

C Sharp Yardimci Metotlar

Tarih, Sayi, Metin, Sifreleme gibi konularda faydali olacak metotlar içeren bir Windows Form Uygulamasi

Table of Contents

Symbol Reference	1
CSharpHelperMethods Namespace	1
CSharpHelperMethods.Properties Namespace	1
Classes	1
Resources Class	1
Settings Class	3
CSharpHelperMethods.YardimciSiniflar Namespace	3
Classes	4
DataTableIslemleri Class	4
DosyaIslemleri Class	6
KisiIslemleri Class	9
MetinIslemleri Class	11
SayiIslemleri Class	16
SifrelemeliIslemleri Class	19
SiralamaIslemleri Class	22
TarihIslemleri Class	24
Classes	29
FrmAnaSayfa Class	29
FrmAnaSayfa.FrmAnaSayfa Constructor	29
FrmAnaSayfa Methods	29
Program Class	30
Files	30
AssemblyInfo.cs	31
CSharpHelperMethods.csproj	32
CSharpHelperMethods.sln	32
DataTableIslemleri.cs	32
DosyaIslemleri.cs	34
FrmAnaSayfa.cs	36
FrmAnaSayfa.Designer.cs	44
KisiIslemleri.cs	56
MetinIslemleri.cs	57
Program.cs	61
Resources.Designer.cs	61
SayiIslemleri.cs	63
Settings.Designer.cs	65
SifrelemeliIslemleri.cs	66

Siralamalslemleri.cs	68
TarihIslemleri.cs	70

Index

a

1 Symbol Reference



1.1 CSharpHelperMethods Namespace

This is namespace CSharpHelperMethods.

Namespaces

Name	Description
Properties (see page 1)	This is namespace CSharpHelperMethods.Properties.
YardimciSiniflar (see page 3)	This is namespace CSharpHelperMethods.YardimciSiniflar.



Classes

	Name	Description
	FrmAnaSayfa (see page 29)	This is class CSharpHelperMethods.FrmAnaSayfa.
	Program (see page 30)	This is class CSharpHelperMethods.Program.

1.1.1 CSharpHelperMethods.Properties Namespace

This is namespace CSharpHelperMethods.Properties.



Classes

	Name	Description
	Resources (see page 1)	A strongly-typed resource class, for looking up localized strings, etc.
	Settings (see page 3)	This is class CSharpHelperMethods.Properties.Settings.

1.1.1.1 Classes

The following table lists classes in this documentation.

Classes

	Name	Description
	Resources (see page 1)	A strongly-typed resource class, for looking up localized strings, etc.
	Settings (see page 3)	This is class CSharpHelperMethods.Properties.Settings.

1.1.1.1.1 Resources Class

A strongly-typed resource class, for looking up localized strings, etc.

Class Hierarchy

CSharpHelperMethods.Properties.Resources

C#

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Resources.Tools.StronglyTypedResourceBuilder",
"4.0.0.0")]
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
[global::System.Runtime.CompilerServices.CompilerGeneratedAttribute()]
internal class Resources;
```


File

Resources.Designer.cs ([see page 61](#))



Description

This class was auto-generated by the StronglyTypedResourceBuilder class via a tool like ResGen or Visual Studio. To add or remove a member, edit your .ResX file then rerun ResGen with the /str option, or rebuild your VS project.

Methods

	Name	Description
	Resources (see page 2)	This is Resources, a member of class Resources.

Resources Properties

	Name	Description
	Culture (see page 2)	Overrides the current thread's CurrentUICulture property for all resource lookups using this strongly typed resource class.
	ResourceManager (see page 2)	Returns the cached ResourceManager instance used by this class.

1.1.1.1.1.1 Resources.Resources Constructor**C#**

```
[global::System.Diagnostics.CodeAnalysis.SuppressMessageAttribute("Microsoft.Performance",
"CA1811:AvoidUncalledPrivateCode")]
internal Resources();
```

Description

This is Resources, a member of class Resources.

Body Source

```
1:
[global::System.Diagnostics.CodeAnalysis.SuppressMessageAttribute("Microsoft.Performance",
"CA1811:AvoidUncalledPrivateCode")]
2: internal Resources()
3: {
4: }
```

1.1.1.1.1.2 Resources Properties**1.1.1.1.1.2.1 Resources.Culture Property**

Overrides the current thread's CurrentUICulture property for all resource lookups using this strongly typed resource class.

C#

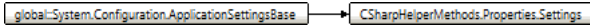
```
[global::System.ComponentModel.EditorBrowsableAttribute(global::System.ComponentModel.EditorBrowsableState.Advanced)]
internal static global::System.Globalization.CultureInfo Culture;
```

1.1.1.1.1.2.2 Resources.ResourceManager Property

Returns the cached ResourceManager instance used by this class.

C#

```
[global::System.ComponentModel.EditorBrowsableAttribute(global::System.ComponentModel.EditorBrowsableState.Advanced)]
internal static global::System.Resources.ResourceManager ResourceManager;
```

1.1.1.1.2 Settings Class**Class Hierarchy****C#**

```
[global::System.Runtime.CompilerServices.CompilerGeneratedAttribute()]
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("Microsoft.VisualStudio.Editors.SettingsDesigner.SettingsSingleFileGenerator",
"11.0.0.0")]
internal sealed class Settings : global::System.Configuration.ApplicationSettingsBase;
```

File

Settings.Designer.cs (🔗 see page 65)

Description

This is class CSharpHelperMethods.Properties.Settings.

Settings Properties

	Name	Description
🔗	Default (🔗 see page 3)	This is Default, a member of class Settings.

1.1.1.1.2.1 Settings Properties**1.1.1.1.2.1.1 Settings.Default Property****C#**

```
public static Settings Default;
```

Description

This is Default, a member of class Settings.

1.1.2 CSharpHelperMethods.YardimciSiniflar Namespace

This is namespace CSharpHelperMethods.YardimciSiniflar.

Classes









	Name	Description
🔗	DataTableIslemleri (🔗 see page 4)	Raporlama için لازم olan DataTable vb. verilerin düzenlenmesini sağlayan sınıflardır.
🔗	DosyaIslemleri (🔗 see page 6)	Dosya işlemleri ile ilgili yardımcı sınıf
🔗	KisiIslemleri (🔗 see page 9)	Kisi ile ilişkili yardımcı metotların tamamını içeren sınıf
🔗	MetinIslemleri (🔗 see page 11)	Metinsel veri kontrol metotlarını barındıran sınıf
🔗	SayıIslemleri (🔗 see page 16)	Sayısal veri kontrol metotlarını barındıran sınıf
🔗	SifrelemeIslemleri (🔗 see page 19)	Sifre ile ilgili işlemler sınıfı
🔗	SiralamaIslemleri (🔗 see page 22)	Alfa numerik sıralama için hazırlanmış bir metod

	Tarihslemleri (see page 24)	Tarihsel veri kontrol metotlarini barindiran sinif
---	---	--

1.1.2.1 Classes

The following table lists classes in this documentation.

Classes

	Name	Description
	DataTableIslemleri (see page 4)	Raporlama için lazım olan DataTable vb. verilerin düzenlenmesini saglayan siniflardir.
	DosyaIslemleri (see page 6)	Dosya islemleri ile ilgili yardimci sinif
	KisiIslemleri (see page 9)	Kisi ile ilgili yardimci metotlarin tamamini içeren sinif
	MetinIslemleri (see page 11)	Metinsel veri kontrol metotlarini barindiran sinif
	SayıIslemleri (see page 16)	Sayisal veri kontrol metotlarini barindiran sinif
	SifrelemeIslemleri (see page 19)	Sifre ile ilgili islemler sinifi
	SiralamaIslemleri (see page 22)	Alfa numerik siralama için hazirlanmis bir metod
	Tarihslemleri (see page 24)	Tarihsel veri kontrol metotlarini barindiran sinif

1.1.2.1.1 DataTableIslemleri Class

Raporlama için lazım olan DataTable vb. verilerin düzenlenmesini saglayan siniflardir.

Class Hierarchy

CSharpHelperMethods.YardimciSiniflar.DataTableIslemleri



C#

```
public static class DataTableIslemleri;
```

File

DataTableIslemleri.cs ([see page 32](#))

DataTableIslemleri Methods

	Name	Description
	BaseType (see page 4)	Nesnenin içindeki her hangi bir degerin (DateTime, decimal vb.) Eger Nullable ise, Nullable olmasini saglayan metottur (Çünkü Crystal Report, Telerik vb. Nullable alan kabul etmiyor)
	ToDataTable (see page 5)	Parametre olarak girilen herhangi bir sinif türünden listeyi Idael bir sekilde Nullable alan içermeyen bir DataTable'a çevirmek için kullanılan metottur.

1.1.2.1.1.1 DataTableIslemleri Methods

1.1.2.1.1.1.1 DataTableIslemleri.BaseType Method

Nesnenin içindeki her hangi bir degerin (DateTime, decimal vb.) Eger Nullable ise, Nullable olmasini saglayan metottur (Çünkü Crystal Report, Telerik vb. Nullable alan kabul etmiyor)

C#

```
public static Type BaseType(Type oType);
```

Parameters

Parameters	Description
Type oType	Veri Tipi

Returns

Verinin Ideal Tipi

1.1.2.1.1.1.2 DataTableIslemleri.ToDataTable Method

Parametre olarak girilen herhangi bir sinif türünden listeyi Idael bir sekilde Nullable alan içermeyen bir DataTable'a çevirmek için kullanılan metottur.

C#

```
public static DataTable ToDataTable<T>(List<T> liste);
```

Parameters

Parameters	Description
List<T> liste	Liste Bilgisi

Type Parameters

Type Parameters	Description
T	Generic Nesne Tipi

Returns

DataTable cinsinde Nesne

Body Source

```
1: public static DataTable ToDataTable<T>(List<T> liste)
2: {
3:     try
4:     {
5:         var sonuc = new DataTable(typeof(T).Name);
6:
7:         var ozellikler = typeof(T).GetFields(BindingFlags.Public |
BindingFlags.CreateInstance | BindingFlags.Default | BindingFlags.DeclaredOnly |
BindingFlags.ExactBinding | BindingFlags.FlattenHierarchy | BindingFlags.IgnoreReturn |
BindingFlags.Instance | BindingFlags.InvokeMethod | BindingFlags.NonPublic |
BindingFlags.OptionalParamBinding | BindingFlags.SuppressChangeType);
8:
9:         var ozellikler2 = typeof(T).GetProperties(BindingFlags.Public |
BindingFlags.CreateInstance | BindingFlags.Default | BindingFlags.DeclaredOnly |
BindingFlags.ExactBinding | BindingFlags.FlattenHierarchy | BindingFlags.IgnoreReturn |
BindingFlags.Instance | BindingFlags.InvokeMethod | BindingFlags.NonPublic |
BindingFlags.OptionalParamBinding | BindingFlags.SuppressChangeType);
10:
11:         foreach (FieldInfo ozellik in ozellikler)
12:         {
13:             var varMi = false;
14:             foreach (PropertyInfo ozellik2 in ozellikler2)
15:             {
16:                 if (ozellik.Name.Contains(ozellik2.Name))
17:                 {
18:                     varMi = true;
19:                     var ozellikAdi = ozellik.Name.Replace("<", "");
20:                     var ozellikDizi = ozellikAdi.Split('>');
21:                     var ozellikTemp = ozellikDizi[0].Trim();
22:                     //ozellikTemp = ozellikTemp.Substring(0, ozellikTemp.Length - 1);
23:                     if (ozellikTemp == ozellik2.Name)
24:                     {
25:                         if (sonuc.Columns.IndexOf(new DataColumn(ozellik2.Name)) < 0)
26:                         {
27:                             sonuc.Columns.Add(ozellik2.Name,
BaseType(ozellik2.PropertyType));
28:                             break;
29:                         }
30:                     }
31:                 }
32:             }
33:         }
34:     }
35: }
```



```

31:         else
32:         {
33:             if (sonuc.Columns.IndexOf(new DataColumn(ozellikTemp)) < 0)
34:             {
35:                 sonuc.Columns.Add(ozellikTemp,
BaseType(ozellik2.PropertyType));
36:                 break;
37:             }
38:         }
39:     }
40: }
41: if (!varMi)
42: {
43:     if (sonuc.Columns.IndexOf(new DataColumn(ozellik.Name)) < 0)
44:     {
45:         sonuc.Columns.Add(ozellik.Name, BaseType(ozellik.FieldType));
46:     }
47: }
48:
49: }
50:
51: foreach (T nesne in liste)
52: {
53:     var degerler = new object[ozellikler.Length];
54:
55:     int i;
56:     for (i = 0; i < ozellikler.Length; i++)
57:     {
58:         var varMi = false;
59:         foreach (var item in ozellikler2)
60:         {
61:             if (ozellikler[i].Name.Contains(item.Name))
62:             {
63:                 varMi = true;
64:                 degerler[i] = item.GetValue(nesne, null);
65:                 break;
66:             }
67:         }
68:         if (!varMi)
69:         {
70:             degerler[i] = ozellikler[i].GetValue(nesne);
71:         }
72:     }
73: }
74:
75: sonuc.Rows.Add(degerler);
76: }
77:
78: return sonuc;
79: }
80: catch (Exception)
81: {
82:     return null;
83: }
84: }

```

1.1.2.1.2 Dosyaislemleri Class

Dosya islemleri ile ilgili yardımcı sınıf

Class Hierarchy

CSharpHelperMethods.YardimciSiniflar.Dosyaislemleri

C#

```
public static class Dosyaislemleri;
```

File

Dosyalslemleri.cs (🔗 see page 34)

Dosyalslemleri Methods

	Name	Description
💎💰💵	DosyaUzantisiUygunMu (🔗 see page 7)	Sisteme yüklenecek dosyanın belirlenen kriterlere uyup uymadığını kontrol eden metottur. Dosyanın uzantisi istediğimiz formatta değilse yanlış olarak geri döndürecektr.s
💎💰💵	GetirDosyaUzantisiMimeTypeIle (🔗 see page 7)	Dosyanın türünden dosyanın uzantisinin ne olduğunu getiren metottur
💎💰💵	GetirIcerikTipiDosyaUzantisille (🔗 see page 8)	Dosya uzantisinden dosyanın türünün ne olduğunu getiren metottur

1.1.2.1.2.1 Dosyalslemleri Methods**1.1.2.1.2.1.1 Dosyalslemleri.DosyaUzantisiUygunMu Method**

Sisteme yüklenecek dosyanın belirlenen kriterlere uyup uymadığını kontrol eden metottur. Dosyanın uzantisi istediğimiz formatta değilse yanlış olarak geri döndürecektr.s

C#

```
public static bool DosyaUzantisiUygunMu(string extension);
```

Parameters

Parameters	Description
string extension	Dosya Uzantisi

Returns

Eklenecek Dosyanın Uygun Olup Olmadığı Bilgisi

Body Source

```
1: public static bool DosyaUzantisiUygunMu(string extension)
2: {
3:     extension = extension.ToLower();
4:
5:     if (extension == ".jpg" || extension == ".bmp" || extension == ".png" || extension
== ".doc" || extension == ".docx" || extension == ".ppt" || extension == ".pptx" ||
extension == ".xls" || extension == ".xlsx" || extension == ".rar" || extension == ".zip"
|| extension == ".7z" || extension == ".pdf" || extension == ".txt" || extension == ".rtf")
6:     {
7:         return true;
8:     }
9:     else
10:    {
11:        return false;
12:    }
13: }
```

1.1.2.1.2.1.2 Dosyalslemleri.GetirDosyaUzantisiMimeTypeIle Method

Dosyanın türünden dosyanın uzantisinin ne olduğunu getiren metottur

C#

```
public static string GetirDosyaUzantisiMimeTypeIle(string mimeType);
```

Parameters

Parameters	Description
string mimeType	Mime Type

Returns

Dosya Türü Bilgisi

Body Source

```

1: public static string GetirDosyaUzantisiMimeTypeIle(string mimeType)
2: {
3:     string dosyaUzantisi;
4:     mimeType = mimeType.ToLower();
5:     switch (mimeType)
6:     {
7:         case "application/pdf": dosyaUzantisi = "pdf"; break;
8:         case "image/jpeg": dosyaUzantisi = "jpeg"; break;
9:         case "image/jpg": dosyaUzantisi = "jpg"; break;
10:        case "image/png": dosyaUzantisi = "png"; break;
11:        case "image/bmp": dosyaUzantisi = "bmp"; break;
12:        case "application/msword": dosyaUzantisi = "doc"; break;
13:        case "application / vnd.openxmlformats -
officedocument.presentationml.presentation": dosyaUzantisi = "pptx"; break;
14:        case "application/vnd.ms-excel": dosyaUzantisi = "xls"; break;
15:        case "text/plain": dosyaUzantisi = "txt"; break;
16:        case "application/vnd.openxmlformats-officedocument.spreadsheetml.sheet":
dosyaUzantisi = "xlsx"; break;
17:        case "application/vnd.openxmlformats-officedocument.wordprocessingml.document":
dosyaUzantisi = "docx"; break;
18:        case "text/xml": dosyaUzantisi = "xml"; break;
19:        case "application/vnd.ms-powerpoint": dosyaUzantisi = "ppt"; break;
20:        case "application/zip": dosyaUzantisi = "zip"; break;
21:        case "application/x-rar-compressed": dosyaUzantisi = "rar"; break;
22:        case "application/x-7z-compressed": dosyaUzantisi = "7z"; break;
23:        default: dosyaUzantisi = "error"; break;
24:    }
25:
26:    return dosyaUzantisi;
27: }

```

1.1.2.1.2.1.3 Dosyalslemleri.GetirIcerikTipiDosyaUzantisille Method

Dosya uzantisindan dosyanin türünün ne oldugunu getiren metottur

C#

```
public static string GetirIcerikTipiDosyaUzantisille(string extension);
```

Parameters

Parameters	Description
string extension	Dosya Uzantisi

Returns

Dosya Türü Bilgisi

Body Source

```

1: public static string GetirIcerikTipiDosyaUzantisille(string extension)
2: {
3:     string contentType = String.Empty;
4:     extension = extension.Replace(".", "").ToLower();
5:     switch (extension)
6:     {
7:         case "pdf": contentType = "application/pdf"; break;
8:         case "jpeg": contentType = "image/jpeg"; break;
9:         case "jpg": contentType = "image/jpg"; break;
10:        case "png": contentType = "image/png"; break;
11:        case "bmp": contentType = "image/bmp"; break;
12:        case "doc": contentType = "application/msword"; break;
13:        case "pptx": contentType =
"application/vnd.openxmlformats-officedocument.presentationml.presentation"; break;

```

```

14:         case "xls": contentType = "application/vnd.ms-excel"; break;
15:         case "txt": contentType = "text/plain"; break;
16:         case "xlsx": contentType =
"application/vnd.openxmlformats-officedocument.spreadsheetml.sheet"; break;
17:         case "rtf": contentType = "application/rtf"; break;
18:         case "docx": contentType =
"application/vnd.openxmlformats-officedocument.wordprocessingml.document"; break;
19:         case "ppt": contentType = "application/vnd.ms-powerpoint"; break;
20:         case "zip": contentType = "application/zip"; break;
21:         case "rar": contentType = "application/x-rar-compressed"; break;
22:         case "7z": contentType = "application/x-7z-compressed"; break;
23:         default: contentType = "application/octet-stream"; break;
24:     }
25:
26:     return contentType;
27: }

```

1.1.2.1.3 Kisilslemleri Class

Kisi ile ilgili yardimci metotların tamamını içeren sınıf

Class Hierarchy

CSharpHelperMethods.YardimciSiniflar.Kisilslemleri




C#

```
public static class KisiIslemleri;
```

File

Kisilslemleri.cs (see page 56)

Kisilslemleri Methods

	Name	Description
	DogrulaEPosta (see page 9)	Ekranlardan girilen bir e-posta adresinin geçerli bir e-posta adresi olup olmadığını Kontrol eden metottur
	DogrulaIBAN (see page 10)	Girilen IBAN'ın standar IBAN algoritmasına uygun olup olmadığını doğrular
	DogrulaTcKimlikNo (see page 10)	Girilen TC Kimlik Numarasının, standart TC Kimlik No algoritmasına uygun olup olmadığını kontrol eden metottur.

1.1.2.1.3.1 Kisilslemleri Methods

1.1.2.1.3.1.1 Kisilslemleri.DogrulaEPosta Method

Ekranlardan girilen bir e-posta adresinin geçerli bir e-posta adresi olup olmadığını Kontrol eden metottur

C#

```
public static bool DogrulaEPosta(string ePosta);
```

Parameters

Parameters	Description
string ePosta	E-Posta Adresi

Returns

E-Posta Adresinin Geçerli Olup Olmadığı Bilgisi

Body Source

```

1: public static bool DogrulaEPosta(string ePosta)
2: {
3:     const string pattern = @"^(([\w-]+\.)+[\w-]+|([a-zA-Z]{1}|[\w-]{2,}))@"

```

```

4:         + @"((( [0-1]?[0-9]{1,2}|25[0-5]|2[0-4][0-9])\."
5:         + @"([0-1]?[0-9]{1,2}|25[0-5]|2[0-4][0-9])\."
6:         + @"([0-1]?[0-9]{1,2}|25[0-5]|2[0-4][0-9])\."
7:         + @"([0-1]?[0-9]{1,2}|25[0-5]|2[0-4][0-9])){1}| "
8:         + @"([a-zA-Z]+[\w-]+\.)+[a-zA-Z]{2,4})$";
9:     return Regex.IsMatch(ePosta, pattern);
10: }

```

1.1.2.1.3.1.2 Kisilslemleri.DogrulaIBAN Method

Girilen IBAN'in standar IBAN algoritmasına uygun olup olmadığını dogrular

C#

```
public static bool DogrulaIBAN(string iban);
```

Parameters

Parameters	Description
string iban	Uluslar arasi banka hesap numarasi (IBAN)

Returns

IBAN uygunsa true, degilse false degeri döner

Description

```
DogrulaIBAN("TR560006200000012990022604")
```

Body Source

```

1: public static bool DogrulaIBAN(string iban)
2: {
3:     iban = iban.ToUpper();
4:
5:     if (string.IsNullOrEmpty(iban))
6:         return false;
7:
8:     if (!Regex.IsMatch(iban, "^[A-Z0-9]"))
9:         return false;
10:
11:     iban = iban.Replace(" ", String.Empty);
12:     var bank = iban.Substring(4, iban.Length - 4) + iban.Substring(0, 4);
13:     var asciiShift = 55;
14:     var sb = new StringBuilder();
15:     foreach (var c in bank)
16:     {
17:         sb.Append(char.IsLetter(c) ? c - asciiShift : int.Parse(c.ToString()));
18:     }
19:
20:     string checksumString = sb.ToString();
21:     int checksum = int.Parse(checksumString.Substring(0, 1));
22:     for (int i = 1; i < checksumString.Length; i++)
23:     {
24:         int v = int.Parse(checksumString.Substring(i, 1));
25:         checksum *= 10;
26:         checksum += v;
27:         checksum %= 97;
28:     }
29:     return checksum == 1;
30: }

```

1.1.2.1.3.1.3 Kisilslemleri.DogrulaTcKimlikNo Method

Girilen TC Kimlik Numarasinin, standart TC Kimlik No algoritmasına uygun olup olmadığını kontrol eden metottur.

C#

```
public static bool DogrulaTcKimlikNo(string tcKimlikNo);
```

Parameters

Parameters	Description
string tcKimlikNo	TC Kimlik No

Returns

TC Kimlik No'nun Uygun Olup Olmadigi Bilgisi

Body Source

```

1: public static bool DogrulaTcKimlikNo(string tcKimlikNo)
2: {
3:     var tcNo = Convert.ToInt64(tcKimlikNo);
4:     var atcno = tcNo / 100;
5:     var btcno = tcNo / 100;
6:     var c1 = atcno % 10; atcno = atcno / 10;
7:     var c2 = atcno % 10; atcno = atcno / 10;
8:     var c3 = atcno % 10; atcno = atcno / 10;
9:     var c4 = atcno % 10; atcno = atcno / 10;
10:    var c5 = atcno % 10; atcno = atcno / 10;
11:    var c6 = atcno % 10; atcno = atcno / 10;
12:    var c7 = atcno % 10; atcno = atcno / 10;
13:    var c8 = atcno % 10; atcno = atcno / 10;
14:    var c9 = atcno % 10; atcno = atcno / 10;
15:    var q1 = ((10 - (((c1 + c3 + c5 + c7 + c9) * 3) + (c2 + c4 + c6 + c8)) % 10)) %
10);
16:    var q2 = ((10 - (((((c2 + c4 + c6 + c8) + q1) * 3) + (c1 + c3 + c5 + c7 + c9)) %
10)) % 10);
17:    var dogrula = ((btcno * 100) + (q1 * 10) + q2 == tcNo);
18:    return dogrula;
19: }

```

1.1.2.1.4 MetinIslemleri Class

Metinsel veri kontrol metotlarini barindiran sinif

Class Hierarchy

CSharpHelperMethods.YardimciSiniflar.MetinIslemleri









C#



