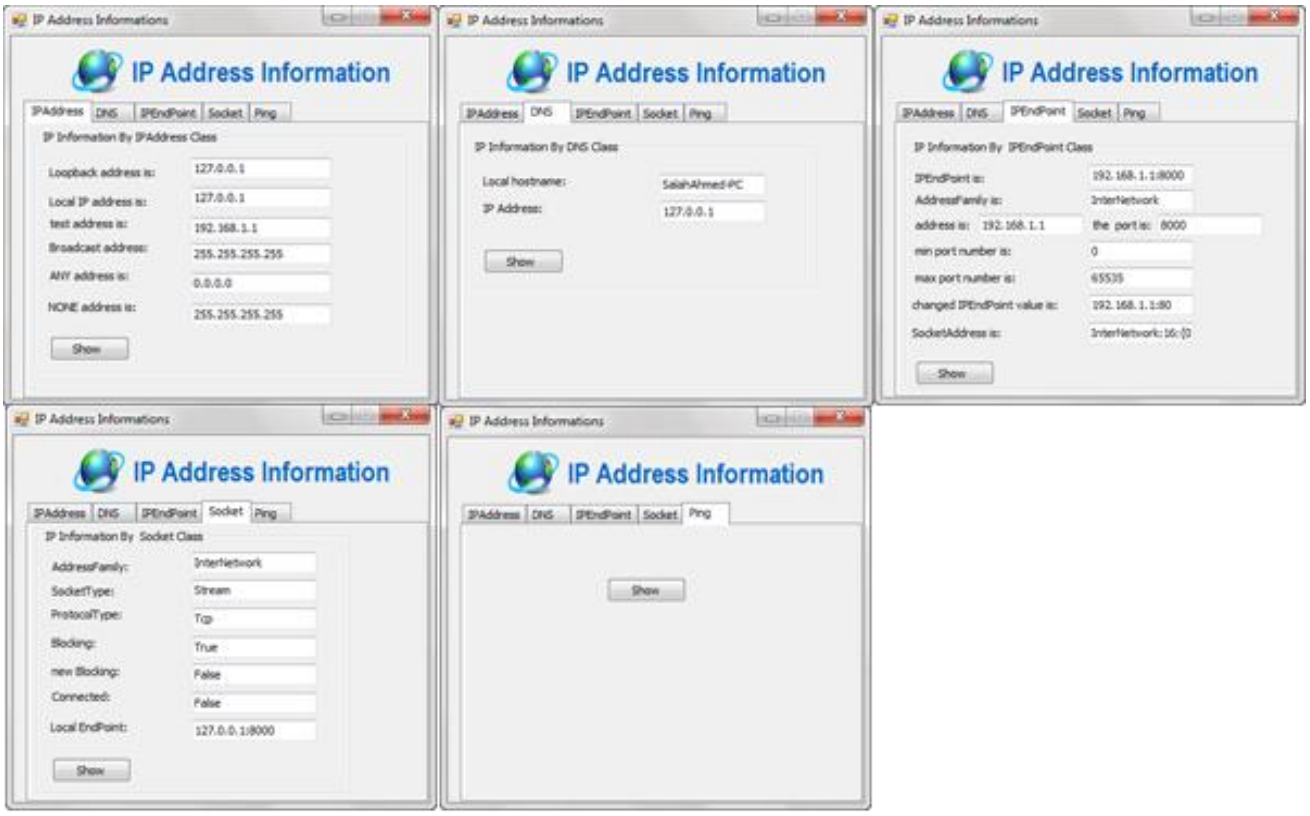
IP ADDRESS INFORMATION USING C#

صلاح أحمد البعداني

سنتكلم عن برنامج الذي هو متخصص في عرض معلومات عن IP Address

يتكون البرنامج من واجهة واحدة فقط من خلال عدة تبويبات موجودة في الواجهة , و الشكل التالي يوضح نظرة عامة على البرنامج بشكل مختصر .



