
  { GéptermiGépek[i - 1] = "7-" + i.ToString();
  }
  // A legördülő listaablak tartalmának törlése.
  cbGépterem.Items.Clear();
  // Nevek elhelyezése a legördülő listaablakban.
  cbGépterem.Items.AddRange(GéptermiGépek);
  // Alapértelmezés szerint az első gép a kiválasztott.
  cbGépterem.SelectedItem = cbGépterem.Items[0];
}
```

Készítsünk a szerkesztőmezőkhöz és a választógombokhoz egy-egy tulajdonságot, ami lehetővé teszi értékük lekérdezését és beállítását.

```
/// <summary>
/// Beállítja/lekérdezi a valós rész alsó határértékét.
/// </summary>
public double dMinRe
{ get
  { return double.Parse(tbMinRe.Text);
  }
  set
  { tbMinRe.Text=value.ToString();
  }
}
/// <summary>
/// Beállítja/lekérdezi a valós rész felső határértékét.
/// </summary>
public double dMaxRe
{ get
  { return double.Parse(tbMaxRe.Text);
  }
  set
  { tbMaxRe.Text=value.ToString();
  }
}
```

## Johanyák Zsolt Csaba: XML Webszolgáltatás alapú osztott alkalmazás fejlesztése (Mandelbrot halmaz számítás)– oktatási segédlet

<http://www.johanyak.hu>

e-mail: [johanyak.csaba@gamf.kefo.hu](mailto:johanyak.csaba@gamf.kefo.hu)

Copyright © 2008 Johanyák Zsolt Csaba

---

```
/// <summary>
/// Beállítja/lekérdezi a képzetes rész alsó határértékét.
/// </summary>
public double dMinIm
{
    get
    { return double.Parse(tbMinIm.Text);
    }
    set
    { tbMinIm.Text=value.ToString();
    }
}
/// <summary>
/// Beállítja/lekérdezi a képzetes rész felső határértékét.
/// </summary>
public double dMaxIm
{
    get
    { return double.Parse(tbMaxIm.Text);
    }
    set
    { tbMaxIm.Text=value.ToString();
    }
}
/// <summary>
/// Beállítja/lekérdezi az iterációk maximális megengedett értékét.
/// </summary>
public int iIterMax
{
    get
    { return int.Parse(tbIterMax.Text);
    }
    set
    { tbIterMax.Text=value.ToString();
    }
}
/// <summary>
/// Beállítja/lekérdezi a webszolgáltatás URL-jét.
/// Az alábbi példa a 7-es gépteremhez készült. Amennyiben a
/// gyakorlatot egy másik gépteremben hajtják végre, akkor
/// értelemszerűen a "7-" szövegrészt módosítani kell.
/// </summary>
public string sURL
{
    get
    { return tbURL.Text;
    }
    set
    { string surl=value;
      // Részekre bontjuk a sztringet. A /-jel az elválasztó.
      // pl. http://localhost/MandelbrotSzerver/MandelbrotSzerver.aspx
      // --> "http:" "localhost" "MandelbrotSzerver" "MandelbrotSzerver"
      string []részek=surl.Split('/');
      // Ha kettőnél kevesebb részünk van, akkor az URL érvénytelen.
      if(részek.Length>1)
      { if(részek[2]=="localhost")
        { rbHelyiGép.Checked=true;
        }
        else
        { if(részek[2].Substring(0,3)=="7-")
          { rbGépterem.Checked=true;
            cbGépterem.SelectedItem=részek[2];
          }
        }
      }
    }
}
```

## Johanyák Zsolt Csaba: XML Webszolgáltatás alapú osztott alkalmazás fejlesztése (Mandelbrot halmaz számítás)– oktatási segédlet

<http://www.johanyak.hu>

e-mail: [johanyak.csaba@gamf.kefo.hu](mailto:johanyak.csaba@gamf.kefo.hu)

Copyright © 2008 Johanyák Zsolt Csaba

---

```
        else
        { rbMásikGép.Checked=true;
          tbMásikGép.Text=részek[2];
        }
    }
    else
    { throw new ApplicationException("A megadott URL érvénytelen!");
      tbURL.Text=value;
    }
}
```

Készítsünk egy eseménykezelőt a helyi gép választógomb kijelölés-változásához (CheckedChanged esemény).

