

Kocaeli Üniversitesi

Bilgisayar Mühendisliği Bölümü

Programlama Laboratuvarı I

Sporcu Kart Oyunu

Alperen Tan
190201054@kocaeli.edu.tr

Oğuz Narlı
180201074@kocaeli.edu.tr

Projenin Özeti:

Programlama Laboratuvarı 2. proje olarak bizlerden “Sporcu kart oyunu” yapmamız beklenmektedir.

Oluşturduğumuz uygulama, önceden oluşturduğumuz 8 basketbolcu ve 8 futbolcu kartını oyunculara rastgele bir biçimde dağıtarak en yüksek puana ulaşanı kazanan belirleyecek şekilde kodlanmıştır.

Giriş:

Oyunumuz java programlama dili ile Netbeans 12.1 geliştirme ortamında kodlanmış olup swing arayüz kütüphanesi ile arayüzü tasarlanmıştır.

Temel Bilgiler:

Sporcu Sınıfı:

Bu sınıf kartlarda belirteceğimiz sporcuların özelliklerini ve işlevlerini içermektedir. Sporcu sınıfı sporculsim, sporcuTakim ve seçildiMi olmak üzere 3 farklı özelliğe sahiptir. Bu sınıfa bağlı olarak “Futbolcu” ve “Basketbolcu” olmak üzere 2 alt sınıfı vardır. Parametrelili ve parametresiz olmak üzere 2 tanede yapıcı metoda sahiptir. Parametrelili constructorin aldığı parametreler sporculsim ve sporcuTakim dir. Ayrıca özelliklerini atayabilmemizi yada alabilmemizi sağlayan setter ve getter metotları tanımlıdır. Sporcu sınıfında yer alan kod aşağıda belirtilmiştir.

```
public class Sporcu {  
  
    private String sporcuIsim = "";  
    Sporcu kartında belirtilen sporcunun ismini belirtir.
```

```
private String sporcuTakim = "";  
Sporcu kartında belirtilen sporcunun takımını belirtir.
```

```
private boolean seçildimi=false;  
Kartın seçilip seçilmediğini belirtir.
```

```
public Sporcu() {}  
Parametre almayan yapıcı metodumuzdur.  
Bu metot ile birlikte Test sınıfında nesnemizi tanımlarken herhangi bir parametre almak zorunda kalmayacaktır.
```

```
public Sporcu(String sporcuIsim, String sporcuTakim) {  
    this.sporcuIsim = sporcuIsim;  
    this.sporcuTakim = sporcuTakim;  
}
```

Parametre alan yapıcı metodumuzdur. Bu metot ile birlikte Test sınıfında nesnemizi tanımlarken aynı zamanda aldığı parametrelere göze nesnemizin özelliğini belirtebiliriz. Aldığımız parametre görüldüğü gibi nesnenin aldığı özelliğe eşitlenmiştir.

```
public String getSporcuIsim() {  
    return sporcuIsim;  
}
```

Tanımladığımız Sporcu nesnesinin ismini çağırabildiğimiz metottur.

```
public void setSporcuIsim(String sporcuIsim) {  
    this.sporcuIsim = sporcuIsim;  
}
```

Tanımladığımız Sporcu nesnesinin ismini atayabildiğimiz methodtur.

```
public String getSporcuTakim() {  
    return sporcuTakim;  
}
```

Tanımladığımız Sporcu nesnesinin takımını çağırabildiğimiz metottur.

```
public void setSporcuTakim(String
sporcuTakim) {
    this.sporcuTakim = sporcuTakim;
}
}
```

Tanımladığımız Sporcu nesnesinin takımını atayabildiğimiz metottur.

Futbolcu Sınıfı:

Futbolcu sınıfımız Sporcu sınıfımızın alt sınıfıdır. “penalti”, “serbestVuruş” ve “kaleciKarsiKarsiya” olmak üzere 3 özelliğimiz vardır. Ayrıca Sporcu sınıfının alt sınıfı olduğundan onun özelliklerini miras alacaktır. Parametrelili ve parametresiz olmak üzere 2 tanede yapıcı metoda sahiptir. Parametrelili metodumuz Sporcu sınıfında miras aldığımız özelliklerle beraber bu sınıfa ait özellikleri 3 parametre daha vardır. Sporcu sınıfından miras aldığımız özellikleri yapıcı metodumuzun içinde super() işlevi ile gönderiyoruz. Aldığı özellikleri atayabilmemizi ya da alabilmemizi sağlayan setter ve getter metotları da içermektedir. Futbolcu sınıfında yer alan kod aşağıda açıklamasıyla belirtilmiştir.

```
public class Futbolcu extends Sporcu {
    “extends” ile Sporcu sınıfının özelliklerini miras
    alıyoruz.
```

```
private int penalti = 0;
private int serbestVuruş = 0;
private int kaleciKarsiKarsiya = 0;
Futbolcu kartında sporcunun penaltı atabilme,
serbest vuruş ve kaleci ile karşı karşıya özelliğini
belirtir.
```

```
public Futbolcu(String sporcuIsim, String
sporcuTakim, int penalti, int serbestAtis,
int kaleciKarsiKarsiya) {
    super(sporcuIsim, sporcuTakim);
    this.penalti = penalti;
    this.serbestVuruş = serbestAtis;
    this.kaleciKarsiKarsiya=kaleciKarsiKarsiya;
}
public Futbolcu() {
    super();
}
```

