1. using System.IO;

// Satır1 dosya kaydetmek için IO kütüphanesinden özellik kullanmak için using System.IO; yazarak IO

// kütüphanesini çağırdık.

1. namespace final1soru
2. {
3. public partial class Form1 : Form
4. {
5. public Form1()
6. {
7. InitializeComponent();
8. }
9. public void Uyari()

// Satır 10 bir alan boş geçilirse uyarı vermesi için Uyari adında bir metot oluşturduk.

1. {
2. MessageBox.Show("Bu alanı boş geçemezsiniz.");

// Satır 12 bir alanı boş geçtiğimizde Uyari() metodunun olduğu yerlerde MessageBox ta göstermek için

// MessageBox.Show(); içine göstermek istediğimiz mesajı yazdık.

1. }
2. private void textBox1\_KeyPress(object sender, KeyPressEventArgs e)

// Satır 14 textBox1 e fiyat yazılacağı için rakamdan başka bir şey girilmemesi istenmektedir. Bundan dolayı

// textBox1 in KeyPress özelliğini kullandık.

1. {
2. if ((Char.IsDigit(e.KeyChar) == false) && (Char.IsControl(e.KeyChar) == false))

// Satır 16 textBox1 e girilen değerlerin harf olmaması ve bunu kontrol ettirdiğimizde false (yani harf

// olmadığı) durumunu koşullandırmak için if kullandık.

1. e.KeyChar = '\0';

// Satır 17 textBox1 e ne kadar harf girilirse girilsin sürekli boş kalması için e.KeyChar ı \0 a eşitledik.

1. }
2. private void textBox1\_Enter(object sender, EventArgs e)

// Satır 19 textBox1 e gelindiğinde textBox1 in arka renginde değişiklik yapabilmek için textBox1 in Enter

// özelliğini kullandık.

1. {
2. textBox1.BackColor = Color.Cyan;

// Satır 21 textBox1 e gelindiğinde textBox1 in arka rengini değiştirmek için BackColor kullanılıyor.

// Rengini seçmek içinde Color.istenilen renge eşitlenerek arka rengi değiştirdik.

1. }
2. private void textBox1\_Leave(object sender, EventArgs e)

// Satır 23 textBox1 den ayrıldığında textBox1 in üzerinde değişiklik yapabilmek için textBox1 in Leave

// özelliğini kullandık.

1. {
2. textBox1.BackColor = Color.White;

// Satır 25 textBox1 den ayrıldığında textBox1 in arka rengini eski rengine döndürmek için BackColor

// kullanılıyor. Eski rengi beyaz olduğu için Color.White a eşitlenerek arka rengi eski haline döner.

1. if (textBox1.Text == "")

// Satır 26 textBox1 de değer yoksa koşulunu oluşturmak için if kullandık.

1. {
2. Uyari();

// Satır 28 Satır 10 da bir alan boş geçilirse uyarı vermesi için Uyari adında bir metot oluşturmuştuk.

// Satır 26 sağlanırsa eğer textBox1 boş geçilmiş olacağı için Uyari(); metodunu kullandık.

1. textBox1.Focus();

// Satır 29 textBox1 boş geçilemediğinden orayı doldurması gerekmekte. Kullanıcıyı yönlendirmek için

// Focus kullanarak textBox1 e gitmesini sağladık.

1. }
2. }
3. private void textBox2\_KeyPress(object sender, KeyPressEventArgs e)

// Satır 32 textBox2 ye ad-soyad yazılacağı için harften başka bir şey girilmemesi istenmektedir. Bundan

// dolayı textBox2 nin KeyPress özelliğini kullandık.

1. {
2. string str = "0123456789";

// Satır 34 string türünde bir değişkene rakamları eşitledik.

1. if (str.IndexOf(e.KeyChar) != -1)

// Satır 35 satır 34 de tanımladığımız değişkeni baz alarak yazdığımız karakterlerin içinde var mı yok mu

// diye koşullandırma için if kullandık.