```
public static class MetinIslemleri;
```

File

MetinIslemleri.cs (see page 57)

MetinIslemleri Methods

	Name	Description
	BoslukKaldir (see page 12)	Girilen metnin bosluklarinin kaldırilmasini saglayan metottur.
	HTMLDuzelt (see page 12)	Girilen metnin basinda virgöl olup olmadigini kontrol eden metottur.
	IdealMetinGetir (see page 13)	Girilen metnin basinda virgöl olup olmadigini kontrol eden metottur.
	MetinIlkKarakterBuyukDigerleriKucukYap (see page 14)	This is MetinIlkKarakterBuyukDigerleriKucukYap, a member of class MetinIslemleri.
	MetinIlkKarakterleriBuyukYap (see page 14)	Girilen bir metindeki tüm kelimelerin ilk harflerinin Büyük yapılması için hazirlanan metottur
	MetinSifrele (see page 14)	This is MetinSifrele, a member of class MetinIslemleri.
	MetinZararliKarakterDuzelt (see page 15)	This is MetinZararliKarakterDuzelt, a member of class MetinIslemleri.
	TelefonNoDuzenle (see page 15)	Girilen telefon numarasinin SMS gönderilecek hale getirilmesini saglayan metottur.

	TurkceKarakterleriDuzelt (see page 15)	Girilen metnin basinda virgöl olup olmadigini kontrol eden metottur.
	UTF8Duzelt (see page 16)	Girilen metnin basinda virgöl olup olmadigini kontrol eden metottur.

1.1.2.1.4.1 MetinIslemleri Methods

1.1.2.1.4.1.1 MetinIslemleri.BoslukKaldir Method

Girilen metnin bosluklarinin kaldırilmasini saglayan metottur.

C#

```
public static string BoslukKaldir(string metin);
```

Parameters

Parameters	Description
string metin	Telefon No Bilgisi

Returns

Girilen Metnin Istene Hali

Body Source

```
1: public static string BoslukKaldir(string metin)
2: {
3:     if (!string.IsNullOrEmpty(metin))
4:     {
5:         metin = metin.Replace("-", "").Replace(" ", "").Replace("(", "").Replace(")",
6:         return metin;
7:     }
8:     else
9:     {
10:        return string.Empty;
11:    }
12: }
```

1.1.2.1.4.1.2 MetinIslemleri.HTMLDuzelt Method

Girilen metnin basinda virgöl olup olmadigini kontrol eden metottur.

C#

```
public static string HTMLDuzelt(string text);
```

Parameters

Parameters	Description
string text	Girdi metni

Returns

Girilen Metnin Olmasi Gerektigi Hali

Body Source

```
1: public static string HTMLDuzelt(string text)
2: {
3:     List<string> sc = new List<string>();
4:     // get rid of unnecessary tag spans (comments and title)
5:     sc.Add(@"<!--(\w|\W)+?-->");
6:     sc.Add(@"<title>(\w|\W)+?</title>");
7:     // Get rid of classes and styles
8:     sc.Add(@"\s?class=\w+");
9:     sc.Add(@"\s+style='[^']*'");
10:    // Get rid of unnecessary tags
11:    sc.Add(
```

```

12:
13: @"<(meta|link|/?o:|/?style|/?div|/?st\d|/?head|/?html|body|/?body|/?span|!\[^\>]*?>");
14: // Get rid of empty paragraph tags
15: sc.Add(@"(<[^\>]+>)+&nbsp;(</\w+>)+");
16: sc.Add(@"<a[^\>]*(<[^\<]+>)</a>");
17: sc.Add(@"<p[^\>]*(<[^\<]+>)</p>");
18: // remove bizarre v: element attached to <img> tag
19: sc.Add(@"\s+v:\w+=\"[^\"]+\"");
20: // remove extra lines
21: sc.Add(@"(\n\r){2,}");
22: sc.Add(@"&nbsp;");
23: foreach (string s in sc)
24: {
25:     text = Regex.Replace(text, s, string.Empty, RegexOptions.IgnoreCase);
26: }
27: text = System.Text.RegularExpressions.Regex.Replace(text, "<.*?>", String.Empty);
28: text = text.Replace("\r", " ").Replace("\n", " ").Replace(" ", " ").Trim();
29: text = text.Replace("&rsquo;", "'");
30: text = text.Replace("&rsquo;", "'");
31: text = text.Replace("&rdquo;", "\"");
32: text = text.Replace("&rdquo;", "\"");
33: text = text.Replace("&lsquo;", "'");
34: text = text.Replace("&lsquo;", "'");
35: text = text.Replace("&ldquo;", "\"");
36: text = text.Replace("&ldquo;", "\"");
37:
38: text = text.Replace("&lt;", "<");
39: text = text.Replace("&gt;", ">");
40: text = text.Replace("&amp;", "&");
41: //text = text.Replace("&lt;", "");
42: //text = text.Replace("&p>", "");
43: //text = text.Replace("&/p>", "");
44: //text = text.Replace("&gt;", "");
45: //return text;
46: return text;
47: }

```

1.1.2.1.4.1.3 MetinIslemleri.IdealMetinGetir Method

Girilen metnin basinda virgöl olup olmadigini kontrol eden metottur.

C#

```
public static string IdealMetinGetir(string text);
```

Parameters

Parameters	Description
string text	Girdi metni

Returns

Girilen Metnin Olmasi Gerektigi Hali

Body Source

```

1: public static string IdealMetinGetir(string text)
2: {
3:     if (!string.IsNullOrEmpty(text) && text[0] == ',')
4:     {
5:         text = text.Substring(1, text.Length - 1);
6:     }
7:     return text;
8: }

```


1.1.2.1.4.1.4 MetinIslemleri.MetinIlkKarakterBuyukDigerleriKucukYap Method**C#**

```
public static string MetinIlkKarakterBuyukDigerleriKucukYap(string text);
```

Description

This is MetinIlkKarakterBuyukDigerleriKucukYap, a member of class MetinIslemleri.

1.1.2.1.4.1.5 MetinIslemleri.MetinIlkKarakterleriBuyukYap Method

Girilen bir metindeki tüm kelimelerin ilk harflerinin Büyük yapılması için hazırlanan metottur

C#

```
public static string MetinIlkKarakterleriBuyukYap(this string input);
```

Parameters

Parameters	Description
this string input	Girilen metin

Returns

Metnin Kelimelerinin İlk Harflerinin Büyülmüş Hali

Body Source

```
1: public static string MetinIlkKarakterleriBuyukYap(this string input)
2: {
3:     string result = "";
4:     var wordList = input.Split(' ');
5:     var cultureInfo = new CultureInfo("tr-TR");
6:     foreach (var item in wordList)
7:     {
8:         if (item.Length > 1)
9:             result = result + " " + item.Substring(0, 1).ToUpper(cultureInfo) +
item.Substring(1).ToLower(cultureInfo) + " ";
10:        else if (item.Length == 1)
11:            result = result + " " + item.ToUpper(cultureInfo) + " ";
12:        }
13:        return result;
14: }
```

1.1.2.1.4.1.6 MetinIslemleri.MetinSifrele Method**C#**

```
public static string MetinSifrele(string metin);
```

Description

This is MetinSifrele, a member of class MetinIslemleri.

Body Source

```
1: public static string MetinSifrele(string metin)
2: {
3:     if (!string.IsNullOrEmpty(metin))
4:     {
5:         var metinDuzenleyici = new StringBuilder(metin);
6:         if (metin.Length == 10)
7:         {
8:             metinDuzenleyici.Remove(2, 6);
9:             metinDuzenleyici.Insert(2, "*****");
10:            metin = metinDuzenleyici.ToString();
11:        }
12:        else if (metin.Length == 11)
13:        {
```

```

14:         metinDuzenleyici.Remove(2, 7);
15:         metinDuzenleyici.Insert(2, "*****");
16:         metin = metinDuzenleyici.ToString();
17:     }
18: }
19: return metin;
20: }

```

1.1.2.1.4.1.7 MetinIslemleri.MetinZararliKarakterDuzelt Method

C#

```
public static string MetinZararliKarakterDuzelt(string text);
```

Description

This is MetinZararliKarakterDuzelt, a member of class MetinIslemleri.

Body Source

```

1: public static string MetinZararliKarakterDuzelt(string text)
2: {
3:     text = text.Replace("&#252;", "ü");
4:     return text;
5: }

```

1.1.2.1.4.1.8 MetinIslemleri.TelefonNoDuzenle Method

Girilen telefon numarasinin SMS gönderilecek hale getirilmesini saglayan metottur.

C#

```
public static string TelefonNoDuzenle(string telefonNo);
```

Parameters

Parameters	Description
string telefonNo	Telefon No Bilgisi

Returns

Girilen Telefon Numarasinin SMS Gönderilecek Hali

Body Source

```

1: public static string TelefonNoDuzenle(string telefonNo)
2: {
3:     if (!string.IsNullOrEmpty(telefonNo))
4:     {
5:         telefonNo = telefonNo.Replace("-", "").Replace(" ", "").Replace("(",
6:         telefonNo.Replace(")", "");
7:         if (telefonNo.Length == 11)
8:         {
9:             telefonNo = telefonNo.Substring(telefonNo.Length - 10);
10:        }
11:        return telefonNo;
12:    }
13:    else
14:    {
15:        return string.Empty;
16:    }
17: }

```

1.1.2.1.4.1.9 MetinIslemleri.TurkceKarakterleriDuzelt Method

Girilen metnin basinda virgöl olup olmadigini kontrol eden metottur.

C#

```
public static string TurkceKarakterleriDuzelt(string text);
```

Parameters

Parameters	Description
string text	Girdi metni

Returns

Girilen Metnin Olmasi Gerektigi Hali

Body Source

```

1: public static string TurkceKarakterleriDuzelt(string text)
2: {
3:     text = text.Replace("ü", "u");
4:     text = text.Replace("İ", "i");
5:     text = text.Replace("ö", "o");
6:     text = text.Replace("ü", "u");
7:     text = text.Replace("ş", "s");
8:     text = text.Replace("ğ", "g");
9:     text = text.Replace("ç", "c");
10:    text = text.Replace("Ü", "U");
11:    text = text.Replace("I", "I");
12:    text = text.Replace("Ö", "O");
13:    text = text.Replace("Ü", "U");
14:    text = text.Replace("S", "S");
15:    text = text.Replace("G", "G");
16:    text = text.Replace("Ç", "C");
17:    return text;
18: }
```

1.1.2.1.4.1.10 MetinIslemleri.UTF8Duzelt Method

Girilen metnin basinda virgül olup olmadigini kontrol eden metottur.

C#

```
public static string UTF8Duzelt(string text);
```

Parameters

Parameters	Description
string text	Girdi metni

Returns

Girilen Metnin Olmasi Gerektigi Hali

Body Source

```

1: public static string UTF8Duzelt(string text)
2: {
3:     text = text.Replace("&cedil;", "ç");
4:     text = text.Replace("&ouml;", "ö");
5:     text = text.Replace("&uuml;", "ü");
6:     text = text.Replace("&cedil;", "ç");
7:     text = text.Replace("&ouml;", "ö");
8:     text = text.Replace("&uuml;", "ü");
9:     return text;
10: }
```

1.1.2.1.5 SayIslemleri Class

Sayisal veri kontrol metotlarini barindiran sinif

Class Hierarchy

CSharpHelperMethods.YardimciSiniflar.SayIslemleri





C#

```
public static class SayiIslemleri;
```

File

SayIslemleri.cs (see page 63)

SayIslemleri Methods

	Name	Description
	AltiHaneliOnayKoduOlustur (see page 17)	Sadece rakamlardan olusan 6 haneli random sayi üreten metottur.
	SayiHalineCevir (see page 17)	Ekrandan alinan string degerde virgöl olduğundan ajax metodu patladigi için Burada string olarak tutar verisi aliniyor Ve decimala çevriliyor
	SayisalMi (see page 18)	Girilen bir metnin sayi olup olmadigini kontrol eden metottur.
	SayiyiYaziyaCevir (see page 18)	Elimizdeki tutar verilerinin Yazı formatında gösterilmesini sağlayacak bir metottur

1.1.2.1.5.1 SayIslemleri Methods**1.1.2.1.5.1.1 SayIslemleri.AltiHaneliOnayKoduOlustur Method**

Sadece rakamlardan olusan 6 haneli random sayi üreten metottur.

C#

```
public static string AltiHaneliOnayKoduOlustur();
```

Returns

6 Haneli Onay Kodu

Body Source

```
1: public static string AltiHaneliOnayKoduOlustur()
2: {
3:     var rastgele = new Random();
4:     var sb = new StringBuilder();
5:     for (var i = 0; i < 6; i++)
6:     {
7:         var ascii = rastgele.Next(48, 57); //Rakamlar
8:         var karakter = Convert.ToChar(ascii);
9:         sb.Append(karakter);
10:    }
11:    return sb.ToString();
12: }
```

1.1.2.1.5.1.2 SayIslemleri.SayiHalineCevir Method

Ekrandan alinan string degerde virgöl olduğundan ajax metodu patladigi için Burada string olarak tutar verisi aliniyor Ve decimala çevriliyor

C#

```
public static decimal SayiHalineCevir(string metin);
```

Parameters

Parameters	Description
incomingValue	Girilen String Deger

Returns

Decimal Deger

Body Source

```

1: public static decimal SayiHalineCevir(string metin)
2: {
3:     decimal val;
4:     if (metin.Contains("_") || metin.Contains(".") || metin.Contains(","))
5:     {
6:         if (!decimal.TryParse(metin.Replace(",", "").Replace(".", "").Replace("_", ""),
NumberStyles.Number, CultureInfo.InvariantCulture, out val))
7:             return 0;
8:         return val / 100;
9:     }
10:    else
11:    {
12:        var value = decimal.TryParse(metin.Replace(",", "").Replace(".",
13:        "").Replace("_", ""), NumberStyles.Number, CultureInfo.InvariantCulture, out val);
14:        if (value)
15:        {
16:            return val;
17:        }
18:        else
19:        {
20:            return 0;
21:        }
22:    }

```

1.1.2.1.5.1.3 SayiIslemleri.SayisalMi Method

Girilen bir metnin sayi olup olmadigini kontrol eden metottur.

C#

```
public static bool SayisalMi(string text);
```

Parameters

Parameters	Description
string text	Girdi metni

Returns

Girilen Metnin Sayisal Olup Olmadigi Bilgisi

1.1.2.1.5.1.4 SayiIslemleri.SayiyiYaziyaCevir Method

Elimizdeki tutar verilerinin Yazı formatında gösterilmesini sağlayacak bir metottur

C#

```
public static string SayiyiYaziyaCevir(decimal tutar);
```

Parameters

Parameters	Description
decimal tutar	Girilen Sayı Bilgisi

Returns

Sayinin Yazıya Dönüştürülmüş Hali

Body Source

```

1: public static string SayiyiYaziyaCevir(decimal tutar)
2: {
3:     var sTutar = tutar.ToString("F2").Replace('.', ','); // Replace('.', ',') ondalik
ayracinin . olma durumu için
4:     var lira = sTutar.Substring(0, sTutar.IndexOf(',')); //tutarin tam kısmi
5:     var kurus = sTutar.Substring(sTutar.IndexOf(',') + 1, 2);
6:     var yazi = "";

```

```

7:
8:     string[] birler = { "", "BIR", "IKI", "ÜÇ", "DÖRT", "BES", "ALTI", "YEDI", "SEKİZ",
"DOKUZ" };
9:     string[] onlar = { "", "ON", "YIRMI", "OTUZ", "KIRK", "ELLI", "ALTMIS", "YETMİS",
"SEKSEN", "DOKSAN" };
10:    string[] binler = { "KATRİLYON", "TRİLYON", "MİLYAR", "MİLYON", "BIN", "" };
//KATRİLYON'un önüne ekleme yapılarak artırabilir.
11:
12:    var grupSayisi = 6;
13:    //sayidaki 3'lü grup sayısı. katrilyon içi 6. (1.234,00 daki grup sayısı 2'dir.)
14:    //KATRİLYON'un basına ekleyeceğiniz her değer için grup sayisini artiriniz.
15:
16:    lira = lira.PadLeft(grupSayisi * 3, '0'); //sayinin soluna '0' eklenerek sayi 'grup
sayisi x 3' basakmakli yapiliyor.
17:
18:    for (int i = 0; i < grupSayisi * 3; i += 3) //sayi 3'erli gruplar halinde ele
aliniyor.
19:    {
20:        var grupDegeri = "";
21:
22:        if (lira.Substring(i, 1) != "0")
23:            grupDegeri += birler[Convert.ToInt32(lira.Substring(i, 1))] + "YÜZ";
//yüzler
24:
25:        if (grupDegeri == "BİRYÜZ") //biryüz düzeltiliyor.
26:            grupDegeri = "YÜZ";
27:
28:        grupDegeri += onlar[Convert.ToInt32(lira.Substring(i + 1, 1))]; //onlar
29:
30:        grupDegeri += birler[Convert.ToInt32(lira.Substring(i + 2, 1))];
//birler
31:
32:        if (grupDegeri != "") //binler
33:            grupDegeri += binler[i / 3];
34:
35:        if (grupDegeri == "BİRBİN") //birbin düzeltiliyor.
36:            grupDegeri = "BİN";
37:
38:        yazi += grupDegeri;
39:    }
40:
41:    if (yazi != "")
42:        yazi += " TL ";
43:
44:    int yaziUzunlugu = yazi.Length;
45:
46:    if (kurus.Substring(0, 1) != "0") //kurus onlar
47:        yazi += onlar[Convert.ToInt32(kurus.Substring(0, 1))];
48:
49:    if (kurus.Substring(1, 1) != "0") //kurus birler
50:        yazi += birler[Convert.ToInt32(kurus.Substring(1, 1))];
51:
52:    if (yazi.Length > yaziUzunlugu)
53:        yazi += " Kr.";
54:    else
55:        yazi += "SİFİR Kr.";
56:
57:    return yazi;
58: }

```

1.1.2.1.6 Sifrelemelslemleri Class

Sifre ile ilgili işlemler sınıfı

Class Hierarchy

CSharpHelperMethods.YardimciSiniflar.Sifrelemelslemleri

C#

```
public static class SifrelemeIslemleri;
```

File

Sifrelemelslemleri.cs (see page 66)

Sifrelemelslemleri Methods

	Name	Description
💡💡💡	HesaplaMD5 (see page 20)	Kullanici sifresinin MD5 yöntemiyle sifrlenmesini saglayan metottur.
💡💡💡	HesaplaSHA256 (see page 20)	Kullanici sifresinin SHA yöntemiyle sifrlenmesini saglayan metottur.
💡💡💡	HesaplaSHA512 (see page 21)	Kullanici sifresinin SHA yöntemiyle sifrlenmesini saglayan metottur.
💡💡💡	RastgeleSifreUret (see page 21)	8 karakter uzunlugunda Rastgele sifre olusturur.
💡💡💡	SifreUygunMu (see page 22)	sifreyi kriterlere göre kontrol edip uymayan kosullarini string listesine geri dönuş yapan methoddur. Uygun olmasi durumunda null döndürecektir.

1.1.2.1.6.1 Sifrelemelslemleri Methods**1.1.2.1.6.1.1 Sifrelemelslemleri.HesaplaMD5 Method**

Kullanici sifresinin MD5 yöntemiyle sifrlenmesini saglayan metottur.

C#

```
public static string HesaplaMD5(string text);
```

Parameters

Parameters	Description
string text	Sifrelenecek Metin

Returns

Girdi Metninin Sifrlenmis Hali

Body Source

```
1: public static string HesaplaMD5(string text)
2: {
3:     MD5 md5 = MD5.Create();
4:     byte[] inputBytes = Encoding.ASCII.GetBytes(text);
5:     byte[] hash = md5.ComputeHash(inputBytes);
6:
7:     StringBuilder sb = new StringBuilder();
8:     foreach (byte t in hash)
9:     {
10:         sb.Append(t.ToString("X2"));
11:     }
12:     return sb.ToString();
13: }
```

1.1.2.1.6.1.2 Sifrelemelslemleri.HesaplaSHA256 Method

Kullanici sifresinin SHA yöntemiyle sifrlenmesini saglayan metottur.

C#

```
public static string HesaplaSHA256(string text);
```

Parameters

Parameters	Description
string text	Sifrelenecek Metin

Returns

Girdi Metninin Sifrelenmis Hali

Body Source

```

1: public static string HesaplaSHA256(string text)
2: {
3:     SHA256 sha256 = SHA256Managed.Create();
4:     byte[] bytes = Encoding.UTF8.GetBytes(text);
5:     byte[] hash = sha256.ComputeHash(bytes);
6:     StringBuilder result = new StringBuilder();
7:     for (int i = 0; i < hash.Length; i++)
8:     {
9:         result.Append(hash[i].ToString("X2"));
10:    }
11:    return result.ToString();
12: }

```

1.1.2.1.6.1.3 Sifrelemelslemleri.HesaplaSHA512 Method

Kullanici sifresinin SHA yöntemiyle sifrelenmesini saglayan metottur.

C#

```
public static string HesaplaSHA512(string text);
```

Parameters

Parameters	Description
string text	Sifrelenecek Metin

Returns

Girdi Metninin Sifrelenmis Hali

Body Source

```

1: public static string HesaplaSHA512(string text)
2: {
3:     SHA512Managed sha512 = new SHA512Managed();
4:     Byte[] encryptedSha512 = sha512.ComputeHash(Encoding.UTF8.GetBytes(text));
5:     string strHex = "";
6:     foreach (byte b in encryptedSha512)
7:     {
8:         strHex += String.Format("{0:x2}", b);
9:     }
10:    return strHex.ToUpper();
11: }

```

1.1.2.1.6.1.4 Sifrelemelslemleri.RastgeleSifreUret Method

8 karakter uzunlugunda Rastgele sifre olusturur.

C#

```
public static string RastgeleSifreUret();
```

Body Source

```

1: public static string RastgeleSifreUret()
2: {
3:     int sifreUzunluk = 8;
4:     string gecerliKarakterler = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890";
5:     StringBuilder strB = new StringBuilder(100);
6:     Random random = new Random();
7:     while (0 < sifreUzunluk--)
8:     {
9:         strB.Append(gecerliKarakterler[random.Next(gecerliKarakterler.Length)]);
10:    }
11:    return strB.ToString();

```



```
12: }
```

1.1.2.1.6.1.5 Sifrelemelslemleri.SifreUygunMu Method

sifreyi kriterlere göre kontrol edip uymayan kosullarini string listesine geri dönüş yapan methoddur. Uygun olmasi durumunda null döndürecektir.

C#

```
public static string SifreUygunMu(string text);
```

Parameters

Parameters	Description
string text	Kullanici Sifresi

Returns

Sifrenin Uygun Olup Olmadigi Bilgisi (Açiklamasi ile beraber)

Body Source

```
1: public static string SifreUygunMu(string text)
2: {
3:     //bu kisimda sifrenin olmasi gereken kosullari degerlendirilecek.
4:     List<string> mesajlar = new List<string>();
5:
6:     if (text.Length < 8)
7:         mesajlar.Add("Parola en az 8 karakter olmalidir.");
8:
9:     if (!Regex.IsMatch(text, @"^(?=.*[a-z])", RegexOptions.ECMAScript))
10:        mesajlar.Add("Parola en az bir küçük harf içermelidir.");
11:
12:     if (!Regex.IsMatch(text, @"^(?=.*[A-Z])", RegexOptions.ECMAScript))
13:        mesajlar.Add("Parola en az bir büyük harf içermelidir.");
14:
15:     if (!Regex.IsMatch(text, @"^(?=.*[!@,##$%&*,?_~,-,£,(,)]",
16:        RegexOptions.ECMAScript))
17:        mesajlar.Add("Parola en az bir noktalama isareti içermelidir.");
18:
19:     if (!Regex.IsMatch(text, @"^(?=.*[0-9])", RegexOptions.ECMAScript))
20:        mesajlar.Add("Parola en az bir rakam içermelidir.");
21:
22:     if (mesajlar.Count < 1)
23:         return "Uygun";
24:     else
25:     {
26:         var sonuc = "";
27:         foreach (var item in mesajlar)
28:             sonuc += item + ", ";
29:         sonuc = sonuc.Substring(0, sonuc.Length - 2);
30:         return sonuc;
31:     }
```

1.1.2.1.7 Siralamalslemleri Class

Alfa numerik siralama için hazirlanmis bir metod

Class Hierarchy

CSharpHelperMethods.YardimciSiniflar.Siralamalslemleri


C#

```
public static class SiralamaIslemleri;
```

File

Siralamalslemleri.cs (see page 68)

Siralamalslemleri Methods

	Name	Description
	Compare (see page 23)	This is Compare, a member of class Siralamalslemleri.

1.1.2.1.7.1 Siralamalslemleri Methods**1.1.2.1.7.1.1 Siralamalslemleri.Compare Method****C#**

