

SPORCU KART OYUNU

Hasan Çolak

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

hasancolak99@gmail.com

ÖZET

Nesneye yönelik programlamanın temel prensipleri kullanılarak futbolcu ve basketbolcu kartlarıyla oynanan basit bir kart oyunu tasarlamamız istenmiştir. Oyun için bir arayüz tasarlamamız ve kullanıcının bilgisayara karşı oynaması istenmiştir. Oyunda kartlar rastgele dağıtılacak olup bilgisayarın kartlarını ortaya random olarak atması istenmiştir.

GİRİŞ

Bu projede amaç ; nesneye yönelik programlamanın yapısını öğrenmek ve uygulamaya geçirebilecek yetkinliğe ulaşmaktır. Projede , nesneye yönelik programlama için çok büyük önem arz eden encapsulation , inheritance , polymorphism , abstraction yapılarını ihtiyaç doğrultusunda mutlaka kullanmamız , her sınıf için biri parametrelili biri parametresiz olmak koşuluyla iki constructor yazmamız ve tüm özellikler için get ile set metodlarını tanımlamamız istenmiştir. Bu bağlamda ; Java , C++ , C# programlama dillerinden birisini seçip , görselleştirme için seçtiğimiz programlama diline uygun kütüphaneyi kullanarak "Sporcu Kart Oyunu" adında bir oyun tasarlamamız istenmiştir. Programlama dili olarak Java'yı , arayüz tasarımı için ise Swing kütüphanesini tercih ettim. Oyunun oluşturulması aşamasında ; "Sporcu" adında bir sınıf oluşturup futbolcu ve basketbolcu kartlarındaki ortak metod ve değişkenlerin bu sınıfta tanımlanması isteniyor. Daha sonra "Basketbolcu" ve "Futbolcu" sınıfları oluşturulup "Sporcu" sınıfına extends edilecek. Ayrıca "Basketbolcu" ve "Futbolcu" sınıflarında , sporcu türüne göre pozisyonları ifade eden değişkenler oluşturulacak. Oyuncuların ellerindeki kartların , skorların ve oyuncu bilgilerinin tutulduğu "Oyuncu" adında sınıf oluşturulup "Kullanıcı" ve "Bilgisayar" sınıfları tarafından extends ettirilecek. "Kullanıcı" ve "Bilgisayar" sınıfları "Oyuncu" sınıfından aldıkları metodlardan bazılarını kendi kullanımlarına göre override edebilecek. Oyunun ilerleyişi ise oluşturacağımız "Test" sınıfında olacak.

Oyun , kullanıcının bilgisayara karşı rekabet edebileceği basit bir kart oyunu şeklindedir. Oyunda ; 8'i futbolcu kartı , 8'i basketbolcu kartı olmak üzere toplam 16 sporcu kartı bulunmaktadır. Futbolcu kartlarının penaltı , serbest atış , kaleciyle karşı karşıya olmak üzere üç özelliği ;

basketbolcu kartlarının ise ikilik , üçlük , serbest atış olmak üzere üç özelliği bulunmaktadır. Kullanıcı ve