1. e.KeyChar = '\0';

// Satır 36 textBox2 e ne kadar sayı girilirse girilsin sürekli boş kalması için e.KeyChar ı \0 a eşitledik.

1. }
2. private void textBox2\_Enter(object sender, EventArgs e)

// Satır 38 textBox2 ye gelindiğinde textBox2 nin arka renginde değişiklik yapabilmek için textBox1 in Enter

// özelliğini kullandık.

1. {
2. textBox2.BackColor = Color.HotPink;

// Satır 40 textBox2 ye gelindiğinde textBox2 nin arka rengini değiştirmek için BackColor kullanılıyor.

// Rengini seçmek içinde Color.istenilen renge eşitlenerek arka rengi değiştirdik.

1. }
2. private void textBox2\_Leave(object sender, EventArgs e)

// Satır 42 textBox2 den ayrıldığında textBox2 nin üzerinde değişiklik yapabilmek için textBox2 nin Leave

// özelliğini kullandık.

1. {
2. textBox2.BackColor = Color.White;

// Satır 44 textBox2 den ayrıldığında textBox2 nin arka rengini eski rengine döndürmek için BackColor

// kullanılıyor. Eski rengi beyaz olduğu için Color.White a eşitlenerek arka rengi eski haline döner.

1. textBox2.Text = textBox2.Text.ToUpper();

// Satır 45 textBox2 den ayrıldığında textBox2 den ayrıldığında textBox2 de ki harflerin hepsinin büyük harf // olması için ToUpper(); kullandık.

1. if (textBox2.Text == "")

// Satır 46 textBox2 de değer yoksa koşulunu oluşturmak için if kullandık.

1. {
2. Uyari();

// Satır 48 Satır 10 da bir alan boş geçilirse uyarı vermesi için Uyari adında bir metot oluşturmuştuk.

// Satır 46 sağlanırsa eğer textBox2 boş geçilmiş olacağı için Uyari(); metodunu kullandık.

1. textBox2.Focus();

// Satır 49 textBox1 boş geçilemediğinden orayı doldurması gerekmekte. Kullanıcıyı yönlendirmek için

// Focus kullanarak textBox1 e gitmesini sağladık.

1. }
2. }
3. ToolTip bilgi = new ToolTip();

// Satır 52 Araç ipucu oluşturup ve Form kapsayıcısıyla ilişkilendirdik.

1. private void comboBox1\_MouseHover(object sender, EventArgs e)

// Satır 53 **fareyle üzerine gelindiğinde bilgi kutusu çıkması için comboBoxın MouseHover özelliğini**

//**kullandık.**

1. {
2. bilgi.ToolTipIcon = ToolTipIcon.Info;

// **Satır 55 araç ipucu metniyle birlikte görüntülenecek simgenin bilgi simgesi (info) tanımladık ve değeri //aldık.**

1. bilgi.ToolTipTitle = "Bilgilendirme";

// **Satır 56 araç ipucu metninin başlığını Bilgilendirme yaptık.**

1. bilgi.Show("1- Halkbank (3), Vakıfbank (4) ve Ziraatbank (5) taksit seçildiğinde peşin fiyatına taksitlendirme yapılır.\n2 " +

"2- Diğer taksit seçeneklerinde aylık % 2 faiz uygulanır.\n3 " +

"3- Diğer kartlar ile aylık % 2 faiz uygulanır.\n4 " +

"4- En fazla 6 taksit seçilebilir.\n5 " +

"5- Nakit ve tek çekimde ürün fiyatı uygulanır", comboBox1);

// **Satır 57 araç ipucu metninde yazacak yazıyı tanımladık.**

1. }
2. private void comboBox1\_MouseLeave(object sender, EventArgs e)

// Satır 59 **fareyle üzerinden ayrıldığında bilgi kutusu kalkması için comboBoxın MouseLeave özelliğini**

//**kullandık.**

1. {
2. bilgi.Hide(comboBox1);

// Satır 61 **fareyle üzerinden ayrıldığında bilgi kutusu kaldırmak için Hide kullandık.**

1. }
2. private void comboBox1\_Enter(object sender, EventArgs e)

// Satır 63 comboBox1 e gelindiğinde comboBox1 in arka renginde değişiklik yapabilmek için

// comboBox1 in Enter özelliğini kullandık.