```
public static int Compare(string x, string y);
```

Description

This is Compare, a member of class Siralamalslemleri.

Body Source

```
1: public static int Compare(string x, string y)
2: {
3:     String s1 = x;
4:     String s2 = y;
5:     if (string.IsNullOrEmpty(s1) || string.IsNullOrEmpty(s2))
6:     {
7:         return 0;
8:     }
9:
10:    int thisMarker = 0;
11:    int thatMarker = 0;
12:
13:    while ((thisMarker < s1.Length) || (thatMarker < s2.Length))
14:    {
15:        if (thisMarker >= s1.Length)
16:        {
17:            return -1;
18:        }
19:        else if (thatMarker >= s2.Length)
20:        {
21:            return 1;
22:        }
23:        char thisCh = s1[thisMarker];
24:        char thatCh = s2[thatMarker];
25:
26:        StringBuilder thisChunk = new StringBuilder();
27:        StringBuilder thatChunk = new StringBuilder();
28:
29:        while ((thisMarker < s1.Length) && (thisChunk.Length == 0 || InChunk(thisCh,
thisChunk[0])))
30:        {
31:            thisChunk.Append(thisCh);
32:            thisMarker++;
33:
34:            if (thisMarker < s1.Length)
35:            {
36:                thisCh = s1[thisMarker];
37:            }
38:        }
39:
40:        while ((thatMarker < s2.Length) && (thatChunk.Length == 0 || InChunk(thatCh,
thatChunk[0])))
41:        {
42:            thatChunk.Append(thatCh);
43:            thatMarker++;
44:
45:            if (thatMarker < s2.Length)
46:            {
```

```

47:         thatCh = s2[thatMarker];
48:     }
49: }
50:
51: int result = 0;
52: // If both chunks contain numeric characters, sort them numerically
53: if (char.IsDigit(thisChunk[0]) && char.IsDigit(thatChunk[0]))
54: {
55:     var thisNumericChunk = Convert.ToInt64(thisChunk.ToString());
56:     var thatNumericChunk = Convert.ToInt64(thatChunk.ToString());
57:
58:     if (thisNumericChunk < thatNumericChunk)
59:     {
60:         result = -1;
61:     }
62:
63:     if (thisNumericChunk > thatNumericChunk)
64:     {
65:         result = 1;
66:     }
67: }
68: else
69: {
70:     var ci = new CultureInfo("tr-TR");
71:     result = String.Compare(thisChunk.ToString(), thatChunk.ToString(), true,
72:         ci);
73:     //result = thisChunk.ToString().CompareTo(thatChunk.ToString());
74: }
75:
76: if (result != 0)
77: {
78:     return result;
79: }
80: }
81:
82: return 0;
83: }

```

1.1.2.1.8 Tarihslemleri Class

Tarihsel veri kontrol metotlarını barındıran sınıf

Class Hierarchy

CSharpHelperMethods.YardimciSiniflar.Tarihslemleri

C#

```
public static class Tarihslemleri;
```

File

Tarihslemleri.cs (see page 70)

Tarihslemleri Methods

	Name	Description
💎💰💵	AyFarki (see page 25)	Girilen iki tarih arasında kaç ay olduğunu bulan metottur
💎💰💵	KidemTazminatıTarihAralığıHesapla (see page 25)	Kidem Tazminatı Hesaplaması Yapılırken ayda kaç gün olursa olsun tamamında çalışmışsa 30 gün üzerinden hesaplanması gerektiği için İsten ayrış ve iş başlangıç tarihleri arasında kaç gün geçtiğini belirtmek için Hesaplama yapılmasını sağlayan metottur.
💎💰💵	TarihAralığıHesapla (see page 26)	İki tarih arasında kaç gün olduğunu hesaplamak için kullanılan metottur
💎💰💵	YasHesapla (see page 27)	Yas hesaplamak için kullanılan metottur
💎💰💵	YasHesaplaMetinsel (see page 27)	Yas hesaplamak için kullanılan metottur

	ZamaniDegistir (see page 28)	Filtre ekranlarında başlangıç ve bitis tarihi üzerinden sorgulama yapıldığında Eger aynı gün için sorgulama yapılıyorsa Bitis tarihini o günün 23.59.59.999'una ayarlamak için Hazırlanan metottur
--	------------------------------	--

1.1.2.1.8.1 TarihIslemleri Methods

1.1.2.1.8.1.1 TarihIslemleri.AyFarki Method

Girilen iki tarih arasında kaç ay olduğunu bulan metottur

C#

```
public static int AyFarki(DateTime lValue, DateTime rValue);
```

Parameters

Parameters	Description
DateTime lValue	İlk tarih bilgisi
DateTime rValue	Son tarih bilgisi

Returns

Ay Farki Bilgisi

1.1.2.1.8.1.2 TarihIslemleri.KidemTazminatıTarihAralığıHesapla Method

Kidem Tazminatı Hesaplaması Yapılırken ayda kaç gün olursa olsun tamamında çalışmıssa 30 gün üzerinden hesaplanması gerektiği için İsten ayrılış ve iş başlangıç tarihleri arasında kaç gün geçtiğini belirtmek için Hesaplama yapılmasını sağlayan metottur.

C#

```
public static int KidemTazminatıTarihAralığıHesapla(DateTime? baslangicTarihi, DateTime? bitisTarihi);
```

Parameters

Parameters	Description
DateTime? baslangicTarihi	Başlangıç Tarihi Bilgisi
DateTime? bitisTarihi	Bitis Tarihi Bilgisi

Returns

Gün Sayısı Bilgisi

Body Source

```
1: public static int KidemTazminatıTarihAralığıHesapla(DateTime? baslangicTarihi,
2: DateTime? bitisTarihi)
3: {
4:     var artacakMi = true;
5:     var result = 0;
6:     var baslangicMi = false;
7:     var bitisMi = false;
8:     var aylıkGünSayısı = 0;
9:     for (DateTime date = (DateTime)baslangicTarihi; date <= (DateTime)bitisTarihi; date
= date.AddDays(1))
10:     {
11:         if (!artacakMi)
12:             result++;
13:         else
14:             artacakMi = false;
15:         if (date.Day == 1)
16:             baslangicMi = true;
17:         else if (date.Day == DateTime.DaysInMonth(date.Year, date.Month))
```

```

18:         bitisMi = true;
19:
20:         if (baslangicMi)
21:             aylikGunSayisi++;
22:         if (baslangicMi && bitisMi && aylikGunSayisi != 30)
23:         {
24:             result = result - aylikGunSayisi + 30;
25:             baslangicMi = false;
26:             bitisMi = false;
27:             aylikGunSayisi = 0;
28:         }
29:         else if (bitisMi)
30:         {
31:             baslangicMi = false;
32:             bitisMi = false;
33:             aylikGunSayisi = 0;
34:         }
35:     }
36: }
37: if (baslangicTarihi?.Day > bitisTarihi?.Day)
38:     result--;
39: return result;
40: }

```

1.1.2.1.8.1.3 TarihIslemleri.TarihAraligiHesapla Method

İki tarih arasında kaç gün olduğunu hesaplamak için kullanılan metottur

C#

```
public static string TarihAraligiHesapla(DateTime baslangicTarihi, DateTime bitisTarihi);
```

Parameters

Parameters	Description
DateTime baslangicTarihi	Baslangıç Tarihi Bilgisi
DateTime bitisTarihi	Bitis Tarihi Bilgisi

Returns

Tarih Araligi Bilgisi

Body Source

```

1: public static string TarihAraligiHesapla(DateTime baslangicTarihi, DateTime bitisTarihi)
2: {
3:     int Years = new DateTime(bitisTarihi.Subtract(baslangicTarihi).Ticks).Year - 1;
4:     DateTime PastYearDate = baslangicTarihi.AddYears(Years);
5:     int Months = 0;
6:     for (int i = 1; i <= 12; i++)
7:     {
8:         if (PastYearDate.AddMonths(i) == bitisTarihi)
9:         {
10:             Months = i;
11:             break;
12:         }
13:         else if (PastYearDate.AddMonths(i) >= bitisTarihi)
14:         {
15:             Months = i - 1;
16:             break;
17:         }
18:     }
19:     int Days = bitisTarihi.Subtract(PastYearDate.AddMonths(Months)).Days;
20:     var result = "";
21:     if (Years < 10)
22:         result = result + "0" + Years;
23:     else
24:         result = result + Years;
25:     result += " Yil, ";

```

```

26:         if (Months < 10)
27:             result = result + "0" + Months;
28:         else
29:             result = result + Months;
30:         result += " Ay, ";
31:         if (Days < 10)
32:             result = result + "0" + Days;
33:         else
34:             result = result + Days;
35:         result += " Gün";
36:         return result;
37:     }

```

1.1.2.1.8.1.4 TarihIslemleri.YasHesapla Method

Yas hesaplamak için kullanılan metottur

C#

```
public static int YasHesapla(DateTime dogumTarihi);
```

Parameters

Parameters	Description
DateTime dogumTarihi	Dogum Tarihi Bilgisi

Returns

Yas Bilgisi

Body Source

```

1: public static int YasHesapla(DateTime dogumTarihi)
2: {
3:     DateTime Now = DateTime.Now;
4:     int Years = new DateTime(DateTime.Now.Subtract(dogumTarihi).Ticks).Year - 1;
5:     DateTime PastYearDate = dogumTarihi.AddYears(Years);
6:     int Months = 0;
7:     for (int i = 1; i <= 12; i++)
8:     {
9:         if (PastYearDate.AddMonths(i) == Now)
10:         {
11:             Months = i;
12:             break;
13:         }
14:         else if (PastYearDate.AddMonths(i) >= Now)
15:         {
16:             Months = i - 1;
17:             break;
18:         }
19:     }
20:     return Years;
21: }

```

1.1.2.1.8.1.5 TarihIslemleri.YasHesaplaMetinsel Method

Yas hesaplamak için kullanılan metottur

C#

```
public static string YasHesaplaMetinsel(DateTime dogumTarihi);
```

Parameters

Parameters	Description
DateTime dogumTarihi	Dogum Tarihi Bilgisi

Returns

Yas Bilgisi (Metinsel olarak)

Body Source

```

1: public static string YasHesaplaMetinsel(DateTime dogumTarihi)
2: {
3:     DateTime Now = DateTime.Now;
4:     int Years = new DateTime(DateTime.Now.Subtract(dogumTarihi).Ticks).Year - 1;
5:     DateTime PastYearDate = dogumTarihi.AddYears(Years);
6:     int Months = 0;
7:     for (int i = 1; i <= 12; i++)
8:     {
9:         if (PastYearDate.AddMonths(i) == Now)
10:        {
11:            Months = i;
12:            break;
13:        }
14:        else if (PastYearDate.AddMonths(i) >= Now)
15:        {
16:            Months = i - 1;
17:            break;
18:        }
19:    }
20:    int Days = Now.Subtract(PastYearDate.AddMonths(Months)).Days;
21:    int Hours = Now.Subtract(PastYearDate).Hours;
22:    int Minutes = Now.Subtract(PastYearDate).Minutes;
23:    int Seconds = Now.Subtract(PastYearDate).Seconds;
24:    return String.Format("Yas: {0} Yil {1} Ay {2} Gün",
25:        Years, Months, Days);
26:    //return String.Format("Yas: {0} Yil {1} Ay {2} Gün {3} Saat {4} Saniye",
27:        //Years, Months, Days, Hours, Seconds);
28: }

```

1.1.2.1.8.1.6 Tarihslemleri.ZamaniDegistir Method

Filtre ekranlarında başlangıç ve bitis tarihi üzerinden sorgulama yapıldığında Eger aynı gün için sorgulama yapılıyorsa Bitis tarihini o günün 23.59.59.999'una ayarlamak için Hazirlanan metottur

C#

```

public static DateTime ZamaniDegistir(this DateTime dateTime, int hours, int minutes, int
seconds, int milliseconds);

```

Parameters

Parameters	Description
this DateTime dateTime	Tarih Bilgisi
int hours	Saat Bilgisi
int minutes	Dakika Bilgisi
int seconds	Saniye Bilgisi
int milliseconds	Mili Saniye Bilgisi

Returns

Yeni Tarih Bilgisi

Body Source

```

1: public static DateTime ZamaniDegistir(this DateTime dateTime, int hours, int minutes,
int seconds, int milliseconds)
2: {
3:     return new DateTime(
4:         dateTime.Year,
5:         dateTime.Month,
6:         dateTime.Day,
7:         hours,
8:         minutes,
9:         seconds,
10:        milliseconds,

```

```



11:         dateTime.Kind);
12:     }

```

1.1.3 Classes

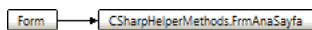
The following table lists classes in this documentation.

Classes

	Name	Description
	FrmAnaSayfa (see page 29)	This is class CSharpHelperMethods.FrmAnaSayfa.
	Program (see page 30)	This is class CSharpHelperMethods.Program.

1.1.3.1 FrmAnaSayfa Class

Class Hierarchy



C#

```
public class FrmAnaSayfa : Form;
```

File

FrmAnaSayfa.Designer.cs ([see page 44](#))


Description

This is class CSharpHelperMethods.FrmAnaSayfa.

Methods

	Name	Description
	FrmAnaSayfa (see page 29)	This is FrmAnaSayfa, a member of class FrmAnaSayfa.

FrmAnaSayfa Methods

	Name	Description
	Dispose (see page 30)	Clean up any resources being used.

1.1.3.1.1 FrmAnaSayfa.FrmAnaSayfa Constructor

C#

```
public FrmAnaSayfa();
```

Description

This is FrmAnaSayfa, a member of class FrmAnaSayfa.

Body Source

```

1: public FrmAnaSayfa()
2: {
3:     InitializeComponent();
4: }

```

1.1.3.1.2 FrmAnaSayfa Methods

1.1.3.1.2.1 FrmAnaSayfa.Dispose Method

Clean up any resources being used.

C#

```
protected override void Dispose(bool disposing);
```

Parameters

Parameters	Description
bool disposing	true if managed resources should be disposed; otherwise, false.

Body Source

```
1: protected override void Dispose(bool disposing)
2: {
3:     if (disposing && (components != null))
4:     {
5:         components.Dispose();
6:     }
7:     base.Dispose(disposing);
8: }
```

1.1.3.2 Program Class

Class Hierarchy

CSharpHelperMethods.Program

C#

```
public static class Program;
```

File

Program.cs ([see page 61](#))

Description

This is class CSharpHelperMethods.Program.

1.2 Files

The following table lists files in this documentation.

Files

Name	Description
AssemblyInfo.cs (see page 31)	This is file AssemblyInfo.cs.
CSharpHelperMethods.csproj (see page 32)	This is file CSharpHelperMethods.csproj.
CSharpHelperMethods.sln (see page 32)	This is file CSharpHelperMethods.sln.
DataTableIslemleri.cs (see page 32)	This is file DataTableIslemleri.cs.
Dosyalslemleri.cs (see page 34)	This is file Dosyalslemleri.cs.
FrmAnaSayfa.cs (see page 36)	This is file FrmAnaSayfa.cs.
FrmAnaSayfa.Designer.cs (see page 44)	This is file FrmAnaSayfa.Designer.cs.
Kisilslemleri.cs (see page 56)	This is file Kisilslemleri.cs.

MetinIslemleri.cs (see page 57)	This is file MetinIslemleri.cs.
Program.cs (see page 61)	This is file Program.cs.
Resources.Designer.cs (see page 61)	This code was generated by a tool. Runtime Version:4.0.30319.42000 Changes to this file may cause incorrect behavior and will be lost if the code is regenerated.
SayIslemleri.cs (see page 63)	This is file SayIslemleri.cs.
Settings.Designer.cs (see page 65)	This code was generated by a tool. Runtime Version:4.0.30319.42000 Changes to this file may cause incorrect behavior and will be lost if the code is regenerated.
Sifrelemelslemleri.cs (see page 66)	This is file Sifrelemelslemleri.cs.
Siralamalslemleri.cs (see page 68)	This is file Siralamalslemleri.cs.
TarihIslemleri.cs (see page 70)	This is file TarihIslemleri.cs.

1.2.1 AssemblyInfo.cs

This is file AssemblyInfo.cs.

Body Source

```

1: ?using System.Reflection;
2: using System.Runtime.CompilerServices;
3: using System.Runtime.InteropServices;
4:
5: // General Information about an assembly is controlled through the following
6: // set of attributes. Change these attribute values to modify the information
7: // associated with an assembly.
8: [assembly: AssemblyTitle("CSharpHelperMethods")]
9: [assembly: AssemblyDescription("")]
10: [assembly: AssemblyConfiguration("")]
11: [assembly: AssemblyCompany("")]
12: [assembly: AssemblyProduct("CSharpHelperMethods")]
13: [assembly: AssemblyCopyright("Copyright © 2021")]
14: [assembly: AssemblyTrademark("")]
15: [assembly: AssemblyCulture("")]
16:
17: // Setting ComVisible to false makes the types in this assembly not visible
18: // to COM components. If you need to access a type in this assembly from
19: // COM, set the ComVisible attribute to true on that type.
20: [assembly: ComVisible(false)]
21:
22: // The following GUID is for the ID of the typelib if this project is exposed to COM
23: [assembly: Guid("be02934f-770b-411c-aa8c-43b2efa18625")]
24:
25: // Version information for an assembly consists of the following four values:
26: //
27: //      Major Version
28: //      Minor Version
29: //      Build Number
30: //      Revision
31: //
32: // You can specify all the values or you can default the Build and Revision Numbers
33: // by using the '*' as shown below:
34: // [assembly: AssemblyVersion("1.0.*")]
35: [assembly: AssemblyVersion("1.0.0.0")]
36: [assembly: AssemblyFileVersion("1.0.0.0")]

```

1.2.2 CSharpHelperMethods.csproj

This is file CSharpHelperMethods.csproj.

1.2.3 CSharpHelperMethods.sln

This is file CSharpHelperMethods.sln.

1.2.4 DataTableIslemleri.cs

This is file DataTableIslemleri.cs.

Body Source

```

1: ?using System;
2: using System.Collections.Generic;
3: using System.Data;
4: using System.Reflection;
5:
6: namespace CSharpHelperMethods.YardimciSiniflar
7: {
8:     /// <summary>
9:     /// Raporlama için lazım olan DataTable vb. verilerin düzenlenmesini saglayan
10:    siniflardir.
11:    /// </summary>
12:    public static class DataTableIslemleri
13:    {
14:        /// <summary>
15:        /// Parametre olarak girilen herhangi bir sinif türünden listeyi
16:        /// Idael bir sekilde Nullable alan içermeyen bir DataTable'a çevirmek için
17:        /// kullanılan metottur.
18:        /// </summary>
19:        /// <typeparam name="T">Generic Nesne Tipi</typeparam>
20:        /// <param name="liste">Liste Bilgisi</param>
21:        /// <returns>DataTable cinsinde Nesne</returns>
22:        public static DataTable ToDataTable<T>(List<T> liste)
23:        {
24:            try
25:            {
26:                var sonuc = new DataTable(typeof(T).Name);
27:
28:                var ozellikler = typeof(T).GetFields(BindingFlags.Public |
29:                BindingFlags.CreateInstance | BindingFlags.Default | BindingFlags.DeclaredOnly |
30:                BindingFlags.ExactBinding | BindingFlags.FlattenHierarchy | BindingFlags.IgnoreReturn |
31:                BindingFlags.Instance | BindingFlags.InvokeMethod | BindingFlags.NonPublic |
32:                BindingFlags.OptionalParamBinding | BindingFlags.SuppressChangeType);
33:
34:                var ozellikler2 = typeof(T).GetProperties(BindingFlags.Public |
35:                BindingFlags.CreateInstance | BindingFlags.Default | BindingFlags.DeclaredOnly |
36:                BindingFlags.ExactBinding | BindingFlags.FlattenHierarchy | BindingFlags.IgnoreReturn |
37:                BindingFlags.Instance | BindingFlags.InvokeMethod | BindingFlags.NonPublic |
38:                BindingFlags.OptionalParamBinding | BindingFlags.SuppressChangeType);
39:
40:                foreach (FieldInfo ozellik in ozellikler)
41:                {
42:                    var varMi = false;

```

```

33:         foreach (PropertyInfo ozellik2 in ozellikler2)
34:         {
35:             if (ozellik.Name.Contains(ozellik2.Name))
36:             {
37:                 varMi = true;
38:                 var ozellikAdi = ozellik.Name.Replace("<", "");
39:                 var ozellikDizi = ozellikAdi.Split('>');
40:                 var ozellikTemp = ozellikDizi[0].Trim();
41:                 //ozellikTemp = ozellikTemp.Substring(0,
ozellikTemp.Length - 1);
42:                 if (ozellikTemp == ozellik2.Name)
43:                 {
44:                     if (sonuc.Columns.IndexOf(new
DataColumn(ozellik2.Name)) < 0)
45:                     {
46:                         sonuc.Columns.Add(ozellik2.Name,
BaseType(ozellik2.PropertyType));
47:                         break;
48:                     }
49:                 }
50:                 else
51:                 {
52:                     if (sonuc.Columns.IndexOf(new DataColumn(ozellikTemp))
< 0)
53:                     {
54:                         sonuc.Columns.Add(ozellikTemp,
BaseType(ozellik2.PropertyType));
55:                         break;
56:                     }
57:                 }
58:             }
59:         }
60:         if (!varMi)
61:         {
62:             if (sonuc.Columns.IndexOf(new DataColumn(ozellik.Name)) < 0)
63:             {
64:                 sonuc.Columns.Add(ozellik.Name,
BaseType(ozellik.FieldType));
65:             }
66:         }
67:     }
68: }
69:
70: foreach (T nesne in liste)
71: {
72:     var degerler = new object[ozellikler.Length];
73:
74:     int i;
75:     for (i = 0; i < ozellikler.Length; i++)
76:     {
77:         var varMi = false;
78:         foreach (var item in ozellikler2)
79:         {
80:             if (ozellikler[i].Name.Contains(item.Name))
81:             {
82:                 varMi = true;
83:                 degerler[i] = item.GetValue(nesne, null);
84:                 break;
85:             }
86:         }
87:         if (!varMi)
88:         {
89:             degerler[i] = ozellikler[i].GetValue(nesne);
90:         }
91:     }
92: }
93:
94: sonuc.Rows.Add(degerler);
95: }

```

```

96:
97:         return sonuc;
98:     }
99:     catch (Exception)
100:    {
101:        return null;
102:    }
103: }
104:
105:     /// <summary>
106:     /// Nesnenin içindeki her hangi bir degerin (DateTime, decimal vb.)
107:     /// Eger Nullable ise, Nullable olmamasini saglayan metottur
108:     /// (Çünkü Crystal Report, Telerik vb. Nullable alan kabul etmiyor)
109:     /// </summary>
110:     /// <param name="oType">Veri Tipi</param>
111:     /// <returns>Verinin Ideal Tipi</returns>
112:     public static Type BaseType(Type oType) => oType != null && oType.IsValueType
&& oType.IsGenericType && oType.GetGenericTypeDefinition() == typeof(Nullable<>) ?
Nullable.GetUnderlyingType(oType) : oType;
113: }
114: }

```

Namespaces

Name	Description
CSharpHelperMethods (see page 1)	This is namespace CSharpHelperMethods.

1.2.5 Dosyalslemleri.cs

This is file Dosyalslemleri.cs.

Body Source

```

1: ?using System;
2:
3: namespace CSharpHelperMethods.YardimciSiniflar
4: {
5:     /// <summary>
6:     /// Dosya islemleri ile ilgili yardimci sinif
7:     /// </summary>
8:     public static class DosyaIslemleri
9:     {
10:         /// <summary>
11:         /// Dosya uzantisindan dosyanin türünün ne oldugunu getiren metottur
12:         /// </summary>
13:         /// <param name="extension">Dosya Uzantisi</param>
14:         /// <returns>Dosya Türü Bilgisi</returns>
15:         public static string GetIcerikTipiDosyaUzantisiIle(string extension)
16:         {
17:             string contentType = String.Empty;
18:             extension = extension.Replace(".", "").ToLower();
19:             switch (extension)
20:             {
21:                 case "pdf": contentType = "application/pdf"; break;
22:                 case "jpeg": contentType = "image/jpeg"; break;
23:                 case "jpg": contentType = "image/jpg"; break;
24:                 case "png": contentType = "image/png"; break;
25:                 case "bmp": contentType = "image/bmp"; break;
26:                 case "doc": contentType = "application/msword"; break;
27:                 case "pptx": contentType =
"application/vnd.openxmlformats-officedocument.presentationml.presentation"; break;
28:                 case "xls": contentType = "application/vnd.ms-excel"; break;
29:                 case "txt": contentType = "text/plain"; break;
30:                 case "xlsx": contentType =
"application/vnd.openxmlformats-officedocument.spreadsheetml.sheet"; break;

```