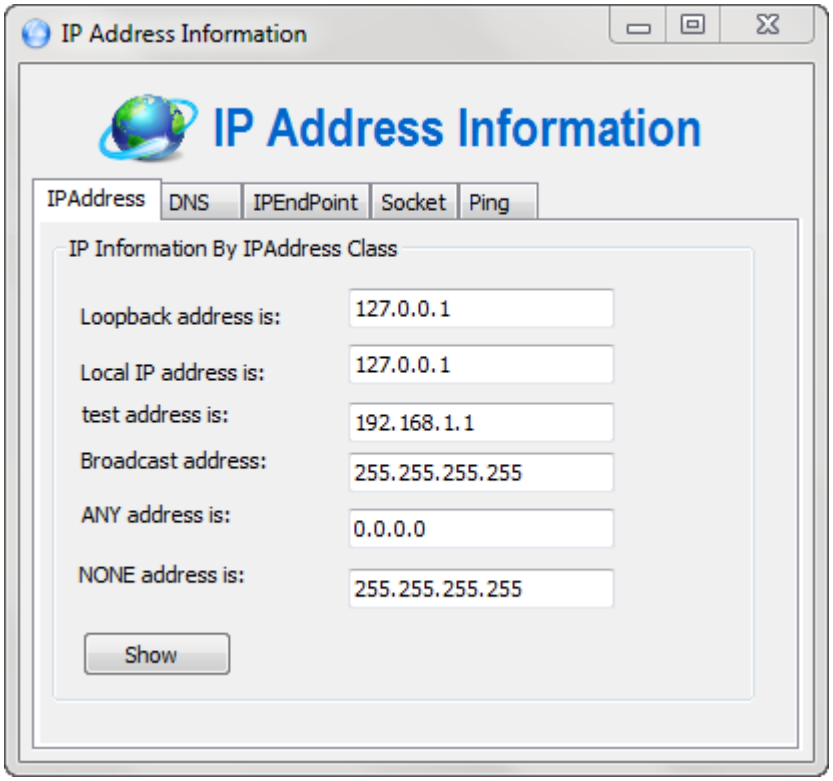
البرنامج يعرض معلومات عن IP Address بإستخدام عدة كلاسات منها :

1- IPAddress Class
2- DNS Class
3- IPEndPoint Class
4- Socket Class

نأتي الآن لشرح عمل هذه الكلاسات بشكل من التفصيل

- **بإستخدام IPAddress Class**

✓ الواجهة المستخدمة لعرض المعلومات عبر هذا الكلاس بشكل التالي :



✓ الأدوات الأساسية في الواجهة

م	الأداة	خصائص الأداة	
		Name	Text
1	TextBox	texIP1	
2	TextBox	texIP2	
3	TextBox	texIP3	
4	TextBox	texIP4	
5	TextBox	texIP5	
6	TextBox	texIP6	
7	Button	ShowIPAddress	Show

✓ الأوامر المستخدمة في الواجهة :

تحتوى الواجهة على أمر واحد يعمل على إظهار المعلومات في الأدوات الموضحة في الجدول السابق , يتم تنفيذ هذا الأمر عند الضغط على الزر التالي :

