

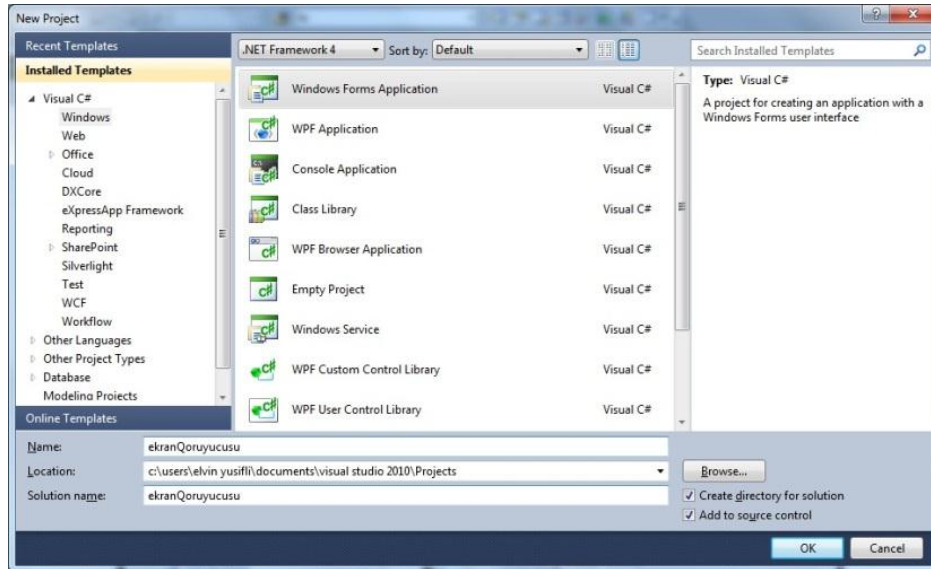
C# ilə Ekran Qoruyucusu Yazmaq.

Salam əziz dostlar!

Bu məqaləmdə sizlərə Windows əməliyyat sistemi üçün ekran qoruyucusu yazmağın mexanizmini izah etməyə çalışacam. İnanıram ki, bu məqalə sizin üçün lazımlı istinad mənbəyi olacaqdır.

Ekran qoruyucusu yazmaq üçün ilk öncə Visual Studio 2010 (və ya digər versiyalar) ekrana gətirilir.

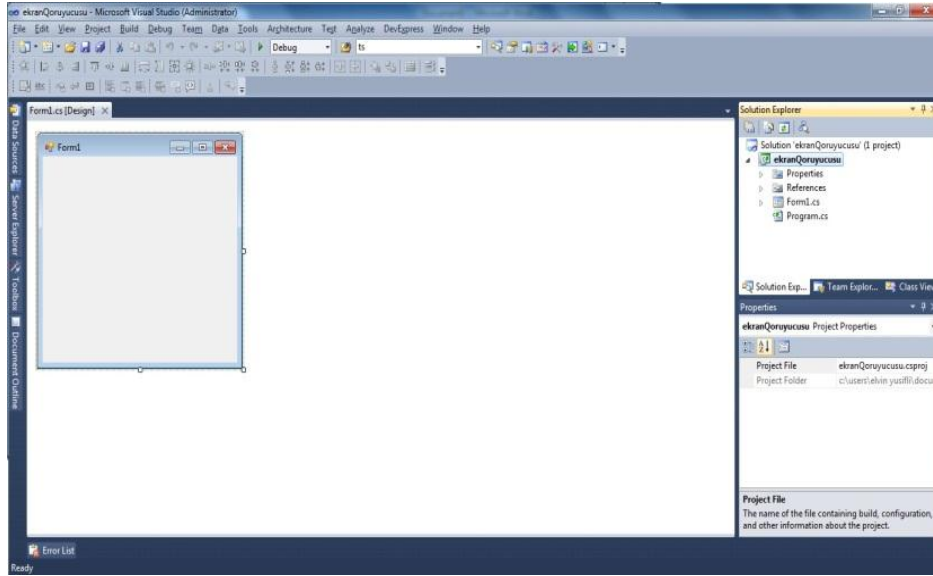
File ->New->Project açdıqdan sonra **Visual C#->Windows->Windows Form Application** seçirik.



Name: ekranQoruyucusu (Bu bölməyə layihəmizin adını daxil edirik).

Solution name: ekranQoruyucusu (Bu bölməni istədiyiniz halda dəyişdirə bilərsiniz, layihə adı ilə eyni olmaya da bilər).

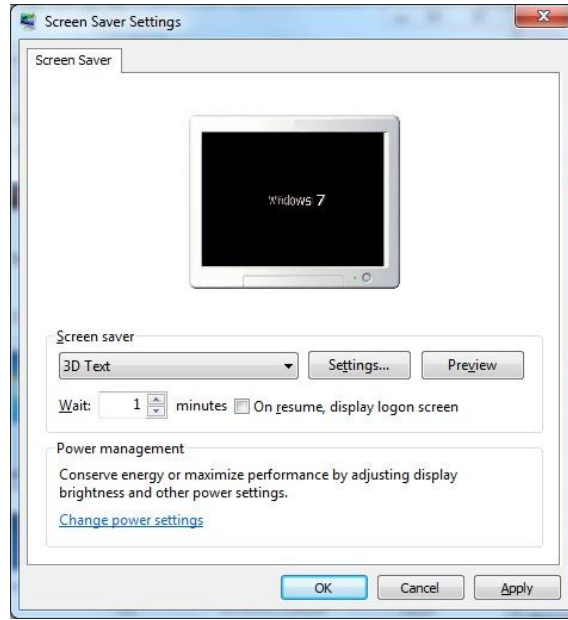
Bu əməliyyatları həyata keçirdikdən sonra **Ok** düyməsinə basın.



