

# PROGRAMLAMA LABORATUVARI 1

## 2. PROJE

Murat Karakurt

Ahmet Burhan Bulut

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

### Özet

Bu doküman Programlama Laboratuvarı 1 dersi 2. Projesi için çözümümü açıklamaya yönelik oluşturulmuştur. Dökümanda projenin tanımı, çözüme yönelik yapılan araştırmalar, kullanılan yöntemler, proje hazırlanırken kullanılan geliştirme ortamı ve kod bilgisi gibi programın oluşumunu açıklayan başlıklara yer verilmiştir. Doküman sonunda projemi hazırlarken kullandığım kaynaklar bulunmaktadır.

### 1. Proje Tanımı

Projede bizden istenen, bir oyuncunun otomatik oyuncuyla (bilgisayar), savaşılabileceği basit bir kart oyunu yaratmamızdır.

Tasarlayacağımız oyunda, toplamda 8 futbolcu ve 8 basketbolcu kartı olacak ve her bir kullanıcıya ilk başta random olarak 4 basketbolcu 4 futbolcu kartı dağıtılacaktır. Dağıtımdan sonra oyun başlayacaktır. Kullanıcı ve bilgisayar kendilerine dağıtılan ilk önce futbolcu kartını ardından basketbolcu kartlarından birini seçerek ortaya koyacaktır (sıralama futbolcu ve basketbolcu olarak devam etmek zorundadır). Kartlarda üçer tane özellikten rastgele olarak bir tanesi belirlenir futbolcular için serbest vuruş,penaltı ve kaleci ile karşı karşıya; basketbolcular için serbest atış, ikilik ve üçlük olmak üzere her kart üç özelliğe sahiptir. Seçilen kartlar ortaya gönderilir seçilen özelliğe göre bilgisayar ya da kullanıcı puan kazanır. Beraberlik durumunda kartlar destenin sonuna eklenir ve sıradaki kart atılır.

Eldeki kartlar bitene kadar oyun devam edecektir. Oyun bittiğinde en yüksek puana sahip oyuncu kazanacaktır. Bu oyunu nesneye yönelik programlama yöntemini kullanarak yapmamız beklenmektedir.

Oyun kurallarına ek olarak proje gereği olan isterler bulunmaktadır. Oyunu ve oyuncuların ellerinde bulunan kartlar görülebilecek ve takip edilebilecek bir arayüz tasarlamamız beklenmektedir.

Oyun işleyişi aşağıdaki gibi olmalıdır:

- ❖ Masadaki destede 16 adet Sporcu kartı bulunmalıdır.
- ❖ Oyuncu desteden 8 rastgele kart alır.
- ❖ Bilgisayar desteden 8 rastgele kart alır.
- ❖ Oyuncu ve bilgisayar elinde bulunan 8 karttan sıralamaya göre seçerek açık bir şekilde ortaya koyar. Bilgisayar seçimi random olarak gerçekleştirir. Burada kullanıcı bilgisayarın hangi kartı seçtiğini bilmeyecek (Fakat sunum sırasında, oyunun doğru çalıştığının kontrolü yapılabilmesi için kartlar gösterilmeli).



Şekil 1. Seçilen Kartlar

- ❖ Bilgisayarın kartları kullanıcıya arayüzde kapalı olarak görünür(Şekil 2.) kullanıcı kart seçtiğinde bilgisayarın kartı açık bir şekilde ortaya gönderilir.



Şekil 2. Kapalı Kartlar

- ❖ Kartlar ortaya gönderildikten sonra rastgele bir pozisyon belirlenir bu pozisyona göre özelliği yüksek olan kart kazanır.
- ❖ Kartların özellikleri eşit ise kartlar ortaya gönderilmez destenin sonuna eklenir, sıradaki karta geçer.
- ❖ Eldeki kartlar bitene kadar bu işlemler uygulanır. Kartlar bitince oyun sonucu ekrana yazdırılır.

### 1.1. Oluşturulması Gereken Sınıflar

Aşağıda belirtilen tüm sınıflarda ortak olarak yapıcı (constructor) metotları, sınıflardaki tüm özellikler için get, set metotları tanımlanmalıdır.

- ❖ **Sporcu Sınıfı**  
Sporcunun puanını göstermek için `sporcuPuanıGoster()` fonksiyonu tanımlanmalıdır. Sınıfta `sporcuIsim` ve `sporcuTakim` özellikleri tutulmaktadır. Bu sınıf abstract bir sınıf olup hiç bir sınıftan kalıtım almamaktadır.

❖ **Basketbolcu Sınıfı**  
Sporcu sınıfından kalıtım almaktadır. `sporcuPuanıGoster()` bu sınıfta override edilmiştir. Ek olarak kart kullanıldığı zaman tekrar kullanmamak için `kartKullanildiMi()` metodu yazılmıştır.