```

31:             case "rtf": contentType = "application/rtf"; break;
32:             case "docx": contentType =
"application/vnd.openxmlformats-officedocument.wordprocessingml.document"; break;
33:             case "ppt": contentType = "application/vnd.ms-powerpoint"; break;
34:             case "zip": contentType = "application/zip"; break;
35:             case "rar": contentType = "application/x-rar-compressed"; break;
36:             case "7z": contentType = "application/x-7z-compressed"; break;
37:             default: contentType = "application/octet-stream"; break;
38:         }
39:
40:         return contentType;
41:     }
42:
43:     /// <summary>
44:     /// Dosyanin türünden dosyanin uzantisinin ne oldugunu getiren metottur
45:     /// </summary>
46:     /// <param name="mimeType">Mime Type</param>
47:     /// <returns>Dosya Türü Bilgisi</returns>
48:     public static string GetirDosyaUzantisiMimeIle(string mimeType)
49:     {
50:         string dosyaUzantisi;
51:         mimeType = mimeType.ToLower();
52:         switch (mimeType)
53:         {
54:             case "application/pdf": dosyaUzantisi = "pdf"; break;
55:             case "image/jpeg": dosyaUzantisi = "jpeg"; break;
56:             case "image/jpg": dosyaUzantisi = "jpg"; break;
57:             case "image/png": dosyaUzantisi = "png"; break;
58:             case "image/bmp": dosyaUzantisi = "bmp"; break;
59:             case "application/msword": dosyaUzantisi = "doc"; break;
60:             case "application / vnd.openxmlformats -
officedocument.presentationml.presentation": dosyaUzantisi = "pptx"; break;
61:             case "application/vnd.ms-excel": dosyaUzantisi = "xls"; break;
62:             case "text/plain": dosyaUzantisi = "txt"; break;
63:             case
"application/vnd.openxmlformats-officedocument.spreadsheetml.sheet": dosyaUzantisi =
"xlsx"; break;
64:             case
"application/vnd.openxmlformats-officedocument.wordprocessingml.document": dosyaUzantisi =
"docx"; break;
65:             case "text/xml": dosyaUzantisi = "xml"; break;
66:             case "application/vnd.ms-powerpoint": dosyaUzantisi = "ppt"; break;
67:             case "application/zip": dosyaUzantisi = "zip"; break;
68:             case "application/x-rar-compressed": dosyaUzantisi = "rar"; break;
69:             case "application/x-7z-compressed": dosyaUzantisi = "7z"; break;
70:             default: dosyaUzantisi = "error"; break;
71:         }
72:
73:         return dosyaUzantisi;
74:     }
75:
76:     /// <summary>
77:     /// Sisteme yüklenecek dosyanin belirlenen kriterlere uyup uymadigini kontrol
eden metottur.
78:     /// Dosyanin uzantisi istedigimiz formatta degilse yanlis olarak geri
döndürecektir.s
79:     /// </summary>
80:     /// <param name="extension">Dosya Uzantisi</param>
81:     /// <returns>Eklencek Dosyanin Uygun Olup Olmadigi Bilgisi</returns>
82:     public static bool DosyaUzantisiUygunMu(string extension)
83:     {
84:         extension = extension.ToLower();
85:
86:         if (extension == ".jpg" || extension == ".bmp" || extension == ".png" ||
extension == ".doc" || extension == ".docx" || extension == ".ppt" || extension == ".pptx"
|| extension == ".xls" || extension == ".xlsx" || extension == ".rar" || extension ==
".zip" || extension == ".7z" || extension == ".pdf" || extension == ".txt" || extension ==
".rtf")
87:         {

```

```

88:         return true;
89:     }
90:     else
91:     {
92:         return false;
93:     }
94: }
95: }
96: }

```

Namespaces

Name	Description
CSharpHelperMethods (see page 1)	This is namespace CSharpHelperMethods.

1.2.6 FrmAnaSayfa.cs

This is file FrmAnaSayfa.cs.

Body Source

```

1: ?using CSharpHelperMethods.YardimciSiniflar;
2: using System.Text;
3: using System.Windows.Forms;
4:
5: namespace CSharpHelperMethods
6: {
7:     public partial class FrmAnaSayfa : Form
8:     {
9:         public FrmAnaSayfa()
10:        {
11:            InitializeComponent();
12:        }
13:
14:         #region Formun Yüklmesi ve Panel Seçimi
15:         /// <summary>
16:         /// Form İlk Yüklendiği zaman ekrandaki bütün panellerin erişilemez hale
17:         /// Bu metotta, çünkü hangi panelin aktifleseyeceğinin
18:         /// Combobox üzerinden seçilmesi lazım
19:         /// </summary>
20:         /// <param name="sender"></param>
21:         /// <param name="e"></param>
22:         private void FrmAnaSayfa_Load(object sender, System.EventArgs e)
23:         {
24:             #region Sayfa İlk Yüklendiğinde Tüm Panellerin Erisime Kapanması
25:             pnlTarihIslemleri.Enabled = false;
26:             pnlSifreIslemleri.Enabled = false;
27:             pnlSayiIslemleri.Enabled = false;
28:             pnlKisiIslemleri.Enabled = false;
29:             pnlMetinIslemleri.Enabled = false;
30:             #endregion
31:         }
32:
33:         /// <summary>
34:         /// Islem Türü Combobox'ında seçilen degere göre
35:         /// Ekrandaki panellerden ilgili olan haricinde hepsinin erişilemez hale
36:         /// Eger bir sey seçilmemisse, tamaminin erişilemez hale getirilmesi için
37:         /// Gerekli işlemlerin gerçekleştirildiği metottur.
38:         /// </summary>
39:         /// <param name="sender"></param>
40:         /// <param name="e"></param>
41:         private void cmbIslemTuru_SelectedIndexChanged(object sender, System.EventArgs
e)

```

```

42:         {
43:             #region Combobox'tan Seçilen Degere Göre İlgili Paneli Aktif Hale Getirme
44:             switch (cmbIslemTuru.SelectedIndex)
45:             {
46:                 // Metin İşlemleri
47:                 case 0:
48:                 {
49:                     pnlMetinIslemleri.Enabled = true;
50:                     pnlTarihIslemleri.Enabled = false;
51:                     pnlSifreIslemleri.Enabled = false;
52:                     pnlSayiIslemleri.Enabled = false;
53:                     pnlKisiIslemleri.Enabled = false;
54:                     break;
55:                 }
56:                 // Tarih İşlemleri
57:                 case 1:
58:                 {
59:                     pnlMetinIslemleri.Enabled = false;
60:                     pnlTarihIslemleri.Enabled = true;
61:                     pnlSifreIslemleri.Enabled = false;
62:                     pnlSayiIslemleri.Enabled = false;
63:                     pnlKisiIslemleri.Enabled = false;
64:                     break;
65:                 }
66:                 // Sayı İşlemleri
67:                 case 2:
68:                 {
69:                     pnlMetinIslemleri.Enabled = false;
70:                     pnlTarihIslemleri.Enabled = false;
71:                     pnlSifreIslemleri.Enabled = false;
72:                     pnlSayiIslemleri.Enabled = true;
73:                     pnlKisiIslemleri.Enabled = false;
74:                     break;
75:                 }
76:                 // Sifre İşlemleri
77:                 case 3:
78:                 {
79:                     pnlMetinIslemleri.Enabled = false;
80:                     pnlTarihIslemleri.Enabled = false;
81:                     pnlSifreIslemleri.Enabled = true;
82:                     pnlSayiIslemleri.Enabled = false;
83:                     pnlKisiIslemleri.Enabled = false;
84:                     break;
85:                 }
86:                 // Kisi İşlemleri
87:                 case 4:
88:                 {
89:                     pnlMetinIslemleri.Enabled = false;
90:                     pnlTarihIslemleri.Enabled = false;
91:                     pnlSifreIslemleri.Enabled = false;
92:                     pnlSayiIslemleri.Enabled = false;
93:                     pnlKisiIslemleri.Enabled = true;
94:                     break;
95:                 }
96:                 // Bunlar haricinde herhangi bir sey
97:                 default:
98:                 {
99:                     pnlMetinIslemleri.Enabled = false;
100:                    pnlTarihIslemleri.Enabled = false;
101:                    pnlSifreIslemleri.Enabled = false;
102:                    pnlSayiIslemleri.Enabled = false;
103:                    pnlKisiIslemleri.Enabled = false;
104:                    break;
105:                }
106:            }
107:            #endregion
108:        }
109:        #endregion
110:    }

```



```

111:         #region Tarih Islemleri
112:
113:         /// <summary>
114:         /// Paneldeki Baslangiç Tarihi seçili ve tarih girilirse, o tarihten bugüne
115:         kaç yas oldugunun hesaplandigi metottur. Ilgili butona tiklandigi zaman çalışır
116:         /// </summary>
117:         /// <param name="sender"></param>
118:         /// <param name="e"></param>
119:         private void btnYasHesapla_Click(object sender, System.EventArgs e)
120:         {
121:             if (!dtpBaslangicTarihi.Checked)
122:                 MessageBox.Show("Baslangiç Tarihi girilmedigi için bu işlem
123: gerçekleştirilemez");
124:             else
125:             {
126:                 lbSonuc.Items.Clear();
127:                 StringBuilder sb = new StringBuilder();
128:                 sb.Append("Dogum tarihi seçilen kisi ");
129:                 sb.Append(TarihIslemleri.YasHesapla(dtpBaslangicTarihi.Value));
130:                 sb.Append(" yasındadır");
131:                 lbSonuc.Items.Add(sb.ToString());
132:             }
133:         }
134:         /// <summary>
135:         /// Paneldeki Baslangiç Tarihi seçili ve tarih girilirse, o tarihten bugüne
136:         kaç yas oldugunun yıl, ay ve gün olarak metinsel bir şekilde hesaplandigi metottur.
137:         /// Ilgili butona tiklandigi zaman çalışır
138:         /// </summary>
139:         /// <param name="sender"></param>
140:         /// <param name="e"></param>
141:         private void btnYasHesaplaMetinsel_Click(object sender, System.EventArgs e)
142:         {
143:             if (!dtpBaslangicTarihi.Checked)
144:                 MessageBox.Show("Baslangiç Tarihi girilmedigi için bu işlem
145: gerçekleştirilemez");
146:             else
147:             {
148:                 lbSonuc.Items.Clear();
149:                 StringBuilder sb = new StringBuilder();
150:                 sb.Append(TarihIslemleri.YasHesaplaMetinsel(dtpBaslangicTarihi.Value));
151:                 lbSonuc.Items.Add(sb.ToString());
152:             }
153:         }
154:         /// <summary>
155:         /// Paneldeki Baslangiç Tarihi ve Bitis Tarihi alanlari seçili ve tarih
156:         girilirse,
157:         /// Söz konusu tarihler arasında kaç yıl, kaç ay ve gün oldugunun bilgisinin
158:         hazirlandigi metottur.
159:         /// Ilgili butona tiklandigi zaman çalışır
160:         /// </summary>
161:         /// <param name="sender"></param>
162:         /// <param name="e"></param>
163:         private void btnTarihAraligi_Click(object sender, System.EventArgs e)
164:         {
165:             if (!dtpBaslangicTarihi.Checked || !dtpBitisTarihi.Checked)
166:                 MessageBox.Show("Baslangiç Tarihi veya Bitis Tarihi girilmedigi için
167: bu işlem gerçekleştirilemez");
168:             else if (dtpBaslangicTarihi.Value.Date > dtpBitisTarihi.Value.Date)
169:                 MessageBox.Show("Baslangiç Tarihi Bitis Tarihinden büyük olamaz");
170:             else
171:             {
172:                 lbSonuc.Items.Clear();
173:                 StringBuilder sb = new StringBuilder();
174:                 sb.Append("İki tarih arası: ");
175:                 sb.Append(TarihIslemleri.TarihAraligiHesapla(dtpBaslangicTarihi.Value.Date,
176: dtpBitisTarihi.Value.Date));

```

```

171:                lbSonuc.Items.Add(sb.ToString());
172:            }
173:        }
174:
175:        /// <summary>
176:        /// Paneldeki Baslangiç Tarihi ve Bitis Tarihi alanlari seçili ve tarih
177:        /// girilmisse,
178:        /// Söz konusu tarihler arasında kaç ay oldugunun bilgisinin hazirlandigi
179:        /// metottur.
180:        /// ilgili butona tiklandigi zaman çalisir
181:        /// </summary>
182:        /// <param name="sender"></param>
183:        /// <param name="e"></param>
184:        private void btnAyFarki_Click(object sender, System.EventArgs e)
185:        {
186:            if (!dtpBaslangicTarihi.Checked || !dtpBitisTarihi.Checked)
187:                MessageBox.Show("Baslangiç Tarihi veya Bitis Tarihi girilmedigi için
188:                bu islem gerçekleştirilemez");
189:            else if (dtpBaslangicTarihi.Value.Date > dtpBitisTarihi.Value.Date)
190:                MessageBox.Show("Baslangiç Tarihi Bitis Tarihinden büyük olamaz");
191:            else
192:            {
193:                lbSonuc.Items.Clear();
194:                StringBuilder sb = new StringBuilder();
195:                sb.Append("Ay Farki: ");
196:                sb.Append(TarihIslemleri.AyFarki(dtpBaslangicTarihi.Value.Date,
197:                dtpBitisTarihi.Value.Date));
198:                lbSonuc.Items.Add(sb.ToString());
199:            }
200:        }
201:        #endregion
202:
203:        #region Sifre Islemleri
204:        /// <summary>
205:        /// Paneldeki metin kutusu dolu ise
206:        /// Girilen metnin MD5 algoritmasi ile sifrelenmis halini
207:        /// Listbox üzerine yazdiran metottur.
208:        /// </summary>
209:        /// <param name="sender"></param>
210:        /// <param name="e"></param>
211:        private void btnMd5Hash_Click(object sender, System.EventArgs e)
212:        {
213:            if (string.IsNullOrEmpty(txtSifrelenecekMetin.Text.Trim()))
214:                MessageBox.Show("Sifrelenecek metin girilmedigi için bu islem
215:                gerçekleştirilemez");
216:            else
217:            {
218:                lbSifreSonuc.Items.Clear();
219:                StringBuilder sb = new StringBuilder();
220:                sb.Append("Girilen Metnin MD5 Karsiligi: ");
221:                sb.Append(SifrelemeIslemleri.HesaplaMD5(txtSifrelenecekMetin.Text.Trim()));
222:                lbSifreSonuc.Items.Add(sb.ToString());
223:            }
224:        }
225:
226:        /// <summary>
227:        /// Paneldeki metin kutusu dolu ise
228:        /// Girilen metnin SHA 256 algoritmasi ile sifrelenmis halini
229:        /// Listbox üzerine yazdiran metottur.
230:        /// </summary>
231:        /// <param name="sender"></param>
232:        /// <param name="e"></param>
233:        private void btnSha256Hash_Click(object sender, System.EventArgs e)
234:        {
235:            if (string.IsNullOrEmpty(txtSifrelenecekMetin.Text.Trim()))
236:                MessageBox.Show("Sifrelenecek metin girilmedigi için bu islem
237:                gerçekleştirilemez");
238:            else

```

```

233:         {
234:             lbSifreSonuc.Items.Clear();
235:             StringBuilder sb = new StringBuilder();
236:             sb.Append("Girilen Metnin SHA 256 Karsiligi: ");
237:
sb.Append(SifrelemeIslemleri.HesaplaSHA256(txtSifrelenecekMetin.Text.Trim()));
238:             lbSifreSonuc.Items.Add(sb.ToString());
239:         }
240:     }
241:
242:     /// <summary>
243:     /// Paneldeki metin kutusu dolu ise
244:     /// Girilen metnin SHA 512 algoritmasi ile sifrelenmis halini
245:     /// Listbox üzerine yazdiran metottur.
246:     /// </summary>
247:     /// <param name="sender"></param>
248:     /// <param name="e"></param>
249:     private void btnSha512Hash_Click(object sender, System.EventArgs e)
250:     {
251:         if (string.IsNullOrEmpty(txtSifrelenecekMetin.Text.Trim()))
252:             MessageBox.Show("Sifrelenecek metin girilmedigi için bu işlem
gerçeklestirilemez");
253:         else
254:         {
255:             lbSifreSonuc.Items.Clear();
256:             StringBuilder sb = new StringBuilder();
257:             sb.Append("Girilen Metnin SHA 512 Karsiligi: ");
258:
sb.Append(SifrelemeIslemleri.HesaplaSHA512(txtSifrelenecekMetin.Text.Trim()));
259:             lbSifreSonuc.Items.Add(sb.ToString());
260:         }
261:     }
262:
263:     /// <summary>
264:     /// Ilgili butona tiklandiginda rastgele sifre üretilmesini saglayan metottur
265:     /// </summary>
266:     /// <param name="sender"></param>
267:     /// <param name="e"></param>
268:     private void btnRastgeleSifreUret_Click(object sender, System.EventArgs e)
269:     {
270:         lbSifreSonuc.Items.Clear();
271:         StringBuilder sb = new StringBuilder();
272:         sb.Append("Rastgele Üretilen Sifre: ");
273:         sb.Append(SifrelemeIslemleri.RastgeleSifreUret());
274:         lbSifreSonuc.Items.Add(sb.ToString());
275:     }
276:
277:     /// <summary>
278:     /// Paneldeki metin kutusu dolu ise
279:     /// Burada yazilan metnin bir sifre olarak belirli kriterlere uygun olup
olmadiginin
280:     /// Gösterilmesini saglayan bir metottur
281:     /// </summary>
282:     /// <param name="sender"></param>
283:     /// <param name="e"></param>
284:     private void btnSifreUygunMu_Click(object sender, System.EventArgs e)
285:     {
286:         if (string.IsNullOrEmpty(txtSifrelenecekMetin.Text.Trim()))
287:             MessageBox.Show("Sifre metni girilmedigi için bu işlem
gerçeklestirilemez");
288:         else
289:         {
290:             lbSifreSonuc.Items.Clear();
291:             StringBuilder sb = new StringBuilder();
292:             sb.Append("Girilen Sifrenin Uygunluk Durumu: ");
293:
sb.Append(SifrelemeIslemleri.SifreUygunMu(txtSifrelenecekMetin.Text.Trim()));
294:             lbSifreSonuc.Items.Add(sb.ToString());
295:         }

```

```

296:     }
297:     #endregion
298:
299:     #region Sayi Islemleri
300:     /// <summary>
301:     /// Paneldeki metin kutusuna girilen metnin sayisal bir deger olup olmadigini
302:     /// Kontrol eden ve sonucu gösteren metottur.
303:     /// </summary>
304:     /// <param name="sender"></param>
305:     /// <param name="e"></param>
306:     private void btnSayisalMi_Click(object sender, System.EventArgs e)
307:     {
308:         if (string.IsNullOrEmpty(txtSayi.Text.Trim()))
309:             MessageBox.Show("Metin girilmedigi için bu islem gerçekleştirilemez");
310:         else
311:         {
312:             lbSayiSonuc.Items.Clear();
313:             StringBuilder sb = new StringBuilder();
314:             sb.Append("Girilen Metin Sayi Mi?: ");
315:             sb.Append(SayiIslemleri.SayisalMi(txtSayi.Text.Trim()) ? "Evet" :
"Hayir");
316:             lbSayiSonuc.Items.Add(sb.ToString());
317:         }
318:     }
319:
320:     /// <summary>
321:     /// Paneldeki metin kutusuna girilen para birimi cinsinden rakamin
322:     /// Yazıya çevrilmesini saglayan ve sonucu listbox üzerinde gösteren bir
323:     /// metottur.
324:     /// </summary>
325:     /// <param name="sender"></param>
326:     /// <param name="e"></param>
327:     private void btnMetneDonustur_Click(object sender, System.EventArgs e)
328:     {
329:         if (string.IsNullOrEmpty(txtSayi.Text.Trim()))
330:             MessageBox.Show("Metin girilmedigi için bu islem gerçekleştirilemez");
331:         else if (!SayiIslemleri.SayisalMi(txtSayi.Text.Trim().Replace(".",
".").Replace(".", "")))
332:             MessageBox.Show("Girilen metin sayisal bir deger olmadigi için bu
islem gerçekleştirilemez");
333:         else
334:         {
335:             var metinSadeceSayiKismi = txtSayi.Text.Trim();
336:             decimal sayi = SayiIslemleri.SayiHaliCevir(metinSadeceSayiKismi);
337:             lbSayiSonuc.Items.Clear();
338:             StringBuilder sb = new StringBuilder();
339:             sb.Append("Girilen Sayinin Metin Hali (Parasal): ");
340:             sb.Append(SayiIslemleri.SayiyiYaziyaCevir(sayi));
341:             lbSayiSonuc.Items.Add(sb.ToString());
342:         }
343:     }
344:
345:     /// <summary>
346:     /// Rastgele olacak sekilde 6 haneli kod olusturmayi saglayan bir metottur.
347:     /// </summary>
348:     /// <param name="sender"></param>
349:     /// <param name="e"></param>
350:     private void btnOnayKoduOlustur_Click(object sender, System.EventArgs e)
351:     {
352:         lbSayiSonuc.Items.Clear();
353:         StringBuilder sb = new StringBuilder();
354:         sb.Append("Olusturulan 6 haneli onay kodu: ");
355:         sb.Append(SayiIslemleri.AltiHaneliOnayKoduOlustur());
356:         lbSayiSonuc.Items.Add(sb.ToString());
357:     }
358:     #endregion
359:
360:     #region Kisi Islemleri
361:     /// <summary>

```

```

361:        /// Paneldeki metin kutusuna girilen TC Kimlik numarasinin
362:        /// Standart TC Kimlik No algoritmasına uygun olup olmadigini kontrol eden bir
metottur.
363:        /// </summary>
364:        /// <param name="sender"></param>
365:        /// <param name="e"></param>
366:        private void btnTcKimlikDogrula_Click(object sender, System.EventArgs e)
367:        {
368:            if (string.IsNullOrEmpty(txtKisiMetni.Text.Trim()))
369:                MessageBox.Show("Metin girilmediği için bu işlem gerçekleştirilemez");
370:            else if (!SayiIslemleri.SayisalMi(txtKisiMetni.Text.Trim()))
371:                MessageBox.Show("Girilen metin sayisal bir deger olmadığı için bu
işlem gerçekleştirilemez");
372:            else if (txtKisiMetni.Text.Trim().Length != 11)
373:                MessageBox.Show("Girilen metin 11 karakter olmadığı için bu işlem
gerçekleştirilemez");
374:            else
375:            {
376:                lbKisiSonuc.Items.Clear();
377:                StringBuilder sb = new StringBuilder();
378:                sb.Append("TC Kimlik No Dogru Mu?: ");
379:                sb.Append(KisiIslemleri.DogrulaTcKimlikNo(txtKisiMetni.Text.Trim()) ?
"Evet" : "Hayir");
380:                lbKisiSonuc.Items.Add(sb.ToString());
381:            }
382:        }
383:
384:        /// <summary>
385:        /// Paneldeki metin kutusuna girilen IBAN Numarasinin
386:        /// Standart IBAN algoritmasına uygun olup olmadigini kontrol eden bir
metottur.
387:        /// </summary>
388:        /// <param name="sender"></param>
389:        /// <param name="e"></param>
390:        private void btnIbanDogrula_Click(object sender, System.EventArgs e)
391:        {
392:            if (string.IsNullOrEmpty(txtKisiMetni.Text.Trim()))
393:                MessageBox.Show("Metin girilmediği için bu işlem gerçekleştirilemez");
394:            else if (txtKisiMetni.Text.Trim().Length != 26)
395:                MessageBox.Show("Girilen metin 26 karakter olmadığı için bu işlem
gerçekleştirilemez. Karakter sayısı: " + txtKisiMetni.Text.Trim().Length);
396:            else
397:            {
398:                lbKisiSonuc.Items.Clear();
399:                StringBuilder sb = new StringBuilder();
400:                sb.Append("IBAN Dogru Mu?: ");
401:                sb.Append(KisiIslemleri.DogrulaIBAN(txtKisiMetni.Text.Trim()) ? "Evet"
: "Hayir");
402:                lbKisiSonuc.Items.Add(sb.ToString());
403:            }
404:        }
405:
406:        /// <summary>
407:        /// Paneldeki metin kutusuna girilen metnin
408:        /// Geçerli bir e-posta adresi olup olmadigini kontrol eden bir metottur.
409:        /// </summary>
410:        /// <param name="sender"></param>
411:        /// <param name="e"></param>
412:        private void btnEPostaDogrula_Click(object sender, System.EventArgs e)
413:        {
414:            if (string.IsNullOrEmpty(txtKisiMetni.Text.Trim()))
415:                MessageBox.Show("Metin girilmediği için bu işlem gerçekleştirilemez");
416:            else
417:            {
418:                lbKisiSonuc.Items.Clear();
419:                StringBuilder sb = new StringBuilder();
420:                sb.Append("E-Posta Dogru Mu?: ");
421:                sb.Append(KisiIslemleri.DogrulaEPosta(txtKisiMetni.Text.Trim()) ?
"Evet" : "Hayir");

```