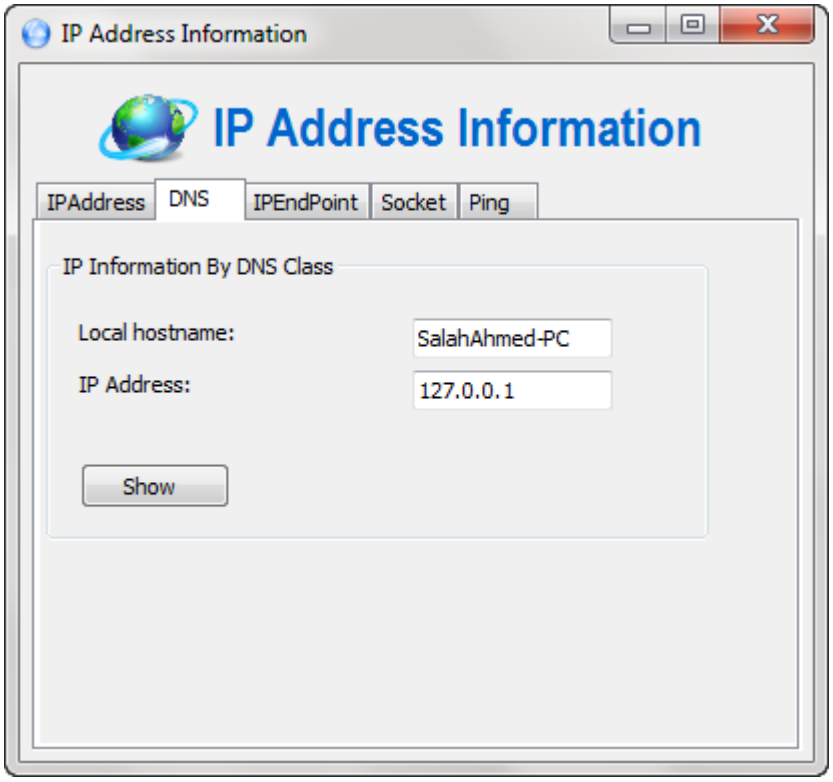
```
IPAddress test1 = IPAddress.Parse("192.168.1.1");
        IPAddress test2 = IPAddress.Loopback;
        IPAddress test3 = IPAddress.Broadcast;
        IPAddress test4 = IPAddress.Any;
        IPAddress test5 = IPAddress.None;
```

✓ يحتوى الكود السابق على بعض الأوامر المندرجه تحت هذا الكلاس وهي التي ترجع معلومات عن IP Address موضحة في الجدول التالي :

م	الأمر	وصفة
1	IPAddress.Parse()	تعمل على تحويل النص إلى كائن من نوع IPAddress
2	IPAddress.Loopback	يستعمل لتمثيل العنوان loopback للنظام
3	IPAddress.Broadcast	يستعمل لتمثيل عنوان IP broadcast للشبكة المحليّة
4	IPAddress.Any	يستعمل لتمثيل أيّ عنوان IP متوفر على النظام المحليّ
5	IPAddress.None	لن يمثل أي شيء

- **بإستخدام DNC Class**

✓ الواجهة المستخدمة لعرض المعلومات عبر هذا الكلاس بشكل التالي :



✓ الأدوات الأساسية في الواجهة

م	الأداة	خصائص الأداة	
		Name	Text
1	TextBox	texDNS1	
2	TextBox	texDNS2	
3	Button	ShowDNS	Show

✓ الأوامر المستخدمة في الواجهة :

تحتوى الواجهة على أمر واحد يعمل على إظهار المعلومات في الأدوات الموضحة في الجدول السابق , يتم تنفيذ هذا الأمر عند الضغط على الزر التالي :

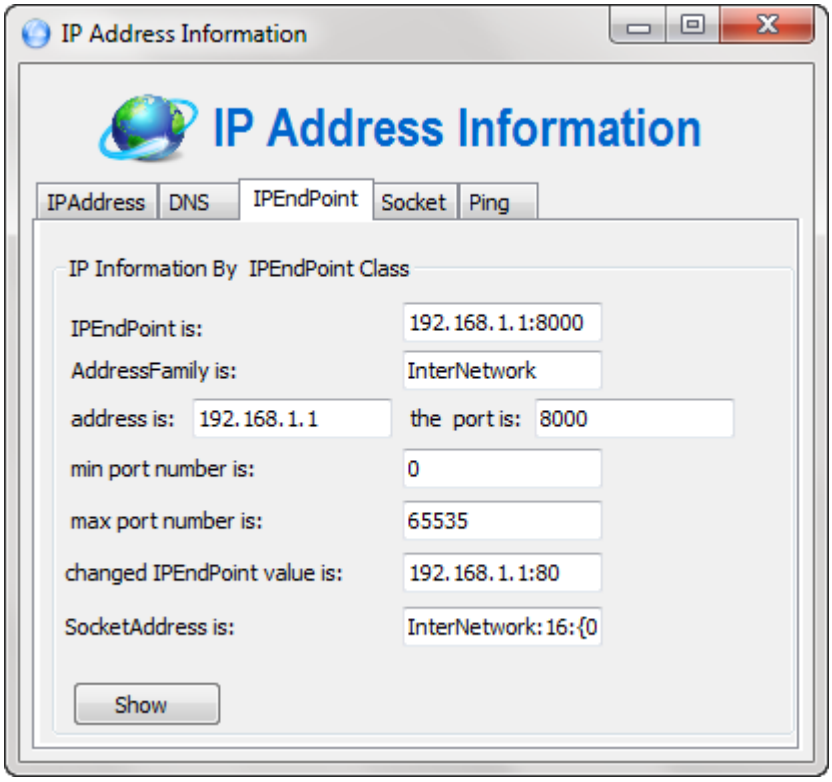
```
string hostName = Dns.GetHostName();
            texDNS1.Text= hostName;
            IPHostEntry myself =
Dns.GetHostByName(hostName);
            foreach (IPAddress address in
myself.AddressList)
            {
                texDNS2.Text= address.ToString();
            }
```

