

# Sporcu Kart Oyunu

*Alperen Ünlü, Kaan Kalaycı*

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

[ahmet.alper96@gmail.com.tr](mailto:ahmet.alper96@gmail.com.tr), [kaan.klyc1@hotmail.com](mailto:kaan.klyc1@hotmail.com)

## Özet:

Programlama Laboratuvarı I Projesi olarak bizden bir oyuncunun otomatik oyuncuyla (bilgisayara karşı) rekabet edebileceği basit bir sporcu kart oyunu ile oyunu ve oyuncuların ellerindeki kartları görüp takip edebileceğimiz bir arayüz tasarlamamız istendi. Tasarlanacak oyunda, 8'i futbolcu, 8'i basketbolcu toplamda 16 adet sporcu kartı olmalı ve her bir kullanıcıya 4'er tane futbolcu ve basketbolcu kartı dağıtılacaktır. Futbolcuların penaltı, serbest vuruş ve çalım; basketbolcuların üçlük, ikilik ve rebound özellikleri bulunmaktadır. Kullanıcı ve bilgisayar kendilerine dağıtılan 8 karttan her hamlede birini seçerek ardışık sırayla bir futbolcu bir basketbolcu kartlarıyla oynanacak şekilde ortaya koyacaklardır. Oyuncular birbirlerinin kartlarını göremeyecektir. Atılan kartın türüne göre pozisyon bilgisi alınır. Pozisyon bilgilerine göre kartlar karşılaştırılır. Pozisyon bilgisine göre yüksek değerli karta sahip kullanıcı 10 puan kazanır. Değerler aynıysa kartlar geri alınmalıdır ve kimse puan kazanmaz. Bu durumda hamle sırası diğer sporcu türünden devam etmelidir. Eldeki kartlar bitene kadar oyun devam edecektir. Eğer eldeki son kartların bütün pozisyon bilgileri aynıysa oyun biter. Bu oyunu tasarlarken bizden "sporcu", "futbolcu", "basketbolcu", "oyuncu", "bilgisayar", "kullanıcı" ve "test" sınıflarını ve onlara ait özellikler ile metotlarını oluşturmamız istendi.

## Giriş:

Bu proje için verilen diller arasından en aşina olduğumuz dil olan Java programlama dilini seçtik ve aynı sebeple Apache Netbeans

geliştirme ortamını seçtik. Arayüz tasarımı için bize kullanımı daha basit gelen "Swing" arayüz tasarım kütüphanesini seçtik. Projede Nesneye Yönelik Programlamanın 4 ana prensiplerinden ikisi olan Encapsulation ve Inheritance yapılarını kullandık.

## Yöntem:

Öncelikle projenin bizden istediği sınıfları oluşturduk. Bunlardan ilki olan "sporcu" üst sınıfını oluşturduk. İçine private olarak sporcuSıfır ve sporcuTakım özelliklerini ve kartlarımızın fotoğraflarına erişebilmek için normal olarak tanımladığımız ImageIcon tipinde kart özelliğini tanımladık ve bu özellikleri atamak için getter-setter metotları ve biri parametresiz biri parametrelili iki constructor oluşturduk. Sporcuların puanını göstermek için sporcuPuanıGöster metodu oluşturduk ancak arayüzümüz bu işleve sahip olduğu için bu metodu kullanmaya ihtiyaç duymadık. Sonra "sporcu" sınıfından kalıtım alan 2 alt sınıf oluşturduk. Bunların ilki olan "futbolcu" sınıfına private olarak penaltı, serbestVuruş ve çalım özelliklerini tanımladık. Bu özellikleri atamak için getter-setter ve biri parametresiz biri parametrelili iki constructor oluşturduk. Ayrıca "sporcu" sınıfının sporcuPuanıGöster metodunu futbolcu özelliklerini uyacak şekilde override ettik. Aynı şekilde kullanmaya ihtiyaç duymadık. Diğer alt sınıf olan "basketbolcu" sınıfına private olarak ikilik, üçlük ve rebound özelliklerini tanımladık ve "futbolcu" sınıfında yaptığımız gibi bu sınıfa getter-setter, constructor ve override işlemlerini uyarladık. Sonra "oyuncu" üst sınıfını oluşturduk. İçine private olarak oyuncuID, oyuncuAdı, skor ile kartları tutacak kartListesi1 ve kartListesi2'yi