```
/// <summary>
/// Ha változik a rbHelyiGép állapota.
/// </summary>
private void rbHelyiGép_CheckedChanged(object sender, EventArgs e)
{ // Ha ki van jelölve.
  if (rbHelyiGép.Checked)
  { // Részekre bontjuk a /-jelek szerint.
    string[] részek = tbURL.Text.Split('/');
    // Ha részek száma kisebb mint kettő, akkor az URL érvénytelen.
    if (részek.Length > 1)
    { részek[2] = "localhost";
      string surl = "";
      // Újra összeállítjuk a teljes URL-t.
      for (int i = 0; i < részek.Length; i++)
        surl += részek[i] + "/";
      // Az utolsó /-jelet el kell távolítani.
      tbURL.Text = surl.Substring(0, surl.Length - 1);
    }
  }
}
```

Készítsünk egy eseménykezelőt a legördülő listaablak kijelölésváltozás eseményéhez (SelectedIndexChanged esemény).

```
/// <summary>
/// Ha változott a legördülő listaablak kiválasztott eleme.
/// </summary>
private void cbGépterem_SelectedIndexChanged(object sender,
  EventArgs e)
{ // Részekre bontjuk az URL-t a /-jelek szerint.
  string[] részek = tbURL.Text.Split('/');
  // Ha a részek száma kisebb kettőnél az URL érvénytelen.
  if (részek.Length > 1)
  { // A második helyre betesszük a kiválasztott gép nevét.
    részek[2] = cbGépterem.SelectedItem.ToString();
    string surl = "";
    // Újból összeállítjuk az URL-t.
    for (int i = 0; i < részek.Length; i++)
      surl += részek[i] + "/";
    // Az utolsó /-jelet el kell távolítani.
    tbURL.Text = surl.Substring(0, surl.Length - 1);
  }
}
```

```
}
```

Készítsünk egy eseménykezelőt a „A gépterem egy gépe” választógomb kijelölés-változásához (CheckedChanged esemény).

```
/// <summary>
/// Ha megváltozik az rgGépterem választógomb állapota.
/// </summary>
private void rbGépterem_CheckedChanged(object sender, EventArgs e)
{ // A legördülő listaablakot a választógomb állapotának megfelelően
  // engedélyezzük/tiltjuk.
  cbGépterem.Enabled = rbGépterem.Checked;
  // Az eseménykezelőt meghívva előidézzük, hogy a megfelelő gépnév
  // jelenjen meg a tbURL mezőben
  cbGépterem_SelectedIndexChanged(this, new EventArgs());
}
```

Készítsünk egy eseménykezelőt a tbMásikGép szerkesztőmező tartalomváltozás eseményéhez (TextChanged esemény).

```
/// <summary>
/// Ha megváltozott a tbMásikGép szerkesztőmezőben a szöveg.
/// </summary>
private void tbMásikGép_TextChanged(object sender, EventArgs e)
{ // Részekre bontjuk az eredeti URL-t a /-jelek szerint.
  string []részek=tbURL.Text.Split('/');
  // Ha részek száma kisebb mint kettő, akkor az URL érvénytelen.
  if(részek.Length>1)
  { // Második részként beépítjük a megadott gépnevet.
    részek[2]=tbMásikGép.Text;
    // Újra összeállítjuk a teljes URL-t.
    string surl="";
    for(int i=0;i<részek.Length;i++)
      surl+=részek[i]+"/";
    // Az utolsó /-jelet el kell távolítani.
    tbURL.Text=surl.Substring(0, surl.Length-1);
  }
}
```

Készítsünk egy eseménykezelőt az „Egy másik gép” választógomb kijelölés-változásához (CheckedChanged esemény).

