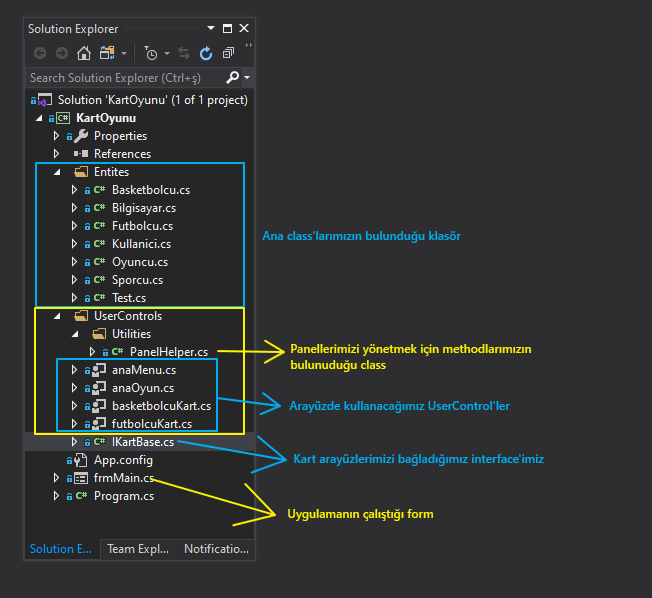
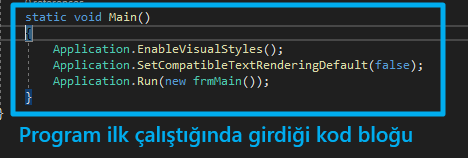
# Projenin DOSYA Yapısı



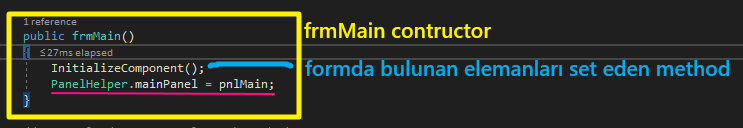
# Uygulamanın Çalışma Sırası

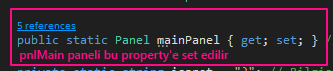
Dosya: Program.cs

Bu blockta uygulamamızın çalışacağı formumuz new edilir. Bu block VisualStudio’daki winform uygulamasının standart kod bloğudur. “Application.Run(new frmMain());” satırına girildiğinde aşağıdaki kodlar çalışır.

Dosya: frmMain.cs

Tür: Form

  
Bu blok formumuzun constructor’ıdır. Formumuz new edildiği anda bu blok çalışır. Mavi işaretli method form elemanlarını oluşturup ekrana formumuzu çıkartan metoddur. Altı işaretli kod satırına girildiğinde aşağıdaki kodlar çalışır.

Dosya: PanelHelper.cs

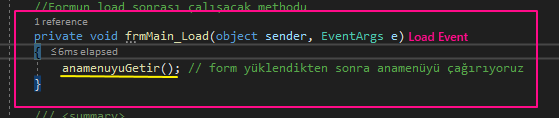
Tür: Static Class

Set edilme işlemi bittikten sonra formun “Load” event’ına girilir.

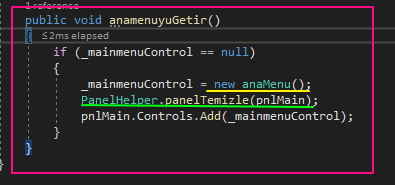
Dosya: frmMain.cs

Event: frmMain\_Load

Tür: Form



Bu event işleminde altı çizili kod satırına girildiğinde aşağıdaki kodlar çalışır.

Dosya: frmMain.cs

Method: anamenuyuGetir()

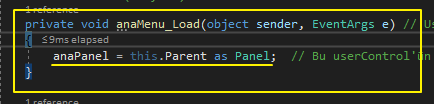
Tür: Form

Bu metoddaki if bloğu anaMenu (UserControl) tipindeki \_mainmenuControl değişkeninin boş oluduğu durumda çalışacaktır.

Sarı renkle altı çizili kod satırına gelindiğinde şu kodlar çalışır => [anaMenu](#anaMenu) (tıkla)

Yeşil renkle altı çizili kod satırına gelindiğinde şu kodlar çalışır => [panelTemizle](#panelTemizle) (tıkla)

Daha sonra pnlMain (Panel) panelinin içine \_mainmenuControl eklenir ve aşağıdaki kodlar çalışır.