1. {
2. comboBox1.BackColor = Color.LightGoldenrodYellow;

// Satır 65 comboBox1 e gelindiğinde comboBox1 in arka rengini değiştirmek için BackColor

// kullanılıyor. Rengini seçmek içinde Color.istenilen renge eşitlenerek arka rengi değiştirdik.

1. }
2. private void comboBox1\_Leave(object sender, EventArgs e)

// Satır 67 comboBox1 den ayrıldığında comboBox1 in üzerinde değişiklik yapabilmek için

// comboBox1 in Leave özelliğini kullandık.

1. {
2. comboBox1.BackColor = Color.White;

// Satır 69 comboBox1 den ayrıldığında comboBox1 in arka rengini eski rengine döndürmek için

// BackColor kullanılıyor. Eski rengi beyaz olduğu için Color.White a eşitlenerek arka rengi eski haline

// döner.

1. if (comboBox1.Text == "")

// Satır 70 comboBox1 de değer yoksa koşulunu oluşturmak için if kullandık.

1. {
2. Uyari();

// Satır 72 Satır 10 da bir alan boş geçilirse uyarı vermesi için Uyari adında bir metot oluşturmuştuk.

// Satır 70 sağlanırsa eğer comboBox1 boş geçilmiş olacağı için Uyari(); metodunu kullandık.

1. comboBox1.Focus();

// Satır 73 comboBox1 boş geçilemediğinden orayı doldurması gerekmekte. Kullanıcıyı yönlendirmek

// için Focus kullanarak comboBox1 e gitmesini sağladık.

1. }
2. }
3. private void comboBox2\_Enter(object sender, EventArgs e)

// Satır 76 comboBox2 ye gelindiğinde comboBox2 nin arka renginde değişiklik yapabilmek için

// comboBox2 in Enter özelliğini kullandık.

1. {
2. comboBox2.BackColor = Color.SkyBlue;

// Satır 78 comboBox2 ye gelindiğinde comboBox2 nin arka rengini değiştirmek için BackColor

// kullanılıyor. Rengini seçmek içinde Color.istenilen renge eşitlenerek arka rengi değiştirdik.

1. }
2. private void comboBox2\_Leave(object sender, EventArgs e)

// Satır 80 comboBox2 den ayrıldığında comboBox2 nin üzerinde değişiklik yapabilmek için

// comboBox2 nin Leave özelliğini kullandık.

1. {
2. comboBox2.BackColor = Color.White;

// Satır 80 comboBox2 den ayrıldığında comboBox2 nin arka rengini eski rengine döndürmek için

// BackColor kullanılıyor. Eski rengi beyaz olduğu için Color.White a eşitlenerek arka rengi eski haline

// döner.

1. if (comboBox2.Text == "")

// Satır 70 comboBox2 de değer yoksa koşulunu oluşturmak için if kullandık.

1. {
2. Uyari();

// Satır 85 Satır 10 da bir alan boş geçilirse uyarı vermesi için Uyari adında bir metot oluşturmuştuk.

// Satır 83 sağlanırsa eğer comboBox2 boş geçilmiş olacağı için Uyari(); metodunu kullandık.

1. comboBox2.Focus();

// Satır 86 comboBox2 boş geçilemediğinden orayı doldurması gerekmekte. Kullanıcıyı yönlendirmek

// için Focus kullanarak comboBox2 ye gitmesini sağladık.

1. }
2. }
3. private void maskedTextBox1\_Enter(object sender, EventArgs e)

// Satır 93 maskedTextBox1 e gelindiğinde maskedTextBox1 in arka renginde değişiklik yapabilmek için

// maskedTextBox1 in Enter özelliğini kullandık.