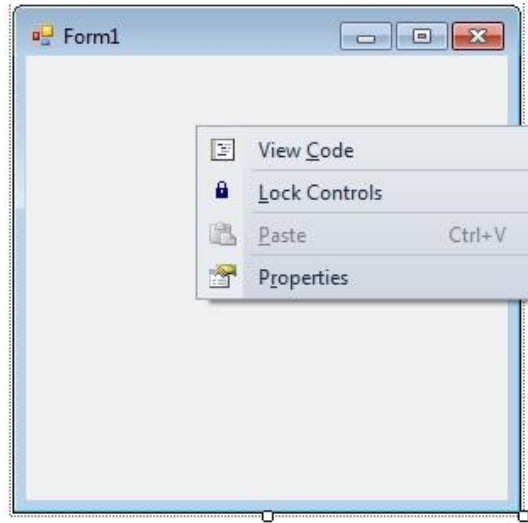
Bundan sonra Visual Studio bizim üçün boş bir Form yaradacaq. Siz **Form1** seçərək istədiyiniz halda **Form1 Name** və **Text** özəlliklərini dəyişə bilərsiniz. (Bu məqaləmizdə biz dəyişmədən davam edəcəyik) .

Bir anlıqda olsa, ekran qoruyucusunu göz önünə gətirib onun nədən ibarət olduğunu təsəvvür edərk. Ekran qoruyucusu açılan zaman bütün ekranı tutur və ekranda hərəkət edən fiqurlar və ya yazılar hərəkət etməyə başlayır. Bunların hamısı bir **Form** üzərində baş verir. Bəzən Maus hərəkət etdirən zaman Ekran Qoruyucusunun yox olduğunu müşahidə edirik. Ekran qoruyucusunu gözdən keçirmək üçün

Start->Control Panel->Appearance and Personalization->Personalization->Screen Saver (Windows 7 üçün) daxil olun.



Bizim ilk görəcəyimiz iş **Form1**-in açılışı zamanı nələrin görünüb görünməyəcəyini təyin etmək olacaq. Bunun üçün **Form1** kod arxasına keçirik.



Form1 üzərində sağ düyməni vuraraq **View Code** daxil oluruq.

```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;
```

```

using System. Drawing;
using System. Linq;
using System. Text;
using System. Windows.Forms;

namespace ekranQoruyucusu
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            //Form1 komponentlərinin yüklənməsi və Form1 qurulması
            InitializeComponent();
        }
    }
}

```

Kod bölməsinə daxil olan zaman ilk gördüyümüz kodlar bundan ibarət olacaq. Burdan kodlarımızı yazmağa başlayırıq. İlk öncə biz Form1 kənarlarını görünməz etməliyik.

FormBorderStyle. None-Bu əmr bizə Form1-in kənarlarını boş şəkildə göstərir.

FormWindowState.Maximized– Bu əmr bizə Form1 tam ekran açmağa imkan verir.

ShowInTaskbar- Bu əmr bizə Form1-in Taskbarda görünüb görünməməsini təyin etməyə imkan verir. Əgər True seçilibsə, görünəcək False seçilibsə, görünməyəcək.

TopMost- Bu əmr bizə Form1-in açıq olan bütün proqramlardan öndə qalmasına imkan yaradır. **True** olan zaman öndə qalır.

Cursor. Hide()- Form1 açılan zaman Mausun işarəsini gizlətmək üçün istifadə olunur.

Yuxarıda yazılan bizə lazım olan əmrləri Form1-lə tətbiq edirik.

```

namespace ekranQoruyucusu
{
    public partial class Form1 : Form
    {
        public Form1()

```

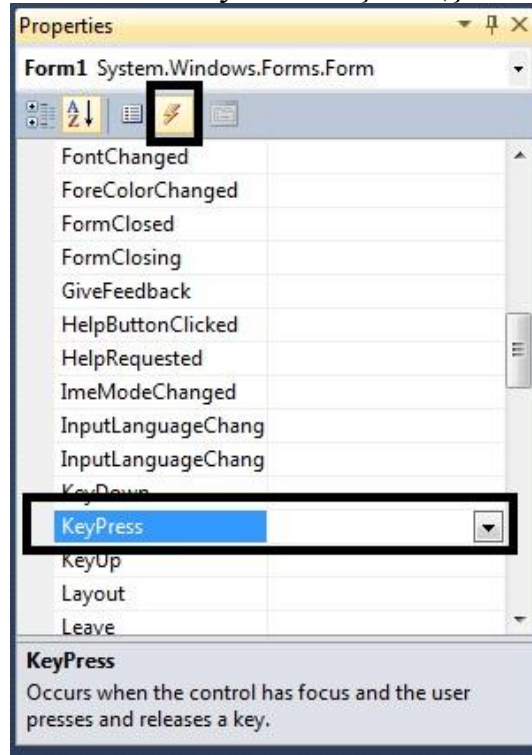
```

{
    InitializeComponent();
    //Form1 kənarları boş formada görsənsin.
    FormBorderStyle =FormBorderStyle.None;
    //Form1 açılan zaman bütün ekranı tutsun.
    WindowState = FormWindowState.Maximized;
    //Ələtlər panelində görsənməsin.
    ShowInTaskbar = false;
    //Bütün proqramlardan öndə açılınsın.
    TopMost = true;
    //Mausun işarəsi gizlədilsin.
    Cursor.Hide();
}
}
}

```

İndi klaviaturadan F5(Run) sıxaraq proqramı işə salırıq. Açılan pəncərə bütün ekranı tutacaq. Bundan çıxmaq üçün Alt-F4 istifadə edin.