```

422:         lbKisiSonuc.Items.Add(sb.ToString());
423:     }
424: }
425: #endregion
426:
427: #region Metin Islemleri
428: /// <summary>
429: /// Paneldeki metin kutusuna girilen metnin
430: /// Kelimelerinin ilk harflerini büyük yapmaya yarayan bir metottur.
431: /// </summary>
432: /// <param name="sender"></param>
433: /// <param name="e"></param>
434: private void btnIlkHarfleriBuyukYap_Click(object sender, System.EventArgs e)
435: {
436:     if (string.IsNullOrEmpty(txtMetin.Text.Trim()))
437:         MessageBox.Show("Metin girilmediği için bu işlem gerçekleştirilemez");
438:     else
439:     {
440:         lbMetinSonuc.Items.Clear();
441:         StringBuilder sb = new StringBuilder();
442:         sb.Append("Metnin Düzenlenmiş Hali: ");
443:         sb.Append(MetinIslemleri.MetinIlkKarakterleriBuyukYap(txtMetin.Text.Trim()));
444:         lbMetinSonuc.Items.Add(sb.ToString());
445:     }
446: }
447:
448: /// <summary>
449: /// Paneldeki metin kutusuna girilen metnin
450: /// Sadece ilk harfini büyük yapmaya yarayan bir metottur.
451: /// </summary>
452: /// <param name="sender"></param>
453: /// <param name="e"></param>
454: private void btnIlkHarfiBuyukYap_Click(object sender, System.EventArgs e)
455: {
456:     if (string.IsNullOrEmpty(txtMetin.Text.Trim()))
457:         MessageBox.Show("Metin girilmediği için bu işlem gerçekleştirilemez");
458:     else
459:     {
460:         lbMetinSonuc.Items.Clear();
461:         StringBuilder sb = new StringBuilder();
462:         sb.Append("Metnin Düzenlenmiş Hali: ");
463:         sb.Append(MetinIslemleri.MetinIlkKarakterBuyukDigerleriKucukYap(txtMetin.Text.Trim()));
464:         lbMetinSonuc.Items.Add(sb.ToString());
465:     }
466: }
467:
468: /// <summary>
469: /// Paneldeki metin kutusuna girilen metnin
470: /// İçerisindeki Türkçe karakterleri İngilizce hallerine çeviren bir metottur.
471: /// </summary>
472: /// <param name="sender"></param>
473: /// <param name="e"></param>
474: private void btnTürkceKarakterDuzelt_Click(object sender, System.EventArgs e)
475: {
476:     if (string.IsNullOrEmpty(txtMetin.Text.Trim()))
477:         MessageBox.Show("Metin girilmediği için bu işlem gerçekleştirilemez");
478:     else
479:     {
480:         lbMetinSonuc.Items.Clear();
481:         StringBuilder sb = new StringBuilder();
482:         sb.Append("Metnin Düzenlenmiş Hali: ");
483:         sb.Append(MetinIslemleri.TürkceKarakterleriDuzelt(txtMetin.Text.Trim()));
484:         lbMetinSonuc.Items.Add(sb.ToString());
485:     }
486: }
487:

```

```

488:         /// <summary>
489:         /// KVKK durumlarından dolayı TC Kimlik No ve Vergi No gibi alanların
490:         /// Matbu çıktılarda gizlenmesi gibi bir durum söz konusu olduğundan
491:         /// Girilen bu 10 veya 11 karakterlik metinleri şifrelemeye yarayan bir
metottur
492:         /// </summary>
493:         /// <param name="sender"></param>
494:         /// <param name="e"></param>
495:         private void btnMetinSifrele_Click(object sender, System.EventArgs e)
496:         {
497:             if (string.IsNullOrEmpty(txtMetin.Text.Trim()))
498:                 MessageBox.Show("Metin girilmediği için bu işlem gerçekleştirilemez");
499:             else if (!SayiIslemleri.SayisalMi(txtMetin.Text.Trim()))
500:                 MessageBox.Show("Girilen metin sayısal bir değer olmadığı için bu
işlem gerçekleştirilemez");
501:             else if (txtMetin.Text.Trim().Length != 11 && txtMetin.Text.Trim().Length
!= 10)
502:                 MessageBox.Show("Girilen metin 10 veya 11 karakter olmadığı için bu
işlem gerçekleştirilemez");
503:             else
504:             {
505:                 lbMetinSonuc.Items.Clear();
506:                 StringBuilder sb = new StringBuilder();
507:                 sb.Append("TC Kimlik No veya Vergi No'nun Şifrelenmiş Hali: ");
508:                 sb.Append(MetinIslemleri.MetinSifrele(txtMetin.Text.Trim()));
509:                 lbMetinSonuc.Items.Add(sb.ToString());
510:             }
511:         }
512:
513:         /// <summary>
514:         /// Ekrandan çeşitli karakterler içeren telefon numaraları alındığı zaman
515:         /// Bu telefon numarasını olması gerektiği hale çevirmeye yarayan bir metottur.
516:         /// </summary>
517:         /// <param name="sender"></param>
518:         /// <param name="e"></param>
519:         private void btnTelefonNoDuzenle_Click(object sender, System.EventArgs e)
520:         {
521:             if (string.IsNullOrEmpty(txtMetin.Text.Trim()))
522:                 MessageBox.Show("Metin girilmediği için bu işlem gerçekleştirilemez");
523:             else
524:             {
525:                 lbMetinSonuc.Items.Clear();
526:                 StringBuilder sb = new StringBuilder();
527:                 sb.Append("Telefon Numarasının Düzenlenmiş Hali: ");
528:                 sb.Append(MetinIslemleri.TelefonNoDuzenle(txtMetin.Text.Trim()));
529:                 lbMetinSonuc.Items.Add(sb.ToString());
530:             }
531:         }
532:         #endregion
533:     }
534: }

```

Namespaces

Name	Description
CSharpHelperMethods (see page 1)	This is namespace CSharpHelperMethods.

1.2.7 FrmAnaSayfa.Designer.cs

This is file FrmAnaSayfa.Designer.cs.

Body Source

```

1: ?namespace CSharpHelperMethods
2: {

```



```

3:     partial class FrmAnaSayfa
4:     {
5:         /// <summary>
6:         /// Required designer variable.
7:         /// </summary>
8:         private System.ComponentModel.IContainer components = null;
9:
10:        /// <summary>
11:        /// Clean up any resources being used.
12:        /// </summary>
13:        /// <param name="disposing">true if managed resources should be disposed;
otherwise, false.</param>
14:        protected override void Dispose(bool disposing)
15:        {
16:            if (disposing && (components != null))
17:            {
18:                components.Dispose();
19:            }
20:            base.Dispose(disposing);
21:        }
22:
23:        #region Windows Form Designer generated code
24:
25:        /// <summary>
26:        /// Required method for Designer support - do not modify
27:        /// the contents of this method with the code editor.
28:        /// </summary>
29:        private void InitializeComponent()
30:        {
31:            this.cmbIslemTuru = new System.Windows.Forms.ComboBox();
32:            this.labell = new System.Windows.Forms.Label();
33:            this.pnlTarihIslemleri = new System.Windows.Forms.Panel();
34:            this.btnAyFarki = new System.Windows.Forms.Button();
35:            this.label8 = new System.Windows.Forms.Label();
36:            this.btnTarihAraligi = new System.Windows.Forms.Button();
37:            this.btnYasHesaplaMetinsel = new System.Windows.Forms.Button();
38:            this.btnYasHesapla = new System.Windows.Forms.Button();
39:            this.label4 = new System.Windows.Forms.Label();
40:            this.lbSonuc = new System.Windows.Forms.ListBox();
41:            this.dtpBitisTarihi = new System.Windows.Forms.DateTimePicker();
42:            this.dtpBaslangicTarihi = new System.Windows.Forms.DateTimePicker();
43:            this.label3 = new System.Windows.Forms.Label();
44:            this.label2 = new System.Windows.Forms.Label();
45:            this.pnlSifreIslemleri = new System.Windows.Forms.Panel();
46:            this.label7 = new System.Windows.Forms.Label();
47:            this.btnSifreUygunMu = new System.Windows.Forms.Button();
48:            this.btnRastgeleSifreUret = new System.Windows.Forms.Button();
49:            this.btnSha512Hash = new System.Windows.Forms.Button();
50:            this.btnSha256Hash = new System.Windows.Forms.Button();
51:            this.label6 = new System.Windows.Forms.Label();
52:            this.lbSifreSonuc = new System.Windows.Forms.ListBox();
53:            this.btnMd5Hash = new System.Windows.Forms.Button();
54:            this.txtSifreleneccekMetin = new System.Windows.Forms.TextBox();
55:            this.label5 = new System.Windows.Forms.Label();
56:            this.pnlSayiIslemleri = new System.Windows.Forms.Panel();
57:            this.label9 = new System.Windows.Forms.Label();
58:            this.btnOnayKoduOlustur = new System.Windows.Forms.Button();
59:            this.btnMetneDonustur = new System.Windows.Forms.Button();
60:            this.label10 = new System.Windows.Forms.Label();
61:            this.lbSayiSonuc = new System.Windows.Forms.ListBox();
62:            this.btnSayisalMi = new System.Windows.Forms.Button();
63:            this.txtSayi = new System.Windows.Forms.TextBox();
64:            this.label11 = new System.Windows.Forms.Label();
65:            this.pnlKisiIslemleri = new System.Windows.Forms.Panel();
66:            this.label12 = new System.Windows.Forms.Label();
67:            this.btnEPostaDogrula = new System.Windows.Forms.Button();
68:            this.btnIbanDogrula = new System.Windows.Forms.Button();
69:            this.label13 = new System.Windows.Forms.Label();
70:            this.lbKisiSonuc = new System.Windows.Forms.ListBox();

```



```

71:         this.btnTcKimlikDogrula = new System.Windows.Forms.Button();
72:         this.txtKisiMetni = new System.Windows.Forms.TextBox();
73:         this.label14 = new System.Windows.Forms.Label();
74:         this.pnlMetinIslemleri = new System.Windows.Forms.Panel();
75:         this.btnTelefonNoDuzenle = new System.Windows.Forms.Button();
76:         this.btnMetinSifrele = new System.Windows.Forms.Button();
77:         this.label15 = new System.Windows.Forms.Label();
78:         this.btnTurkceKarakterDuzelt = new System.Windows.Forms.Button();
79:         this.btnIlkHarfiBuyukYap = new System.Windows.Forms.Button();
80:         this.label16 = new System.Windows.Forms.Label();
81:         this.lbMetinSonuc = new System.Windows.Forms.ListBox();
82:         this.btnIlkHarfleriBuyukYap = new System.Windows.Forms.Button();
83:         this.txtMetin = new System.Windows.Forms.TextBox();
84:         this.label17 = new System.Windows.Forms.Label();
85:         this.pnlTarihIslemleri.SuspendLayout();
86:         this.pnlSifreIslemleri.SuspendLayout();
87:         this.pnlSayiIslemleri.SuspendLayout();
88:         this.pnlKisiIslemleri.SuspendLayout();
89:         this.pnlMetinIslemleri.SuspendLayout();
90:         this.SuspendLayout();
91:         //
92:         // cmbIslemTuru
93:         //
94:         this.cmbIslemTuru.FormattingEnabled = true;
95:         this.cmbIslemTuru.Items.AddRange(new object[] {
96:             "Metin Islemleri",
97:             "Tarih Islemleri",
98:             "Sayi Islemleri",
99:             "Sifre Islemleri",
100:            "Kisi Islemleri"});
101:         this.cmbIslemTuru.Location = new System.Drawing.Point(132, 13);
102:         this.cmbIslemTuru.Name = "cmbIslemTuru";
103:         this.cmbIslemTuru.Size = new System.Drawing.Size(200, 21);
104:         this.cmbIslemTuru.TabIndex = 0;
105:         this.cmbIslemTuru.SelectedIndexChanged += new
System.EventHandler(this.cmbIslemTuru_SelectedIndexChanged);
106:         //
107:         // label1
108:         //
109:         this.label1.AutoSize = true;
110:         this.label1.Location = new System.Drawing.Point(12, 16);
111:         this.label1.Name = "label1";
112:         this.label1.Size = new System.Drawing.Size(93, 13);
113:         this.label1.TabIndex = 1;
114:         this.label1.Text = "Islem Türü Seçiniz";
115:         //
116:         // pnlTarihIslemleri
117:         //
118:         this.pnlTarihIslemleri.BorderStyle =
System.Windows.Forms.BorderStyle.FixedSingle;
119:         this.pnlTarihIslemleri.Controls.Add(this.btnAyFarki);
120:         this.pnlTarihIslemleri.Controls.Add(this.label8);
121:         this.pnlTarihIslemleri.Controls.Add(this.btnTarihAraligi);
122:         this.pnlTarihIslemleri.Controls.Add(this.btnYasHesaplaMetinsel);
123:         this.pnlTarihIslemleri.Controls.Add(this.btnYasHesapla);
124:         this.pnlTarihIslemleri.Controls.Add(this.label4);
125:         this.pnlTarihIslemleri.Controls.Add(this.lbSonuc);
126:         this.pnlTarihIslemleri.Controls.Add(this.dtpBitisTarihi);
127:         this.pnlTarihIslemleri.Controls.Add(this.dtpBaslangicTarihi);
128:         this.pnlTarihIslemleri.Controls.Add(this.label3);
129:         this.pnlTarihIslemleri.Controls.Add(this.label2);
130:         this.pnlTarihIslemleri.Location = new System.Drawing.Point(15, 56);
131:         this.pnlTarihIslemleri.Name = "pnlTarihIslemleri";
132:         this.pnlTarihIslemleri.Size = new System.Drawing.Size(374, 181);
133:         this.pnlTarihIslemleri.TabIndex = 2;
134:         //
135:         // btnAyFarki
136:         //
137:         this.btnAyFarki.Location = new System.Drawing.Point(286, 76);

```

```

138:         this.btnAyFarki.Name = "btnAyFarki";
139:         this.btnAyFarki.Size = new System.Drawing.Size(79, 23);
140:         this.btnAyFarki.TabIndex = 12;
141:         this.btnAyFarki.Text = "Ay Farki";
142:         this.btnAyFarki.UseVisualStyleBackColor = true;
143:         this.btnAyFarki.Click += new System.EventHandler(this.btnAyFarki_Click);
144:         //
145:         // label8
146:         //
147:         this.label8.AutoSize = true;
148:         this.label8.Dock = System.Windows.Forms.DockStyle.Top;
149:         this.label8.Location = new System.Drawing.Point(0, 0);
150:         this.label8.Name = "label8";
151:         this.label8.Size = new System.Drawing.Size(71, 13);
152:         this.label8.TabIndex = 11;
153:         this.label8.Text = "Tarih Islemleri";
154:         //
155:         // btnTarihAraligi
156:         //
157:         this.btnTarihAraligi.Location = new System.Drawing.Point(204, 76);
158:         this.btnTarihAraligi.Name = "btnTarihAraligi";
159:         this.btnTarihAraligi.Size = new System.Drawing.Size(76, 23);
160:         this.btnTarihAraligi.TabIndex = 10;
161:         this.btnTarihAraligi.Text = "Tarih Araligi";
162:         this.btnTarihAraligi.UseVisualStyleBackColor = true;
163:         this.btnTarihAraligi.Click += new
System.EventHandler(this.btnTarihAraligi_Click);
164:         //
165:         // btnYasHesaplaMetinsel
166:         //
167:         this.btnYasHesaplaMetinsel.Location = new System.Drawing.Point(84, 76);
168:         this.btnYasHesaplaMetinsel.Name = "btnYasHesaplaMetinsel";
169:         this.btnYasHesaplaMetinsel.Size = new System.Drawing.Size(113, 23);
170:         this.btnYasHesaplaMetinsel.TabIndex = 9;
171:         this.btnYasHesaplaMetinsel.Text = "Yas Hesapla (Metin)";
172:         this.btnYasHesaplaMetinsel.UseVisualStyleBackColor = true;
173:         this.btnYasHesaplaMetinsel.Click += new
System.EventHandler(this.btnYasHesaplaMetinsel_Click);
174:         //
175:         // btnYasHesapla
176:         //
177:         this.btnYasHesapla.Location = new System.Drawing.Point(3, 76);
178:         this.btnYasHesapla.Name = "btnYasHesapla";
179:         this.btnYasHesapla.Size = new System.Drawing.Size(75, 23);
180:         this.btnYasHesapla.TabIndex = 8;
181:         this.btnYasHesapla.Text = "Yas Hesapla";
182:         this.btnYasHesapla.UseVisualStyleBackColor = true;
183:         this.btnYasHesapla.Click += new
System.EventHandler(this.btnYasHesapla_Click);
184:         //
185:         // label4
186:         //
187:         this.label4.AutoSize = true;
188:         this.label4.Location = new System.Drawing.Point(51, 109);
189:         this.label4.Name = "label4";
190:         this.label4.Size = new System.Drawing.Size(38, 13);
191:         this.label4.TabIndex = 7;
192:         this.label4.Text = "Sonuç";
193:         //
194:         // lbSonuc
195:         //
196:         this.lbSonuc.FormattingEnabled = true;
197:         this.lbSonuc.HorizontalScrollbar = true;
198:         this.lbSonuc.Location = new System.Drawing.Point(117, 109);
199:         this.lbSonuc.Name = "lbSonuc";
200:         this.lbSonuc.Size = new System.Drawing.Size(200, 69);
201:         this.lbSonuc.TabIndex = 6;
202:         //
203:         // dtpBitisTarihi

```

```

204:         //
205:         this.dtpBitisTarihi.Format =
System.Windows.Forms.DateTimePickerFormat.Short;
206:         this.dtpBitisTarihi.Location = new System.Drawing.Point(117, 43);
207:         this.dtpBitisTarihi.Name = "dtpBitisTarihi";
208:         this.dtpBitisTarihi.ShowCheckBox = true;
209:         this.dtpBitisTarihi.Size = new System.Drawing.Size(200, 20);
210:         this.dtpBitisTarihi.TabIndex = 5;
211:         //
212:         // dtpBaslangicTarihi
213:         //
214:         this.dtpBaslangicTarihi.Format =
System.Windows.Forms.DateTimePickerFormat.Short;
215:         this.dtpBaslangicTarihi.Location = new System.Drawing.Point(117, 17);
216:         this.dtpBaslangicTarihi.Name = "dtpBaslangicTarihi";
217:         this.dtpBaslangicTarihi.ShowCheckBox = true;
218:         this.dtpBaslangicTarihi.Size = new System.Drawing.Size(200, 20);
219:         this.dtpBaslangicTarihi.TabIndex = 4;
220:         //
221:         // label3
222:         //
223:         this.label3.AutoSize = true;
224:         this.label3.Location = new System.Drawing.Point(44, 47);
225:         this.label3.Name = "label3";
226:         this.label3.Size = new System.Drawing.Size(55, 13);
227:         this.label3.TabIndex = 3;
228:         this.label3.Text = "Bitis Tarihi";
229:         //
230:         // label2
231:         //
232:         this.label2.AutoSize = true;
233:         this.label2.Location = new System.Drawing.Point(17, 24);
234:         this.label2.Name = "label2";
235:         this.label2.Size = new System.Drawing.Size(82, 13);
236:         this.label2.TabIndex = 2;
237:         this.label2.Text = "Baslangiç Tarihi";
238:         //
239:         // pnlSifreIslemleri
240:         //
241:         this.pnlSifreIslemleri.BorderStyle =
System.Windows.Forms.BorderStyle.FixedSingle;
242:         this.pnlSifreIslemleri.Controls.Add(this.label7);
243:         this.pnlSifreIslemleri.Controls.Add(this.btnSifreUygunMu);
244:         this.pnlSifreIslemleri.Controls.Add(this.btnRastgeleSifreUret);
245:         this.pnlSifreIslemleri.Controls.Add(this.btnSha512Hash);
246:         this.pnlSifreIslemleri.Controls.Add(this.btnSha256Hash);
247:         this.pnlSifreIslemleri.Controls.Add(this.label6);
248:         this.pnlSifreIslemleri.Controls.Add(this.lbSifreSonuc);
249:         this.pnlSifreIslemleri.Controls.Add(this.btnMd5Hash);
250:         this.pnlSifreIslemleri.Controls.Add(this.txtSifrelenecekMetin);
251:         this.pnlSifreIslemleri.Controls.Add(this.label5);
252:         this.pnlSifreIslemleri.Location = new System.Drawing.Point(411, 56);
253:         this.pnlSifreIslemleri.Name = "pnlSifreIslemleri";
254:         this.pnlSifreIslemleri.Size = new System.Drawing.Size(377, 181);
255:         this.pnlSifreIslemleri.TabIndex = 3;
256:         //
257:         // label7
258:         //
259:         this.label7.AutoSize = true;
260:         this.label7.Dock = System.Windows.Forms.DockStyle.Top;
261:         this.label7.Location = new System.Drawing.Point(0, 0);
262:         this.label7.Name = "label7";
263:         this.label7.Size = new System.Drawing.Size(68, 13);
264:         this.label7.TabIndex = 18;
265:         this.label7.Text = "Sifre Islemleri";
266:         this.label7.TextAlign = System.Drawing.ContentAlignment.TopCenter;
267:         //
268:         // btnSifreUygunMu
269:         //

```

```

270:         this.btnSifreUygunMu.Location = new System.Drawing.Point(173, 71);
271:         this.btnSifreUygunMu.Name = "btnSifreUygunMu";
272:         this.btnSifreUygunMu.Size = new System.Drawing.Size(145, 23);
273:         this.btnSifreUygunMu.TabIndex = 17;
274:         this.btnSifreUygunMu.Text = "Sifre Uygun Mu?";
275:         this.btnSifreUygunMu.UseVisualStyleBackColor = true;
276:         this.btnSifreUygunMu.Click += new
System.EventHandler(this.btnSifreUygunMu_Click);
277:         //
278:         // btnRastgeleSifreUret
279:         //
280:         this.btnRastgeleSifreUret.Location = new System.Drawing.Point(21, 71);
281:         this.btnRastgeleSifreUret.Name = "btnRastgeleSifreUret";
282:         this.btnRastgeleSifreUret.Size = new System.Drawing.Size(145, 23);
283:         this.btnRastgeleSifreUret.TabIndex = 16;
284:         this.btnRastgeleSifreUret.Text = "Rastgele Sifre Üret";
285:         this.btnRastgeleSifreUret.UseVisualStyleBackColor = true;
286:         this.btnRastgeleSifreUret.Click += new
System.EventHandler(this.btnRastgeleSifreUret_Click);
287:         //
288:         // btnSha512Hash
289:         //
290:         this.btnSha512Hash.Location = new System.Drawing.Point(243, 42);
291:         this.btnSha512Hash.Name = "btnSha512Hash";
292:         this.btnSha512Hash.Size = new System.Drawing.Size(75, 23);
293:         this.btnSha512Hash.TabIndex = 15;
294:         this.btnSha512Hash.Text = "SHA 512";
295:         this.btnSha512Hash.UseVisualStyleBackColor = true;
296:         this.btnSha512Hash.Click += new
System.EventHandler(this.btnSha512Hash_Click);
297:         //
298:         // btnSha256Hash
299:         //
300:         this.btnSha256Hash.Location = new System.Drawing.Point(131, 42);
301:         this.btnSha256Hash.Name = "btnSha256Hash";
302:         this.btnSha256Hash.Size = new System.Drawing.Size(75, 23);
303:         this.btnSha256Hash.TabIndex = 14;
304:         this.btnSha256Hash.Text = "SHA 256";
305:         this.btnSha256Hash.UseVisualStyleBackColor = true;
306:         this.btnSha256Hash.Click += new
System.EventHandler(this.btnSha256Hash_Click);
307:         //
308:         // label6
309:         //
310:         this.label6.AutoSize = true;
311:         this.label6.Location = new System.Drawing.Point(74, 104);
312:         this.label6.Name = "label6";
313:         this.label6.Size = new System.Drawing.Size(38, 13);
314:         this.label6.TabIndex = 13;
315:         this.label6.Text = "Sonuç";
316:         //
317:         // lbSifreSonuc
318:         //
319:         this.lbSifreSonuc.FormattingEnabled = true;
320:         this.lbSifreSonuc.HorizontalScrollbar = true;
321:         this.lbSifreSonuc.Location = new System.Drawing.Point(118, 100);
322:         this.lbSifreSonuc.Name = "lbSifreSonuc";
323:         this.lbSifreSonuc.Size = new System.Drawing.Size(200, 69);
324:         this.lbSifreSonuc.TabIndex = 12;
325:         //
326:         // btnMd5Hash
327:         //
328:         this.btnMd5Hash.Location = new System.Drawing.Point(21, 42);
329:         this.btnMd5Hash.Name = "btnMd5Hash";
330:         this.btnMd5Hash.Size = new System.Drawing.Size(75, 23);
331:         this.btnMd5Hash.TabIndex = 11;
332:         this.btnMd5Hash.Text = "MD5";
333:         this.btnMd5Hash.UseVisualStyleBackColor = true;
334:         this.btnMd5Hash.Click += new System.EventHandler(this.btnMd5Hash_Click);

```