1. {
2. maskedTextBox1.BackColor = Color.LightSalmon;

// Satır 95 maskedTextBox1 e gelindiğinde maskedTextBox1 in arka rengini değiştirmek için BackColor

// kullanılıyor. Rengini seçmek içinde Color.istenilen renge eşitlenerek arka rengi değiştirdik.

1. }
2. private void maskedTextBox1\_Leave(object sender, EventArgs e)

// Satır 97 maskedTextBox1 den ayrıldığında maskedTextBox1 in üzerinde değişiklik yapabilmek için

// maskedTextBox1 in Leave özelliğini kullandık.

1. {
2. maskedTextBox1.BackColor = Color.White;

// Satır 99 maskedTextBox1 den ayrıldığında maskedTextBox1 in arka rengini eski rengine döndürmek için

// BackColor kullanılıyor. Eski rengi beyaz olduğu için Color.White a eşitlenerek arka rengi eski haline

// döner.

1. if (maskedTextBox1.Text == "")

// Satır 100 maskedTextBox1 de değer yoksa koşulunu oluşturmak için if kullandık.

1. {
2. Uyari();

// Satır 102 Satır 10 da bir alan boş geçilirse uyarı vermesi için Uyari adında bir metot oluşturmuştuk.

// Satır 100 sağlanırsa eğer maskedTextBox1 boş geçilmiş olacağı için Uyari(); metodunu kullandık.

1. maskedTextBox1.Focus();

// Satır 103 maskedTextBox1 boş geçilemediğinden orayı doldurması gerekmekte. Kullanıcıyı yönlendirmek

// için Focus kullanarak maskedTextBox1 e gitmesini sağladık.

1. }
2. }
3. private void Form1\_Load\_1(object sender, EventArgs e)

// Satır 102 form yüklendiğinde üzerinde gözükmesini istediğimiz yazıları tanımlıyoruz.

1. {
2. comboBox1.Items.Add("Nakit");
3. comboBox1.Items.Add("Halkbank");
4. comboBox1.Items.Add("Vakıfbank");
5. comboBox1.Items.Add("Ziraat Bankası");
6. comboBox1.Items.Add("Diğer kart seçenekleri");

// Satır 104-105-106-107-108 form yüklendiğinde comboBox1 in seçeneklerinde görünmesini istediğimiz //yazıları ekliyoruz.

1. comboBox2.Items.Add("Tek çekim");
2. comboBox2.Items.Add("2");
3. comboBox2.Items.Add("3");
4. comboBox2.Items.Add("4");
5. comboBox2.Items.Add("5");
6. comboBox2.Items.Add("6");

// Satır 109-110-111-112-113-113 form yüklendiğinde comboBox2 nin seçeneklerinde görünmesini istediğimiz //yazıları ekliyoruz.

1. timer1.Start();

// Satır 115 form yüklendiğinde zamanın başlatılıyor.

1. }
2. private string mesaj()

// Satır 117 Goster buttonuna basıldığında çıkacak mesaj ve hesaplanacak de taksitlendirme için bir mesaj()

//metodu oluşturduk.

1. {
2. string mesaj;

// Satır 119 yazdıracağımız mesajı atamak için string tütünde bir değişken oluşturduk.

1. if (string.Compare(textBox1.Text, "") > 0 &&string.Compare(textBox2.Text, "") > 0 &&comboBox1.SelectedIndex > -1 &&comboBox2.SelectedIndex > -1 &&string.Compare(maskedTextBox1.Text, "") > 0)

// Satır 120 textBox1 ve boşluğu compare ile karşılaştırıp 0 dan büyük ise yani karakterler varsa koşulu için //if kullandık. Bunu diğer boşluk değer girme yerlei içinde ifin içine yazdık.

1. {
2. string banka = comboBox1.Text;

// Satır 122 comboBox1 in içerisinde bankalarımız vardı. Onları çekmek ve karşılaştırmak için string

//türünde bir değişkene atadık.

1. string taksitSecim = comboBox2.Text;

// Satır 123 comboBox2 nin içerisinde taksit seçeneklerimiz vardı. Onları çekmek ve karşılaştırmak için

//string türünde bir değişkene atadık.