İlk əvvəl bu açılan pəncərənin bağlanması üçün kod yazaq. Adətən Ekran qoruyucusu işləyəən zaman klaviaturadan hər hansı bir düymənin sıxılması ilə və ya Mausun hərəkəti ilə qoruyucunun bağlandığını görürük. Bunun üçün **Form1 -> Properties -> Events** daxil olur və KeyPress seçirik.(Şəkil.5)

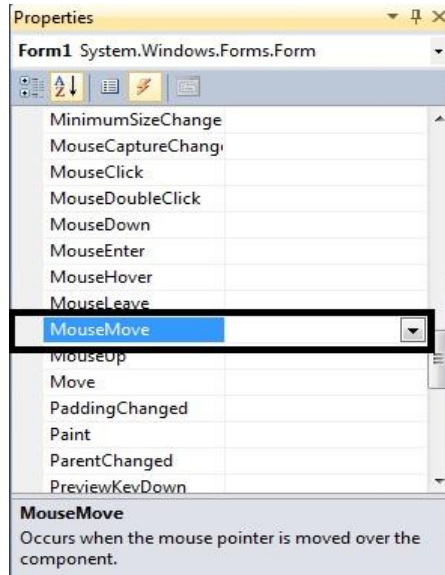


Şəkil.5

KeyPress-ə aşağıdakı kodu yazırıq.

```
private void Form1_KeyPress(object sender, KeyPressEventArgs e)
{
    //Klaviaturadan hər hansı düymə vurularsa, Form1 bağla.
    this.Close();
}
```

Bunu da yazdıqdan sonra keçirik Mausun hərəkəti zamanı Form1 bağlanmasının kodlarını yazmağa. Bunun üçün **Form1 -> Properties -> Events** daxil oluruq və **MouseMove** seçirik.(Şəkil.6)



Şəkil.6

Əgər biz **Form1**-in **MouseMove** eventinə `this.Close()` bunu yazsaq, Form1 açılan kimi bağlanacaq, çünki hər hansı vəziyyətdə olursa olsun, Maus ilə bunu edən zaman Mausun x və y artıb azalır.

```
private void Form1_MouseMove(object sender, MouseEventArgs e)
{
    //Maus hərəkət edən zaman Form1 bağla.
    this.Close();
}
```

Siz yuxarıda yazılanı edib sonra layihəni Run edin. Əgər bunu Mausla etmiş olsanız, Form1 açılan kimi bağlanacaq. Klaviaturadan F5 vuraraq proqramı işə salın. Bu zaman proqram işləyəcək. İndi fikirləşirik: bunu necə edə bilərik ki, Mausla açan zaman Form1 bağlanmasın. Bunun üçün biz Form1 açılan zaman Mausun koordinatlarını götürməliyik və bu koordinatları artıraraq, həmin koordinatdan çıxan zaman Form-u bağlamalıyıq.



Şəkil.7

Qırmızı nöqtə bizim proqram açılan zaman olan koordinatımızdır. Biz indi elə etməliyik ki, çevrədən qırmızı nöqtədən hərəkət edən zaman çevrə kənarına çıxanda proqramımız bağlansın, yəni şəkildə göstərilən sarı hissəyə keçən zaman (Şəkil.7). Bunun üçün aşağıdakılardan istifadə edirik.

//Mausun ilk koordinatlarını götürmək üçün.

Point mausunIlkYeri;

//Mausun aktivliyini yoxlamaq üçün

bool yoxlama = false;

private void Form1_MouseMove(object sender, MouseEventArgs e)

{

if (!yoxlama)

{

//Mausun ilk koordinatları

mausunIlkYeri = MousePosition;

//İlk koordinatları götürdükdən sonra aktivliyi True edirik.

yoxlama = true;

}

else if (Math.Abs(MousePosition.X - mausunIlkYeri.X) > 20 ||

Math.Abs(MousePosition.Y - mausunIlkYeri.Y) > 20)

{

this.Close();

```
}  
}
```

Aşağıdakı kod hissəsində yerinə yetirdiyimiz əməliyyatlara nəzər salaq.

```
else if (Math.Abs(MousePosition.X - MausunIlkYeri.X) > 20 ||  
Math.Abs(MousePosition.Y - MausunIlkYeri.Y) > 20)
```

Bu hissədə `MousePosition.X` (indi dayandığı X koordinatı) çıxırıq `Mausun` əvvəlki koordinatına. Əgərə koordinat 20-dən böyükdürsə və ya `MousePosition.Y` (indi dayandığı Y koordinatı) çıxırıq, əvvəlki Y koordinatın 20-dən böyükdürsə, onda Formanı bağla (`this.Close()`)əmrini icra edirik.

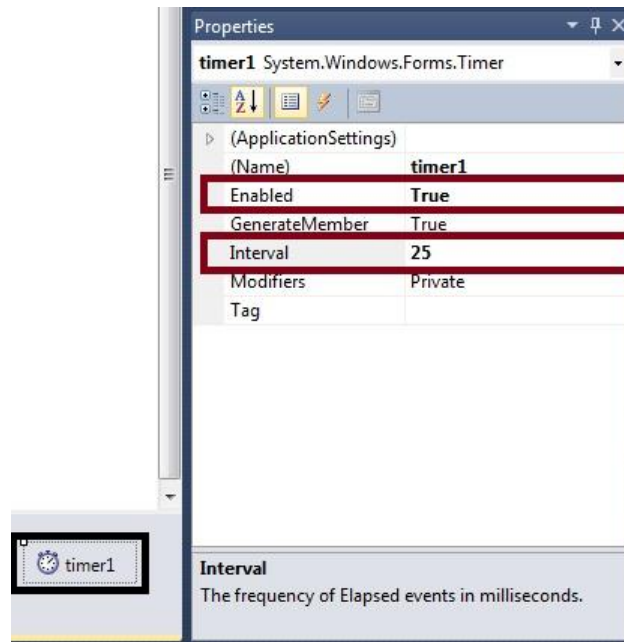
`Math.Abs` bu funksiya sayəsində istədiyimiz bir mənfi ədədin müsbət dəyərini tapa bilərik. Məsələn, `Math.Abs(-125) = 125`

İndi **Form1** üzərinə bir **Label** əlavə edirik və **Label1-in** mətnini dəyişib istənilən mətn daxil edirik (Siz digər parametrlərdən də istifadə edib dəyişikliklər edə və ya rəng seçə bilərsiniz(Şəkil.8).