```

335:         //
336:         // txtSifrelenecekMetin
337:         //
338:         this.txtSifrelenecekMetin.Location = new System.Drawing.Point(118, 16);
339:         this.txtSifrelenecekMetin.Name = "txtSifrelenecekMetin";
340:         this.txtSifrelenecekMetin.Size = new System.Drawing.Size(199, 20);
341:         this.txtSifrelenecekMetin.TabIndex = 4;
342:         //
343:         // label5
344:         //
345:         this.label5.AutoSize = true;
346:         this.label5.Location = new System.Drawing.Point(17, 19);
347:         this.label5.Name = "label5";
348:         this.label5.Size = new System.Drawing.Size(95, 13);
349:         this.label5.TabIndex = 3;
350:         this.label5.Text = "Sifrelenecek Metin";
351:         //
352:         // pnlSayiIslemleri
353:         //
354:         this.pnlSayiIslemleri.BorderStyle =
System.Windows.Forms.BorderStyle.FixedSingle;
355:         this.pnlSayiIslemleri.Controls.Add(this.label9);
356:         this.pnlSayiIslemleri.Controls.Add(this.btnOnayKoduOlustur);
357:         this.pnlSayiIslemleri.Controls.Add(this.btnMetneDonustur);
358:         this.pnlSayiIslemleri.Controls.Add(this.label10);
359:         this.pnlSayiIslemleri.Controls.Add(this.lbSayiSonuc);
360:         this.pnlSayiIslemleri.Controls.Add(this.btnSayisalMi);
361:         this.pnlSayiIslemleri.Controls.Add(this.txtSayi);
362:         this.pnlSayiIslemleri.Controls.Add(this.label11);
363:         this.pnlSayiIslemleri.Location = new System.Drawing.Point(15, 257);
364:         this.pnlSayiIslemleri.Name = "pnlSayiIslemleri";
365:         this.pnlSayiIslemleri.Size = new System.Drawing.Size(377, 181);
366:         this.pnlSayiIslemleri.TabIndex = 19;
367:         //
368:         // label9
369:         //
370:         this.label9.AutoSize = true;
371:         this.label9.Dock = System.Windows.Forms.DockStyle.Top;
372:         this.label9.Location = new System.Drawing.Point(0, 0);
373:         this.label9.Name = "label9";
374:         this.label9.Size = new System.Drawing.Size(67, 13);
375:         this.label9.TabIndex = 18;
376:         this.label9.Text = "Sayi Islemleri";
377:         this.label9.TextAlign = System.Drawing.ContentAlignment.TopCenter;
378:         //
379:         // btnOnayKoduOlustur
380:         //
381:         this.btnOnayKoduOlustur.Location = new System.Drawing.Point(243, 42);
382:         this.btnOnayKoduOlustur.Name = "btnOnayKoduOlustur";
383:         this.btnOnayKoduOlustur.Size = new System.Drawing.Size(75, 23);
384:         this.btnOnayKoduOlustur.TabIndex = 15;
385:         this.btnOnayKoduOlustur.Text = "Onay Kodu";
386:         this.btnOnayKoduOlustur.UseVisualStyleBackColor = true;
387:         this.btnOnayKoduOlustur.Click += new
System.EventHandler(this.btnOnayKoduOlustur_Click);
388:         //
389:         // btnMetneDonustur
390:         //
391:         this.btnMetneDonustur.Location = new System.Drawing.Point(102, 42);
392:         this.btnMetneDonustur.Name = "btnMetneDonustur";
393:         this.btnMetneDonustur.Size = new System.Drawing.Size(135, 23);
394:         this.btnMetneDonustur.TabIndex = 14;
395:         this.btnMetneDonustur.Text = "Para Metnine Dönüştür";
396:         this.btnMetneDonustur.UseVisualStyleBackColor = true;
397:         this.btnMetneDonustur.Click += new
System.EventHandler(this.btnMetneDonustur_Click);
398:         //
399:         // label10
400:         //

```



```

401:         this.label10.AutoSize = true;
402:         this.label10.Location = new System.Drawing.Point(18, 78);
403:         this.label10.Name = "label10";
404:         this.label10.Size = new System.Drawing.Size(38, 13);
405:         this.label10.TabIndex = 13;
406:         this.label10.Text = "Sonuç";
407:         //
408:         // lbSayiSonuc
409:         //
410:         this.lbSayiSonuc.FormattingEnabled = true;
411:         this.lbSayiSonuc.HorizontalScrollbar = true;
412:         this.lbSayiSonuc.Location = new System.Drawing.Point(62, 78);
413:         this.lbSayiSonuc.Name = "lbSayiSonuc";
414:         this.lbSayiSonuc.Size = new System.Drawing.Size(254, 82);
415:         this.lbSayiSonuc.TabIndex = 12;
416:         //
417:         // btnSayisalMi
418:         //
419:         this.btnSayisalMi.Location = new System.Drawing.Point(21, 42);
420:         this.btnSayisalMi.Name = "btnSayisalMi";
421:         this.btnSayisalMi.Size = new System.Drawing.Size(75, 23);
422:         this.btnSayisalMi.TabIndex = 11;
423:         this.btnSayisalMi.Text = "Sayisal Mi?";
424:         this.btnSayisalMi.UseVisualStyleBackColor = true;
425:         this.btnSayisalMi.Click += new
System.EventHandler(this.btnSayisalMi_Click);
426:         //
427:         // txtSayi
428:         //
429:         this.txtSayi.Location = new System.Drawing.Point(118, 16);
430:         this.txtSayi.Name = "txtSayi";
431:         this.txtSayi.Size = new System.Drawing.Size(199, 20);
432:         this.txtSayi.TabIndex = 4;
433:         //
434:         // label11
435:         //
436:         this.label11.AutoSize = true;
437:         this.label11.Location = new System.Drawing.Point(17, 19);
438:         this.label11.Name = "label11";
439:         this.label11.Size = new System.Drawing.Size(95, 13);
440:         this.label11.TabIndex = 3;
441:         this.label11.Text = "Sifrelenecek Metin";
442:         //
443:         // pnlKisiIslemleri
444:         //
445:         this.pnlKisiIslemleri.BorderStyle =
System.Windows.Forms.BorderStyle.FixedSingle;
446:         this.pnlKisiIslemleri.Controls.Add(this.label12);
447:         this.pnlKisiIslemleri.Controls.Add(this.btnEPostaDogrula);
448:         this.pnlKisiIslemleri.Controls.Add(this.btnIbanDogrula);
449:         this.pnlKisiIslemleri.Controls.Add(this.label13);
450:         this.pnlKisiIslemleri.Controls.Add(this.lbKisiSonuc);
451:         this.pnlKisiIslemleri.Controls.Add(this.btnTcKimlikDogrula);
452:         this.pnlKisiIslemleri.Controls.Add(this.txtKisiMetni);
453:         this.pnlKisiIslemleri.Controls.Add(this.label14);
454:         this.pnlKisiIslemleri.Location = new System.Drawing.Point(411, 257);
455:         this.pnlKisiIslemleri.Name = "pnlKisiIslemleri";
456:         this.pnlKisiIslemleri.Size = new System.Drawing.Size(377, 181);
457:         this.pnlKisiIslemleri.TabIndex = 20;
458:         //
459:         // label12
460:         //
461:         this.label12.AutoSize = true;
462:         this.label12.Dock = System.Windows.Forms.DockStyle.Top;
463:         this.label12.Location = new System.Drawing.Point(0, 0);
464:         this.label12.Name = "label12";
465:         this.label12.Size = new System.Drawing.Size(63, 13);
466:         this.label12.TabIndex = 18;
467:         this.label12.Text = "Kisi Islemleri";

```

```

468:         this.label12.TextAlign = System.Drawing.ContentAlignment.TopCenter;
469:         //
470:         // btnEPostaDogrula
471:         //
472:         this.btnEPostaDogrula.Location = new System.Drawing.Point(223, 42);
473:         this.btnEPostaDogrula.Name = "btnEPostaDogrula";
474:         this.btnEPostaDogrula.Size = new System.Drawing.Size(95, 23);
475:         this.btnEPostaDogrula.TabIndex = 15;
476:         this.btnEPostaDogrula.Text = "E-Posta Dogrula";
477:         this.btnEPostaDogrula.UseVisualStyleBackColor = true;
478:         this.btnEPostaDogrula.Click += new
System.EventHandler(this.btnEPostaDogrula_Click);
479:         //
480:         // btnIbanDogrula
481:         //
482:         this.btnIbanDogrula.Location = new System.Drawing.Point(131, 42);
483:         this.btnIbanDogrula.Name = "btnIbanDogrula";
484:         this.btnIbanDogrula.Size = new System.Drawing.Size(86, 23);
485:         this.btnIbanDogrula.TabIndex = 14;
486:         this.btnIbanDogrula.Text = "IBAN Dogrula";
487:         this.btnIbanDogrula.UseVisualStyleBackColor = true;
488:         this.btnIbanDogrula.Click += new
System.EventHandler(this.btnIbanDogrula_Click);
489:         //
490:         // label13
491:         //
492:         this.label13.AutoSize = true;
493:         this.label13.Location = new System.Drawing.Point(18, 78);
494:         this.label13.Name = "label13";
495:         this.label13.Size = new System.Drawing.Size(38, 13);
496:         this.label13.TabIndex = 13;
497:         this.label13.Text = "Sonuç";
498:         //
499:         // lbKisiSonuc
500:         //
501:         this.lbKisiSonuc.FormattingEnabled = true;
502:         this.lbKisiSonuc.HorizontalScrollbar = true;
503:         this.lbKisiSonuc.Location = new System.Drawing.Point(62, 78);
504:         this.lbKisiSonuc.Name = "lbKisiSonuc";
505:         this.lbKisiSonuc.Size = new System.Drawing.Size(254, 82);
506:         this.lbKisiSonuc.TabIndex = 12;
507:         //
508:         // btnTcKimlikDogrula
509:         //
510:         this.btnTcKimlikDogrula.Location = new System.Drawing.Point(21, 42);
511:         this.btnTcKimlikDogrula.Name = "btnTcKimlikDogrula";
512:         this.btnTcKimlikDogrula.Size = new System.Drawing.Size(104, 23);
513:         this.btnTcKimlikDogrula.TabIndex = 11;
514:         this.btnTcKimlikDogrula.Text = "TC Kimlik Dogrula";
515:         this.btnTcKimlikDogrula.UseVisualStyleBackColor = true;
516:         this.btnTcKimlikDogrula.Click += new
System.EventHandler(this.btnTcKimlikDogrula_Click);
517:         //
518:         // txtKisiMetni
519:         //
520:         this.txtKisiMetni.Location = new System.Drawing.Point(118, 16);
521:         this.txtKisiMetni.Name = "txtKisiMetni";
522:         this.txtKisiMetni.Size = new System.Drawing.Size(199, 20);
523:         this.txtKisiMetni.TabIndex = 4;
524:         //
525:         // label14
526:         //
527:         this.label14.AutoSize = true;
528:         this.label14.Location = new System.Drawing.Point(17, 19);
529:         this.label14.Name = "label14";
530:         this.label14.Size = new System.Drawing.Size(33, 13);
531:         this.label14.TabIndex = 3;
532:         this.label14.Text = "Metin";
533:         //

```

```

534:         // pnlMetinIslemleri
535:         //
536:         this.pnlMetinIslemleri.BorderStyle =
System.Windows.Forms.BorderStyle.FixedSingle;
537:         this.pnlMetinIslemleri.Controls.Add(this.btnTelefonNoDuzenle);
538:         this.pnlMetinIslemleri.Controls.Add(this.btnMetinSifrele);
539:         this.pnlMetinIslemleri.Controls.Add(this.label15);
540:         this.pnlMetinIslemleri.Controls.Add(this.btnTurkceKarakterDuzelt);
541:         this.pnlMetinIslemleri.Controls.Add(this.btnIlkHarfiBuyukYap);
542:         this.pnlMetinIslemleri.Controls.Add(this.label16);
543:         this.pnlMetinIslemleri.Controls.Add(this.lbMetinSonuc);
544:         this.pnlMetinIslemleri.Controls.Add(this.btnIlkHarfleriBuyukYap);
545:         this.pnlMetinIslemleri.Controls.Add(this.txtMetin);
546:         this.pnlMetinIslemleri.Controls.Add(this.label17);
547:         this.pnlMetinIslemleri.Location = new System.Drawing.Point(15, 458);
548:         this.pnlMetinIslemleri.Name = "pnlMetinIslemleri";
549:         this.pnlMetinIslemleri.Size = new System.Drawing.Size(773, 181);
550:         this.pnlMetinIslemleri.TabIndex = 21;
551:         //
552:         // btnTelefonNoDuzenle
553:         //
554:         this.btnTelefonNoDuzenle.Location = new System.Drawing.Point(503, 42);
555:         this.btnTelefonNoDuzenle.Name = "btnTelefonNoDuzenle";
556:         this.btnTelefonNoDuzenle.Size = new System.Drawing.Size(134, 23);
557:         this.btnTelefonNoDuzenle.TabIndex = 20;
558:         this.btnTelefonNoDuzenle.Text = "Telefon No Düzenle";
559:         this.btnTelefonNoDuzenle.UseVisualStyleBackColor = true;
560:         this.btnTelefonNoDuzenle.Click += new
System.EventHandler(this.btnTelefonNoDuzenle_Click);
561:         //
562:         // btnMetinSifrele
563:         //
564:         this.btnMetinSifrele.Location = new System.Drawing.Point(363, 42);
565:         this.btnMetinSifrele.Name = "btnMetinSifrele";
566:         this.btnMetinSifrele.Size = new System.Drawing.Size(134, 23);
567:         this.btnMetinSifrele.TabIndex = 19;
568:         this.btnMetinSifrele.Text = "Metin Sifrele";
569:         this.btnMetinSifrele.UseVisualStyleBackColor = true;
570:         this.btnMetinSifrele.Click += new
System.EventHandler(this.btnMetinSifrele_Click);
571:         //
572:         // label15
573:         //
574:         this.label15.AutoSize = true;
575:         this.label15.Dock = System.Windows.Forms.DockStyle.Top;
576:         this.label15.Location = new System.Drawing.Point(0, 0);
577:         this.label15.Name = "label15";
578:         this.label15.Size = new System.Drawing.Size(73, 13);
579:         this.label15.TabIndex = 18;
580:         this.label15.Text = "Metin Islemleri";
581:         this.label15.TextAlign = System.Drawing.ContentAlignment.TopCenter;
582:         //
583:         // btnTurkceKarakterDuzelt
584:         //
585:         this.btnTurkceKarakterDuzelt.Location = new System.Drawing.Point(223, 42);
586:         this.btnTurkceKarakterDuzelt.Name = "btnTurkceKarakterDuzelt";
587:         this.btnTurkceKarakterDuzelt.Size = new System.Drawing.Size(134, 23);
588:         this.btnTurkceKarakterDuzelt.TabIndex = 15;
589:         this.btnTurkceKarakterDuzelt.Text = "Türkçe Karakter Düzelt";
590:         this.btnTurkceKarakterDuzelt.UseVisualStyleBackColor = true;
591:         this.btnTurkceKarakterDuzelt.Click += new
System.EventHandler(this.btnTurkceKarakterDuzelt_Click);
592:         //
593:         // btnIlkHarfiBuyukYap
594:         //
595:         this.btnIlkHarfiBuyukYap.Location = new System.Drawing.Point(131, 42);
596:         this.btnIlkHarfiBuyukYap.Name = "btnIlkHarfiBuyukYap";
597:         this.btnIlkHarfiBuyukYap.Size = new System.Drawing.Size(86, 23);
598:         this.btnIlkHarfiBuyukYap.TabIndex = 14;

```



```

599:         this.btnIlkHarfiBuyukYap.Text = "Ilk Harf Büyük";
600:         this.btnIlkHarfiBuyukYap.UseVisualStyleBackColor = true;
601:         this.btnIlkHarfiBuyukYap.Click += new
System.EventHandler(this.btnIlkHarfiBuyukYap_Click);
602:         //
603:         // label16
604:         //
605:         this.label16.AutoSize = true;
606:         this.label16.Location = new System.Drawing.Point(18, 78);
607:         this.label16.Name = "label16";
608:         this.label16.Size = new System.Drawing.Size(38, 13);
609:         this.label16.TabIndex = 13;
610:         this.label16.Text = "Sonuç";
611:         //
612:         // lbMetinSonuc
613:         //
614:         this.lbMetinSonuc.FormattingEnabled = true;
615:         this.lbMetinSonuc.HorizontalScrollbar = true;
616:         this.lbMetinSonuc.Location = new System.Drawing.Point(62, 78);
617:         this.lbMetinSonuc.Name = "lbMetinSonuc";
618:         this.lbMetinSonuc.Size = new System.Drawing.Size(652, 82);
619:         this.lbMetinSonuc.TabIndex = 12;
620:         //
621:         // btnIlkHarfleriBuyukYap
622:         //
623:         this.btnIlkHarfleriBuyukYap.Location = new System.Drawing.Point(21, 42);
624:         this.btnIlkHarfleriBuyukYap.Name = "btnIlkHarfleriBuyukYap";
625:         this.btnIlkHarfleriBuyukYap.Size = new System.Drawing.Size(104, 23);
626:         this.btnIlkHarfleriBuyukYap.TabIndex = 11;
627:         this.btnIlkHarfleriBuyukYap.Text = "Ilk Harfler Büyük";
628:         this.btnIlkHarfleriBuyukYap.UseVisualStyleBackColor = true;
629:         this.btnIlkHarfleriBuyukYap.Click += new
System.EventHandler(this.btnIlkHarfleriBuyukYap_Click);
630:         //
631:         // txtMetin
632:         //
633:         this.txtMetin.Location = new System.Drawing.Point(119, 12);
634:         this.txtMetin.Name = "txtMetin";
635:         this.txtMetin.Size = new System.Drawing.Size(595, 20);
636:         this.txtMetin.TabIndex = 4;
637:         //
638:         // label17
639:         //
640:         this.label17.AutoSize = true;
641:         this.label17.Location = new System.Drawing.Point(17, 19);
642:         this.label17.Name = "label17";
643:         this.label17.Size = new System.Drawing.Size(33, 13);
644:         this.label17.TabIndex = 3;
645:         this.label17.Text = "Metin";
646:         //
647:         // FrmAnaSayfa
648:         //
649:         this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
650:         this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
651:         this.AutoSizeMode = System.Windows.Forms.AutoSizeMode.GrowAndShrink;
652:         this.ClientSize = new System.Drawing.Size(800, 686);
653:         this.Controls.Add(this.pnlMetinIslemleri);
654:         this.Controls.Add(this.pnlKisiIslemleri);
655:         this.Controls.Add(this.pnlSayiIslemleri);
656:         this.Controls.Add(this.pnlSifreIslemleri);
657:         this.Controls.Add(this.pnlTarihIslemleri);
658:         this.Controls.Add(this.label1);
659:         this.Controls.Add(this.cmbIslemTuru);
660:         this.Name = "FrmAnaSayfa";
661:         this.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;
662:         this.Text = "Islemler Ekrani";
663:         this.Load += new System.EventHandler(this.FrmAnaSayfa_Load);
664:         this.pnlTarihIslemleri.ResumeLayout(false);
665:         this.pnlTarihIslemleri.PerformLayout();

```

```

666:         this.pnlSifreIslemleri.ResumeLayout(false);
667:         this.pnlSifreIslemleri.PerformLayout();
668:         this.pnlSayiIslemleri.ResumeLayout(false);
669:         this.pnlSayiIslemleri.PerformLayout();
670:         this.pnlKisiIslemleri.ResumeLayout(false);
671:         this.pnlKisiIslemleri.PerformLayout();
672:         this.pnlMetinIslemleri.ResumeLayout(false);
673:         this.pnlMetinIslemleri.PerformLayout();
674:         this.ResumeLayout(false);
675:         this.PerformLayout();
676:
677:     }
678:
679:     #endregion
680:
681:     private System.Windows.Forms.ComboBox cmbIslemTuru;
682:     private System.Windows.Forms.Label label1;
683:     private System.Windows.Forms.Panel pnlTarihIslemleri;
684:     private System.Windows.Forms.Label label2;
685:     private System.Windows.Forms.Label label3;
686:     private System.Windows.Forms.DateTimePicker dtpBaslangicTarihi;
687:     private System.Windows.Forms.DateTimePicker dtpBitisTarihi;
688:     private System.Windows.Forms.ListBox lbSonuc;
689:     private System.Windows.Forms.Label label4;
690:     private System.Windows.Forms.Button btnYasHesapla;
691:     private System.Windows.Forms.Button btnYasHesaplaMetinsel;
692:     private System.Windows.Forms.Button btnTarihAraligi;
693:     private System.Windows.Forms.Panel pnlSifreIslemleri;
694:     private System.Windows.Forms.Label label5;
695:     private System.Windows.Forms.TextBox txtSifrelenecekMetin;
696:     private System.Windows.Forms.Button btnMd5Hash;
697:     private System.Windows.Forms.Label label6;
698:     private System.Windows.Forms.ListBox lbSifreSonuc;
699:     private System.Windows.Forms.Button btnSha256Hash;
700:     private System.Windows.Forms.Button btnSha512Hash;
701:     private System.Windows.Forms.Button btnRastgeleSifreUret;
702:     private System.Windows.Forms.Button btnSifreUygunMu;
703:     private System.Windows.Forms.Label label8;
704:     private System.Windows.Forms.Label label7;
705:     private System.Windows.Forms.Button btnAyFarki;
706:     private System.Windows.Forms.Panel pnlSayiIslemleri;
707:     private System.Windows.Forms.Label label9;
708:     private System.Windows.Forms.Button btnOnayKoduOlustur;
709:     private System.Windows.Forms.Button btnMetneDonustur;
710:     private System.Windows.Forms.Label label10;
711:     private System.Windows.Forms.ListBox lbSayiSonuc;
712:     private System.Windows.Forms.Button btnSayisalMi;
713:     private System.Windows.Forms.TextBox txtSayi;
714:     private System.Windows.Forms.Label label11;
715:     private System.Windows.Forms.Panel pnlKisiIslemleri;
716:     private System.Windows.Forms.Label label12;
717:     private System.Windows.Forms.Button btnEPostaDogrula;
718:     private System.Windows.Forms.Button btnIbanDogrula;
719:     private System.Windows.Forms.Label label13;
720:     private System.Windows.Forms.ListBox lbKisiSonuc;
721:     private System.Windows.Forms.Button btnTcKimlikDogrula;
722:     private System.Windows.Forms.TextBox txtKisiMetni;
723:     private System.Windows.Forms.Label label14;
724:     private System.Windows.Forms.Panel pnlMetinIslemleri;
725:     private System.Windows.Forms.Label label15;
726:     private System.Windows.Forms.Button btnTurkceKarakterDuzelt;
727:     private System.Windows.Forms.Button btnIlkHarfiBuyukYap;
728:     private System.Windows.Forms.Label label16;
729:     private System.Windows.Forms.ListBox lbMetinSonuc;
730:     private System.Windows.Forms.Button btnIlkHarfleriBuyukYap;
731:     private System.Windows.Forms.TextBox txtMetin;
732:     private System.Windows.Forms.Label label17;
733:     private System.Windows.Forms.Button btnMetinSifrele;
734:     private System.Windows.Forms.Button btnTelefonNoDuzenle;

```

```

735:     }
736: }
737:

```

Namespaces

Name	Description
CSharpHelperMethods (see page 1)	This is namespace CSharpHelperMethods.

1.2.8 Kisilslemleri.cs

This is file Kisilslemleri.cs.

Body Source