```
/// <summary>
/// Ha megváltozott az rbMásikGép választógomb állapota.
/// </summary>
private void rbMásikGép_CheckedChanged(object sender, EventArgs e)
{ // A szerkesztőmező engedélyezettsége a választógomb bekapcsolt/
  // kikapcsolt állapotától függ.
  if ((tbMásikGép.Enabled = rbMásikGép.Checked) == true)
  { // Ha be van kapcsolva, akkor meghívjuk a szerkesztőmező
    // szövegváltozási eseménykezelőjét, ami által a tbURL
    // tartalma a megadott gépnévnek megfelelően módosul.
    tbMásikGép_TextChanged(rbMásikGép, new System.EventArgs());
  }
}
```

## Johanyák Zsolt Csaba: XML Webszolgáltatás alapú osztott alkalmazás fejlesztése (Mandelbrot halmaz számítás)– oktatási segédlet

<http://www.johanyak.hu>

e-mail: [johanyak.csaba@gamf.kefo.hu](mailto:johanyak.csaba@gamf.kefo.hu)

Copyright © 2008 Johanyák Zsolt Csaba

---

Készítsünk ellenőrző eseménykezelőket a numerikus adatok bevitelére szolgáló szövegmezőkhöz (Validating) esemény. Ha az adatbevitel hibás, akkor az e.Cancel adattagot true-ra kell állítani, ami megakadályozza, hogy a felhasználó kilépjen a szövegmezőből. Emellett egy üzenetablakkal tájékoztassuk a felhasználót a hibáról.

```
/// <summary>
/// Ellenőrző eseménykezelő. Ha hibás az adatbevitel, akkor a
/// felhasználó nem léphet ki a szerkesztőmezőből.
/// </summary>
private void tbMinRe_Validating(object sender,
    System.ComponentModel.CancelEventArgs e)
{ try
    { // Megpróbáljuk valós számmá alakítani a megadott adatot.
      // Ha nem sikerül, akkor kivétel keletkezik.
      double d=double.Parse(tbMinRe.Text);
      // Ha a szám túl kicsi vagy túl nagy, akkor kivételt idézünk elő.
      if(d<-2 || d>=dMaxRe)
          throw new ApplicationException("Hibás szám");
    }
    catch
    { // Ha a megadott adat nem megfelelő, akkor hibaüzenet, és
      // a szerkesztőmezőből történő kilépés megtiltása.
      MessageBox.Show("Az érték a [-2,2] intervallumba kell eszen, "+
          "és kisebb kell legyen mint a felső határérték!",
          "Adatbeviteli hiba", MessageBoxButtons.OK,
          MessageBoxIcon.Error);
      e.Cancel=true;
    }
}

/// <summary>
/// Ellenőrző eseménykezelő. Ha hibás az adatbevitel, akkor a
/// felhasználó nem léphet ki a szerkesztőmezőből.
/// </summary>
private void tbMaxRe_Validating(object sender,
    System.ComponentModel.CancelEventArgs e)
{ try
    { // Megpróbáljuk valós számmá alakítani a megadott adatot.
      // Ha nem sikerül, akkor kivétel keletkezik.
      double d=double.Parse(tbMaxRe.Text);
      // Ha a szám túl kicsi vagy túl nagy, akkor kivételt idézünk elő.
      if(d>2 || d<=dMinRe)
          throw new ApplicationException("Hibás szám");
    }
    catch
    { // Ha a megadott adat nem megfelelő, akkor hibaüzenet, és
      // a szerkesztőmezőből történő kilépés megtiltása.
      MessageBox.Show("Az érték a [-2,2] intervallumba kell eszen, "+
          "és nagyobb kell legyen mint az alsó határérték!",
          "Adatbeviteli hiba", MessageBoxButtons.OK,
          MessageBoxIcon.Error);
      e.Cancel=true;
    }
}
```

## Johanyák Zsolt Csaba: XML Webszolgáltatás alapú osztott alkalmazás fejlesztése (Mandelbrot halmaz számítás)– oktatási segédlet

<http://www.johanyak.hu>

e-mail: [johanyak.csaba@gamf.kefo.hu](mailto:johanyak.csaba@gamf.kefo.hu)

Copyright © 2008 Johanyák Zsolt Csaba

---

```
/// <summary>
/// Ellenőrző eseménykezelő. Ha hibás az adatbevitel, akkor a
/// felhasználó nem léphet ki a szerkesztőmezőből.
/// </summary>
private void tbMinIm_Validating(object sender,
    System.ComponentModel.CancelEventArgs e)
{ try
    { // Megpróbáljuk valós számmá alakítani a megadott adatot.
      // Ha nem sikerül, akkor kivétel keletkezik.
      double d=double.Parse(tbMinIm.Text);
      // Ha a szám túl kicsi vagy túl nagy, akkor kivételt idézünk elő.
      if(d<-2 || d>=dMaxIm)
          throw new ApplicationException("Hibás szám");
    }
    catch
    { // Ha a megadott adat nem megfelelő, akkor hibaüzenet, és
      // a szerkesztőmezőből történő kilépés megtiltása.
      MessageBox.Show("Az érték a [-2,2] intervallumba kell essen, "+
          "és kisebb kell legyen mint a felső határérték!",
          "Adatbeviteli hiba", MessageBoxButtons.OK,
          MessageBoxIcon.Error);
      e.Cancel=true;
    }
}