bilgisayardan her birine rastgele biçimde 4'ü futbolcu kartı , 4'ü basketbolcu kartı olmak üzere toplamda 8'er kart dağıtılacak ve her oyuncu yalnızca kendi kartlarını görebilecektir. Oyun başlangıcında oyunculardan birisi ortaya diğer oyuncunun göremeyeceği şekilde futbolcu veya basketbolcu kartı atar. Diğer oyuncu da ortaya atılan kartla aynı türde kart - futbolcu kartı atılmışsa futbolcu kartı , basketbolcu kartı atılmışsa basketbolcu kartı - atar. Sonrasında kartlar açılır ve atılan kartların türündeki üç özellikten her hangi birisi bilgisayar tarafından rastgele seçilerek o özelliğin karşılaştırılması iki kart için yapılır. Karşılaştırılan özellikte daha yüksek puanı olan kartı atan oyuncu 10 puan kazanır , ortadaki kartlar ise oyun dışı kalır. Eğer karşılaştırma sonucunda kartların seçili özellikte puanları eşitse oyuncular puan kazanamaz ve ortaya konulan kartlar bir başka hamlede kullanılmak üzere oyunculara geri verilir. Sonuçlanan elden sonra hamle sırası diğer sporcu türüne geçerek (bir önceki elde futbolcular karşılaştırıldıysa basketbolculara , basketbolcular karşılaştırıldıysa futbolculara) devam eder. Oyun , bu şekilde oyuncuların ellerindeki kartlar bitene dek devam eder. Eğer oyuncuların attıkları son kartların karşılaştırıldığı özellikler eşitse kartlar geri alınmadan diğer özellikleri karşılaştırılır. Diğer özellikleri de eşitse oyun eldeki kartlar ile sonlandırılır.

YÖNTEM

Oyunun ilerleyişi "Test" sınıfından takip edileceği için "Test" sınıfı JForm şeklinde oluşturulur , arayüz ekranı burada dizayn edilir. "Test" sınıfı içinde "Futbolcu" sınıfını kullanarak 8 tane futbolcu kartı , "Basketbolcu" sınıfını kullanarak 8 tane basketbolcu kartı oluştururuz. Futbolcu kartlarını "futbolcuKartlari" arrayListine , basketbolcu kartlarını "basketbolcuKartlari" arrayListine ekleriz. Sporcu fotoğraflarını dosyadan alıp tanımladıktan sonra her kartın "fotograf" değişkenine kendi fotoğrafını atarız. 8 elemanlı "buton" ve "label1" arrayListlerini oluştururuz. buton arrayListi arayüz ekranında kullanıcının kartlarını tutar. 0,1,2,3 indeksleri sırasıyla birinci , ikinci , üçüncü , dördüncü futbolcu kartlarını ; 4,5,6,7 indeksleri ise sırasıyla birinci , ikinci , üçüncü , dördüncü basketbolcu kartlarını tutacak şekilde ayarlanacaktır. label1 arrayListi de aynı şekilde arayüz ekranında bilgisayarın kartlarını tutacaktır. Fakat bilgisayarın kartları kullanıcıya gözükmeyeceğinden dolayı label1 arrayListinde kartların tersi gösterilir. Daha sonra oyuncular (kullanıcı ve bilgisayar) tanımlanır ve kartların dağıtılmasına futbolcu kartlarından başlanır. Döngü öncesinde oluşturulan "kullaniciIndex" ve

"bilgisayarIndex" değişkenleri , kartların , oyuncuların kart listelerindeki kaçınıcı indekse konulacağını tutar. While döngüsü oluşturulur ve "kullaniciIndex" ile "bilgisayarIndex" değerleri 4 olana dek çalıştırılır. Çünkü "kullaniciIndex" ve "bilgisayarIndex" değerleri 4 olduğunda her iki oyuncu için de 4 futbolcu kartı dağıtılmış olacaktır. Dağıtımı yaparken "sayi" değişkenine 0'dan 8'e kadar olan rakamlardan birisi atanır. "futbolcuKartlari" arraylistinin "sayi" indeksindeki kart daha önceden dağıtılmamışsa (kartın "kartDagitildiMi" değişkeni false ise) ve oyuncuya 4 futbolcu kartı dağıtılmamışsa (kullanıcı için "kullaniciIndex"!=4 , bilgisayar için "bilgisayarIndex"!=4) o oyuncunun "futbolcuKartlari" arraylistindeki kart sırasına (kullanıcı için "kullaniciIndex" indeksine , bilgisayar için "bilgisayarIndex" indeksine) eklenir. Kullanıcıya dağıtılan kartlar , aynı "futbolcuKartlari" arraylistinde olduğu sırayla butonlara (arayüz ekranında kullanıcının kartlarının bulunduğu yer) eklenir. Bilgisayara dağıtılan kartlar için ise label'e (arayüz ekranında bilgisayar kartlarının bulunduğu yer) ters kart fotoğrafı eklenir. Kartın tekrar dağıtılmaması için kartın "kartDagitildiMi" değişkenine true değeri girilir. Kart kullanıcıya verilmişse "kullaniciIndex" , bilgisayara verilmişse "bilgisayarIndex" değeri 1 artar. Futbolcu kartlarının dağıtımı tamamlandığında ("bilgisayarIndex"==4 && "kullaniciIndex"==4) döngü sona erer. Sıra basketbolcu kartlarının dağıtımına gelir. Basketbolcu kartlarının dağıtımında da futbolcu kartlarının dağıtımında uyguladığımız yöntem kullanılır.