Şəkil.8

Form1 formamıza bir **Timer** əlavə edir və **timer1-in Enable=true** düyməsinə basırıq. Form açılan kimi işləməyə başlayır və **Intervala** qiymət daxil edirik.(Şəkil.9)



Şəkil.9

Bunları təyin etdikdən sonra **Timer1**-in üzərində Mausla sıxaraq kod arxasına keçirik və **timer1_Tick**-inə aşağıdakı kodları yazırıq.

```
//X oxu üzrə hərəkəti təyin etmək üçün.
bool hərəkət =true;
//X oxunda hansı intervalla hərəkət etsin.
int x = 3;
```

```
private void timer1_Tick(object sender, EventArgs e)
{
```

```
    // Hərəkət true alırsa, sola hərəkət etsin.
```

```
    if (hərəkət == true)
```

```
    {
```

```
        label.Left=label.Left-x;// Bu vəziyyətdə label sola doğru hərəkət edəcək.
```

```
    // Əgərə label-in soldan məsafəsi 0-a bərabədirsə, onda istiqamət dəyişib sağa doğru hərəkət etsin:
```

```
    if (label.Left == 0)
```

```
    {
```

```
        // Label saga doğru hərəkət etsin.
```

```
        hərəkət= false;
```

```
    }
```

```
}
```

```

// yox əgər false dəyər alırsa, sağa doğru hərəkət etsin.
else
{
    // Bu vəziyyətdə isə Label sağa doğru hərəkət edəcək.
    label.Left=label.Left+x;

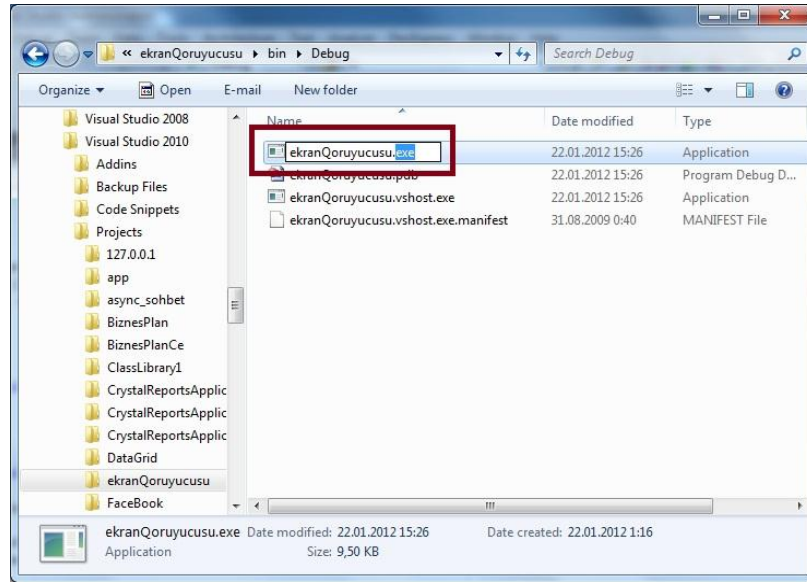
    // Label sağa çatan anda istiqamət dəyişib sola doğru hərəkət etsin.
    if (label.Left == (this.Width - label.Width))

        //Sola doğru hərəkət etsin.
        hərəkət= true;
}
}

```

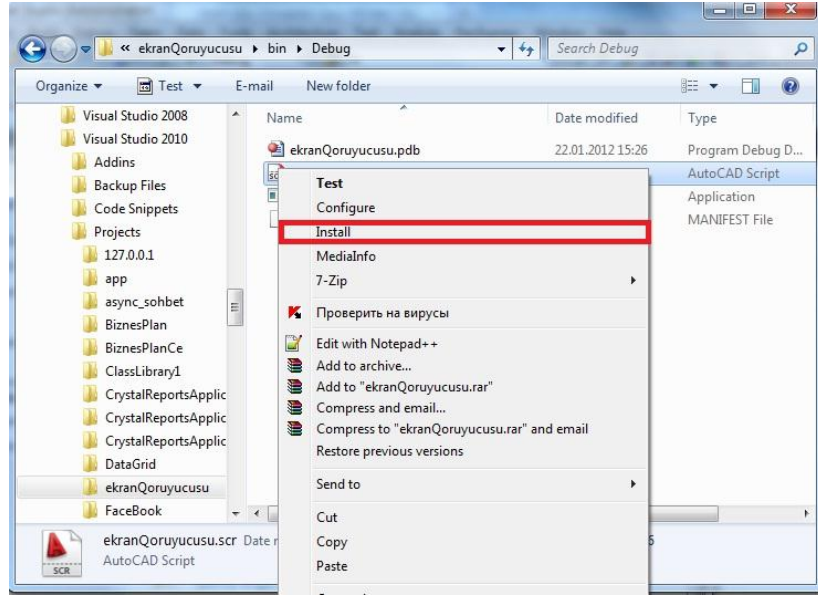
Artıq proqramı işə salaraq görə bilərik ki, label sağdan sola, soldan sağa hərəkət etməyə başlayır. Sola çatan anda istiqamət dəyişərək geriye sağa doğru hərəkət etməyə başlayır və əksinə.