Sporcu sınıfında tanımlanan parametrelili yapıcı metot ile aynı amaçla kullanır. super() metodu ile Sporcu sınıfında yapıcı metodunu aldığı parametreye göre veri almaktadır. super() metodu kullanımı ayrıca üst sınıfın özelliklerini miras aldığını belirtir.

```
public int getPenalti() {
    return penalti;
}
public void setPenalti(int penalti) {
    this.penalti = penalti;
}
public int getserbestVuruş() {
    return serbestVuruş;
}
```

```
public void setSerbestAtis(int serbestAtis)
{
    this.serbestVuruş = serbestAtis;
}
public int getKaleciKarsiKarsiya() {
    return kaleciKarsiKarsiya;
}
public void setKaleciKarsiKarsiya(int
kaleciKarsiKarsiya) {
    this.kaleciKarsiKarsiya=kaleciKarsiKarsiya;
}}
```

Futbolcu nesnesinin sahip olduğu penaltı, serbest vuruş ve kaleci ile karşı karşıya özelliğini atamamıza ve almamıza yarar.

Basketbolcu Sınıfı:

Basketbolcu sınıfımız Sporcu sınıfımızın alt sınıfıdır. “ikilik”, “ucluk” ve “serbestAtis” olmak üzere 3 özelliğimiz vardır. Ayrıca Sporcu sınıfının alt sınıfı olduğundan onun özelliklerini miras alacaktır. Parametrelili ve parametresiz olmak üzere 2 tane de yapıcı metoda sahiptir. Parametrelili metodumuz Sporcu sınıfında miras aldığımız özelliklerle beraber bu sınıfa ait özellikleri 3 parametre daha vardır. Aldığı özellikleri atayabilmemizi ya da alabilmemizi sağlayan setter ve getter metotları da içermektedir. Futbolcu sınıfında yer alan kod aşağıda açıklamasıyla belirtilmiştir.

```
public class Basketbolcu extends Sporcu {
```

```
private int ikilik = 0;
private int ucluk = 0;
private int serbestAtis = 0;
```

Basketbolcu kartında sporcunun ikilik, üçlük ve serbest atış özelliğini belirtir.

```
public Basketbolcu(String sporcuIsim,
String sporcuTakim, int ikilik, int ucluk,
int serbestAtis) {
    super(sporcuIsim, sporcuTakim);
    this.ikilik = ikilik;
    this.ucluk = ucluk;
    this.serbestAtis = serbestAtis;
}
public Basketbolcu() {
    super();
}
```

Sporcu sınıfında tanımlanan parametrelili yapıcı metot ile aynı amaçla kullanır. super() metodu ile Sporcu sınıfında yapıcı metodunun aldığı parametreye göre veri almaktadır.

```
public int getIkilik() {
    return ikilik;
}
public void setIkilik(int ikilik) {
    this.ikilik = ikilik;
}
public int getUcluk() {
    return ucluk;
}
public void setUcluk(int ucluk) {
    this.ucluk = ucluk;
}
public int getSerbestAtis() {
```

```

        return serbestAtis;
    }
    public void setSerbestAtis(int serbestAtis)
    {
        this.serbestAtis = serbestAtis;
    }
}

```

Basketbolcu nesnesinin sahip olduğu ikilik, üçlük ve serbest atış özelliğini atamamıza ve almamızı sağlar.

Oyuncu Sınıfı:

Oyuncu sınıfımız oyunu oynayan oyuncuları temsil etmektedir. “id”, “isim”, “skor”, “flist” ve “blist” olmak üzere 5 farklı özelliğe sahiptir. 3 tane parametrelili 1 tane parametresiz olmak üzere 4 farklı yapıcı metoda sahiptir. İçeriği kod açıklandığında belirtilecektir. Sınıfımız setter ve getter metotlarını da içermektedir. Ayrıca oyuncuların skorunu gösteren “skorGoster()” ve oyuncunun kart seçmesi için “kartSec()” metodu da yer almaktadır. Kodun açıklaması aşağıda belirtilmiştir.

```

public class Oyuncu {
    private int oyuncuID;
    private String isim;
    private int skor=0;

```

Oyuncunun ID sini, ismini ve aldığı toplam skoru belirtir.

```

    public ArrayList<Futbolcu> fList=new
    ArrayList<Futbolcu>();
    public ArrayList<Basketbolcu> bList=new
    ArrayList<Basketbolcu>();