```

1: ?using System;
2: using System.Text;
3: using System.Text.RegularExpressions;
4:
5: namespace CSharpHelperMethods.YardimciSiniflar
6: {
7:     /// <summary>
8:     /// Kisi ile iliskili yardimci metotlarin tamamini içeren sinif
9:     /// </summary>
10:    public static class KisiIslemleri
11:    {
12:        /// <summary>
13:        /// Girilen TC Kimlik Numarasinin, standart TC Kimlik No algoritmasina uygun
14:        olup
15:        /// olmadigini kontrol eden metottur.
16:        /// </summary>
17:        /// <param name="tcKimlikNo">TC Kimlik No</param>
18:        /// <returns>TC Kimlik No'nun Uygun Olup Olmadigi Bilgisi</returns>
19:        public static bool DogrulaTcKimlikNo(string tcKimlikNo)
20:        {
21:            var tcNo = Convert.ToInt64(tcKimlikNo);
22:            var atcno = tcNo / 100;
23:            var btcno = tcNo / 100;
24:            var c1 = atcno % 10; atcno = atcno / 10;
25:            var c2 = atcno % 10; atcno = atcno / 10;
26:            var c3 = atcno % 10; atcno = atcno / 10;
27:            var c4 = atcno % 10; atcno = atcno / 10;
28:            var c5 = atcno % 10; atcno = atcno / 10;
29:            var c6 = atcno % 10; atcno = atcno / 10;
30:            var c7 = atcno % 10; atcno = atcno / 10;
31:            var c8 = atcno % 10; atcno = atcno / 10;
32:            var c9 = atcno % 10; atcno = atcno / 10;
33:            var q1 = ((10 - (((c1 + c3 + c5 + c7 + c9) * 3) + (c2 + c4 + c6 + c8)) %
34:            10)) % 10);
35:            var q2 = ((10 - (((((c2 + c4 + c6 + c8) + q1) * 3) + (c1 + c3 + c5 + c7 +
36:            c9)) % 10)) % 10)) % 10);
37:            var dogrula = ((btcno * 100) + (q1 * 10) + q2 == tcNo);
38:            return dogrula;
39:        }
40:        /// <summary>
41:        /// Girilen IBAN'in standar IBAN algoritmasina uygun olup olmadigini dogrular
42:        /// </summary>
43:        /// <code>DogrulaIBAN("TR560006200000012990022604")</code>
44:        /// <param name="iban">Uluslar arasi banka hesap numarasi (IBAN) </param>
45:        /// <returns>IBAN uygunsa true, degilse false degeri döner</returns>
46:        public static bool DogrulaIBAN(string iban)
47:        {
48:            iban = iban.ToUpper();

```

```

48:         if (string.IsNullOrEmpty(iban))
49:             return false;
50:
51:         if (!Regex.IsMatch(iban, "^[A-Z0-9]"))
52:             return false;
53:
54:         iban = iban.Replace(" ", String.Empty);
55:         var bank = iban.Substring(4, iban.Length - 4) + iban.Substring(0, 4);
56:         var asciiShift = 55;
57:         var sb = new StringBuilder();
58:         foreach (var c in bank)
59:         {
60:             sb.Append(char.IsLetter(c) ? c - asciiShift : int.Parse(c.ToString()));
61:         }
62:
63:         string checksumString = sb.ToString();
64:         int checksum = int.Parse(checksumString.Substring(0, 1));
65:         for (int i = 1; i < checksumString.Length; i++)
66:         {
67:             int v = int.Parse(checksumString.Substring(i, 1));
68:             checksum *= 10;
69:             checksum += v;
70:             checksum %= 97;
71:         }
72:         return checksum == 1;
73:     }
74:
75:     /// <summary>
76:     /// Ekranlardan girilen bir e-posta adresinin geçerli bir e-posta adresi olup
    olmadigini
77:     /// Kontrol eden metottur
78:     /// </summary>
79:     /// <param name="ePosta">E-Posta Adresi</param>
80:     /// <returns>E-Posta Adresinin Geçerli Olup Olmadigi Bilgisi</returns>
81:     public static bool DogrulaEPosta(string ePosta)
82:     {
83:         const string pattern = @"^(([\w-]+\.[\w-]+|([a-zA-Z]{1}|[\w-]{2,}))@"
84:             + @"((([0-1]?[0-9]{1,2}|25[0-5]|2[0-4][0-9])\.")
85:             + @"([0-1]?[0-9]{1,2}|25[0-5]|2[0-4][0-9])\.")
86:             + @"([0-1]?[0-9]{1,2}|25[0-5]|2[0-4][0-9])\.")
87:             + @"([0-1]?[0-9]{1,2}|25[0-5]|2[0-4][0-9])){1}|")
88:             + @"([a-zA-Z]+[\w-]+\.[\w-]+[a-zA-Z]{2,4})$";
89:         return Regex.IsMatch(ePosta, pattern);
90:     }
91: }
92: }

```

Namespaces

Name	Description
CSharpHelperMethods (see page 1)	This is namespace CSharpHelperMethods.

1.2.9 MetinIslemleri.cs

This is file MetinIslemleri.cs.

Body Source

```

1: ?using System;
2: using System.Collections.Generic;
3: using System.Globalization;
4: using System.Linq;
5: using System.Text;
6: using System.Text.RegularExpressions;
7:

```

```

8: namespace CSharpHelperMethods.YardimciSiniflar
9: {
10:     /// <summary>
11:     /// Metinsel veri kontrol metotlarini barindiran sinif
12:     /// </summary>
13:     public static class MetinIslemleri
14:     {
15:         /// <summary>
16:         /// Girilen metnin basinda virgöl olup olmadigini kontrol eden metottur.
17:         /// </summary>
18:         /// <param name="text">Girdi metni</param>
19:         /// <returns>Girilen Metnin Olmasi Gerektigi Hali</returns>
20:         public static string IdealMetinGetir(string text)
21:         {
22:             if (!string.IsNullOrEmpty(text) && text[0] == ',')
23:             {
24:                 text = text.Substring(1, text.Length - 1);
25:             }
26:             return text;
27:         }
28:
29:         public static string MetinZararliKarakterDuzelt(string text)
30:         {
31:             text = text.Replace("&#252;", "ü");
32:             return text;
33:         }
34:
35:         public static string MetinSifrele(string metin)
36:         {
37:             if (!string.IsNullOrEmpty(metin))
38:             {
39:                 var metinDuzenleyici = new StringBuilder(metin);
40:                 if (metin.Length == 10)
41:                 {
42:                     metinDuzenleyici.Remove(2, 6);
43:                     metinDuzenleyici.Insert(2, "*****");
44:                     metin = metinDuzenleyici.ToString();
45:                 }
46:                 else if (metin.Length == 11)
47:                 {
48:                     metinDuzenleyici.Remove(2, 7);
49:                     metinDuzenleyici.Insert(2, "*****");
50:                     metin = metinDuzenleyici.ToString();
51:                 }
52:             }
53:             return metin;
54:         }
55:
56:         /// <summary>
57:         /// Girilen metnin basinda virgöl olup olmadigini kontrol eden metottur.
58:         /// </summary>
59:         /// <param name="text">Girdi metni</param>
60:         /// <returns>Girilen Metnin Olmasi Gerektigi Hali</returns>
61:         public static string TurkceKarakterleriDuzelt(string text)
62:         {
63:             text = text.Replace("ü", "u");
64:             text = text.Replace("i", "i");
65:             text = text.Replace("ö", "o");
66:             text = text.Replace("ü", "u");
67:             text = text.Replace("s", "s");
68:             text = text.Replace("g", "g");
69:             text = text.Replace("ç", "c");
70:             text = text.Replace("Ü", "U");
71:             text = text.Replace("İ", "I");
72:             text = text.Replace("Ö", "O");
73:             text = text.Replace("Ü", "U");
74:             text = text.Replace("S", "S");
75:             text = text.Replace("G", "G");
76:             text = text.Replace("Ç", "C");

```

```

77:         return text;
78:     }
79:
80:     /// <summary>
81:     /// Girilen metnin basinda virgöl olup olmadigini kontrol eden metottur.
82:     /// </summary>
83:     /// <param name="text">Girdi metni</param>
84:     /// <returns>Girilen Metnin Olmasi Gerektigi Hali</returns>
85:     public static string UTF8Duzelt(string text)
86:     {
87:         text = text.Replace("&cedil;", "ç");
88:         text = text.Replace("&ouml;", "ö");
89:         text = text.Replace("&uuml;", "ü");
90:         text = text.Replace("&ccedil;", "ç");
91:         text = text.Replace("&ouml;", "ö");
92:         text = text.Replace("&uuml;", "ü");
93:         return text;
94:     }
95:
96:     /// <summary>
97:     /// Girilen metnin basinda virgöl olup olmadigini kontrol eden metottur.
98:     /// </summary>
99:     /// <param name="text">Girdi metni</param>
100:    /// <returns>Girilen Metnin Olmasi Gerektigi Hali</returns>
101:    public static string HTMLDuzelt(string text)
102:    {
103:        List<string> sc = new List<string>();
104:        // get rid of unnecessary tag spans (comments and title)
105:        sc.Add(@"<!--(\w|\W)+?-->");
106:        sc.Add(@"<title>(\w|\W)+?</title>");
107:        // Get rid of classes and styles
108:        sc.Add(@"\s?class=\w+");
109:        sc.Add(@"\s+style='[^']*'");
110:        // Get rid of unnecessary tags
111:        sc.Add(
112:            @"<(meta|link|?o:|/?style|/?div|/?st\d|/?head|/?html|body|/?body|/?span|![][>]*?>");
113:        // Get rid of empty paragraph tags
114:        sc.Add(@"(<[>]+>)+&nbsp;(</\w+>)+");
115:        sc.Add(@"<a[>]*>([<]+)</a>");
116:        sc.Add(@"<p[>]*>([<]+)</p>");
117:        // remove bizarre v: element attached to <img> tag
118:        sc.Add(@"\s+v:\w+=" + "[^"]+" + "");
119:        // remove extra lines
120:        sc.Add(@"(\n\r){2,}");
121:        sc.Add("&nbsp;");
122:        foreach (string s in sc)
123:        {
124:            text = Regex.Replace(text, s, string.Empty, RegexOptions.IgnoreCase);
125:        }
126:        text = System.Text.RegularExpressions.Regex.Replace(text, "<.*?>",
String.Empty);
127:        text = text.Replace("\r", " ").Replace("\n", " ").Replace(" ", "
").Trim();
128:        text = text.Replace("&rsquo;", "'");
129:        text = text.Replace("&rsquo;", "'");
130:        text = text.Replace("&rdquo;", "\"");
131:        text = text.Replace("&rdquo;", "\"");
132:
133:        text = text.Replace("&lsquo;", "'");
134:        text = text.Replace("&lsquo;", "'");
135:        text = text.Replace("&ldquo;", "\"");
136:        text = text.Replace("&ldquo;", "\"");
137:
138:        text = text.Replace("&lt;", "<");
139:        text = text.Replace("&gt;", ">");
140:        text = text.Replace("&", "&");
141:        //text = text.Replace("&lt;", "");
142:        //text = text.Replace("&p&gt;", "");

```

```

143:         //text = text.Replace("/p>";, "");
144:         //text = text.Replace("&";, "");
145:         //return text;
146:         return text;
147:     }
148:
149:     public static string MetinIlkKarakterBuyukDigerleriKucukYap(string text) =>
text.First().ToString().ToUpper() + text.Substring(1);
150:
151:     /// <summary>
152:     /// Girilen bir metindeki tüm kelimelerin ilk harflerinin
153:     /// Büyük yapilmasi için hazirlanan metottur
154:     /// </summary>
155:     /// <param name="input">Girilen metin</param>
156:     /// <returns>Metnin Kelimelerinin İlk Harflerinin Büyüültülmüs Hali</returns>
157:     public static string MetinIlkKarakterleriBuyukYap(this string input)
158:     {
159:         string result = "";
160:         var wordList = input.Split(' ');
161:         var cultureInfo = new CultureInfo("tr-TR");
162:         foreach (var item in wordList)
163:         {
164:             if (item.Length > 1)
165:                 result = result + " " + item.Substring(0, 1).ToUpper(cultureInfo) +
item.Substring(1).ToLower(cultureInfo) + " ";
166:             else if (item.Length == 1)
167:                 result = result + " " + item.ToUpper(cultureInfo) + " ";
168:         }
169:         return result;
170:     }
171:
172:     /// <summary>
173:     /// Girilen telefon numarasinin SMS gönderilecek hale getirilmesini saglayan
metottur.
174:     /// </summary>
175:     /// <param name="telefonNo">Telefon No Bilgisi</param>
176:     /// <returns>Girilen Telefon Numarasinin SMS Gönderilecek Hali</returns>
177:     public static string TelefonNoDuzenle(string telefonNo)
178:     {
179:         if (!string.IsNullOrEmpty(telefonNo))
180:         {
181:             telefonNo = telefonNo.Replace("-", "").Replace(" ", "").Replace("(",
"").Replace(")", "");
182:             if (telefonNo.Length == 11)
183:             {
184:                 telefonNo = telefonNo.Substring(telefonNo.Length - 10);
185:             }
186:             return telefonNo;
187:         }
188:         else
189:         {
190:             return string.Empty;
191:         }
192:     }
193:
194:     /// <summary>
195:     /// Girilen metnin bosluklarinin kaldırilmasini saglayan metottur.
196:     /// </summary>
197:     /// <param name="metin">Telefon No Bilgisi</param>
198:     /// <returns>Girilen Metnin Istenen Hali</returns>
199:     public static string BoslukKaldir(string metin)
200:     {
201:         if (!string.IsNullOrEmpty(metin))
202:         {
203:             metin = metin.Replace("-", "").Replace(" ", "").Replace("(",
"").Replace(")", "");
204:             return metin;
205:         }
206:         else

```



```
207:         {  
208:             return string.Empty;  
209:         }  
210:     }  
211: }  
212: }
```

Namespaces

Name	Description
CSharpHelperMethods (see page 1)	This is namespace CSharpHelperMethods.

1.2.10 Program.cs

This is file Program.cs.

Body Source

```
1: ?using System;  
2: using System.Collections.Generic;  
3: using System.Linq;  
4: using System.Windows.Forms;  
5:  
6: namespace CSharpHelperMethods  
7: {  
8:     static class Program  
9:     {  
10:         /// <summary>  
11:         /// The main entry point for the application.  
12:         /// </summary>  
13:         [STAThread]  
14:         static void Main()  
15:         {  
16:             Application.EnableVisualStyles();  
17:             Application.SetCompatibleTextRenderingDefault(false);  
18:             Application.Run(new FrmAnaSayfa());  
19:         }  
20:     }  
21: }
```

Namespaces

Name	Description
CSharpHelperMethods (see page 1)	This is namespace CSharpHelperMethods.

1.2.11 Resources.Designer.cs

This code was generated by a tool. Runtime Version:4.0.30319.42000

Changes to this file may cause incorrect behavior and will be lost if the code is regenerated.

Body Source

```
1: ?//-----  
2: // <auto-generated>  
3: //     This code was generated by a tool.  
4: //     Runtime Version:4.0.30319.42000  
5: //  
6: //     Changes to this file may cause incorrect behavior and will be lost if  
7: //     the code is regenerated.  
8: // </auto-generated>
```



```

 9:  //---------------------------------------------------------------------
10:
11:  namespace CSharpHelperMethods.Properties
12:  {
13:
14:
15:      /// <summary>
16:      ///     A strongly-typed resource class, for looking up localized strings, etc.
17:      /// </summary>
18:      // This class was auto-generated by the StronglyTypedResourceBuilder
19:      // class via a tool like ResGen or Visual Studio.
20:      // To add or remove a member, edit your .ResX file then rerun ResGen
21:      // with the /str option, or rebuild your VS project.
22:
23:      [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Resources.Tools.StronglyTypedResourceBuilder",
24:          "4.0.0.0")]
25:      [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
26:      [global::System.Runtime.CompilerServices.CompilerGeneratedAttribute()]
27:      internal class Resources
28:      {
29:
30:          private static global::System.Resources.ResourceManager resourceMan;
31:
32:          private static global::System.Globalization.CultureInfo resourceCulture;
33:
34:          [global::System.Diagnostics.CodeAnalysis.SuppressMessageAttribute("Microsoft.Performance",
35:              "CA1811:AvoidUncalledPrivateCode")]
36:          internal Resources()
37:          {
38:
39:          }
40:
41:          [global::System.ComponentModel.EditorBrowsableAttribute(global::System.ComponentModel.EditorBrowsableState.Advanced)]
42:          internal static global::System.Resources.ResourceManager ResourceManager
43:          {
44:              get
45:              {
46:                  if ((resourceMan == null))
47:                  {
48:                      global::System.Resources.ResourceManager temp = new
49:                      global::System.Resources.ResourceManager("CSharpHelperMethods.Properties.Resources",
50:                          typeof(Resources).Assembly);
51:                      resourceMan = temp;
52:                  }
53:                  return resourceMan;
54:              }
55:          }
56:
57:          /// <summary>
58:          ///     Overrides the current thread's CurrentUICulture property for all
59:          ///     resource lookups using this strongly typed resource class.
60:          /// </summary>
61:
62:          [global::System.ComponentModel.EditorBrowsableAttribute(global::System.ComponentModel.EditorBrowsableState.Advanced)]
63:          internal static global::System.Globalization.CultureInfo Culture
64:          {
65:              get
66:              {
67:                  return resourceCulture;
68:              }
69:              set
70:              {
71:
72:              }
73:          }
74:      }
75:  }

```

```

67:         resourceCulture = value;
68:     }
69: }
70: }
71: }

```

Namespaces

Name	Description
CSharpHelperMethods (see page 1)	This is namespace CSharpHelperMethods.

1.2.12 Sayislemeleri.cs

This is file Sayislemeleri.cs.

Body Source

```

1: ?using System;
2: using System.Globalization;
3: using System.Linq;
4: using System.Text;
5:
6: namespace CSharpHelperMethods.YardimciSiniflar
7: {
8:     /// <summary>
9:     /// Sayisal veri kontrol metotlarini barindiran sinif
10:    /// </summary>
11:    public static class SayiIslemeleri
12:    {
13:        /// <summary>
14:        /// Girilen bir metnin sayi olup olmadigini kontrol eden metottur.
15:        /// </summary>
16:        /// <param name="text">Girdi metni</param>
17:        /// <returns>Girilen Metnin Sayisal Olup Olmadigi Bilgisi</returns>
18:        public static bool SayisalMi(string text) => text.All(char.IsNumber);
19:
20:        /// <summary>
21:        /// Ekrandan alınan string degerde virgöl olduğundan ajax metodu patladigi için
22:        /// Burada string olarak tutar verisi aliniyor
23:        /// Ve decimala çevriliyor
24:        /// </summary>
25:        /// <param name="incomingValue">Girilen String Deger</param>
26:        /// <returns>Decimal Deger</returns>
27:        public static decimal SayiHaliCevir(string metin)
28:        {
29:            decimal val;
30:            if (metin.Contains("_") || metin.Contains(".") || metin.Contains(","))
31:            {
32:                if (!decimal.TryParse(metin.Replace(",", "").Replace(".", "",
33:                " ").Replace("_", " "), NumberStyles.Number, CultureInfo.InvariantCulture, out val))
34:                    return 0;
35:                return val / 100;
36:            }
37:            else
38:            {
39:                var value = decimal.TryParse(metin.Replace(",", "").Replace(".", "",
40:                " ").Replace("_", " "), NumberStyles.Number, CultureInfo.InvariantCulture, out val);
41:                if (value)
42:                {
43:                    return val;
44:                }
45:                else
46:                {
47:                    return 0;
48:                }
49:            }
50:        }
51:    }
52: }

```

```

47:         }
48:     }
49:
50:     /// <summary>
51:     /// Elimizdeki tutar verilerinin
52:     /// Yazı formatında gösterilmesini sağlayacak bir metottur
53:     /// </summary>
54:     /// <param name="tutar">Girilen Sayı Bilgisi</param>
55:     /// <returns>Sayının Yazıya Dönüştürülmüş Hali</returns>
56:     public static string SayıYazıyaCevir(decimal tutar)
57:     {
58:         var sTutar = tutar.ToString("F2").Replace('.', ','); // Replace('.', ',')
ondalik ayracının . olma durumu için
59:         var lira = sTutar.Substring(0, sTutar.IndexOf(',')); //tutarın tam kısmı
60:         var kurus = sTutar.Substring(sTutar.IndexOf(',') + 1, 2);
61:         var yazi = "";
62:
63:         string[] birler = { "", "BİR", "İKİ", "ÜÇ", "DÖRT", "BES", "ALTI", "YEDİ",
"SEKİZ", "DOKUZ" };
64:         string[] onlar = { "", "ON", "YIRMI", "OTUZ", "KIRK", "ELLİ", "ALTMIS",
"YETMİS", "SEKSEN", "DOKSAN" };
65:         string[] binler = { "KATRİLYON", "TRİLYON", "MİLYAR", "MİLYON", "BİN", ""
}; //KATRİLYON'un önüne ekleme yapılarak artırabilir.
66:
67:         var grupSayisi = 6;
68:         //sayıdaki 3'lü grup sayısı. katrilyon içi 6. (1.234,00 daki grup sayısı
2'dir.)
69:         //KATRİLYON'un basına ekleyeceğiniz her değer için grup sayisini artırırsınız.
70:
71:         lira = lira.PadLeft(grupSayisi * 3, '0'); //sayının soluna '0' eklenerek
sayı 'grup sayısı x 3' basakmakli yapıyor.
72:
73:         for (int i = 0; i < grupSayisi * 3; i += 3) //sayı 3'erli gruplar halinde
ele alınıyor.
74:         {
75:             var grupDegeri = "";
76:
77:             if (lira.Substring(i, 1) != "0")
78:                 grupDegeri += birler[Convert.ToInt32(lira.Substring(i, 1))] +
"YÜZ"; //yüzler
79:
80:             if (grupDegeri == "BİR YÜZ") //biryüz düzeltiliyor.
81:                 grupDegeri = "YÜZ";
82:
83:             grupDegeri += onlar[Convert.ToInt32(lira.Substring(i + 1, 1))]; //onlar
84:
85:             grupDegeri += birler[Convert.ToInt32(lira.Substring(i + 2, 1))];
//birler
86:
87:             if (grupDegeri != "") //binler
88:                 grupDegeri += binler[i / 3];
89:
90:             if (grupDegeri == "BİR BİN") //birbin düzeltiliyor.
91:                 grupDegeri = "BİN";
92:
93:             yazi += grupDegeri;
94:         }
95:
96:         if (yazi != "")
97:             yazi += " TL ";
98:
99:         int yaziUzunlugu = yazi.Length;
100:
101:         if (kurus.Substring(0, 1) != "0") //kurus onlar
102:             yazi += onlar[Convert.ToInt32(kurus.Substring(0, 1))];
103:
104:         if (kurus.Substring(1, 1) != "0") //kurus birler
105:             yazi += birler[Convert.ToInt32(kurus.Substring(1, 1))];
106:

```

```

107:         if (yazi.Length > yaziUzunlugu)
108:             yazi += " Kr.";
109:         else
110:             yazi += "SIFIR Kr.";
111:
112:         return yazi;
113:     }
114:
115:     /// <summary>
116:     /// Sadece rakamlardan olusan 6 haneli random sayi üreten metottur.
117:     /// </summary>
118:     /// <returns>6 Haneli Onay Kodu</returns>
119:     public static string AltiHaneliOnayKoduOlustur()
120:     {
121:         var rastgele = new Random();
122:         var sb = new StringBuilder();
123:         for (var i = 0; i < 6; i++)
124:         {
125:             var ascii = rastgele.Next(48, 57); //Rakamlar
126:             var karakter = Convert.ToChar(ascii);
127:             sb.Append(karakter);
128:         }
129:         return sb.ToString();
130:     }
131: }
132: }

```

Namespaces

Name	Description
CSharpHelperMethods (see page 1)	This is namespace CSharpHelperMethods.

1.2.13 Settings.Designer.cs

This code was generated by a tool. Runtime Version:4.0.30319.42000

Changes to this file may cause incorrect behavior and will be lost if the code is regenerated.

Body Source

```

1:  ?//-----
2:  // <auto-generated>
3:  //     This code was generated by a tool.
4:  //     Runtime Version:4.0.30319.42000
5:  //
6:  //     Changes to this file may cause incorrect behavior and will be lost if
7:  //     the code is regenerated.
8:  // </auto-generated>
9:  //-----
10:
11: namespace CSharpHelperMethods.Properties
12: {
13:
14:
15:     [global::System.Runtime.CompilerServices.CompilerGeneratedAttribute()]
16:
17:     [global::System.CodeDom.Compiler.GeneratedCodeAttribute("Microsoft.VisualStudio.Editors.SettingsDesigner.SettingsSingleFileGenerator",
18:     "11.0.0.0")]
19:     internal sealed partial class Settings :
20:     global::System.Configuration.ApplicationSettingsBase
21:     {
22:
23:         private static Settings defaultInstance =
24:         ((Settings)(global::System.Configuration.ApplicationSettingsBase.Synchronized(new

```

```

Settings())));
21:
22:         public static Settings Default
23:         {
24:             get
25:             {
26:                 return defaultInstance;
27:             }
28:         }
29:     }
30: }

```

Namespaces

Name	Description
CSharpHelperMethods (see page 1)	This is namespace CSharpHelperMethods.

1.2.14 Sifrelemelslemleri.cs

This is file Sifrelemelslemleri.cs.

Body Source