Kart seçiminde kullandığımız "kartSec" metoduyla, oyuncular için tanımlanmış "secilenFutbolcuKarti" ve "secilenBasketbolcuKarti" nesnelere seçilmiş kartı (bilgisayar için random) atamaktır. "kartSec" metodunun kullanımı , kullanıcı ve bilgisayar için farklı olmasından dolayı "Oyuncu" sınıfında abstract olarak ("Kullanıcı" ve "Bilgisayar" sınıflarında override edilmesi zorunlu olarak) yazılmıştır. "Bilgisayar" sınıfı için "kartSec" metodu , belirtilen kart türünde (futbolcu veya basketbolcu) rastgele olacak şekilde kartı seçip seçtiği kartı futbolcuysa "secilenFutbolcuKarti"na , basketbolcuysa "secilenBasketbolcuKarti"na atar. Parametre olarak hangi türde kart seçeceğini ve o türde daha önce bilgisayarın kullandığı kart sayısını alır. Kullanılan kart sayısı , bilgisayarın kaç kart içerisinden seçim yapacağını belirler. Örneğin kullanılan kart sayısı 0 ise 0-4 kart numaraları arasından seçim yaparken , kullanılan kart sayısı 2 olduğunda 0-2 kart numaraları arasından seçim yapabilecektir. "kartSec" metodu "Kullanıcı" sınıfı için ise , kullanıcının istediği kartın "secilenFutbolcuKarti" veya "secilenBasketbolcuKarti"na atanacağı şekilde override edilmiştir. Parametre olarak seçilecek kart türü (futbolcu veya basketbolcu) ve kartın numarası gönderilir. Futbolcu kartı seçildiyse , "futbolcuKartlari"nın "numara" indeksindeki kart "secilenFutbolcuKarti"na atanır. Basketbolcu kartı seçildiyse , "basketbolcuKartlari"nın "numara"-4 indeksindeki kart "secilenBasketbolcuKarti"na atanır. Basketbolcu seçilirken indeksin "numara"-4 olmasının sebebi basketbolcu kartlarının butonlarda 4. numaradan itibaren olmasıdır. Kart dağıtımını tamamladıktan sonra