1. float faiz = 0;

// Satır 124 bazı taksit seçeneklerinde fiyatın faizlenmesi istenmektedir. Bunun için faizi alabilmek için float //türünde bir değişken oluşturduk.

1. int taksit = 0;

// Satır 125 taksitlendirme işlemi için taksit sayısını eşitlemek için int türünde bir değişken oluşturduk.

1. switch (taksitSecim)

// Satır 126 taksit seçimine bakmak için switch kullandık.

1. {
2. case "Tek Çekim":

// Satır 128 seçim Tek çekim olduğu zamanki durumları göstermek için case yazdık.

1. taksit = 1;

// Satır 129 taksit seçimi tek çekim olduğundan taksit 1 olmalıdır. Faiz uygulanmamaktadır.

1. break;

// Satır 130 seçim tek çekim değilse seçimden çıkılır.

1. default:

// Satır 131 satır 128 sağlanmama durumu için default kullandık.

1. taksit = int.Parse(taksitSecim);

// Satır 132 seçilen taksiti satır 125 te tanımladığımız değişkene eşitledik.

1. break;

// Satır 133 seçim tek çekim ve dışındaki değerler yoksa seçimlerden çıkılır.

1. }
2. switch (banka)

// Satır 135 banka seçimine bakmak için switch kullandık.

1. {
2. case "Halkbank":

// Satır 137 seçim halkbank olduğu zamanki durumları göstermek için case yazdık.

1. if (taksit > 3) faiz = 0.02f;

// Satır 138 halkbankta 3. taksitten sonrası için faiz uygulanması koşulunu if kullanarak yazdık ve //uygulanacak faiz değerini değişkene atadık.

1. break;

// Satır 139 seçim halkbank değilse seçimden çıkılır.

1. case "Vakıfbank":

// Satır 140 seçim vakıfbank olduğu zamanki durumları göstermek için case yazdık.

1. if (taksit > 4) faiz = 0.02f;

// Satır 141 vakıfbankta 4. taksitten sonrası için faiz uygulanması koşulunu if kullanarak yazdık ve //uygulanacak faiz değerini değişkene atadık.

1. break;

// Satır 142 seçim vakıfbank değilse seçimden çıkılır.

1. case "Ziraat Bankası":

// Satır 143 seçim ziraat bankası olduğu zamanki durumları göstermek için case yazdık.

1. if (taksit > 5) faiz = 0.02f;

// Satır 144 vakıfbankta 5. taksitten sonrası için faiz uygulanması koşulunu if kullanarak yazdık ve //uygulanacak faiz değerini değişkene atadık.

1. break;

// Satır 145 seçim ziraat bankası değilse seçimden çıkılır.

1. case "Nakit":

// Satır 146 seçim nakit olduğu zamanki durumları göstermek için case yazdık.

1. taksit = 1;

// Satır 147 ödeme şekli nakit olduğundan taksit 1 olmalıdır. Faiz uygulanmamaktadır.

1. faiz = 0f;

// Satır 148 Faiz uygulanmayacağı için 0f eşitledik.

1. break;

// Satır 149 seçim nakit değilse seçimden çıkılır.

1. case “Diğer kart seçenekleri”:

// Satır 150 seçim diğer kart seçenekleri olduğu zamanki durumları göstermek için case yazdık.

1. faiz = 0.02f;

// Satır 151 diğer kart seçenekleri için her taksitlendirmede faizlendirme yapılacak.

1. break;

// Satır 152 seçim diğer kart seçenekleri değilse seçimden çıkılır.

1. }
2. float tutar = float.Parse(textBox1.Text);

// Satır 154 textBox1 in textine girilen fiyatı çekmek için float türünde bir değişkene eşitledik.

1. float toplamTutar = tutar + tutar \* faiz \* taksit;

// Satır 155 de tutar için üzerine uygulanacak faizlendirme işlemini tanımladık.