- ❖ **Futbolcu Sınıfı**  
Sporcu sınıfından kalıtım almaktadır. `sporcuPuanıGoster()` bu sınıfta override edilmiştir. Ek olarak kart kullanıldığı zaman tekrar kullanmamak için `kartKullanildiMi()` metodu yazılmıştır.

- ❖ **Oyuncu Sınıfı**

Bu sınıf abstract bir sınıf olup hiç bir sınıftan kalıtım almamaktadır. `kartSayısıFutbol()` ve `kartSayısıBasketbol()` metotlarında kullanılmamış kartların sayısı tutulur. `kartVarMiFutbol()` ve `kartVarMiBasketbol()` metotları da gönderilen kartın kullanılıp kullanılmadığını kontrol eder.

`kartKullanFutbol()` ve `kartKullanBasketbol()` metotları kartların deste içerisinde mi kontrolünü yapar ve kartı `kullan()` metodu yardımıyla kullanır. 2 tane abstract `kartSec` ve `kartSec1` metodu tanımlanır.

- ❖ **Bilgisayar Sınıfı**

`kartSec` ve `kartSec1` fonksiyonları override edilerek bilgisayar random kart seçer.

- ❖ **Kullanıcı Sınıfı**

`kartSec` metodu kullanılarak gönderilen kart seçilir.

## 2. YÖNTEM

- Arayüz tasarımı için JavaFX ve scenebuilder kullanılmıştır.
- Random kart seçme ise bütün kartlar bir diziye atanıp Random kullanılarak dağıtılmıştır.

## 3.KABA KOD

- ❖ **Sporcu Sınıfı**

- 1) Sporcu sınıfı abstract olarak tanımlanmıştır.
- 2) String olarak `sporcuIsim` ve `sporcuTakim` tanımlanmıştır.
- 3) Constructor, getter ve setter metotları tanımlanmıştır.
- 4) `sporcuPuanıGoster()` metodu tanımlanmıştır.

- ❖ **Futbolcu Sınıfı**

- 1) Futbolcu sınıfı Sporcu sınıfından kalıtım almaktadır.
- 2) Private olarak `penalti`, `serbesAtis`, `kaleciKarsiKarsiya`, `kartKullanildiMi` değişkenleri tanımlanmıştır.
- 3) Constructor, getter ve setter metotları tanımlanmıştır.
- 4) `kullan()` metodu kartı kullanır.
- 5) `sporcuPuanıGoster()` override edilmiştir.

- ❖ **Basketbolcu Sınıfı**

- 1) Basketbolcu sınıfı Sporcu sınıfından kalıtım almaktadır.
- 2) Private olarak `ikilik`, `ucluk`, `serbestAtis`, `kartKullanildiMi` değişkenleri tanımlanmıştır.
- 3) Constructor, getter ve setter metotları tanımlanmıştır.
- 4) `kullan()` metodu kartı kullanır.
- 5) `sporcuPuanıGoster()` override edilmiştir.

- ❖ **Oyuncu Sınıfı**

- 1) Oyuncu sınıfı abstract olarak tanımlanmıştır.
- 2) Private olarak `oyuncuID`, `oyuncuIsim`, `Skor`, `kartListesi`, `kartListesi2` tanımlanmıştır.
- 3) Constructor, getter ve setter metotları tanımlanmıştır.
- 4) `kartSayısıFutbol()` ve `kartSayısıBasketbol()` metotları kullanılmamış kart sayısını bulur.

5)kartVarMiBasketbol( ) ve kartVarMiFutbol( ) metotları eğer destede kart kalmamışsa false döndürür. Destede kart var ise gönderilen kartın kullanılma durumunu döndürür.

6) kartKullanFutbol( ) ve kartKullanBasketbol( ) metotları destede kart yoksa veya gönderilen kart yok ise değer döndürmez. Kart destede ise kartı desteden bulur ve kullanır.

7) Abstract olarak Futbolcu kartSec( ) ve Basketbolcu kartsec1( ) oluşturulmuştur.

#### ❖ Kullanıcı Sınıfı

- 1)Kullanıcı sınıfı Oyuncu sınıfından kalıtım almaktadır.
- 2)Parametrelili ve parametresiz constructor metotları oluşturulmuştur.
- 3)Futbolcu kartSec( ) ve Basketbolcu kartsec1( ) override edilmiştir.

#### ❖ Bilgisayar Sınıfı

- 1)Bilgisayar sınıfı Oyuncu sınıfından kalıtım almaktadır.
- 2)Parametrelili ve parametresiz constructor metotları oluşturulmuştur.
- 3)Futbolcu kartSec( ) ve Basketbolcu kartsec1( ) override edilerek bilgisayara rastgele kart seçtirilmiştir.