```

Yukarıdaki arraylistler oyunculara rastgele dağıtılan kartları temsil etmektedir. Kartları temsil ettiğinden dolayı alacakları nesne tipleri Sporcu sınıfına bağlı Futbolcu ve Basketbolcu sınıfları olacaktır.

```

    public int getOyuncuID() {
        return oyuncuID;
    }
    public void setOyuncuID(int oyuncuID) {
        this.oyuncuID = oyuncuID;
    }
    public String getIsim() {
        return isim;
    }
    public void setIsim(String isim) {
        this.isim = isim;
    }
    public int getSkor() {
        return skor;
    }
    public void setSkor(int skor) {
        this.skor = skor;
    }
}

```

Oyuncu nesnesinin ID, isim ve skor özelliklerini atamamıza ve almamıza yarar.

```

public Oyuncu(int oyuncuID, String isim,
int skor) {
    this.oyuncuID = oyuncuID;
    this.isim = isim;
    this.skor = skor;
}
public Oyuncu() {
}

```

```

    public Oyuncu(int oyuncuID, String isim)
    {
        this.oyuncuID = oyuncuID;
        this.isim = isim;
    }

```

```

    public Oyuncu(int oyuncuID) {
        this.oyuncuID = oyuncuID;
    }

```

Yukarıda oyuncuID, isim ve skor parametreleri almaktadır. Aşağısında ise diğer yapıcı metotlar olan oyuncuID, isim, oyuncuID parametreleri ve parametresiz yapıcı metotları da yer almaktadır. Farklı parametreler alan yapıcı metot tanımlama sebebimiz nesne tanımladığımızda istediğimiz herhangi bir özelliğe göre değer verebilme olanağımız olmasıdır. Bu şekilde diğer özellikleri daha sonrada verebiliriz.

```

    public void skorGoster(){
        System.out.println(oyuncuID+"
oyuncunun ("+isim+") skoru:"+skor);
    }

```

Oyuncunun sahip olduğu özelliği ekrana bastıran bir fonksiyondur.

```

    public int kartSec(int kartId){
        return kartId;
    }
}

```

Oyuncunun kart seçmesini sağlayan metottur. Oyuncu sınıfının alt sınıflarında override işlemi ile isteğimize göre işlevi şekillendirilmiştir.

Kullanıcı Sınıfı:

Kullanıcı sınıfı oyuncunun gerçek alemde oynayan kişiyi temsil etmektedir. Oyuncu sınıfının özelliklerini miras almıştır.2 parametrelili ve parametresiz olmak üzere 2 yapıcı metoda sahiptir. Bunun yanında “kartSec()” metodu Oyuncu sınıfınınkinden override edilmiştir. Fakat içeri Oyuncu sınıfı ile aynıdır. Kod ve açıklaması aşağıda belirtilmiştir.

```

public class Kullanici extends Oyuncu{
    public Kullanici(int oyuncuID, String
isim, int skor) {

```

```

        super(oyuncuID, isim, skor);
    }

```

```

    public Kullanici() {
    }
}

```

Nesne tanımlanırken Oyuncu sınıfından miras aldığı özelliğe göre şekillenmesini sağlar. `super()` metodu ile üst sınıftan aldığı özelliği yollar.

```
public Kullanici(int oyuncuID) {  
    super(oyuncuID);  
}
```

Nesne tanımlarken sadece id tanımlamamızı sağlar.

```
@Override  
public int kartSec(int kartId) {  
    return super.kartSec(kartId);  
}  
}
```

“kartId” parametresini alır ve aldığı parametreyi gönderir.

Bilgisayar Sınıfı:

Oyunu sanal kısımda yani oyuncunu bilgisayar olduğu kısmı temsil eder. “basketboladım” ve “futboladım” olmak üzere 2 parametre almaktadır. İşlevleri aşağıda kodun açıklandığı yerde belirtilecektir. Ayrıca parametrelili ve parametresiz olmak üzere 2 tane yapıcı metoda sahiptir. Ayrıca Oyuncu sınıfından “kartSec()” metodu override edilmiştir. Yalnız bilgisayar seçtiği kartı rastgele seçtiğinden dolayı metodun içeri istegimize göre tekrar biçimlendirilmiştir. Bilgisayar sınıfına dair kodun açıklaması aşağıda belirtilmiştir.

```
public class Bilgisayar extends Oyuncu {  
    public int basketboladim=4;  
    public int futboladim=4;
```

Yukarıdaki iki özelliğin tanımlama sebebimiz seçilecek kartın hangi sayı aralığında kart seçeceğini `kartSec()` metodunda belirtir. Kartlar karşılaştırıldığında sahip olduğu kart arraylistten çıkartılacağından seçebileceği sayı aralığı azalacaktır. Bundan dolayı tanımlama ihtiyacı duyulmuştur.

```
public Bilgisayar(int oyuncuID, String isim, int skor) {  
    super(oyuncuID, isim, skor);  
}  
public Bilgisayar() {}
```

Nesne tanımlanırken Oyuncu sınıfından miras aldığı özelliğe göre şekillenmesini sağlar. `super()` metodu ile üst sınıftan aldığı özelliği yollar.

```
@Override  
public int kartSec(int durum) {  
    Random random=new Random();  
    if(futboladim==0||basketboladim==0){  
        return super.kartSec(0);  
    }  
    if(durum%2==0){  
        int kartId=  
            random.nextInt(futboladim);  
        return super.kartSec(kartId);  
    }  
    else{  
        int kartId=  
            random.nextInt(basketboladim);
```

```
        return super.kartSec(kartId);  
    }  
}
```