tanımladık ve bu özellikleri atamak için getter-setter metotları ve biri parametresiz biri parametrelili iki constructor oluşturduk. Oyuncuların skorlarını göstermek için skorGöster metodu oluşturduk ama benzer şekilde arayüzümüz bu işleve sahip olduğu için metodu kullanmaya ihtiyaç duymadık. Futbolcu kartlarını dağıtmak için KartDagit1 ve basketbolcu kartlarını dağıtmak için KartDagit2 metodlarını oluşturduk. Oyuncuların kart seçmesi için kartSeç metodunu oluşturduk. Sonra “oyuncu” sınıfından kalıtım alan 2 alt sınıf oluşturduk. Bunların ilki olan “kullanıcı” sınıfını ve biri parametrelili biri parametresiz olan constructorlarını oluşturduk. Kullanıcı arayüzünden kart seçeceği için override ettiğimiz kartSeç metodu kullanılmadı. Diğer alt sınıf olan “bilgisayar” sınıfına kartSeç metodunu kullanabilmek için private olarak sayıListesi ve sayıListesi2’yi tanımladık ve bunların getter-setter metotlarını oluşturduk. Biri parametrelili biri parametresiz olan constructorlarını tanımladık. “oyuncu” sınıfından override ettiğimiz kartSeç metodunu tanımladığımız sayı listelerini kullanarak bilgisayarın rastgele kart seçeceği şekilde oluşturduk. Son olarak oyunun ilerleyişini takip etmek için “test” sınıfını oluşturduk. İlk önce kullanıcı için “kullanıcı”, bilgisayar için “bot” objelerini tanımlayıp özelliklerini atadık. Sonra kartlarımız için kullanacağımız sporcuları tanımlayıp özelliklerini ve kart görsellerini atadık. Ayrıca bilgisayarın kartlarını görmememiz için arka adında bir kart arkası görseli tanımladık. Sonra futbolcuları ve basketbolcuları daha rahat işleme sokmak için 2 ayrı arraylist tanımlayıp onlara yolladık. Bir de oyunun ilerleyişini takip etmek için 2 adet futbolcu ve basketbolcuya ait sayaç tanımladık. Bilgisayarın kartSeç metodunu kullanmak için sayı listelerine sayılar yolladık. Kartların hangi özelliklerinin kıyaslanacağını rastgele olarak belirleyen pozisyonBelirle metodunu tanımladık. Sonra arayüz işlemlerine başladık. İlk başta arkaplan ve başlat butonu dışında bütün elemanları görünmez yaptık. Başlat butonuna basıldığında buton görünmez oldu ve arraylistlerdeki sporcular kullanıcı ve bilgisayara rastgele dağıtıldı. Kullanıcı ve bilgisayarın kartları, kullanıcının kartları açık;