#### ❖ Controller Sınıfı

- 1)sample.fxml dosyasından ImageView ve Text'ler çekilmiştir.
- 2)initialize( ) metodu içinde Main içerisindeki diziler çekilmiştir.
- 3)kullanıcıPuan, bilgisayarPuan, ve gameState private olarak tanımlanmıştır.
- 4)userClicked( ) metodu içerisinde asıl oyun oynanmaktadır. Dağıtılmış kartlar alınarak oyuna başlanır. Spor branşı belirlenip branşa göre bilgisayara rastgele kart ve rastgele pozisyon seçtirilir. Kullanıcının oynanan branştaki bir sporcu seçmesi beklenir. Seçilen sporcular seçilen pozisyonda karşılaştırılır eğer bir üstünlük durumu varsa üstün olan 10 puan kazanır. Beraberlik durumunda ise kartlar ortaya gönderilmez ve sıradaki branşa geçilir. Tüm kartlar bitene kadar bu olay tekrarlanır. Son olarak kazanan belirlenir.

#### ❖ Main Sınıfı

- 1)Public olarak kullanıcıFutbol, bilgisayarFutbol, kullanıcıbasketbol, bilgisayarBasketbol dizileri tanımlanır.
- 2)start metodu override edilerek arayüz penceresi açılır.
- 3)Arayüz penceresine sample.fxml entegre edilir
- 4)Sekiz futbol ve sekiz basketbolcu olmak üzere on altı tane sporcu tanımlanır.
- 5)Kullanıcıya dört futbol ve dört basketbol olmak üzere sekiz tane rastgele kart verilir.
- 6)Kalan sekiz kart bilgisayara verilir.
- 7)Images klasöründeki oyuncu kartları ile dizideki oyuncular eşleştirilir.
- 8)Arayüz main içerisinde çalıştırılır.

## 4.DENEYSEL SONUÇLAR



#### KULLANICI KARTLARI:

Hakan Çalhanoğlu  
Sergen Yalçın  
Cristiano Ronaldo  
Lionel Messi  
Bogdan Bogdanovic  
Cedi Osman  
Göksenin Köksal  
Şehmus Hazer

#### BİLGİSAYAR KARTLARI

Neymar da Silva Santos Júnior  
Zlatan Ibrahimovic  
Ozan Tufan  
Sabri Sarıoğlu  
Stephen Curry  
Kobe Bryant  
Lebron James  
Melih Mahmutoğlu



Hakan Çalhanoğlu VS Ozan Tufan  
 --Pozisyon -> Serbest Vuruş  
 --Kullanıcı 10 Puan Kazandı

Bogdan Bogdanovic VS Melih Mahmutoğlu  
 --Pozisyon -> Serbest Atış  
 --Bilgisayar 10 Puan Kazandı

Sergen Yalçın VS Neymar da Silva Santos Júnior  
 --Pozisyon -> Serbest Vuruş  
 --Bilgisayar 10 Puan Kazandı

Göksenin Köksal VS LeBron James  
 --Pozisyon -> İkilik  
 --Bilgisayar 10 Puan Kazandı

Cristiano Ronaldo VS Zlatan Ibrahimovic  
 --Pozisyon -> Penaltı  
 Berabere Kalındı Lütfen Sıradaki Basketbol Kartını Seçiniz...

Cedi Osman VS Kobe Bryant  
 --Pozisyon -> İkilik  
 --Bilgisayar 10 Puan Kazandı

Lionel Messi VS Sabri Sarıoğlu  
 --Pozisyon -> Kaleci İle Karşı Karşıya  
 --Kullanıcı 10 Puan Kazandı

Şehmus Hazer VS Stephen Curry  
 --Pozisyon -> Üçlük  
 --Bilgisayar 10 Puan Kazandı

Cristiano Ronaldo VS Zlatan Ibrahimovic  
 --Pozisyon -> Serbest Vuruş  
 --Bilgisayar 10 Puan Kazandı

Bilgisayar KAZANDI...

## 5.SONUÇ

Nesne yönelimli programlama pratiği yapmak amacıyla Java dili kullanılarak istenilen kart oyunu yapılmıştır.

Bu projenin kazanımı nesne yönelimli programlamanın prensiplerini öğrenmek ve uygularken nelere dikkat edilmesi gerektiğini tecrübe etmektir.