/// <summary>
/// Ellenőrző eseménykezelő. Ha hibás az adatbevitel, akkor a
/// felhasználó nem léphet ki a szerkesztőmezőből.
/// </summary>
private void tbMaxIm_Validating(object sender,
    System.ComponentModel.CancelEventArgs e)
{ try
    { // Megpróbáljuk valós számmá alakítani a megadott adatot.
      // Ha nem sikerül, akkor kivétel keletkezik.
      double d=double.Parse(tbMaxIm.Text);
      // Ha a szám túl kicsi vagy túl nagy, akkor kivételt idézünk elő.
      if(d>2 || d<=dMinIm)
          throw new ApplicationException("Hibás szám");
    }
    catch
    { // Ha a megadott adat nem megfelelő, akkor hibaüzenet, és
      // a szerkesztőmezőből történő kilépés megtiltása.
      MessageBox.Show("Az érték a [-2,2] intervallumba kell essen, "+
          "és nagyobb kell legyen mint az alsó határérték!",
          "Adatbeviteli hiba", MessageBoxButtons.OK,
          MessageBoxIcon.Error);
      e.Cancel=true;
    }
}

/// <summary>
/// Ellenőrző eseménykezelő. Ha hibás az adatbevitel, akkor a
/// felhasználó nem léphet ki a szerkesztőmezőből.
/// </summary>
private void tbIterMax_Validating(object sender,
    System.ComponentModel.CancelEventArgs e)
{ try
    { // Megpróbáljuk előjel nélküli egész számmá alakítani a megadott
```



## Johanyák Zsolt Csaba: XML Webszolgáltatás alapú osztott alkalmazás fejlesztése (Mandelbrot halmaz számítás)– oktatási segédlet

<http://www.johanyak.hu>

e-mail: [johanyak.csaba@gamf.kefo.hu](mailto:johanyak.csaba@gamf.kefo.hu)

Copyright © 2008 Johanyák Zsolt Csaba

---

```
// adatot. Ha nem sikerül, akkor kivétel keletkezik.
uint u=uint.Parse(tbIterMax.Text);
// Ha a szám túl kicsi, akkor kivételt idézünk elő.
if(u<1)
    throw new ApplicationException("Hibás szám");
}
catch
{ // Ha a megadott adat nem megfelelő, akkor hibaüzenet, és
  // a szerkesztőmezőből történő kilépés megtiltása.
  MessageBox.Show("Az iterációk maximálisan megengedett "+
    "számánál megadott "+
    "érték nem egy pozitív egész szám!",
    "Adatbeviteli hiba", MessageBoxButtons.OK,
    MessageBoxIcon.Error);
  e.Cancel=true;
}
}
```

Készítsünk egy eseménykezelőt a főablak Beállítás menüpontjához, melyben létrehozunk egy példányt az frmBeállítások párbeszédablakból, beállítjuk a vezérlők értékeit az érvényes adatok alapján, majd megjelenítjük a párbeszédablakot. Ha a felhasználó az OK gombbal zárta le az ablakot, akkor kiolvassuk a beállított értékeket.

```
/// <summary>
/// Műveletek menü Beállítás menüpont.
/// </summary>
private void tsmiBeállítások_Click(object sender, EventArgs e)
{ // Létrehozunk egy példányt a párbeszédablakból.
  frmBeállítások bf=new frmBeállítások();
  // Beállítjuk a tulajdonságokat a tárolt adatok alapján.
  bf.dMinRe=min_re;
  bf.dMaxRe=max_re;
  bf.dMinIm=min_im;
  bf.dMaxIm=max_im;
  bf.iIterMax=max_iter;
  bf.sURL=Szerver.Url;
  // Megjelenítjük a párbeszédablakot
  if(bf.ShowDialog()==DialogResult.OK)
  { // Ha a felhasználó az OK gombbal zárta be a párbeszédablakot,
    // akkor kiolvassuk a megadott értékeket.
    min_re=bf.dMinRe;
    max_re=bf.dMaxRe;
    min_im=bf.dMinIm;
    max_im=bf.dMaxIm;
    max_iter=bf.iIterMax;
    Szerver.Url=bf.sURL;
  }
}
```

Helyezzünk el egy PictureBox komponenst az frmMandelbrotKliens-en úgy, hogy neve legyen pbKép, és töltsé ki a teljes kliens ablakterületet (Dock=Fill).

A form konstruktorában a kép\_szél és kép\_mag adattagokban tároljuk el a komponens kezdeti szélességét és magasságát.