✓ يحتوى الكود السابق على بعض الأوامر المندرجه تحت هذا الكلاس وهي التي ترجع معلومات عن IP Address موضحة في الجدول التالي :

م	الأمر	وصفة
1	`Dns.GetHostName();`	سترجع اسم مضيّف النظام المحليّ
2	`Dns.GetHostByName(hostName);`	يحاول إيجاد عنوان IP باستخدام اسم مضيّف
3	`myself.AddressList`	مصفوفة من نوع IPAddress ترجع جميع العنوان حسب اسم المضيف

- **بإستخدام IPEndPoint Class**

✓ الواجهة المستخدمة لعرض المعلومات عبر هذا الكلاس بشكل التالي :



✓ الأدوات الأساسية في الواجهة

م	الأداة	خصائص الأداة	
		Name	Text
1	TextBox	texPoint1	
2	TextBox	texPoint2	
3	TextBox	texPoint3	
4	TextBox	texPoint4	
5	TextBox	texPoint5	
6	TextBox	texPoint6	
7	TextBox	texPoint7	
8	TextBox	texPoint8	
9	Button	ShowIPEndPoint	Show

✓ الأوامر المستخدمة في الواجهة :

تحتوى الواجهة على أمر واحد يعمل على إظهار المعلومات في الأدوات الموضحة في الجدول السابق , يتم تنفيذ هذا الأمر عند الضغط على الزر التالي :

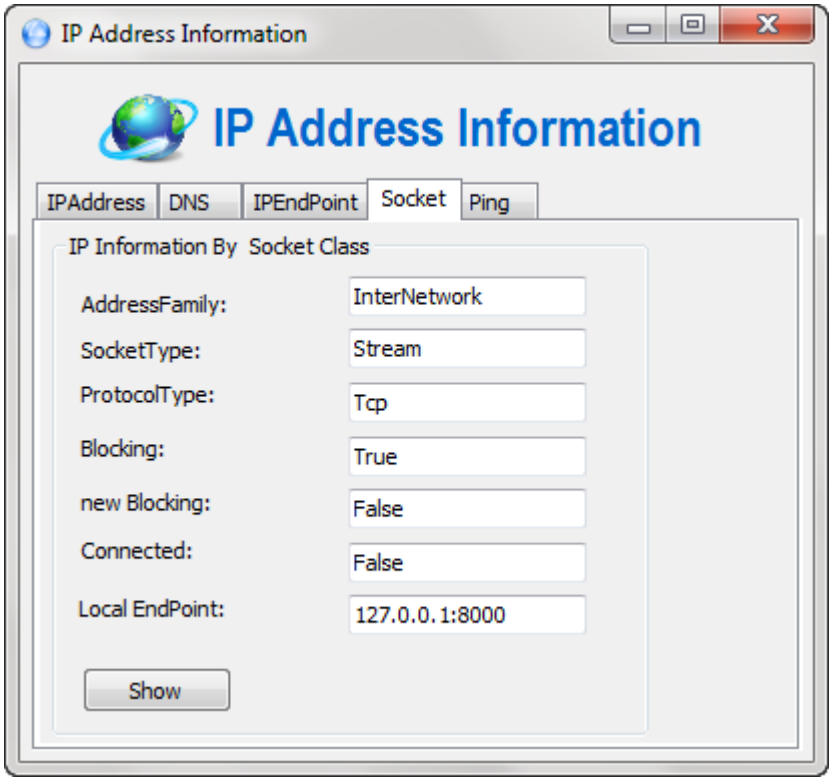
```
IPAddress test1 = IPAddress.Parse("192.168.1.1");
            IPEndPoint ie = new IPEndPoint(test1, 8000);
            texPoint1.Text= ie.ToString();
            texPoint2.Text= ie.AddressFamily.ToString();
            texPoint3.Text= ie.Address.ToString();
            texPoint4.Text= ie.Port.ToString();
            texPoint5.Text= IPEndPoint.MinPort.ToString();
            texPoint6.Text= IPEndPoint.MaxPort.ToString();
            ie.Port = 80;
            texPoint7.Text= ie.ToString();
            SocketAddress sa = ie.Serialize();
            texPoint8.Text= sa.ToString();
```