“kartSec” metodumuz bu sınıf için rastgele kart seçmemizi sağlar. Durum parametresi oyunun basketbolda mı yoksa futbolda mı olduğunu belirtir. 2'ye tam bölünüyorsa futbol, bölünmüyorsa sıranın basketbolda olduğunu belirtir. Sahip oldukları akış kontrol yapısında(if else) Random yapısı kullanılarak atama işlemleri yapılır. İlk if yapısı ise eğer elimizde son kart kaldıysa direk 0. indeksteki değeri atamasını sağlar.

Test Sınıfı:

“karsilastir() Metodu:

Bu metodumuzda oyuncularımızın seçtiği kartları karşılaştırıp kazanana puan verme görevini üstlenen metottur. “Bilgisayar pc”, “Kullanici kul”, “int sıra”, “int kartidpc” ve “int kartidkul” olmak üzere 4 farklı parametreye sahiptir. “Bilgisayar pc” oyuncu olan bilgisayarı, “Kullanici kul” oyuncu olan kullanıcıyı, “int sıra” sıranın futbol kartında yoksa basketbol kartında mı olduğunu temsil eder, “int kartidkul” ise kullanıcının seçtiği kartın idsini temsil eder. Metodumuzun işleyişi gelen sıra parametresi 2'ye tam bölünüyorsa sıra futbolda değilse sıra basketbolda olduğu ifade edilir. Daha sonra seçilen kartların hangi özelliği seçeceğini rastgele belirlenir ve seçilen kartlar karşılaştırılır. Metodumuzun içeriği ve açıklaması aşağıda belirtilmiştir.

```
static ArrayList<Futbolcu> futbolcu=new  
ArrayList<Futbolcu>();  
static ArrayList<Basketbolcu>  
basketbolcu=new ArrayList<Basketbolcu>();  
static Bilgisayar bilgisayar=new  
Bilgisayar(2,"Bilgisayar",0);  
static Kullanici oyuncu=new Kullanici(1);  
static int tiklanan=4;
```

```
static void karsilastir(Bilgisayar  
pc,Kullanici kul,int sıra,int kartidpc,int  
kartidkul){  
    Random r=new Random();  
    int pozisyon=r.nextInt(3);
```

Rastgele 3 farklı pozisyonlardan biri seçilir ve aşağıdaki değişkenimize atanır.

```
if (sıra%2 == 0) {  
    Eğer "sıra" parametresi 2 ye tam bölünüyorsa sıra  
    futbolda demektir. Böylece seçilen futbolcu kartları  
    karşılaştırılır."sıra" parametremiz tam bölünmez ise  
    else yapısında basketbol kartları seçilen basketbol  
    kartları için karşılaştırma yapacaktır.
```

```
Futbolcu fpc=pc.fList.get(kartidpc);  
Futbolcu fkul=kul.fList.get(kartidkul);  
Seçilen kartların atandığı geçici nesnelerdir.
```

```

oyananankartisimpc.setText(fpc.getSporcuIsim());
oyananankarttakimpc.setText(fpc.getSporcuTakim());
ucuncupuandegeripc.setText(""+fpc.getPenalti());
dorduncupuandegeripc.setText(""+fpc.getSerbestAtis());
besincipuandegeripc.setText(""+fpc.getKaleciKarsiKarsiya());
ikinciucuncuyazi.setText("Penalti:");
ikincidorduncuyazi.setText("SerbestAtis:");
ikincibesinciyazi.setText("Kaleci ile karsi karsiya:");

```

Yukarıdaki kod parçacıkları, arayüzde kullanılan yazıları basketbolcu veya futbolcu turuna göre değiştiriyor.

```

if (pozisyon == 0) {

    pozisyonText.setText("Kaleci ile karsi karsiya");
    oyPuanText.setText(""+fkul.getKaleciKarsiKarsiya());
    pcPuanText.setText(""+fpc.getKaleciKarsiKarsiya());

    if (fpc.getKaleciKarsiKarsiya() > fkul.getKaleciKarsiKarsiya()) {
        pc.setSkor(pc.getSkor()+10);
        kul.fList.remove(kartidkul);
        pc.fList.remove(kartidpc);
        pc.futboladim--;
    }
    else if (fpc.getKaleciKarsiKarsiya() < fkul.getKaleciKarsiKarsiya()) {
        kul.setSkor(kul.getSkor()+10);
        kul.fList.remove(kartidkul);
        pc.fList.remove(kartidpc);
        pc.futboladim--;
    }
}

else if (pozisyon == 1) {

    pozisyonText.setText("Penalti");
    oyPuanText.setText(""+fkul.getPenalti());
    pcPuanText.setText(""+fpc.getPenalti());

    if (fpc.getPenalti() > fkul.getPenalti()) {
        pc.setSkor(pc.getSkor()+10);

        kul.fList.remove(kartidkul);
        pc.fList.remove(kartidpc);

        pc.futboladim--;
    }

    else if (fpc.getPenalti() < fkul.getPenalti()) {
        kul.setSkor(kul.getSkor()+10);
        kul.fList.remove(kartidkul);
        pc.fList.remove(kartidpc);
        pc.futboladim--;
    }
}