```
// Eltároljuk a PictureBox komponens kezdeti szélességét és
```

## Johanyák Zsolt Csaba: XML Webszolgáltatás alapú osztott alkalmazás fejlesztése (Mandelbrot halmaz számítás)– oktatási segédlet

<http://www.johanyak.hu>

e-mail: [johanyak.csaba@gamf.kefo.hu](mailto:johanyak.csaba@gamf.kefo.hu)

Copyright © 2008 Johanyák Zsolt Csaba

---

```
// magasságát.  
kép_szél = pbKép.Width;  
kép_mag = pbKép.Height;
```

A form konstruktorában hozzunk létre egy olyan bitkép (Bitmap) objektumot, ami az előzőekben meghatározott szélességgel és magassággal rendelkezik.

```
// Létrehozunk egy bitkép objektumot.  
bmKép = new Bitmap(kép_szél, kép_mag);
```

A konstruktorban hozzuk létre a szerveret jelképező proxy objektumot.

```
// Létrehozzuk a szerveret jelképező objektumot.  
Szerver = new MandelbrotKliens.localhost.MandelbrotSzerver();
```

Készítsünk egy eseménykezelőt az frmMandelbrotKliens Indít menüpontjához. Itt történik a távoli metódushívás és az eredmény megjelenítése.

```
/// <summary>  
/// Műveletek menü Indít menüpont.  
/// </summary>  
private void tsmiIndít_Click(object sender, EventArgs e)  
{ // Beállítjuk a homokóra kurzort, jelezve, hogy dolgozik a program  
  Cursor = Cursors.WaitCursor;  
  // Eltároljuk a PictureBox komponens aktuális szélességét és  
  // magasságát.  
  kép_szél = pbKép.Width;  
  kép_mag = pbKép.Height;  
  // Létrehozunk egy bitkép objektumot, ami pont ekkora méretű.  
  bmKép = new Bitmap(kép_szél, kép_mag);  
  try  
  { // Meghívjuk a szerveren a számításokat végző metódust.  
    iSzínek = Szerver.MandelSzámít(kép_szél, kép_mag, min_re,  
      min_im, max_re, max_im, max_iter);  
    // Beállítjuk a bitkép pontjainak színét a kiszámított értékek  
    // alapján.  
    for (int y = 0; y < kép_mag; y++)  
      for (int x = 0; x < kép_szél; x++)  
        bmKép.SetPixel(x, y,  
          ColorTranslator.FromWin32(iSzínek[y * kép_szél + x]));  
    // Az előállított képet hozzárendeljük a komponens  
    // Image tulajdonságához.  
    pbKép.Image = bmKép;  
  }  
  catch (System.Web.Services.Protocols.SoapException se)  
  { // Ha túl nagy a képpontok színeit leíró tömb.  
    MessageBox.Show("Változtassa meg a beállításokat vagy az " +  
      "ablak méretét!" +  
      "\n\nHiba típusa: " + se + "\n\nAz ablak jelenelegi mérete: " +  
      this.Width.ToString() + "x" + this.Height.ToString() +  
      "\nA kép mérete: " + pbKép.Width.ToString() + "x" +  
      pbKép.Height.ToString(),  
      "Hiba", MessageBoxButtons.OK, MessageBoxIcon.Error);  
  }  
  catch (System.Net.WebException)
```

## Johanyák Zsolt Csaba: XML Webszolgáltatás alapú osztott alkalmazás fejlesztése (Mandelbrot halmaz számítás)– oktatási segédlet

<http://www.johanyak.hu>

e-mail: [johanyak.csaba@gamf.kefo.hu](mailto:johanyak.csaba@gamf.kefo.hu)

Copyright © 2008 Johanyák Zsolt Csaba

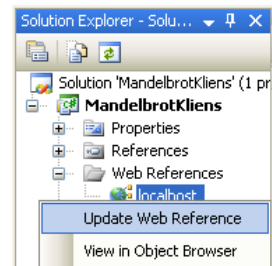
---