1. float aylikTaksit = toplamTutar / taksit;

// Satır 156 satır 155 de faizlendirme yapılan paranın aylık kaç tlye denk geldiğini bulup float türünde

//değişkene atadık.

1. mesaj = "Seçiminiz\t\t: " + banka +"\nTaksit Sayısı\t\t:" + taksit +"\n" + toplamTutar + " borcunuzu aylık " + aylikTaksit + " olarak ödeyebilirsiniz." +"\n Tarih:\t\t" + dateTimePicker1.Text +"\n Saat:\t\t" + label6.Text;

// Satır 157 mesajda neler gösterilmesini istiyorsak mesaj değişkenine eşitledik.

1. }
2. else

// Satır 159 satır 120 de tanımladığımız koşul sağlanmıyorsa için else kullandık.

1. {
2. mesaj = "Tüm bilgileri giriniz!";

// Satır 161 satır 120 deki koşul sağlanmadığında eksik bilgiler vardır. Onu anımsatmak için mesaja tüm

//bilgileri girinize eşitledik.

1. }
2. return mesaj;

// Satır 163 burda yukarıdaki koşullanrdan sonra oluşan mesajı tuttuk.

1. }
2. private void button1\_Click(object sender, EventArgs e)

// Satır 165 button1 Goster butonudur. Basıldığında hesaplanan değerleri göstermesi için buttonun click

//özelliğini kullandık.

1. {
2. DialogResult sonuc = MessageBox.Show(mesaj(), "", MessageBoxButtons.OK);

// Satır 167 DialogResult dönüş değeri bir iletişim kutusunda göstermek için tanımlayıcıları belirler. sonuc //adında bir nesne oluşturup bunu MessageBox.Showa eşitleyerek messagebox ta mesaj metodunda //yazılan işlemleri gösterdik.

1. }
2. Font fontum;

// Satır 169 Font ailesinden fontum adında bir değişken oluşturduk.

1. SolidBrush renk;

// Satır 170 SolidBrush ailesinden renk adında bir değişken oluşturduk.

1. private void button2\_Click(object sender, EventArgs e)

// Satır 171 button2 PRINT butonudur. Basıldığında yazdırılma işlemlerini, font seçimini verenk seçimini yapabilmek için button click özelliğini kullandık.

1. {
2. fontDialog1.ShowDialog();

// Satır 173 font seçimi için bir kutu açılır.

1. fontum = new Font(fontDialog1.Font.Name, fontDialog1.Font.Size, fontDialog1.Font.Style);

// Satır 174 satır 169 da tanımladığım fontum değişkenine yaptığım seçimleri eşitliyorum.

1. colorDialog1.ShowDialog();

// Satır 175 renk seçimi için bir kutu açılır.

1. renk = new SolidBrush(colorDialog1.Color);

// Satır 176 satır 170 de tanımladığım değişkene seçtiğim renge eşitledim.

1. printDialog1.ShowDialog();

// Satır 177 yazdırma seçenekleri için bir kutu açılır.

1. printDocument1.Print();

// Satır 178 print özleliği ile doküman yazdırılır. printDocument1\_PrintPage metodunu tetikleyecek

1. }
2. private void printDocument1\_PrintPage(object sender, System.Drawing.Printing.PrintPageEventArgs e)

// Satır 180 yazdırılacak doküman için sayfa işlemleri yapılmakta.

1. {
2. float leftMargin = e.MarginBounds.Left;

// Satır 182 soldan boşluk ayarlanır.

1. float topMargin = e.MarginBounds.Top;

// Satır 183 üstten boşluk ayarlanır.

1. if (fontDialog1.ShowDialog() != DialogResult.Cancel)

// Satır 184 bir formu kapattığınızda geçersiz kılınmadıkça DialogResult.Cancel olacaktır.

1. {
2. Font printFont = fontDialog1.Font;

// Satır 186 yazdırılacak fontu seçilen fontDialog1.Fonta eşitledik.

1. float yPos = topMargin + (1 \*printFont.GetHeight(e.Graphics));

// Satır 187 satır 183 te üstten boşluk üzerine yükseklik ekledik.