oyunun oynanması aşamasına geçilir. Oyun kullanıcının kartlarından birini seçmesiyle ilerleyecek şekilde tasarlanmıştır. Kart görselleri sırasıyla butonların üzerine eklenmiştir. Butonlar , kullanıcının kart seçimi yapmasıyla birlikte gerçekleşecek aksiyonları içerir. Kullanıcıya ait 8 tane buton bulunur. kullanici0-1-2-3 butonları futbolcu kartlarını , kullanici4-5-6-7 butonları basketbolcu kartlarını tutar. Hamle sırasının hangi kart türünde olduğunu belirleyebilmek için başlangıçta "futbolcuHamleSayisi" ve "basketbolcuHamleSayisi" adlı iki değişken oluştururuz ve başlangıç değeri olarak 0 değerini atarız. Ortaya futbolcu kartı atıldığında "futbolcuHamleSayisi"nın , basketbolcu kartı atıldığında "basketbolcuHamleSayisi"nın değeri 1 artar. Ortaya atılan kartların beraberlik durumunda "Kartları geri al" yazan butona tıklayıp attığımız kartları geri alınız. Bu butona tıklayıp tıklamadığımızı kontrol eden "kartlarGeriAlindiMi" değişkeni tanımlanmıştır. Beraberlik durumunda "kartlarGeriAlindiMi" değişkeni false değerini alırken , "Kartları geri al" yazan butona tıkladığımızda true değerini alır. Butonlardan birisine tıkladığımızda kartın ortaya atılmasını ve devamındaki aksiyonların gerçekleşmesini 2 şart kontrol eder. Birincisi şart , "kartlarGeriAlindiMi" değişkeninin true değerde olması. İkinci şart ise futbolcu kartlarını temsil eden butonlara ve basketbolcu kartlarını temsil eden butonlara göre farklılık gösteriyor. Eğer futbolcu kartı seçimi yapacaksak (kullanici0-1-2-3 butonlarından birisine basacaksak) , "futbolcuHamleSayisi"nın "basketbolcuHamleSayisi"na eşit olması veya "kullanılanBasketbolcu" değişkeninin 4 olması gerekir. Başlangıçta "futbolcuHamleSayisi" ve "basketbolcuHamleSayisi"nın değerleri 0 olduğu için şartta bahsedilen eşitlik sağlanır ve kart atılabilir. Kartı attıktan sonra "futbolcuHamleSayisi" 1 artırıldığından artık "futbolcuHamleSayisi" == "basketbolcuHamleSayisi" eşitliği sağlanamaz ve bir daha futbolcu kartı atabilmek için basketbolcu kartı atmak gerekir. Basketbolcu kartı (kullanici4-5-6-7 butonları) atabilmek için ise "futbolcuHamleSayisi"nın "basketbolcuHamleSayisi"ndan bir fazla olması gerekir. Başlangıçta bu iki değer eşitken , futbolcu kartı atıldıktan sonra "futbolcuHamleSayisi" bir fazla olur ve basketbolcu kartı seçimi yapılabilir duruma gelir. Basketbolcu kartı seçildikten sonra da "basketbolcuHamleSayisi" değeri 1 artar ve yeniden "futbolcuHamleSayisi" ile eşit duruma geldiğinden futbolcu hamlesi yapılabilir. Böylece ; bir futbolcu kartı , bir basketbolcu kartı atılmak koşuluyla oyunu sürdürürüz. Fakat oyuncuların elindeki futbolcu kartları bittiğinde ("kullanılanFutbolcuKarti"==4 olması durumu) kullanıcı elindeki basketbolcu kartlarını teker teker atabilir , bir futbolcu bir basketbolcu kart atılma zorunluluğu oluşmaz. Aynı şey basketbolcu kartlarının bittiği durumlar ("kullanılanBasketbolcuKarti"==4 olduğunda) için de geçerlidir. Bu sefer de futbolcu kartları teker teker atılabilir. Atmak istediğimiz karta ait butona tıkladığımızda , "kartSec" metoduyla seçtiğimiz kartı ve bilgisayar için de "kartSec" metoduyla seçilen rastgele kart ortaya atılır. Eğer seçilen kart futbolcu kartıysa (kullanici0-1-2-3 butonlarından biri) "futbolcuKarsilastir" metodu , basketbolcu kartıysa "basketbolcuKarsilastir"