İndi proqramın yerləşdiyi qovluğu açıyıq və orda yaranmış olan ekranQoruyucusu.exe uzantısını dəyişib src edirik, yəni ekranQoruyucusu.src olur.(Şəkil.10)



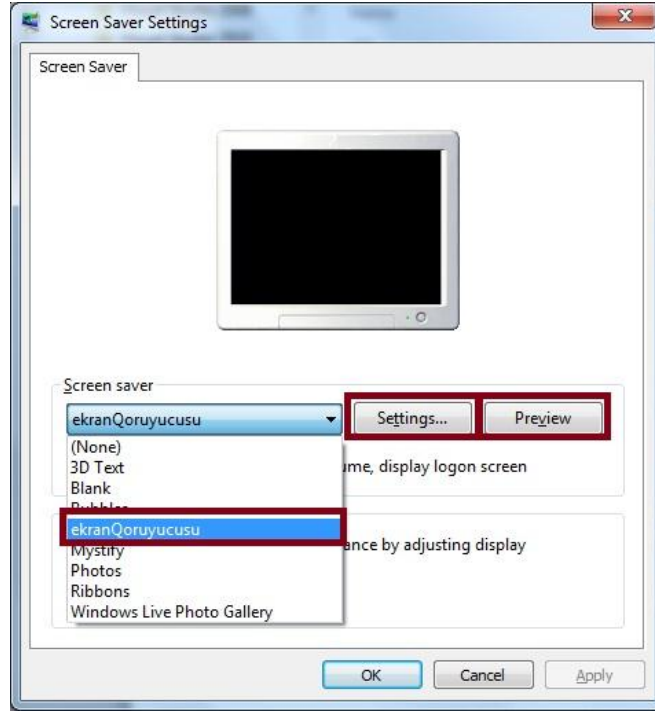
Şəkil.10

Aşağıda göstiriləni yerinə yetirib Ekran qoruyucusun İnstall edirik.(Şəkil.11)



Şəkil.11

İnstall etdikdən sonra artıq ekran qorucuyumuzu Window-sun ekran qoruyucuları sırasında görə bilirik.(Şəkil.12)

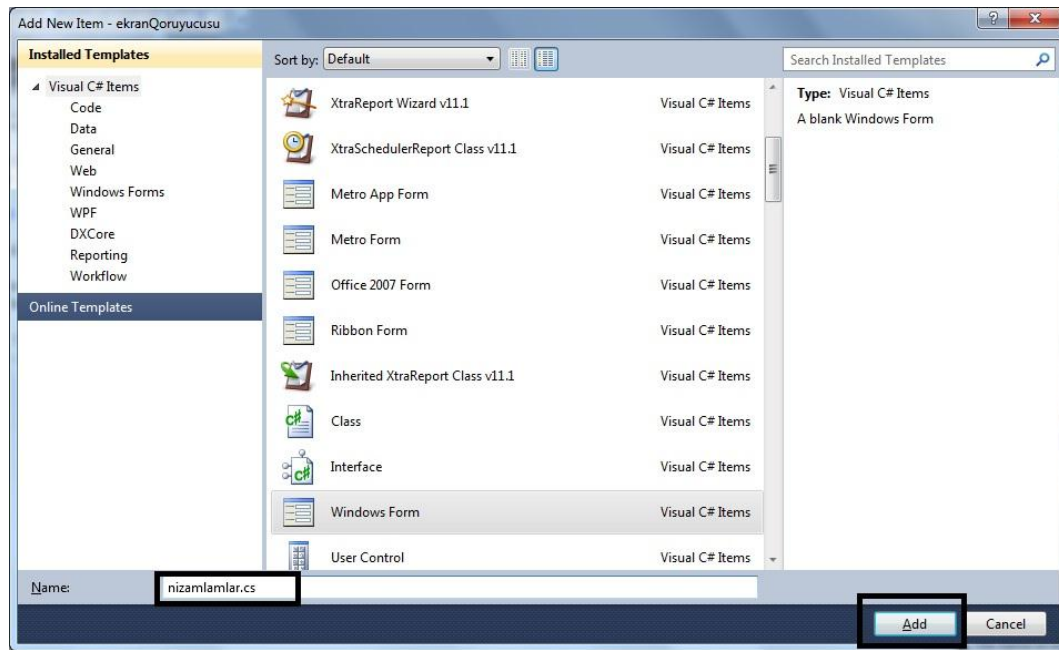


Şəkil.12

Əgər fikir versək görərik ki, digər ekran qoruyucularında **Settings** və **Preview** vuran zaman bir funksiya icra edilir. Ekran qoruyucusuna düzəlişlər etmək, rəng seçmək, yazı dəyişmək kimi düzəlişlər **Settings** bölməsindən edilir. Gəlin biz də kodlarımızda bunları nəzərə alaq. **WindowsunSettings**-ə və ya **Preview**-a vuran zaman bizə göndərmiş olduğu məlumatları **Program.cs** ilə alıb bunun üzərindən kodlarımızı yazmağa davam edəcəyik.

İlk əvvəl ekran qoruyucusuna nizamlamaları çağırmaq üçün layihəmizə yeni bir Form əlavə edirik. Formun adını Nizamlamalar kimi təyin edirik.