1. e.Graphics.DrawString(mesaj(), fontum, renk, leftMargin, yPos, new StringFormat());

// Satır 188 yazılcak yazının fontunu, rengini, bırakılacak boşlukları ve metin düzeni bilgilerini,

//düzenlemeleri ve OpenType özelliklerini e.Graphics.DrawString içine yazılır ve sayfaya yazdırılır.

1. }
2. }
3. private void button3\_Click(object sender, EventArgs e)

// Satır 191 button3 Dosyaya Yaz butonudur. Basıldığında oluşan dosyanın txt halinde kaydedilmesini

//sağlamak için buttonun Click özelliğini kullandık.

1. {
2. Stream dosya;

// Satır 193 Streami verinin bir bütün olarak değil de parça parça alınması, işlenmesi için bir dosya adında

// bir nesne oluşturduk.

1. SaveFileDialog dosyakaydet = new SaveFileDialog();

// Satır 194 SaveFileDialog ile dosya kaydet adında bir nesne oluşturduk. SaveFileDialog bir dialog ekranı

// ile dosya kaydedilir.

1. dosyakaydet.Filter = "txt files (\*.txt)|\*.txt|All files (\*.\*)|\*.\*";

// Satır 195 soruda bizden dosyanın .txt uzantılı kaydedilmesini istemektedir. Bunu yapmak içinde. Filter //özelliği kullanılarak hangi özellikte bulmak istiyorsak ona eşitlemekteyiz

1. dosyakaydet.FilterIndex = 2;

// Satır 196 dosyayı kaydederken uzantı kısmına filitere de ki 1. Uzantı getirilir. Bunu değiştirmek için

// FilterIndexi 2 ye eşitleyerek ekran açıldığında direk .txt uzantısı görülecek.

1. dosyakaydet.RestoreDirectory = true;

// Satır 197 RestoreDirectory bazen sürekli aynı dizini değil de seçilen son dizinin görüntülenmesi

// için kullanılır. True olarak setlediğimizde dialog bir önceki dosyanın seçildiği dizinden başlatılacaktır.

1. if (dosyakaydet.ShowDialog() == DialogResult.OK)

// Satır 198 dosyaKaydet için ShowDialog çağırarak satır DialogResult.OK a eşitledik.

1. {
2. if ((dosya = dosyakaydet.OpenFile()) != null)

// Satır 200 satır 193 te oluşturduğumuz dosya nesnesini dosyaKaydet.OpenFile a eşitliyoruz. dosya

// nesnesini eşitleyip dosyayı açacağız. Ve bu değeri de !null a eşitleyeceğiz. Eğer boş bir değere eşit

// olmadığında yani istediğimiz koşullar olduğunda neler yaptıracağımızı aşağıda yazdık.

1. {
2. StreamWriter a = new StreamWriter(dosya);

// Satır 202 StreamWriter sınıfı ile metin dosyalarına karakter türünde değer girişi yapılır.

1. a.Write(comboBox1.Text + comboBox2.Text);

// Satır 203 satır 2052 de oluşturulan a nesnesini yazdımak için a.Write yazarak içinde comboBox1.Text + //comboBox2.Text i görmek istediğimiz için yazıyoruz.

1. a.Close();

// Satır 207 açıldığı gibi kapanması için Close(); kullandık.

1. }
2. }
3. }
4. private void timer1\_Tick(object sender, EventArgs e)

//Satır 211 zamanı tam gösterebilmek için timerın tick özelliğini kullanıyoruz.

1. {
2. DateTime datetime = DateTime.Now;

// Satır 213 DateTime sınıfından bir nesne oluşturduk ve onun şimdi ki zamanı göstermesi için DateTime ın //Now özelliğini kullandık.

1. label6.Text = datetime.ToString("HH:mm:ss");

//Satır 394 zamanı label6 da gösterebilmek için labelın textini

//DateTime.Now.ToString("HH:mm:ss");eşitledik.

1. }
2. }
3. }