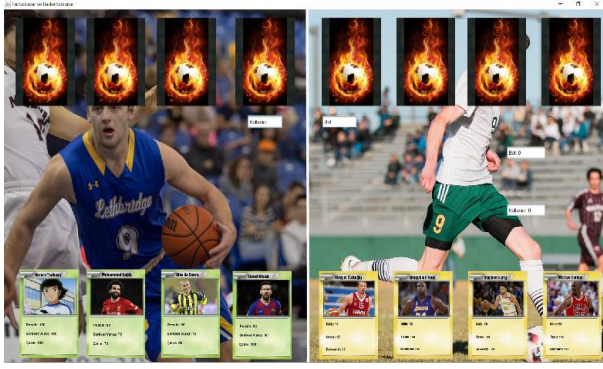
bilgisayarın kartları kapalı olacak şekilde, skorları ve adları ekranda belirdi. Kullanıcı bir kartı seçtiğinde, öncelikle bilgisayara ait kartSeç metodu çalışır ve bilgisayar için rastgele bir kart seçer. Kullanıcının ve bilgisayarın seçtiği kart arayüzün ortasına atılır. Sonra pozisyonBelirle metodu çalışır ve atılan kartların türüne göre rastgele bir pozisyon belirler ve ekranda pozisyonun ne olduğu yazılır. Pozisyona göre kartların puanları karşılaştırılır, kazanan taraf belirtilir, kazanan tarafa 10 puan eklenir ve o kartların tekrar kullanılmaması için görünmez hale getirilir. Ayrıca kullanılan kartın türüne göre sayaç 1 artar. Seçilen kartlar hangi türe ait ise o türe ait diğer kartlar seçilemez hale gelir, diğer türden kartlar seçilebilir hale getirilir. Beraberlik durumunda ise iki tarafa da puan eklenmez ve kartları çek butonu belirir. Kartları çek butonu ortaya atılan kartları tekrar ait olduğu yere gönderir ve diğer türden kart seçilmesini zorunlu hale getirir. Eğer bir türün bütün kartları bitmişse ardışık sıra olmaz ve diğer türün kartlarıyla oyun devam eder. Bütün kartlar bittiğinde skora göre kazanan oyuncu belirtilir. Eğer atılan son kartların bütün puanları aynıysa iki tarafa da puan eklenmez ve oyun bitirilir, kazanan oyuncu belirtilir.

## Deneyisel Sonuçlar:

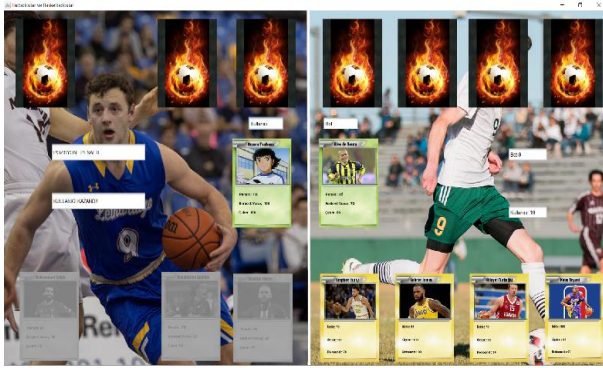
Başlat ekranı:



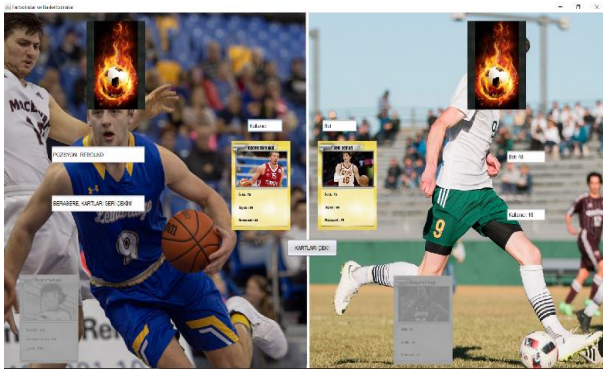
Başlattıktan sonra:



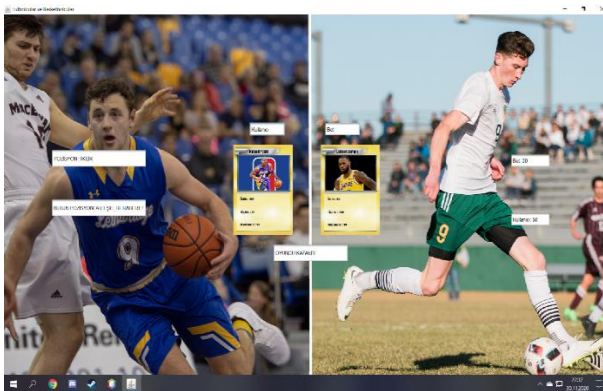
Kart atınca:



Berberlik olursa:



Son beraberlik durumu:



**Sonuç:**

Bu proje sayesinde Java dilini kullanmakta deneyim kazandık ve Nesneye Yönelik Programlamanın ana prensiplerinin önemini öğrendik. Kullandığımız Swing arayüz kütüphanesiyle de basit düzeyde bir arayüzü nasıl tasarlayacağımızı öğrendik. Projemizde eksik olan tek kısım son beraberlik durumunda geri kalan pozisyonların rastgele seçilmemesi oldu.