✓ يحتوى الكود السابق على بعض الأوامر المندرجه تحت هذا الكلاس وهي التي ترجع معلومات عن IP Address موضحة في الجدول التالي :

م	الأمر	وصفة
1	IPEndPoint ie	كائن لتمثيل عنوان آي بي المعيّن / و رقم Port . كائن Ipendpoint يستعمل عندما ربط Socket إلى العناوين المحليّة، أو متى اتصال Socket إلى العناوين البعيدة.
2	ie.AddressFamily	يعيد عائلة عنوان الأي بي
3	ie.Address	يعيد و يعين عنوان الأي بي
4	ie.Port	يعيد و يعين رقم Port
5	IPEndPoint.MinPort	أصغر قيمة يمكن تخصيصها لرقم Port
6	IPEndPoint.MaxPort	أكبر قيمة يمكن تخصيصها لرقم Port

- **بإستخدام Socket Class**

✓ الواجهة المستخدمة لعرض المعلومات عبر هذا الكلاس بشكل التالي :



✓ الأدوات الأساسية في الواجهة

م	الأداة	خصائص الأداة	
		Name	Text
1	TextBox	texSocket1	
2	TextBox	texSocket2	
3	TextBox	texSocket3	
4	TextBox	texSocket4	
5	TextBox	texSocket5	
6	TextBox	texSocket6	
7	TextBox	texSocket7	
8	Button	ShowSocket	Show

✓ الأوامر المستخدمة في الواجهة :

تحتوى الواجهة على أمر واحد يعمل على إظهار المعلومات في الأدوات الموضحة في الجدول السابق , يتم تنفيذ هذا الأمر عند الضغط على الزر التالي :

```
IPAddress ia = IPAddress.Parse("127.0.0.1");
          IPEndPoint ie = new IPEndPoint(ia, 8000);
          Socket test = new Socket(AddressFamily.InterNetwork,
                 SocketType.Stream, ProtocolType.Tcp);
          texSocket1.Text= test.AddressFamily.ToString();
          texSocket2.Text= test.SocketType.ToString();
          texSocket3.Text=  test.ProtocolType.ToString();
          texSocket4.Text= test.Blocking.ToString();
          test.Blocking = false;
          texSocket5.Text= test.Blocking.ToString();
          texSocket6.Text= test.Connected.ToString();
          test.Bind(ie);
          IPEndPoint iep = (IPEndPoint)test.LocalEndPoint;
          texSocket7.Text= iep.ToString();
          test.Close();
```

✓ يحتوى الكود السابق على بعض الأوامر المندرجه تحت هذا الكلاس وهي التي ترجع معلومات عن IP Address موضحة في الجدول التالي :

م	الأمر	وصفة
2	test.AddressFamily	يعيد عائلة الأي بي
3	test.SocketType	يعيد نوع البيانات المستخدمة في الأتصال
4	test.ProtocolType	يعيد نوع البروتوكول المستخدمة للإرسال
5	test.Blocking	يحدد إذا كان socket في نمط blocking
6	test.Connected	تتصل بالأجهزة البعيدة
7	test.LocalEndPoint	يعيد عنوان الجهاز المحلي
8	test.Close();	تقوم بإغلاق Socket