```

```

}

else {
    pozisyonText.setText("Serbest Atis");
    oyPuanText.setText(""+fkul.getSerbestAtis());
    pcPuanText.setText(""+fpc.getSerbestAtis());
    ;
    if (fpc.getSerbestAtis() > fkul.getSerbestAtis()) {
        pc.setSkor(pc.getSkor()+10);
        kul.fList.remove(kartidkul);
        pc.fList.remove(kartidpc);
        pc.futboladim--;
    }

    else if (fpc.getSerbestAtis() < fkul.getSerbestAtis()) {
        kul.setSkor(kul.getSkor()+10);
        kul.fList.remove(kartidkul);
        pc.fList.remove(kartidpc);
        pc.futboladim--;
    }
}

```

Yukarıdaki if else yapılarımızda rastgele seçilen pozisyona göre işlem yapılacaktır. Seçilen pozisyonda öncelikle arayüzdeki yazılar ayarlanır. Ardından oyuncuların seçtiği kartların özellik puanları karşılaştırılır. Eğer herhangi üstünlük durum varsa kazanana puan verilir(örn:pc.setSkor(pc.getSkor()+1)). Daha sonra kartlar oyuncuların sahip oldu kart listesinden çıkarılır ve de bilgisayarın rastgele kart seçme alanı daraldığından "pc.futboladim" yada "pc.basketboladim" bir azaltılır. Eğer özelliklerde eşitlik sağlanırsa kartlar geri dağıtılır ve herhangi azaltma işlemi yapılmaz.

```

else{

    Basketbolcu bpc=pc.bList.get(kartidpc);
    Basketbolcu bkul=kul.bList.get(kartidkul);
    oynananankartisimpc.setText(bpc.getSporcuIsim());
    oynananankarttakimpc.setText(bpc.getSporcuTakim());
    ucuncupuandegeripc.setText(""+bpc.getIkilik());
    dorduncupuandegeripc.setText(""+bpc.getUcluk());
    besincipuandegeripc.setText(""+bpc.getSerbestAtis());
    ikinciucuncuyazi.setText("Ikilik:");
    ikincidorduncuyazi.setText("Ucluk:");
    ikincibesinciyazi.setText("Serbest Atis:");

    if (pozisyon == 0) {
        pozisyonText.setText("Ikilik");
        oyPuanText.setText(""+bkul.getIkilik());
        pcPuanText.setText(""+bpc.getIkilik());

        if (bpc.getIkilik()>bkul.getIkilik()) {

            pc.setSkor(pc.getSkor()+10);

```

```

kul.bList.remove(kartidkul);
pc.bList.remove(kartidpc);
pc.basketboladim--;

}

else if (bpc.getIkilik()<bkul.getIkilik())
{
kul.setSkor(kul.getSkor()+10);
kul.bList.remove(kartidkul);
pc.bList.remove(kartidpc);
pc.basketboladim--;

}

}

else if (pozisyon == 1) {

pozisyonText.setText("Ucluk");
oyPuanText.setText(""+bkul.getUcluk());
pcPuanText.setText(""+bpc.getUcluk());

if (bpc.getUcluk() > bkul.getUcluk()) {
pc.setSkor(pc.getSkor()+10);
kul.bList.remove(kartidkul);
pc.bList.remove(kartidpc);
pc.basketboladim--;

}

else if (bpc.getUcluk() < bkul.getUcluk())
{
kul.setSkor(kul.getSkor()+10);
kul.bList.remove(kartidkul);
pc.bList.remove(kartidpc);
pc.basketboladim--;

}

}

else {
pozisyonText.setText("Serbest Atis");
oyPuanText.setText(""+bkul.getSerbestAtis());
pcPuanText.setText(""+bpc.getSerbestAtis());
;

if (bpc.getSerbestAtis() >
bkul.getSerbestAtis()) {
pc.setSkor(pc.getSkor()+10);
kul.bList.remove(kartidkul);
pc.bList.remove(kartidpc);
pc.basketboladim--;

}

else if (bpc.getSerbestAtis() <
bkul.getSerbestAtis()) {
kul.setSkor(kul.getSkor()+10);
kul.bList.remove(kartidkul);
pc.bList.remove(kartidpc);
pc.basketboladim--;

} } }

```

```

pcPuan.setText("Bilgisayar
Puani:"+bilgisayar.getSkor());
oyuncuPuan.setText("Oyuncu
Puani:"+oyuncu.getSkor()); }

```

Yukarıdaki else kısmı basketbol kartları için yapılan işlemi temsil etmektedir. İçeri en son belirttiğimiz gibidir.