Dosya: anaMenu.cs

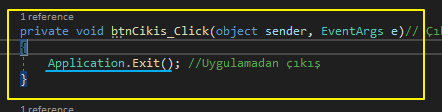
Event: anaMenu\_Load

Tür: UserControl  
Altı çizili satıra girildiğinde Panel tipindeki anaPanel değişkenine kendi içinde bulunduğu paneli atar. Çünkü içinde bulunduğu paneli daha sonra bu UserControl içinde kullanabiliriz.

Sonrasında ekranda ana menüyü görmüş oluruz.

Butonlar

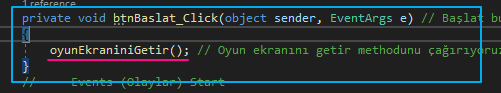
Ana menüdeki çıkış butonuna tıklandığında aşağıdaki kodlar çalışır

****Dosya: anaMenu.cs

Tür: UserControl

Altı çizili kod satırına girildiğinde uygulamadan çıkış yapar.

Ana menüdeki oyunu başlat butonuna tıklandığında aşağıdaki kodlar çalışır

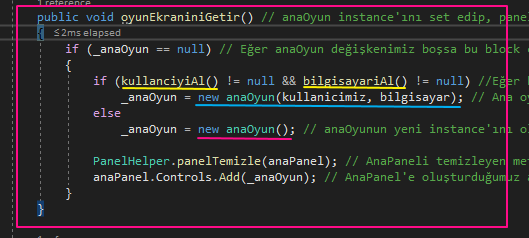
****Dosya: anaMenu.cs

Tür: UserControl

Altı çizili kod satırına gelindiğinde aşağıdaki kodlar çalışır

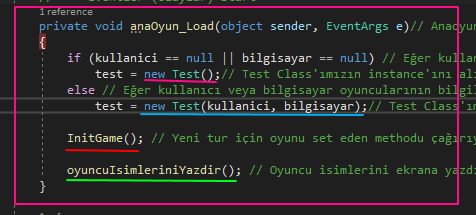
Dosya: anaMenu.cs

Method: oyunEkraniniGetir()

****  
Eğer \_anaoyun değişkenimiz set edilmemişse bu if bloğu çalışır. Eğer altı sarı çizili kullanıcı ve bilgisayar bilgilerini kontrol eden methodlar boş değer dönemez ise altı mavi çizili kod satırına girilir ve şu kodlar çalışır => [anaOyun](#anaOyunParametreli) (parametreli) (tıkla)

Eğer kullanıcı ve bilgisayar bilgileri girilmemişse altı pembe çizili kod satırı çalışır => [anaOyun](#anaOyunParametresiz) (tıkla)

Daha sonra anaPanel içerisindeki elemanları siliyoruz ve anaPanel’e yeni oluşturduğumuz anaOyun UserControl’ünü ekliyoruz.

****Dosya: anaOyun.cs

Event: anaOyun\_Load  
Altı çizili pembe renkli satıra girildiğinde şu kodlar çalışır => [test](#TestParametresiz) (tıkla)

Altı çizili mavi renkli satıra girildiğinde şu kodlar çalışır => [test](#TestParametreli)(Parametreli) (tıkla)

Altı çizili kırmızı renkli satıra girildiğinde şu kodlar çalışır => [InitGame](#initGame) (tıkla)

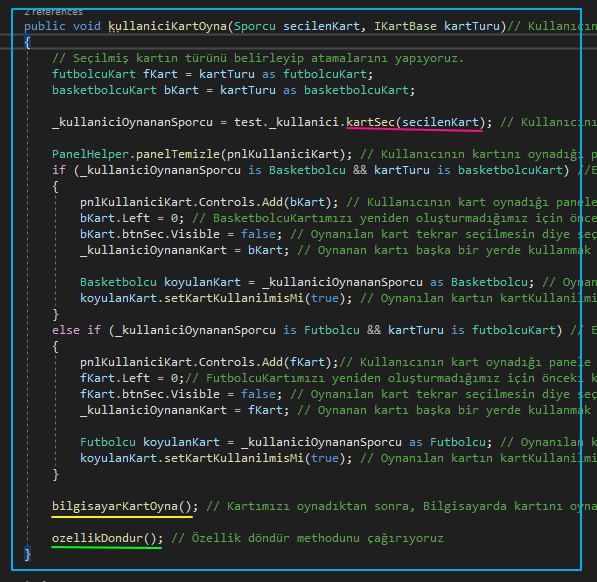
Altı çizili yeşil renkli satıra girildiğinde oyuncuların isimlerinin ekrana yazıldığı method çağırılır.