```
{ // Ha nem sikerült elérni a távoli gépen a webszolgáltatást.
  MessageBox.Show("Nem sikerült elérni a megadott gépen a " +
    "megadott webszolgáltatást!" +
    "\nMódosítsa a gép nevét a Műveletek " +
    "menü Beállítás pontjában!",
    "Hiba", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
finally
{ // Beállítjuk a normál nyíl kurzort, jelezve, hogy a program
  // befejezte a feladatát.
  Cursor = Cursors.Arrow;
}
}
```

Készítünk a pbKép komponenshez egy eseménykezelőt, amely akkor aktivizálódik, ha érvénytelenné vált a kép (Paint esemény).

```
/// <summary>
/// Érvénytelenné vált a PictureBox területe. Fel kell másolni rá a
/// bitkép objektumban tárolt képet.
/// </summary>
private void pbKép_Paint(object sender, PaintEventArgs e)
{ // Az előállított képet
  // hozzárendeljük a komponens Image tulajdonságához.
  pbKép.Image = bmKép;
}
```

Fordítsuk le az alkalmazást, majd indítsuk el. Válasszuk ki az Indít menüpontot, ezzel indítjuk a számításokat. Alapértelmezés szerint a localhost gépen keresi a program a webszolgáltatást. Időnként előfordul, hogy első indításnál rossz szolgáltatásnévteret használ a program a szervernél megadott <http://Felhasználónév> helyett az alapértelmezés szerinti <http://tempuri.org>-ot. Ilyenkor kis várakozás után az ábrán látható hibaüzenetet kapjuk. A megoldás a következő. Lépünk ki a programból, majd a Solution Explorerben nyissuk meg a



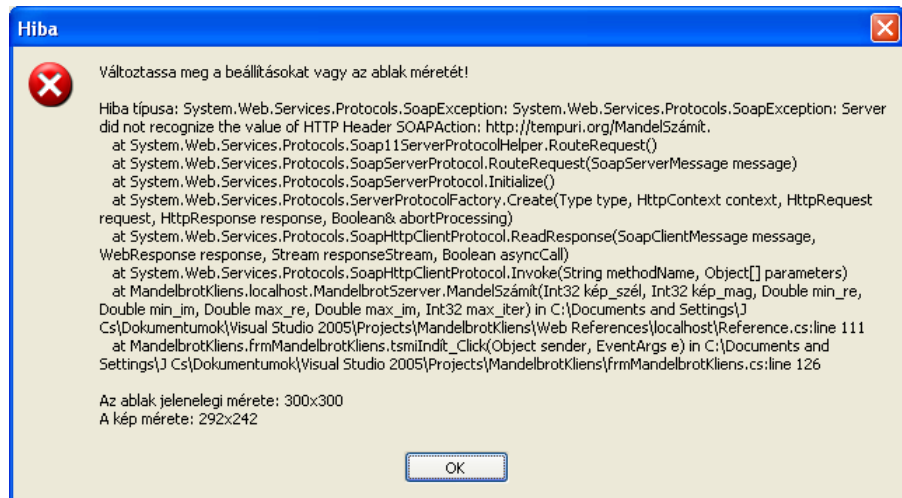
Web References mappát. Az ott megjelenő localhost elemen kattintsunk jobb egérgombbal, majd a gyorsmenüben válasszuk ki az Update Web Reference menüpontot. Ez frissíti a szerverre vonatkozó információt, újra lekéri a webszolgáltatás leírását a webszervertől.

# Johanyák Zsolt Csaba: XML Webszolgáltatás alapú osztott alkalmazás fejlesztése (Mandelbrot halmaz számítás)– oktatási segédlet

<http://www.johanyak.hu>

e-mail: [johanyak.csaba@gamf.kefo.hu](mailto:johanyak.csaba@gamf.kefo.hu)

Copyright © 2008 Johanyák Zsolt Csaba



Helyes beállítások esetén némi várakozás után az alábbiakhoz hasonló ábrát kapunk.