"kartDagit()" Metodu:

Bu metodumuz oyunculara rastgele kart dağıtma işlevini üstlenmektedir. Metodumuz "Bilgisayar pc", "Kullanici kul", "ArrayList<Futbolcu> futbolcu" ve "ArrayList<Basketbolcu> basketbolcu" şeklinde 4 farklı parametre almaktadır. "Bilgisayar pc" oyuncu olan bilgisayarı "Kullanici kul" parateremiz oyuncu olan kullanıcımız, "ArrayList<Futbolcu> futbolcu" ve "ArrayList<Basketbolcu> basketbolcu" ise dağıtılacak kartları temsil etmektedir. Dağıtılan kartlar açılma kısmında belirteceğimiz gibi listeden çıkarılacaktır. Metodumuzun içeriği ve açıklaması aşağıda belirtilmiştir.

```

static void kartDagit(Bilgisayar
pc,Kullanici kul,ArrayList<Futbolcu>
futbolcu,ArrayList<Basketbolcu>
basketbolcu){

```

```

int ind=7;

```

Yukarıdaki değişkenimiz indis listede hangi indis aralığında tahmin yapacağımızı temsil etmektedir. Her tahmin yaptığıında kart listeden çıkarıldığında 1 azaltılacaktır.

```

Random r=new Random();

```

Yukarıdaki nesnemiz rastgele sayı atamamızı sağlar.

```

for (int i = 0; i < 4 ; i++) {
int ata=r.nextInt(ind);
pc.fList.add(futbolcu.get(ata));
futbolcu.remove(ata);
ind--; }

```

Yukarıdaki for döngüsü ile beraber oyuncularımıza kartlarını dağıtıyoruz. Döngünün 4'e kadar dönme sebebi her oyuncunu her spor dalından 4'er kart sahibi olmasından kaynaklıdır."r.nextInt(ind)" yapısı ile beraber

yukarıda belirlediğimiz ind değişkeni ile 0 arasında rastgele sayı atanacaktır.

"pc.fList.add(futbolcu.get(ata));" yapısı ile bilgisayar kullanıcısına rastgele belirlenen sayıya göre futbolcu listesinden kart seçilir. Daha sonra "futbolcu.remove(ata);" yapısı ile beraber listeden çıkılır ind bir azaltılarak tahmin aralığı daraltılır.

```
for (int i = 0; i < 4; i++) {
    kul.fList.add(futbolcu.get(ind));
    futbolcu.remove(ind);
    ind--;
}
```

Yukarıdaki for döngümüzde futbolcu listesinde kalan kartlarımızı kullanıcının futbolcu kartı olarak atar.

ind=7;
"ind" değişkenimiz tekrar 7'ye eşitleme sebebimiz bu sefer basketbolcu kartları için atama yapmamızdan kaynaklıdır.

```
for (int i = 0; i < 4; i++) {
    int ata=r.nextInt(ind);
    pc.bList.add(basketbolcu.get(ata));
    basketbolcu.remove(ata);
    ind--;
}
for (int i = 0; i < 4; i++) {
    kul.bList.add(basketbolcu.get(ind));
    basketbolcu.remove(ind);
    ind--;
}
}
```

Yukarıdaki döngülerimizin içeriğinin mantığı futbolcular için yaptığımızla aynıdır.

“Main” Metodu:

Bu metodumuzun içerisinde oyun kartlarımız ve oyunu oynayan kullanıcılarımız tanımlanmıştır. Oyun kartlarımızı futbolcu ve basketbolcu referans ismiyle ArrayList yapısı kullanılarak 2 ayrı liste ile tutulmuştur. Daha sonra belirteceğimiz kartDagit metodu ile bu listedeki elemanlarımızı oyuncularımıza 4’ er 4’er dağıtılacak ve Oyuncu sınıfını anlattığımız bölümde belirtildiği gibi oyuncuların sahip olduğu “fList” ve “bList” ArrayListlerinde tutulmuştur. “main” metodunun kod içeriği ve açıklaması aşağıdaki gibidir.

```
public static void main(String[] args) {
    Bilgisayar bilgisayar=new
    Bilgisayar(2, "Bilgisayar");
    Kullanici oyuncu=new Kullanici(1);
```

Yukarıda tanımlanan nesneler oyuncuları temsil etmektedir. "bilgisayar" nesnesi sahip olduğu parametrelili yapıcı metodlardan birine göre 3 parametre almaktadır. İlki oyuncu idsi ikicisi oyuncu ismidir. "oyuncu" objemiz ise sadece 1 parametre kullanılmaktadır sebebi ismi kullanıcının girdiği değere göre belirlenecektir.

```
ArrayList<Futbolcu> futbolcu=new
ArrayList<Futbolcu>();
ArrayList<Basketbolcu> basketbolcu=new
ArrayList<Basketbolcu>();
```