```

1: ?using System;
2: using System.Collections.Generic;
3: using System.Security.Cryptography;
4: using System.Text;
5: using System.Text.RegularExpressions;
6:
7: namespace CSharpHelperMethods.YardimciSiniflar
8: {
9:     /// <summary>
10:    /// Sifre ile ilgili islemler sinifi
11:    /// </summary>
12:    public static class SifrelemeIslemleri
13:    {
14:        /// <summary>
15:        /// Kullanici sifresinin SHA yöntemiyle sifrelenmesini saglayan metottur.
16:        /// </summary>
17:        /// <param name="text">Sifrelenecek Metin</param>
18:        /// <returns>Girdi Metninin Sifrelenmis Hali</returns>
19:        public static string HesaplaSHA512(string text)
20:        {
21:            SHA512Managed sha512 = new SHA512Managed();
22:            Byte[] encryptedSha512 = sha512.ComputeHash(Encoding.UTF8.GetBytes(text));
23:            string strHex = "";
24:            foreach (byte b in encryptedSha512)
25:            {
26:                strHex += String.Format("{0:x2}", b);
27:            }
28:            return strHex.ToUpper();
29:        }
30:
31:        /// <summary>
32:        /// Kullanici sifresinin SHA yöntemiyle sifrelenmesini saglayan metottur.
33:        /// </summary>
34:        /// <param name="text">Sifrelenecek Metin</param>
35:        /// <returns>Girdi Metninin Sifrelenmis Hali</returns>
36:        public static string HesaplaSHA256(string text)
37:        {
38:            SHA256 sha256 = SHA256Managed.Create();
39:            byte[] bytes = Encoding.UTF8.GetBytes(text);
40:            byte[] hash = sha256.ComputeHash(bytes);
41:            StringBuilder result = new StringBuilder();
42:            for (int i = 0; i < hash.Length; i++)

```

```

43:         {
44:             result.Append(hash[i].ToString("X2"));
45:         }
46:         return result.ToString();
47:     }
48:
49:     /// <summary>
50:     /// Kullanici sifresinin MD5 yöntemiyle sifrelenmesini saglayan metottur.
51:     /// </summary>
52:     /// <param name="text">Sifrelenecek Metin</param>
53:     /// <returns>Girdi Metninin Sifrelenmis Hali</returns>
54:     public static string HesaplaMD5(string text)
55:     {
56:         MD5 md5 = MD5.Create();
57:         byte[] inputBytes = Encoding.ASCII.GetBytes(text);
58:         byte[] hash = md5.ComputeHash(inputBytes);
59:
60:         StringBuilder sb = new StringBuilder();
61:         foreach (byte t in hash)
62:         {
63:             sb.Append(t.ToString("X2"));
64:         }
65:         return sb.ToString();
66:     }
67:
68:     /// <summary>
69:     /// sifreyi kriterlere göre kontrol edip
70:     /// uymayan kosullarini string listesine geri dönüş yapan methoddur.
71:     /// Uygun olmasi durumunda null döndürecektir.
72:     /// </summary>
73:     /// <param name="text">Kullanici Sifresi</param>
74:     /// <returns>Sifrenin Uygun Olup Olmadigi Bilgisi (Açiklamasi ile
beraber)</returns>
75:     public static string SifreUygunMu(string text)
76:     {
77:         //bu kisimda sifrenin olmasi gereken kosullari degerlendirilecek.
78:         List<string> mesajlar = new List<string>();
79:
80:         if (text.Length < 8)
81:             mesajlar.Add("Parola en az 8 karakter olmalıdır.");
82:
83:         if (!Regex.IsMatch(text, @"^(?=.*[a-z])", RegexOptions.ECMAScript))
84:             mesajlar.Add("Parola en az bir küçük harf içermelidir.");
85:
86:         if (!Regex.IsMatch(text, @"^(?=.*[A-Z])", RegexOptions.ECMAScript))
87:             mesajlar.Add("Parola en az bir büyük harf içermelidir.");
88:
89:         if (!Regex.IsMatch(text, @"^(?=.*[!@#,$%&*,?_~,-,£,(,)]",
RegexOptions.ECMAScript))
90:             mesajlar.Add("Parola en az bir noktalama isareti içermelidir.");
91:
92:         if (!Regex.IsMatch(text, @"^(?=.*[0-9])", RegexOptions.ECMAScript))
93:             mesajlar.Add("Parola en az bir rakam içermelidir.");
94:
95:         if (mesajlar.Count < 1)
96:             return "Uygun";
97:         else
98:         {
99:             var sonuc = "";
100:             foreach (var item in mesajlar)
101:                 sonuc += item + ", ";
102:             sonuc = sonuc.Substring(0, sonuc.Length - 2);
103:             return sonuc;
104:         }
105:     }
106:
107:     /// <summary>
108:     /// 8 karakter uzunlugunda Rastgele sifre olusturur.
109:     /// </summary>

```

```

110:         /// <returns></returns>
111:         public static string RastgeleSifreUret()
112:         {
113:             int sifreUzunluk = 8;
114:             string gecerliKarakterler = "abcdefghijklmnopqrstuvwxyz1234567890";
115:             StringBuilder strB = new StringBuilder(100);
116:             Random random = new Random();
117:             while (0 < sifreUzunluk--)
118:             {
119:                 strB.Append(gecerliKarakterler[random.Next(gecerliKarakterler.Length)]);
120:             }
121:             return strB.ToString();
122:         }
123:     }
124: }

```

Namespaces

Name	Description
CSharpHelperMethods (see page 1)	This is namespace CSharpHelperMethods.

1.2.15 Siralamalslemleri.cs

This is file Siralamalslemleri.cs.

Body Source

```

1: ?using System;
2: using System.Globalization;
3: using System.Text;
4:
5: namespace CSharpHelperMethods.YardimciSiniflar
6: {
7:     /// <summary>
8:     /// Alfa numerik siralama için hazirlanmis bir metod
9:     /// </summary>
10:    public static class SiralamaIslemleri
11:    {
12:        private enum ChunkType { Alphanumeric, Numeric };
13:
14:        private static bool InChunk(char ch, char otherCh)
15:        {
16:            ChunkType type = ChunkType.Alphanumeric;
17:
18:            if (char.IsDigit(otherCh))
19:            {
20:                type = ChunkType.Numeric;
21:            }
22:
23:            if ((type == ChunkType.Alphanumeric && char.IsDigit(ch))
24:                || (type == ChunkType.Numeric && !char.IsDigit(ch)))
25:            {
26:                return false;
27:            }
28:
29:            return true;
30:        }
31:
32:        public static int Compare(string x, string y)
33:        {
34:            String s1 = x;
35:            String s2 = y;
36:            if (string.IsNullOrEmpty(s1) || string.IsNullOrEmpty(s2))
37:            {

```

```

38:         return 0;
39:     }
40:
41:     int thisMarker = 0;
42:     int thatMarker = 0;
43:
44:     while ((thisMarker < s1.Length) || (thatMarker < s2.Length))
45:     {
46:         if (thisMarker >= s1.Length)
47:         {
48:             return -1;
49:         }
50:         else if (thatMarker >= s2.Length)
51:         {
52:             return 1;
53:         }
54:         char thisCh = s1[thisMarker];
55:         char thatCh = s2[thatMarker];
56:
57:         StringBuilder thisChunk = new StringBuilder();
58:         StringBuilder thatChunk = new StringBuilder();
59:
60:         while ((thisMarker < s1.Length) && (thisChunk.Length == 0 ||
InChunk(thisCh, thisChunk[0])))
61:         {
62:             thisChunk.Append(thisCh);
63:             thisMarker++;
64:
65:             if (thisMarker < s1.Length)
66:             {
67:                 thisCh = s1[thisMarker];
68:             }
69:         }
70:
71:         while ((thatMarker < s2.Length) && (thatChunk.Length == 0 ||
InChunk(thatCh, thatChunk[0])))
72:         {
73:             thatChunk.Append(thatCh);
74:             thatMarker++;
75:
76:             if (thatMarker < s2.Length)
77:             {
78:                 thatCh = s2[thatMarker];
79:             }
80:         }
81:
82:         int result = 0;
83:         // If both chunks contain numeric characters, sort them numerically
84:         if (char.IsDigit(thisChunk[0]) && char.IsDigit(thatChunk[0]))
85:         {
86:             var thisNumericChunk = Convert.ToInt64(thisChunk.ToString());
87:             var thatNumericChunk = Convert.ToInt64(thatChunk.ToString());
88:
89:             if (thisNumericChunk < thatNumericChunk)
90:             {
91:                 result = -1;
92:             }
93:
94:             if (thisNumericChunk > thatNumericChunk)
95:             {
96:                 result = 1;
97:             }
98:         }
99:         else
100:         {
101:             var ci = new CultureInfo("tr-TR");
102:             result = String.Compare(thisChunk.ToString(),
thatChunk.ToString(), true,
103:                                     ci);

```



```

104:         //result = thisChunk.ToString().CompareTo(thatChunk.ToString());
105:     }
106:
107:     if (result != 0)
108:     {
109:         return result;
110:     }
111: }
112:
113: return 0;
114: }
115: }
116: }

```

Namespaces

Name	Description
CSharpHelperMethods (see page 1)	This is namespace CSharpHelperMethods.

1.2.16 TarihIslemleri.cs

This is file TarihIslemleri.cs.

Body Source

```

1: ?using System;
2:
3: namespace CSharpHelperMethods.YardimciSiniflar
4: {
5:     /// <summary>
6:     /// Tarihsel veri kontrol metotlarini barindiran sinif
7:     /// </summary>
8:     public static class TarihIslemleri
9:     {
10:         /// <summary>
11:         /// Filtre ekranlarında başlangıç ve bitis tarihi üzerinden sorgulama
12:         /// Eger aynı gün için sorgulama yapiliyorsa
13:         /// Bitis tarihini o günün 23.59.59.999'una ayarlamak için
14:         /// Hazirlanan metottur
15:         /// </summary>
16:         /// <param name="dateTime">Tarih Bilgisi</param>
17:         /// <param name="hours">Saat Bilgisi</param>
18:         /// <param name="minutes">Dakika Bilgisi</param>
19:         /// <param name="seconds">Saniye Bilgisi</param>
20:         /// <param name="milliseconds">Mili Saniye Bilgisi</param>
21:         /// <returns>Yeni Tarih Bilgisi</returns>
22:         public static DateTime ZamaniDegistir(this DateTime dateTime, int hours, int
23: minutes, int seconds, int milliseconds)
24:         {
25:             return new DateTime(
26:                 dateTime.Year,
27:                 dateTime.Month,
28:                 dateTime.Day,
29:                 hours,
30:                 minutes,
31:                 seconds,
32:                 milliseconds,
33:                 dateTime.Kind);
34:         }
35:         /// <summary>
36:         /// Yas hesaplamak için kullanılan metottur
37:         /// </summary>
38:         /// <param name="dogumTarihi">Dogum Tarihi Bilgisi</param>

```

```

39:         /// <returns>Yas Bilgisi</returns>
40:         public static int YasHesapla(DateTime dogumTarihi)
41:         {
42:             DateTime Now = DateTime.Now;
43:             int Years = new DateTime(DateTime.Now.Subtract(dogumTarihi).Ticks).Year -
1;
44:             DateTime PastYearDate = dogumTarihi.AddYears(Years);
45:             int Months = 0;
46:             for (int i = 1; i <= 12; i++)
47:             {
48:                 if (PastYearDate.AddMonths(i) == Now)
49:                 {
50:                     Months = i;
51:                     break;
52:                 }
53:                 else if (PastYearDate.AddMonths(i) >= Now)
54:                 {
55:                     Months = i - 1;
56:                     break;
57:                 }
58:             }
59:             return Years;
60:         }
61:
62:         /// <summary>
63:         /// Yas hesaplamak için kullanılan metottur
64:         /// </summary>
65:         /// <param name="dogumTarihi">Dogum Tarihi Bilgisi</param>
66:         /// <returns>Yas Bilgisi (Metinsel olarak)</returns>
67:         public static string YasHesaplaMetinsel(DateTime dogumTarihi)
68:         {
69:             DateTime Now = DateTime.Now;
70:             int Years = new DateTime(DateTime.Now.Subtract(dogumTarihi).Ticks).Year -
1;
71:             DateTime PastYearDate = dogumTarihi.AddYears(Years);
72:             int Months = 0;
73:             for (int i = 1; i <= 12; i++)
74:             {
75:                 if (PastYearDate.AddMonths(i) == Now)
76:                 {
77:                     Months = i;
78:                     break;
79:                 }
80:                 else if (PastYearDate.AddMonths(i) >= Now)
81:                 {
82:                     Months = i - 1;
83:                     break;
84:                 }
85:             }
86:             int Days = Now.Subtract(PastYearDate.AddMonths(Months)).Days;
87:             int Hours = Now.Subtract(PastYearDate).Hours;
88:             int Minutes = Now.Subtract(PastYearDate).Minutes;
89:             int Seconds = Now.Subtract(PastYearDate).Seconds;
90:             return String.Format("Yas: {0} Yil {1} Ay {2} Gün",
91: Years, Months, Days);
92:             //return String.Format("Yas: {0} Yil {1} Ay {2} Gün {3} Saat {4} Saniye",
93:             //Years, Months, Days, Hours, Seconds);
94:         }
95:
96:         /// <summary>
97:         /// İki tarih arasında kaç gün olduğunu hesaplamak için kullanılan metottur
98:         /// </summary>
99:         /// <param name="baslangicTarihi">Baslangıç Tarihi Bilgisi</param>
100:        /// <param name="bitisTarihi">Bitis Tarihi Bilgisi</param>
101:        /// <returns>Tarih Araligi Bilgisi</returns>
102:        public static string TarihAraligiHesapla(DateTime baslangicTarihi, DateTime
bitisTarihi)
103:        {
104:            int Years = new DateTime(bitisTarihi.Subtract(baslangicTarihi).Ticks).Year

```

```

- 1;
105:         DateTime PastYearDate = baslangicTarihi.AddYears(Years);
106:         int Months = 0;
107:         for (int i = 1; i <= 12; i++)
108:         {
109:             if (PastYearDate.AddMonths(i) == bitisTarihi)
110:             {
111:                 Months = i;
112:                 break;
113:             }
114:             else if (PastYearDate.AddMonths(i) >= bitisTarihi)
115:             {
116:                 Months = i - 1;
117:                 break;
118:             }
119:         }
120:         int Days = bitisTarihi.Subtract(PastYearDate.AddMonths(Months)).Days;
121:         var result = "";
122:         if (Years < 10)
123:             result = result + "0" + Years;
124:         else
125:             result = result + Years;
126:         result += " Yil, ";
127:         if (Months < 10)
128:             result = result + "0" + Months;
129:         else
130:             result = result + Months;
131:         result += " Ay, ";
132:         if (Days < 10)
133:             result = result + "0" + Days;
134:         else
135:             result = result + Days;
136:         result += " Gün";
137:         return result;
138:     }
139:
140:     /// <summary>
141:     /// Kidem Tazminatı Hesaplaması Yapılırken ayda kaç gün olursa olsun tamamında
142:     /// 30 gün üzerinden hesaplanması gerektiği için
143:     /// Isten ayrılış ve iş başlangıç tarihleri arasında kaç gün geçtiğini
144:     /// Hesaplama yapılmasını sağlayan metottur.
145:     /// </summary>
146:     /// <param name="baslangicTarihi">Başlangıç Tarihi Bilgisi</param>
147:     /// <param name="bitisTarihi">Bitiş Tarihi Bilgisi</param>
148:     /// <returns>Gün Sayısı Bilgisi</returns>
149:     public static int KidemTazminatıTarihAralığıHesapla(DateTime? baslangicTarihi,
150:         DateTime? bitisTarihi)
151:     {
152:         var artacakMi = true;
153:         var result = 0;
154:         var baslangicMi = false;
155:         var bitisMi = false;
156:         var aylıkGünSayısı = 0;
157:         for (DateTime date = (DateTime)baslangicTarihi; date <=
158:             (DateTime)bitisTarihi; date = date.AddDays(1))
159:         {
160:             if (!artacakMi)
161:                 result++;
162:             else
163:                 artacakMi = false;
164:
165:             if (date.Day == 1)
166:                 baslangicMi = true;
167:             else if (date.Day == DateTime.DaysInMonth(date.Year, date.Month))
168:                 bitisMi = true;
169:
170:             if (baslangicMi)

```

```

169:         aylikGunSayisi++;
170:         if (baslangicMi && bitisMi && aylikGunSayisi != 30)
171:         {
172:             result = result - aylikGunSayisi + 30;
173:             baslangicMi = false;
174:             bitisMi = false;
175:             aylikGunSayisi = 0;
176:         }
177:         else if (bitisMi)
178:         {
179:             baslangicMi = false;
180:             bitisMi = false;
181:             aylikGunSayisi = 0;
182:         }
183:     }
184: }
185: if (baslangicTarihi?.Day > bitisTarihi?.Day)
186:     result--;
187: return result;
188: }
189:
190: /// <summary>
191: /// Girilen iki tarih arasinda kaç ay oldugunu bulan metottur
192: /// </summary>
193: /// <param name="lValue">Ilk tarih bilgisi</param>
194: /// <param name="rValue">Son tarih bilgisi</param>
195: /// <returns>Ay Farki Bilgisi</returns>
196: public static int AyFarki(DateTime lValue, DateTime rValue) =>
197:     Math.Abs((lValue.Month - rValue.Month) + 12 * (lValue.Year - rValue.Year));
198: }

```

Namespaces

Name	Description
CSharpHelperMethods (see page 1)	This is namespace CSharpHelperMethods.

Index

A

AssemblyInfo.cs 31

C

CSharpHelperMethods 1

CSharpHelperMethods namespace 1

Classes 29

CSharpHelperMethods.csproj 32

CSharpHelperMethods.FrmAnaSayfa 29

CSharpHelperMethods.FrmAnaSayfa.Dispose 30

CSharpHelperMethods.FrmAnaSayfa.FrmAnaSayfa 29

CSharpHelperMethods.Program 30

CSharpHelperMethods.Properties 1

CSharpHelperMethods.Properties namespace 1

Classes 1

CSharpHelperMethods.Properties.Resources 1

CSharpHelperMethods.Properties.Resources.Culture 2

CSharpHelperMethods.Properties.Resources.ResourceManager 2

CSharpHelperMethods.Properties.Resources.Resources 2

CSharpHelperMethods.Properties.Settings 3

CSharpHelperMethods.Properties.Settings.Default 3

CSharpHelperMethods.sln 32

CSharpHelperMethods.YardimciSiniflar 3

CSharpHelperMethods.YardimciSiniflar namespace 3

Classes 4

CSharpHelperMethods.YardimciSiniflar.DataTableIslemleri 4

CSharpHelperMethods.YardimciSiniflar.DataTableIslemleri.BaseType 4

CSharpHelperMethods.YardimciSiniflar.DataTableIslemleri.ToTable 5

CSharpHelperMethods.YardimciSiniflar.DosyaIslemleri 6

CSharpHelperMethods.YardimciSiniflar.DosyaIslemleri.DosyaUzantisiUygunMu 7

CSharpHelperMethods.YardimciSiniflar.DosyaIslemleri.GetirDosyaUzantisiMimeType 7

CSharpHelperMethods.YardimciSiniflar.DosyaIslemleri.GetirLerikTipiDosyaUzantisille 8

CSharpHelperMethods.YardimciSiniflar.KisiIslemleri 9

CSharpHelperMethods.YardimciSiniflar.KisiIslemleri.DogrulaEPosta 9

CSharpHelperMethods.YardimciSiniflar.KisiIslemleri.DogrulaBAN 10

CSharpHelperMethods.YardimciSiniflar.KisiIslemleri.DogrulaTcKimlikNo 10

CSharpHelperMethods.YardimciSiniflar.MetinIslemleri 11

CSharpHelperMethods.YardimciSiniflar.MetinIslemleri.BoslukKaldir 12

CSharpHelperMethods.YardimciSiniflar.MetinIslemleri.HTMLDuzelt 12

CSharpHelperMethods.YardimciSiniflar.MetinIslemleri.IdealMetinGetir 13

CSharpHelperMethods.YardimciSiniflar.MetinIslemleri.MetinIlkKarakterBuyukDigerleriKucukYap 14

CSharpHelperMethods.YardimciSiniflar.MetinIslemleri.MetinIlkKarakterleriBuyukYap 14

CSharpHelperMethods.YardimciSiniflar.MetinIslemleri.MetinSifrele 14

CSharpHelperMethods.YardimciSiniflar.MetinIslemleri.MetinZararliKarakterDuzelt 15

CSharpHelperMethods.YardimciSiniflar.MetinIslemleri.TelefonNoDuzenle 15

CSharpHelperMethods.YardimciSiniflar.MetinIslemleri.TurkceKarakterleriDuzelt 15

CSharpHelperMethods.YardimciSiniflar.MetinIslemleri.UTF8Duzelt 16

CSharpHelperMethods.YardimciSiniflar.SayisIslemleri 16

CSharpHelperMethods.YardimciSiniflar.SayisIslemleri.AltiHaneliOnayKoduOlustur 17

CSharpHelperMethods.YardimciSiniflar.SayisIslemleri.SayiHaliNeCevir 17

CSharpHelperMethods.YardimciSiniflar.SayisIslemleri.SayisalM

i	Dosyalslemleri class 6
18	about Dosyalslemleri class 6
CSharpHelperMethods.YardimciSiniflar.Sayilslemleri.SayiyiYaziyaCevir	DosyaUzantisiUygunMu 7
18	GetirDosyaUzantisiMimeTypeelle 7
CSharpHelperMethods.YardimciSiniflar.Sifrelemelslemleri 19	GetirIcerikTipiDosyaUzantisille 8
CSharpHelperMethods.YardimciSiniflar.Sifrelemelslemleri.HesaplaMD5	Dosyalslemleri.cs 34
20	
CSharpHelperMethods.YardimciSiniflar.Sifrelemelslemleri.HesaplaSHA256	
20	
CSharpHelperMethods.YardimciSiniflar.Sifrelemelslemleri.HesaplaSHA512	
21	
CSharpHelperMethods.YardimciSiniflar.Sifrelemelslemleri.RastgeleSifreUret	
21	
CSharpHelperMethods.YardimciSiniflar.Sifrelemelslemleri.SifreUygunMu	
22	
CSharpHelperMethods.YardimciSiniflar.Siralamalslemleri 22	
CSharpHelperMethods.YardimciSiniflar.Siralamalslemleri.Compare	
23	
CSharpHelperMethods.YardimciSiniflar.TarihIslemleri 24	
CSharpHelperMethods.YardimciSiniflar.TarihIslemleri.AyFarki	
25	
CSharpHelperMethods.YardimciSiniflar.TarihIslemleri.KidemTazminatiTarihAraligiHesapla	
25	
CSharpHelperMethods.YardimciSiniflar.TarihIslemleri.TarihAraligiHesapla	
26	
CSharpHelperMethods.YardimciSiniflar.TarihIslemleri.YasHesapla	
27	
CSharpHelperMethods.YardimciSiniflar.TarihIslemleri.YasHesaplaMetinsel	
27	
CSharpHelperMethods.YardimciSiniflar.TarihIslemleri.ZamaniDegistir	
28	
D	
DataTableIslemleri class 4	
about DataTableIslemleri class 4	
BaseType 4	
ToDataTable 5	
DataTableIslemleri.cs 32	
	F
	Files 30
	FrmAnaSayfa class 29
	about FrmAnaSayfa class 29
	Dispose 30
	FrmAnaSayfa 29
	FrmAnaSayfa.cs 36
	FrmAnaSayfa.Designer.cs 44
	K
	Kisilslemleri class 9
	about Kisilslemleri class 9
	DogrulaEPosta 9
	DogrulaIBAN 10
	DogrulaTcKimlikNo 10
	Kisilslemleri.cs 56
	M
	MetinIslemleri class 11
	about MetinIslemleri class 11
	BoslukKaldir 12
	HTMLDuzelt 12
	IdealMetinGetir 13
	MetinIlkKarakterBuyukDigerleriKucukYap 14
	MetinIlkKarakterleriBuyukYap 14
	MetinSifrele 14
	MetinZararliKarakterDuzelt 15
	TelefonNoDuzenle 15
	TurkceKarakterleriDuzelt 15
	UTF8Duzelt 16
	MetinIslemleri.cs 57

P

Program class 30
 about Program class 30
Program.cs 61

R

Resources class 1
 about Resources class 1
 Culture 2
 ResourceManager 2
 Resources 2
Resources.Designer.cs 61

S

SayIslemleri class 16
 about SayIslemleri class 16
 AltiHaneliOnayKoduOlustur 17
 SayiHalineCevir 17
 SayisalMi 18
 SayiyiYaziyaCevir 18
SayIslemleri.cs 63
Settings class 3
 about Settings class 3
 Default 3
Settings.Designer.cs 65
Sifrelemelslemleri class 19
 about Sifrelemelslemleri class 19
 HesaplaMD5 20
 HesaplaSHA256 20
 HesaplaSHA512 21
 RastgeleSifreUret 21
 SifreUygunMu 22
Sifrelemelslemleri.cs 66
Siralamalslemleri class 22
 about Siralamalslemleri class 22
 Compare 23
Siralamalslemleri.cs 68

T

TarihIslemleri class 24
 about TarihIslemleri class 24
 AyFarki 25
 KidemTazminatiTarihAraligiHesapla 25
 TarihAraligiHesapla 26
 YasHesapla 27
 YasHesaplaMetinsel 27
 ZamaniDegistir 28
TarihIslemleri.cs 70