Bunun üçün siyahıdan **Project->Add Windows Forms** seçib yeni forma ad verib **Add** edirik.(Şəkil.13)



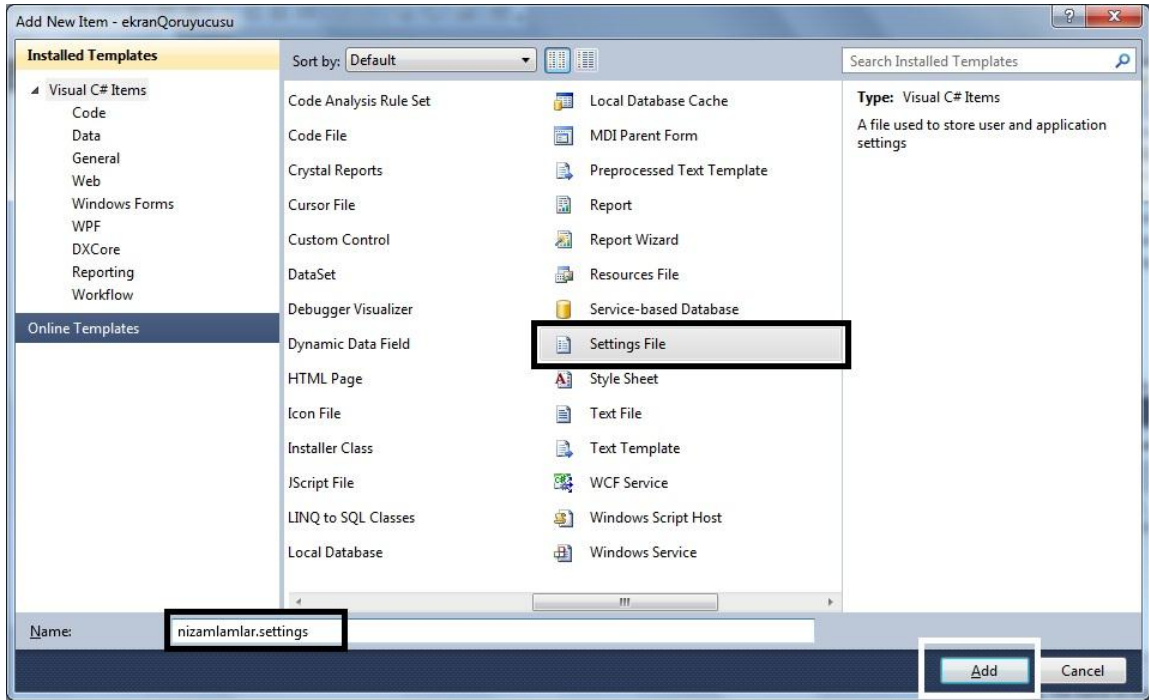
Şəkil.13

Əlavə etdiyimiz forumu aşağıda göstərilən şəkildə hazırlayırıq.(Şəkil.14)



Şəkil.14

Yuxarıda gördüyünüz parametrləri bizim hardasa saxlamağımız və hardansa oxuyub ekrana gətirməyimiz lazımdır. Bunun üçün Visual Studionun imkanlarından istifadə edirik. Belə ki, layihəmizə bir **Project->Add New Item->Settings File** seçib adını daxil edib **Add** edirik .(Şəkil.15)



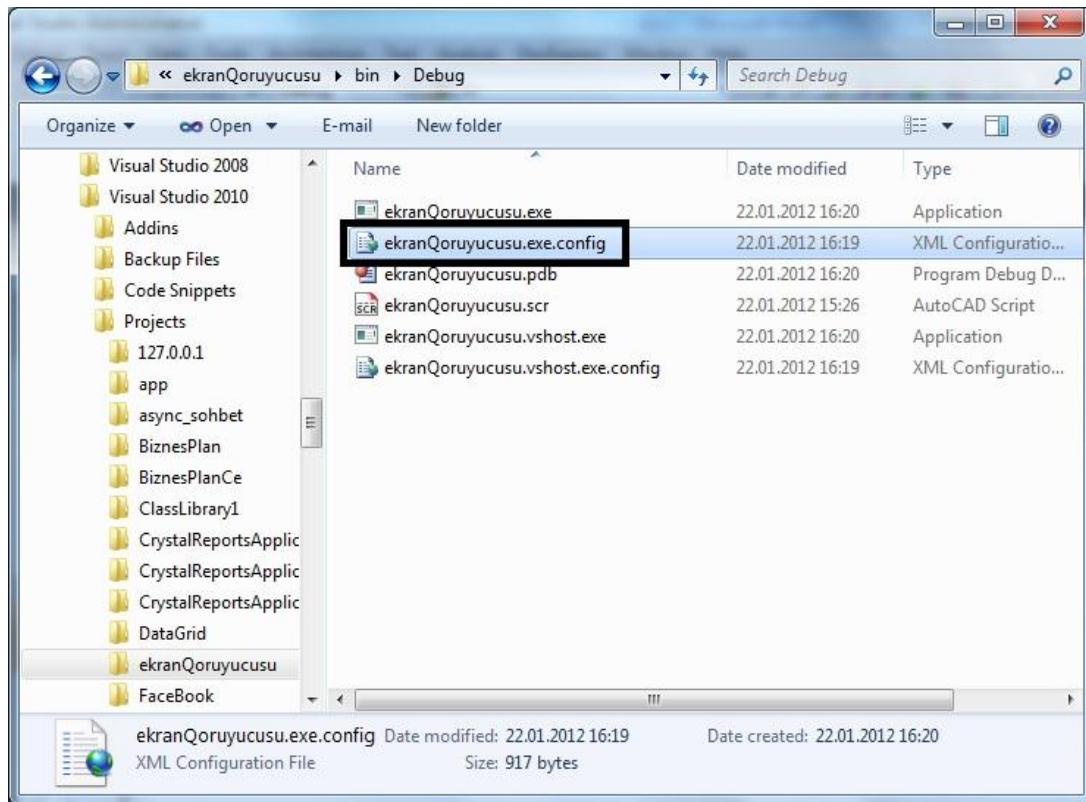
Şəkil.15

Bunu etdikdən sonra qarşımıza aşağıdakı pəncərə gələcək. Bizim yaddaşda saxlayacağımız iki parametir var: bunlardan biri **mətn**, digəri isə **rəngdir**. Bunun üçün aşağıda göstərilənləri edirik.(Şəkil.16)