## Yalancı Kod:

1-Başla.

2-Oyuncuları tanımla.

3-Sporcuları tanımla.

4-Futbolcu ve basketbolcular için 2 arraylist oluştur.

5-Kart kullanımı için sayaçlar tanımla.

6-Arraylistlere sporcuları ekle.

7-Bilgisayarın sayı listelerine sayı ekle.

8-Arayüzün elemanlarını gizle.

9-Başla butonuna bas.

10-Başla butonunu gizle.

11-Oyunculara kartlarını rastgele dağıt.

12-Oyuncuların kartlarını göster.

13-Skorları ve adları göster.

14-Kullanıcının kartına tıkla.

15-Bilgisayar için rastgele bir kart seç.

16-Seçilen kartları ortaya at.

17-Seçilen kartın türüne göre rastgele pozisyon belirle ve ekranda göster.

18-Puanları karşılaştır, kart puanı daha büyük olan oyuncunun skoruna 10 puan ekle, sayacı 1 artır ve kartları sil.

19-Puanlar eşitse, kartları geri çek ve puan ekleme.

20-Diğer kart türünü seçilebilir, aynı kart türünü seçilemez hale getir.

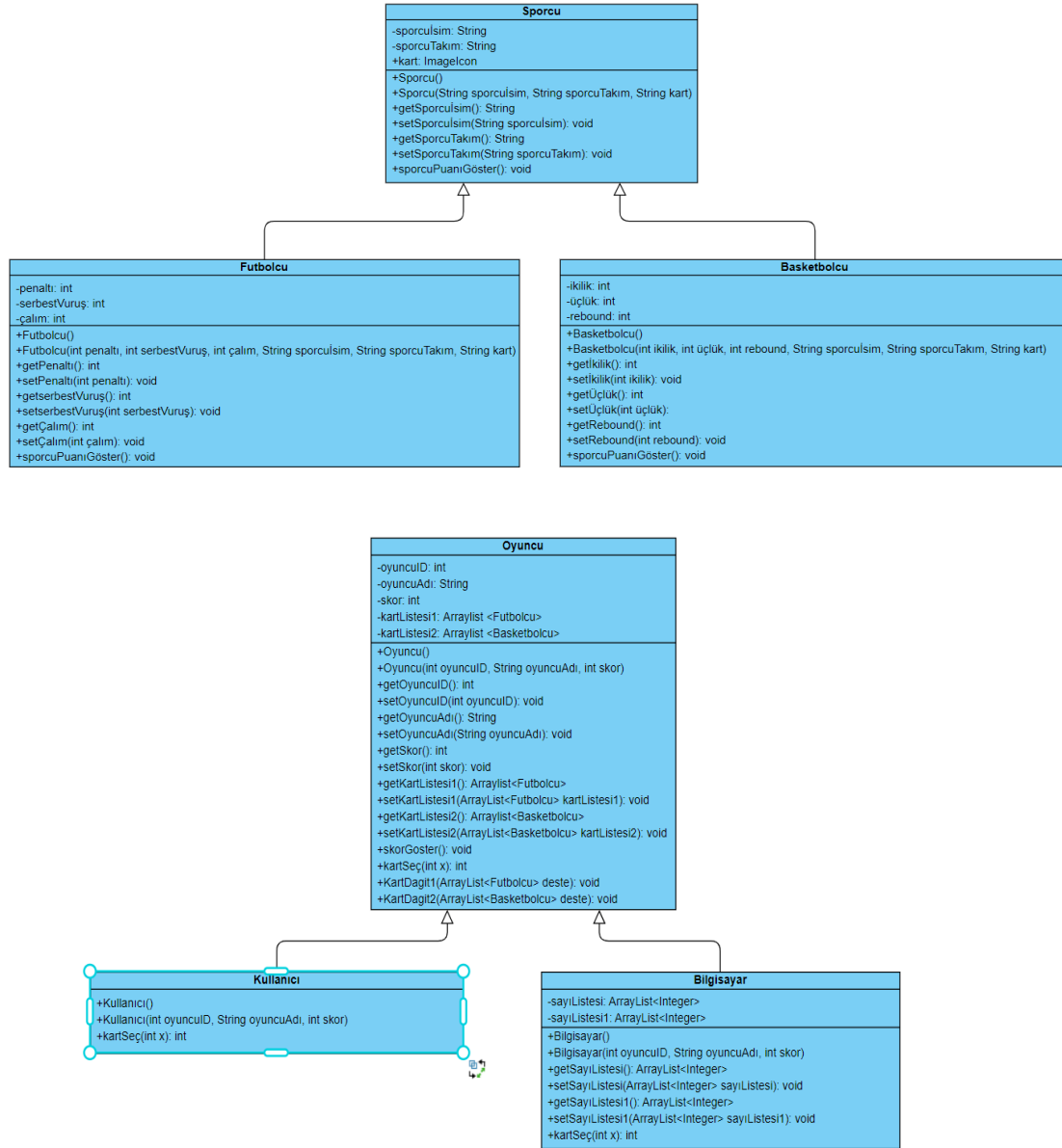
21- Bu işlemi diğer kullanıcı kartları için uygula.