metodu kullanılarak rastgele seçilen bir özellik için kartların karşılaştırması yapılır. "ozellikSec" metodu , futbolcu kartları için 0 indekisinde penaltı , 1 indekisinde serbest atış , 2 indekisinde kaleciyle karşı karşıya ; basketbolcu kartları için ise 3 indekisinde ikilik , 4 indekisinde üçlük , 5 indekisinde serbest atış özelliğinin puanını dönderir. "futbolcuKarsilastir" metodunda 0,1,2 değerlerinden biri rastgele seçilerek karşılaştırılacak olan her iki kartın "ozellikSec" metodunda bu değer parametre olarak verilip kartların seçili özellikteki gücü alarak karşılaştırma yapılır. Karşılaştırma sonucunda üstünlük sağlanırsa , kazananoyuncunun skoruna 10 puan eklenir , "kullanilanFutbolcuKarti"nın değeri 1 arttırılır. "basketbolcuKarsilastir" metodunda da olaylar aynı şekilde gerçekleşir. Bu sefer "ozellikSec" metodu için , basketbolcu özelliklerini temsil eden 3,4,5 değerlerinden biri rastgele seçilerek karşılaştırılacak özellik belirlenmiş olup kartın o özellikteki değeri alınır. Her iki karşılaştırma metodunda da beraberlik durumunda (son kart hariç) ekranda "Kartları geri al" butonu oluşur ve "kullanilanFutbolcuSayisi" ile "kullanilanBasketbolcuSayisi"nin değeri değişmez. Karşılaştırmalar tamamlandıktan sonra "panel2" olarak adlandırdığımız karşılaştırma sonuçlarının yer aldığı bölüm ekranda görünür. Her kart butonunda kullandığımız "sonucuAcikla" metodu , kartlar bittiğinde ("kullanilanFutbolcuKarti" == 4 && "kullanilanBasketbolcuKarti" == 4) yeni pencere açarak kimin kazandığını açıklamak içindir.

SONUÇLAR

Bu projede bizden istenen tüm isterler gerçekleştirilerek arayüz ekranında bir kart oyunu oluşturulmuştur. Oyun defalarca kez test edilmiş ve herhangi bir hataya rastlanılmamıştır.

ÇIKTILAR



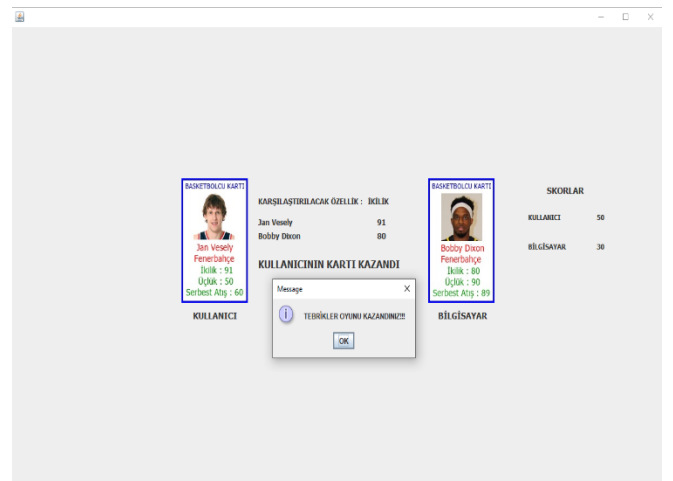
Oyun başlangıcında kartların ekrandaki görüntüsü