Yukarıdaki ArrayListler kartların tutulduğu listeleri temsil etmektedir.

```
Futbolcu F1 = new Futbolcu("Cristiano
Ronaldo", "Juventus FC", 95, 85, 90);
Futbolcu F2 = new Futbolcu("Lionel Messi",
"FC Barcelona", 95, 90, 85);
Futbolcu F3 = new Futbolcu("Antoine
Griezmann", "FC Barcelona", 85, 75, 80);
Futbolcu F4 = new Futbolcu("Sergio Ramos",
"Real Madrid CF", 85, 75, 85);
Futbolcu F5 = new Futbolcu("Paulo Dybala",
"Juventus FC", 80, 80, 85);
Futbolcu F6 = new Futbolcu("Robert
Lewandowski", "FC Bayern München", 90, 90,
80);
Futbolcu F7 = new Futbolcu("Kylian Mbappé",
"Paris Saint-Germain", 80, 75, 90);
Futbolcu F8 = new Futbolcu("Luis Suárez",
"Atlético Madrid", 90, 85, 80);
```

```
futbolcu.add(F1);
futbolcu.add(F2);
futbolcu.add(F3);
futbolcu.add(F4);
futbolcu.add(F5);
futbolcu.add(F6);
futbolcu.add(F7);
futbolcu.add(F8);
```

```
Basketbolcu B1 = new Basketbolcu("Giannis
Antetokounmpo", "Milwaukee Bucks", 95, 90,
95);
Basketbolcu B2 = new Basketbolcu("Kawhi
Leonard", "Los Angeles Clippers", 90, 80,
90);
Basketbolcu B3 = new Basketbolcu("LeBron
James", "Los Angeles Lakers", 100, 95, 95);
Basketbolcu B4 = new Basketbolcu("James
Harden", "Houston Rockets", 95, 95, 95);
Basketbolcu B5 = new Basketbolcu("Anthony
Davis", "Los Angeles Lakers", 90, 85, 85);
Basketbolcu B6 = new Basketbolcu("Nikola
Jokic", "Denver Nuggets", 85, 80, 85);
Basketbolcu B7 = new Basketbolcu("Joel
Embiid", "Philadelphia 76ers", 85, 85, 90);
Basketbolcu B8 = new Basketbolcu("Luka
Doncic", "Dallas Mavericks", 90, 85, 95);
```

```
basketbolcu.add(B1);
basketbolcu.add(B2);
basketbolcu.add(B3);
basketbolcu.add(B4);
basketbolcu.add(B5);
basketbolcu.add(B6);
basketbolcu.add(B7);
basketbolcu.add(B8);
```

```
kartDagit(bilgisayar, oyuncu, futbolcu, basket
bolcu); }
```

Yukarıdaki nesneler Sporcu sınıfının alt sınıfları olan "Futbolcu" ve "Basketbolcu" sınıfından türetilmiştir. Bu nesneler oyunculara dağıtılacak oyun kartlarını temsil etmektedir. Nesneler tanımlandıktan sonra yukarıda tanımladığımız listelere futbolcu.add(nesne ismi) ve basketbolcu.add(nesne ismi) şeklinde ekliyoruz.

“kartDagitButonuActionPerformed” Metodu:

Bu metodun içinde başta kartlar dağıtılmadığı için boş olan kart kutucuklarının içine önceden belirlenen sporcu kartlarını yazması için gerekli

kodlar bulunmaktadır. Tek seferlik bir kullanıma sahip olduğu için ve oyunumuz futbolcu turundan başladığı için sadece futbolcu kartlarını yazdırdık.

```
ArrayList<String> oyfutarraylist = new
ArrayList<String>();
for(var i : oyuncu.fList){

String icerik =
i.getSporcuIsim()+"\n"+i.getSporcuTakim()+"
\nPenalti: "
+i.getPenalti()+"\nSerbest atis:
"+i.getSerbestAtis()+"\nKaleciyle karsi
karsiya:"+
i.getKaleciKarsiKarsiya();
oyfutarraylist.add(icerik);

}

oyKart1.setText(oyfutarraylist.get(0));
oyKart2.setText(oyfutarraylist.get(1));
oyKart3.setText(oyfutarraylist.get(2));
oyKart4.setText(oyfutarraylist.get(3));

ArrayList<String> pcfutarraylist = new
ArrayList<String>();

for(var i: bilgisayar.fList){

String icerik =
i.getSporcuIsim()+"\n"+i.getSporcuTakim()+"
\nPenalti: "
+i.getPenalti()+"\nSerbest atis:
"+i.getSerbestAtis()+"\nKaleciyle karsi
karsiya:"+
i.getKaleciKarsiKarsiya();
pcfutarraylist.add(icerik);    }

pcKart1.setText(pcfutarraylist.get(0));
pcKart2.setText(pcfutarraylist.get(1));
pcKart3.setText(pcfutarraylist.get(2));
pcKart4.setText(pcfutarraylist.get(3));

kartDagitButonu.setVisible(false);
oynaButonu.setVisible(true);    }
```

“oynaButonuActionPerformed” Metodu:

Bu metodumuzda karta tıklanıp oyna butonuna basılınca gerçekleşecek olan olayları kodladık. Tura göre oynanan kartları görünmez yaptık, kartların içeriğini değiştirdik ve skora puan ekledik. Kod içeriği çok uzun olduğu için kodu buraya yazmayı uygun görmedik. Yalancı kod kısmında metodun çalışma mantığı anlatılacaktır.