22-Bütün kartlar bitince oyuncuların skorlarını karşılaştır, ekrana kazananı yaz.

24-Bitir.

23-Eğer son kartların bütün pozisyon puanları birbirine eşitse oyunculara puan ekleme, oyunu bitir.

## UML Sınıf Diyagramı:





Test
+kullanici: Kullanici +bot: Bilgisayar +alex: Futbolcu +messi: Futbolcu +ronaldo: Futbolcu +tsubasa: Futbolcu +neymar: Futbolcu +salah: Futbolcu +maradona: Futbolcu +ronaldinho: Futbolcu +kobe: Basketbolcu +lebron: Basketbolcu +jordan: Basketbolcu +shaquille: Basketbolcu +cedi: Basketbolcu +hidayet: Basketbolcu +kagami: Basketbolcu +stephen: Basketbolcu +arka: ImageIcon +list1: ArrayList<Futbolcu> +list2: ArrayList<Basketbolcu> +fkartKontrol: int +bkartKontrol: int +pozisyonBelirle(): int +Test() -Kullanici1ActionPerformed(): void -Kullanici2ActionPerformed(): void -Kullanici3ActionPerformed(): void -Kullanici4ActionPerformed(): void -Kullanici5ActionPerformed(): void -Kullanici6ActionPerformed(): void -Kullanici7ActionPerformed(): void -Kullanici8ActionPerformed(): void -kartlariCekActionPerformed(): void +main(String args[]): void -BaslaActionPerformed(): void

## Kaynakça:

<https://stackoverflow.com/questions/21695175/jtextfield-setenabled-vs-seteditable/21696772>

<https://docs.oracle.com/javase/7/docs/api/java/awt/Component.html>

<https://stackoverflow.com/questions/8557716/how-to-return-multiple-values#:~:text=You%20can%20return%20an%20object,Map%2C%20List%2C%20Set%20etc.>

<https://stackoverflow.com/search?q=return+multiple+values+in+java%3F&s=37413c74-9db1-4b0a-b335-3159741eba8a>

<https://medium.com/@kamer.dev/oopnin-4-ana-prensibi-encapsulation-inheritance-abstraction-polymorphism-712ed2fbac7e>

<https://www.javatpoint.com/java-jbutton>

<https://www.javatpoint.com/java-jlabel>

<https://stackoverflow.com/questions/16763356/java-getclass-getresource-on-a-png-returning-null-pointer/16763469>