Daha sonra oyunun oynandığı yer ekrana gelir.

Kart seçildikten sonra;

Dosya: anaOyun.cs

Method: kullaniciKartOyna

****

Bu method kullanıcının seçtiği kartı görsel olarak ekrana yansıtır.

Altı pembe çizili kod satırına girildiğinde seçtiğimiz kartı geri döndüren method çalışır.Sonrasına kartın türüne göre (Basketbolcu veya Futbolcu) if bloklarından birine girer.

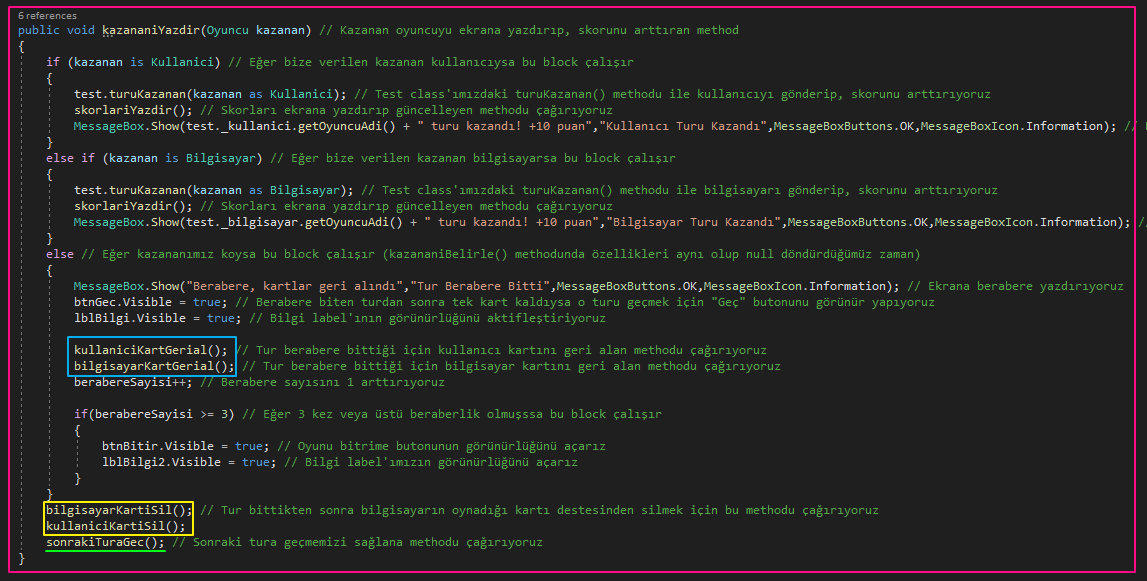
Altı sarı çizili kod satırına girildiğinde bilgisayarın kart oynamasını sağlayan method çağırılır. Çünkü biz kart oynadıktan sonra bilgisayarında kart oynamasını istiyoruz.

=> [bilgisayarKartOyna](#bilgisayarKartOyna)(tıkla)

Altı yeşil çizili kod satırına girildiğinde şuanki tura bağlı olarak (Basketbolcu veya Futbolcu turu) rastgele özellik döndürüp kazananı belirleyen method çağırılır.

**Dosya: anaOyun.cs**

**Method: kazananiYazdir**

****

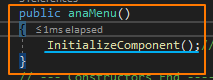
Bu method kazanan oyuncuyu parametre ile alıp ekrana yazdırır.

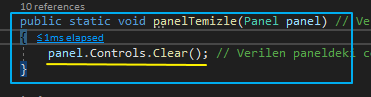
Mavi ile seçilmiş alandaki kodlar beraberlik durumunda kullanıcı ve bilgisayarın kartlarını tekrar destelerine alan methodları çağırır.

Sarı ile seçilmiş alandaki kodlar tur sonunda kullanıcının ve bilgisayarın kartlarını destelerinden silen metodları çağırır.

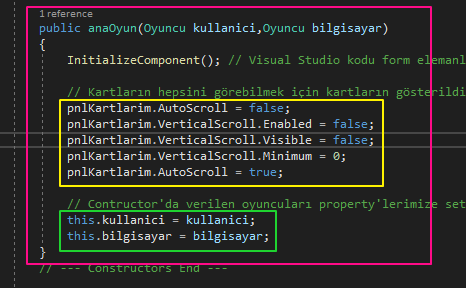
Altı yeşil ile çizilmiş kod satırına girildiğinde sonraki tura geçmemizi sağlayan method çağırılır.