Name	Type	Scope	Value
metn	string	User	Test
reng	System.Drawing.Color	User	Red
*			

Şəkil.16

Burada **mətn** mətn formasında olduğuna görə **Type String**, **reng** isə öz **Type-na** uyğun olaraq, **Color** seçirik və Save edirik. İlk başdan mətni və ya rəngi daxil etməyiniz məcburi deyil. Bunu yaratdıqdan sonra biz Debug qovluğunun içinə baxdığımız zaman **ekranQoruyucusu.exe.config**-adında bir fayl yarandığını görmüş olacağıq.(Şəkil.17)



Şəkil.17

Bizim nizamlamalar **ekranQoruyucusu.exe.config** faylı içində saxlanılacaq. **Scr** faylın götürən zaman mütləq **ekranQoruyucusu.exe.config** faylı ilə birlikdə götürmək lazımdır, əks təqdirdə bu parametrlər yaddaşa yazılmayacaq.

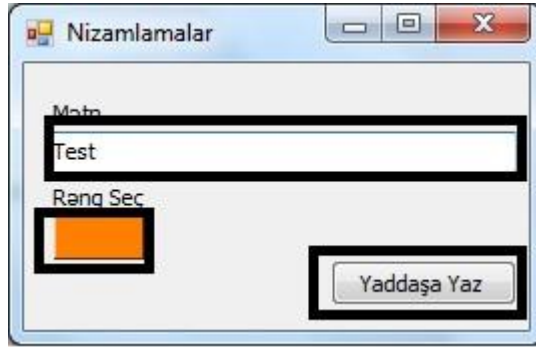
İndi biz **Form1** yüklənən zaman **settings** faylından məlumatları alıb **Label** və **Form** tətbiq edək. Bunun üçün **Form1 Load**-na aşağıdakı kodu yazırıq.

```
private void Form1_Load(object sender, EventArgs e)
{
    //Settings-dəki mətni oxuyub label-ə yazırıq.
    label.Text = nizamlamlar11.Default.metn;
    //Settings-dəki rəngi oxuyub form-un arxa fonuna yazırıq.
    this.BackColor = nizamlamlar11.Default.reng;
}
```

Bunları etdikdən sonra keçirik bizim parametrləri saxlamaq və baxmaq üçün hazırlamış olduğumuz **nizamlamar** form-una(Şəkil.18).

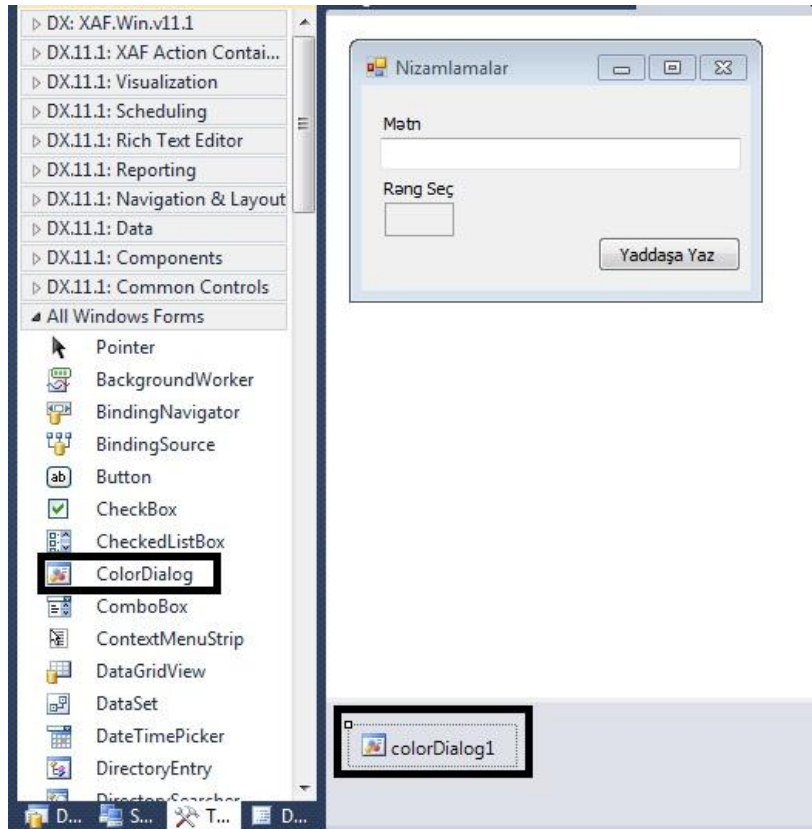
İlk əvvəl **nizamlamar** forumunun load-na aşağıdakı kodları yazırıq.

```
private void nizamlamlar_Load (object sender, EventArgs e)
{
    metn.Text = nizamlamlar11.Default.metn;//Textboxa parametrlərin çağırılması.
    reng.BackColor = nizamlamlar11.Default.reng;//Rəngin göstərilməsi.
}
```



Şəkil.18

Rəng seçmək üçün **Nizamlamlar** forumuna **Toolbox-dancolorDialog1** əlavə edirik(Şəkil.19).