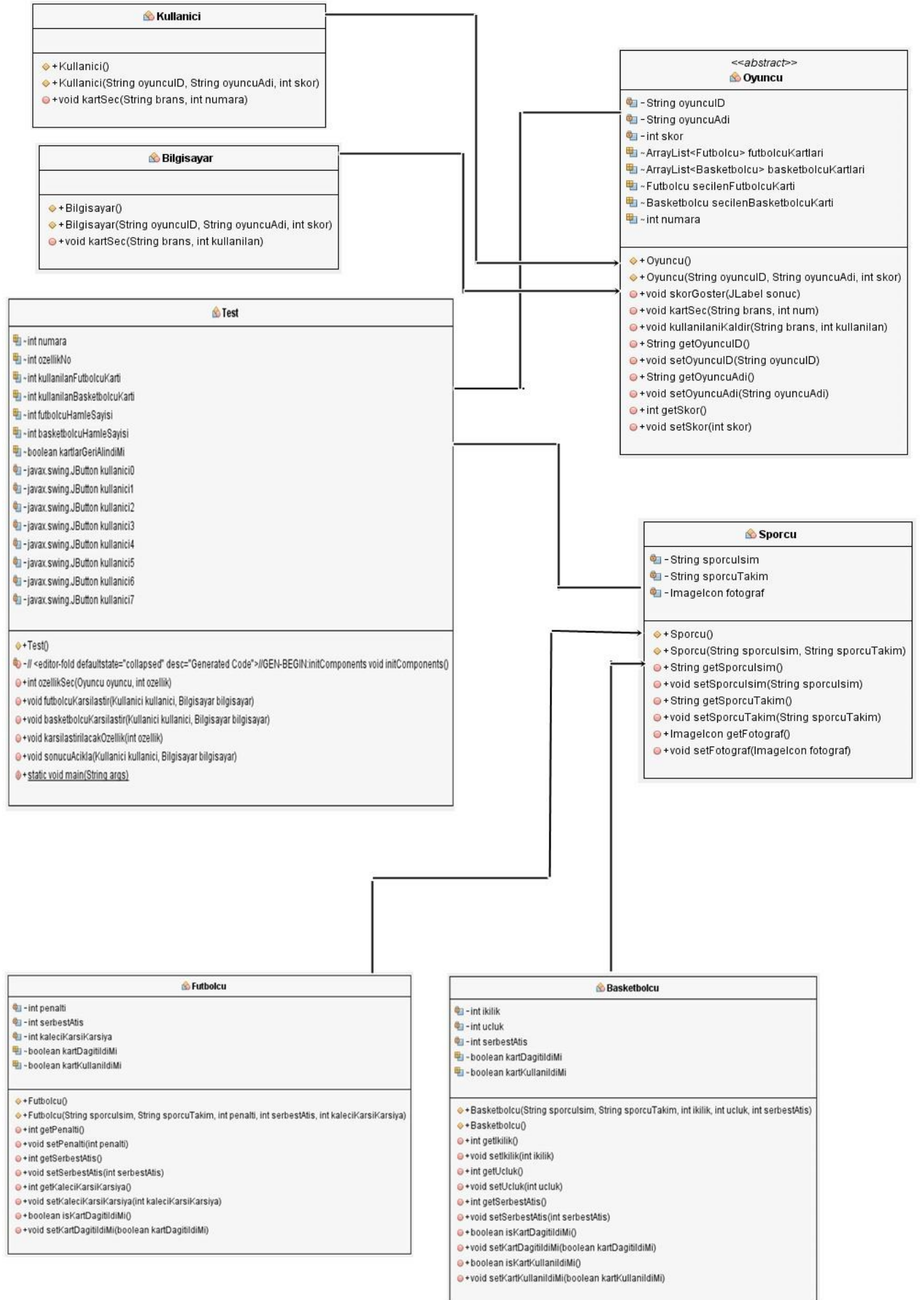
Futbolcu kartı seçimiyle oyun başlar ve kartların karşılaştırması yapılır.



Kartların karşılaştırılan özellikleri eşitse "Kartları geri al" butonuna basarak atılan kartlar geri alınır.



Kartlar bitince oyun sonlanır



KAYNAKÇA

<https://www.youtube.com/playlist?list=PLqG356ExoxZUGwbqoJEKSMnaxVJe4Uvf8>

https://www.youtube.com/playlist?list=PLh9ECzBB8tJPFTpuHKhYayis0H9pS6_rI

<https://www.youtube.com/playlist?list=PL4yfBYtaNjbRRGLQnrTPU2eiAB5rgi9lx>

<http://www.furkanozbay.com/2014/05/pencereyeframe-resim-ekleme.html>

<http://bilgisayarkavramlari.com/2011/06/01/swing-kutuphanesi/>