# YER İŞARETLERİ

 Dosya: anaMenu.cs  
anaMenu (UserControl) new edildiğinde girilen kod bloğu. Altı çizili kod satırına girildiğinde UserControl elemanları (butonlar,labellar,vs) oluşturulur.

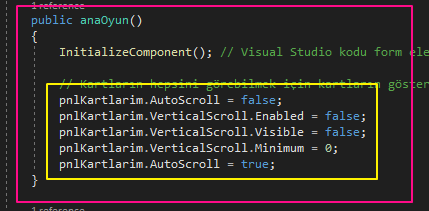
Dosya: PanelHelper.cs

Tür: Static Class  
Parametre olarak verilen panelin içerisindeki tüm kontrolleri silen method. Altı çizili kod satırına gelindiğinde, verilen parametrenin controllerini temizler.

Dosya: anaOyun.cs

Tür: UserControl

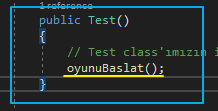
Parametreli Constructor  
Sarı çizili alanda kullanıcının kartlarının gösterildiği panele scrollbar ekliyoruz. Yeşil çizili alanda ise kullanıcı ve bilgisayar değişkenlerine parametrelerden gelen değerleri atıyoruz.

Dosya: anaOyun.cs

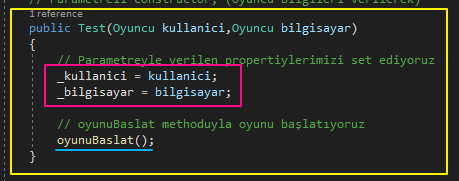
Tür: UserControl

Parametresiz Constructor

Sarı çizili alanda kullanıcının kartlarının gösterildiği panele scrollbar ekliyoruz.

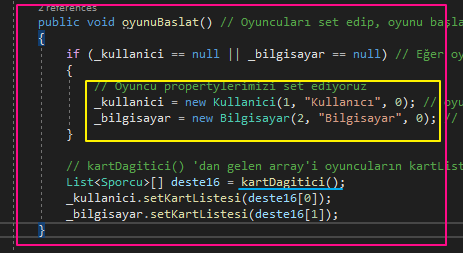
Dosya: Test.cs

Parametresiz Constructor  
Altı sarı çizili satıra girildiğinde şu kodlar çalışır => oyunuBaslat (tıkla)

Dosya: Test.cs

Parametreli contructor  
Pembe alan içindeki kod satırlarında girildiğinde parametreyle gelen değerleri, bu class içinde tanımladığımız kullanıcı ve bilgisayar değişkenlerine atıyoruz.

Altı mavi çizili kod satırına girildiğinde şu kodlar çalışır => oyunuBaslat (tıkla)

Dosya: Test.cs

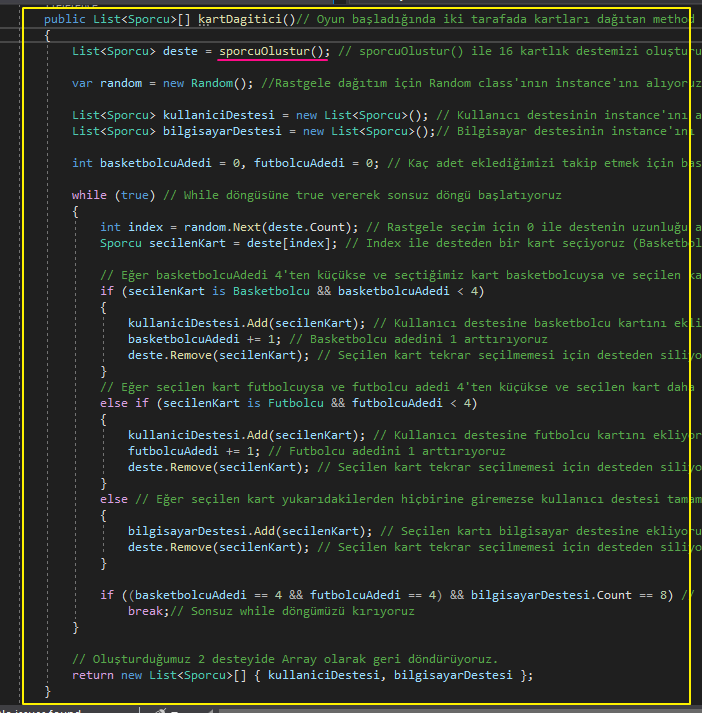
Method: oyunuBaslat  
  
Eğer kullanıcı veya bilgisayar bilgilerimiz girilmemişse sarı alana girip yeni oyuncuları kendimiz tanımlıyoruz.