Şəkil.19

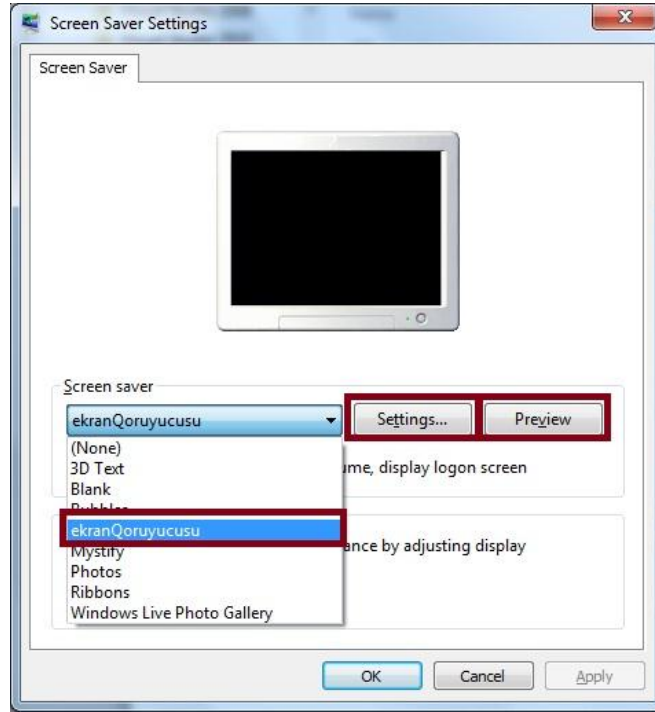
Rəng seçimi üçün aşağıdakı kodları daxil edirik. Burda `colorDialog1.ShowDialog()` bizə rəng pəncərəsini açır, əgər hansısa rəngi seçib OK ediriksə, onda `reng.BackColor = colorDialog1.Color` bərabər edirik.

```
private void reng_Click(object sender, EventArgs e)
{
    if (colorDialog1.ShowDialog() == DialogResult.OK)
    {
        //Rəngin arxa fonunu seçilən rəng edirik.
        reng.BackColor = colorDialog1.Color;
    }
}
```

İndi isə Yaddaşa Yaz üçün kodunu yazaq. Biz burda Mətnə daxil edilən və seçilən rəngi settings faylına yazmalıyıq. Bunun üçün aşağıdakı koddan istifadə edirik.

```
private void yaddasayaz_Click(object sender, EventArgs e)
{
    //Daxil edilən mətni settings faylına yazırıq
    nizamlamlar11.Default.metn=metn.Text;
    //Seçilən rəngi settings faylına yazırıq
    nizamlamlar11.Default.reng=reng.BackColor;
    //Məlumatları settings faylında saxla
    nizamlamlar11.Default.Save();
}
```

İndi isə Setting və Preview vuran zaman icra olunacaq əməllərə baxaq (Şəkil.20).



Şəkil.20

Burda **Settings**-ə vuran zaman bizə **/configure**, **Preview** vuran zaman isə **/show** parametrlərini göndərir. Biz bunu kodumuzda SubString edərək, ilk iki simvolu götürəcəyik(Şəkil.21).

/c configure

/p preview

/p:123

/c:1122

/S 1234

SubString(0,2)

ToLower()

Şəkil.21

Biz kodlarımızı **Program.cs** içində yazırıq.

```
static void Main(string[] parametr)
{
    //Əgər Windows bizə parametr göndərsə
    if (parametr.Count() > 0)
    {
        //Windowsun bizə göndərdiyi parametrlər
        string yeniParametr = parametr[0].Substring(0, 2).ToLower();
        switch (yeniParametr)
        {
            case "/c":
                //Əgər Settings vurulubsa Nizamlamaları aç.
                Application.EnableVisualStyles();
                Application.SetCompatibleTextRenderingDefault(false);
                Application.Run(newnizamlamlar());
                break;

            case "/p":
                //Əgər Preview vurulubsa Form1 aç.
                Application.EnableVisualStyles();
                Application.SetCompatibleTextRenderingDefault(false);
                Application.Run(newForm1());
                break;
        }
    }
    else
    {
        //Program adı halda işləyirsə Form1 aç.
        Application.EnableVisualStyles();
        Application.SetCompatibleTextRenderingDefault(false);
        Application.Run(newForm1());
    }
}
```

`string[]` parametr- Windowsun bizə göndərdiyi parametrləri massiv formasında almağa imkan verir.

Bu məqaləmizdə məqsəd, Ekran Qoruyucusu hazırlamaqla yanaşı bəzi funksiyaların istifadə olunması qaydaları ilə tanış oldunuz. İstəsəniz, siz özünüzdə digər funksiyalardan istifadə edərək təkmilləşdirə bilərsiniz.

Bu məqaləmizdə nə öyrəndik:

- 1.Kodlarla işləmə bacarığına yiyələndik (if,else,switch,case ilə işləmək).
- 2.Form-un funksiyalarından istifadə etməklə forumun görünüşünün dəyişdirilməsi.
- 3.Mausun istiqamətinin dəyişməsinə görə əmrlərin icara olunması.
- 4.Tiplərdən istifadə edilməsi (İnt,Bool).
- 5.Timer-dən istifadə etməklə Labelin hərəkət etməsinin icrası.
- 6.Settings File-dən istifadə etmək.
- 7.ColorDialog-dan istifadə etmək.
- 8.SubString funksiyasından istifadə etmək.
- 9.Windows-un bizə göndərdiyi parametrlərdən istifadə etmək.

Elvin Yusifli

Müəllif hüquqları qorunur