## 5.1 EKSİKLİKLER

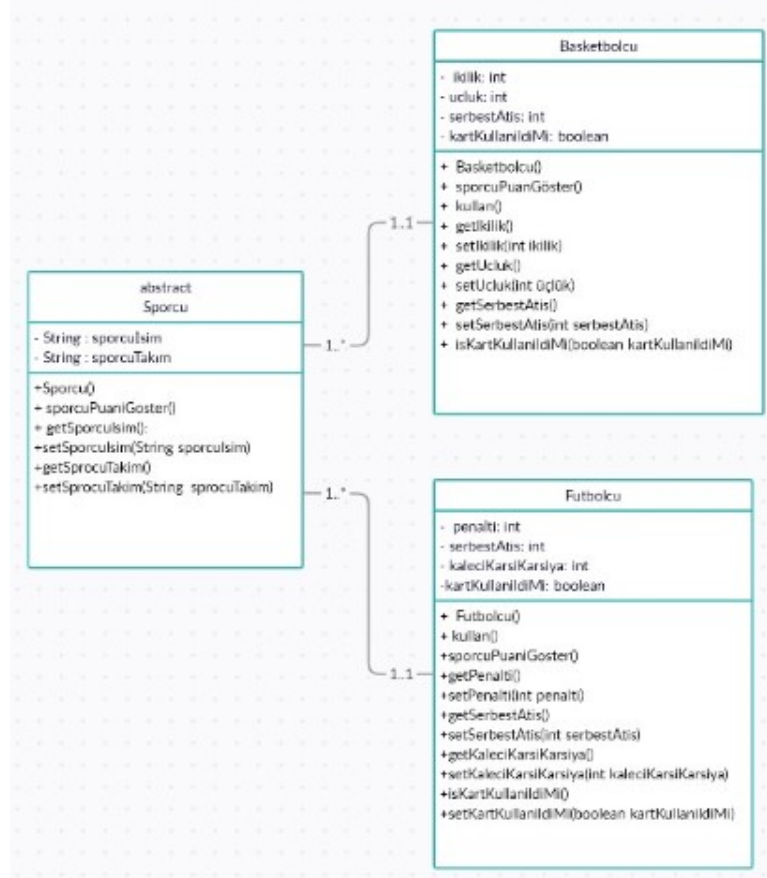
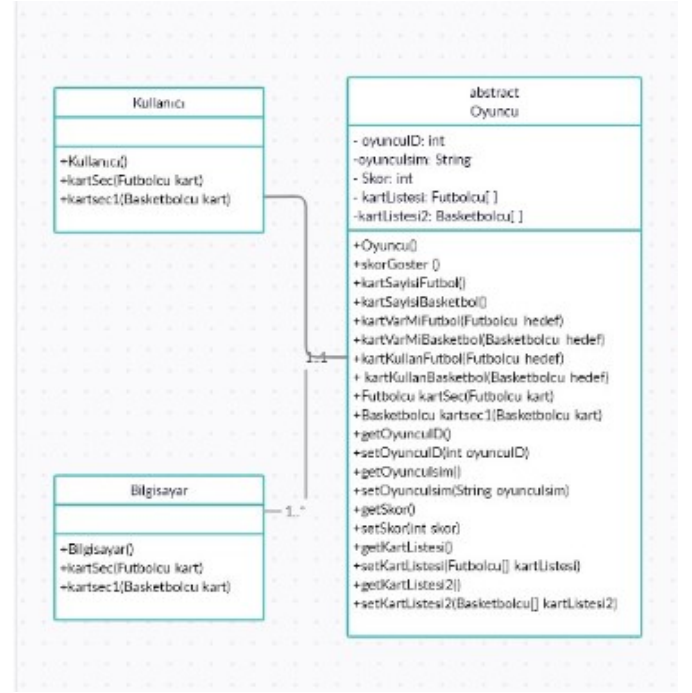
sporcuPuanGoster metodu override edilmiştir fakat arayüzde bu metot kullanılmadan direkt kullanıcının skoru yazdırılmıştır.

İstenilen kart dağıtımı ve oyun işleyişi controller içerisinde kontrol edilmiştir fakat arayüz açılımı main içerisinde yapılmıştır.

## UML DİYAGRAMI İÇİN(TIKLAYIN)

Oyun içi daha fazla görsel için(Tıklayın)

## 6.UML DİYAGRAMI



## 7.KAYNAKÇA

- [https://docs.oracle.com/javase/8/javafx/get-started-tutorial/get\\_start\\_apps.htm#JFXST804](https://docs.oracle.com/javase/8/javafx/get-started-tutorial/get_start_apps.htm#JFXST804)
- [https://www.youtube.com/watch?v=BHj6zbH3inI&list=PLfu\\_Bpi\\_zcDNYL6171Op3S1ABtuyFV7Nr](https://www.youtube.com/watch?v=BHj6zbH3inI&list=PLfu_Bpi_zcDNYL6171Op3S1ABtuyFV7Nr)