Altı mavi çizili kod satırına girildiğinde şu kodlar çalışır => kartDagitici (tıkla)

Daha sonra kullanıcının ve bilgisayarın kart listelerini setter’lar ile kartDagitici() methodundan dönen değerlere atıyoruz.

Dosya: Test.cs

Method: kartDagitici

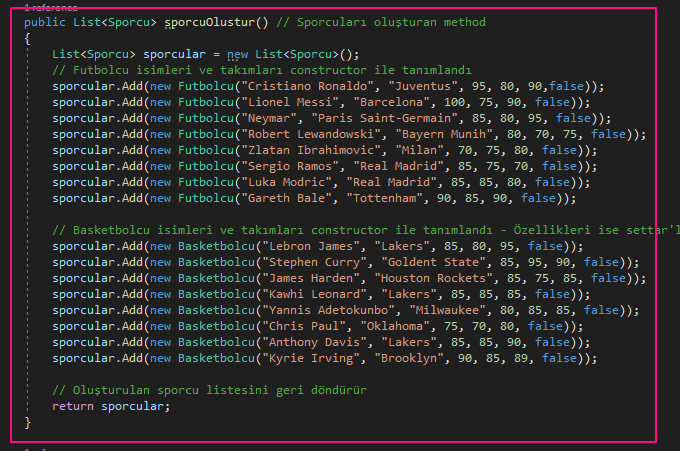


Bu method rastgele bir şekilde 16 kart oluşturup eşit bir şekilde 4 basketbolcu ve 4 futbolcu olarak her bir oyuncuya 8 adet dağıtır.

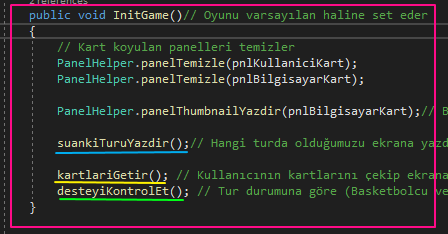
Altı çizili pembe kod satırına girildiğinde şu kodlar çalışır => sporcuOlustur (tıkla)

Dosya: Test.cs

Method: sporcuOlustur



Bu method, 8 futbolcu ve 8 basketbolcu olmak üzere 16 sporcu oluşturup geriye döndürür.

Dosya: anaOyun.cs

Method: InitGame   
Bu method oyunu varsayılan haline set eder.

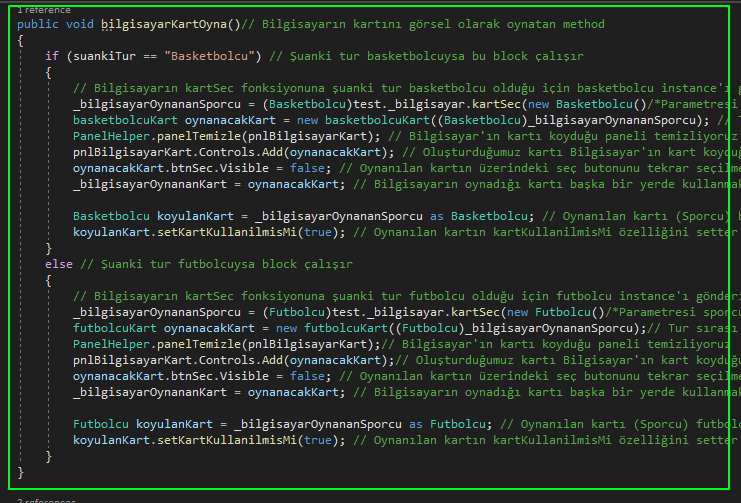
Altı mavi çizili kod satırına girildiğinde şuanki tur sayısına ekrana yazan method çalışır.

Altı sarı çizgili kod satırına girildiğinde kullanıcının kartlarını getirip ekrana yazdıran method çalışır. Ayrıca kartların seç butonlarına kullaniciKartOyna() methodunu ekler.

Altı yeşil çizgili kod satırına girildiğinde desteyiKontrolEt() methodu şuanki tura göre kullanıcının kartları kilitlenip açılır. Basketbolcu turuysa futbolcu kartları kilitlenir, futbolcu turuysa basketbolcu kartları kilitlenir.

Dosya: anaOyun.cs

Method: bilgisayarKartOyna



Bu method şuanki tura göre bilgisayarın destesinden rastgele kart seçilmesini sağlar.