“oyKartMouseClicked” Metodu:

Bu metodumuzda ise oyuncunun kendisine sunulan 4 karttan birine tıkladığı zaman gerçekleşecek olayları kodladık. Kısaca özetlemek gerekirse tıklanan değeri ile hangi kartın

oynandığını tutup oyun ekranının ortasında önceki turdan kalan karşılaştırma değerleri yazılarını sildik. Aynı zamanda tıkladığı zaman kartı görünmez yapıp, ekranın ortasına oynanan kartın değerlerini yazdırdık. Mevcut olan 4 kart için de aynı kodlama mantığını kullandık.

```
tiklanan=0;
oyKart1.setVisible(false);
oynanankartisimpc.setText("****");
oynanankarttakimpc.setText("****");
ucuncupuandegeripc.setText("****");
dorduncupuandegeripc.setText("****");
besincipuandegeripc.setText("****");
ikinciucuncuyazi.setText("Pozisyon 1: ");
ikincidorduncuyazi.setText("Pozisyon 2:");
ikincibesinciyazi.setText("Pozisyon 3:");

if(say%2==0){
oynanankartisimoy.setText(oyuncu.fList.get(
0).getSporcuIsim());
oynanankarttakimoy.setText(oyuncu.fList.get(
0).getSporcuTakim());
ucuncupuandegerioy.setText(""+oyuncu.fList.
get(0).getPenalti());
dorduncupuandegerioy.setText(""+oyuncu.fList
t.get(0).getSerbestAtis());
besincipuandegerioy.setText(""+oyuncu.fList
.get(0).getKaleciKarsiKarsiya());
ucuncuyazi.setText("Penalti:");
dorduncuyazi.setText("Serbest Atis:");
besinciyazi.setText("Kaleci ile karsi
karsiya:");

}

else{
oynanankartisimoy.setText(oyuncu.bList.get(
0).getSporcuIsim());

oynanankarttakimoy.setText(oyuncu.bList.get(
0).getSporcuTakim());

ucuncupuandegerioy.setText(""+oyuncu.bList.
get(0).getIkilik());

dorduncupuandegerioy.setText(""+oyuncu.bList
t.get(0).getUcluk());

besincipuandegerioy.setText(""+oyuncu.bList
.get(0).getSerbestAtis());

ucuncuyazi.setText("Ikilik:");
dorduncuyazi.setText("Ucluk:");
besinciyazi.setText("Serbest Atis:");

}

}
```

Test Sınıfı Yalancı Kod:

```
public class Test extends JFrame{

static ArrayList futbolcu;
static ArrayList basketbolcu;
static Bilgisayar(2,"Bilgisayar",0);
static Kullanici(1); static int
tiklanan=4;
```



```

static void
karsilastir(Bilgisayar,oyuncu,int sira,int
kartidpc,kartidkul){

Random r = new Random();

if(sira%2==0){

Futbolcu fpc=pc.fList'ten kartid al

Futbolcu fkul=kul.fList'ten kart id al

Kartdegiskenleri.setText ekranin ortasına;

if(pozisyon=0){

setText("kaleci ile karsi karsiya");

if(pc kkk>oy kkk){

pc.setSkor(pc.getSkor+10);

remove(kartidkul ve kartidpc);

else{

kul.setSkor(kul.getSkor+10);

remove(kartidkul ve kartidpc);} }

Bütün pozisyonlar için aynı kodu uyarla;

}

public static void main(){

program çalıştığında(){

oynabutonunu.setvisible false;

kazanantext.setvisible false;}

Futbolcu ve basketbolcu kartlarını tanımla;

kartDagit(bilgisayar,oyuncu,futbolcu,basket
bolcu); }

private int say=0;//tur sayısını tutan
değişken

private void kartdagitbutonutiklandi{

ArrayList oyfutarraylist;

for(var i:oyuncu.flist){

String içerik = i.getbutunkartdegerleri;

Oyfutarraylist.add(içerik);

}

oyKart1-2-3-4.setText(oyfutarraylist.get(0-
1-2-3));

Aynı yöntemi pc futbolcu, kul basketbolcu
ve pc futbolcu kartları için de yaz;

Kartdagitbutonu.setvisible(false);
Oynabutonunu.setvisible(true); }

private void oynaButonuTiklandi(){

if(say%2==0){

if(tıklanan==0){

```

```

karsilastir(bilgisayar, oyuncu, say,
bilgisayar.kartSec(say) ,0); }

else if(tıklanan==1){

karsilastir(bilgisayar, oyuncu, say,
bilgisayar.kartSec(say), 1); }

else if(tıklanan==2) "

else if (tıklanan==3) "

if(!(oyuncu.bList.isEmpty())){

ArrayList oybasarraylist;

For(var i:oyuncu.blist){

String
kartloybas=i.getbutunsporcodegerleri;
oybasarraylist.add(kartloybas);

}

if(oybasarraylist.size()==4){

oyKart1.setText(oybasarraylist.get(0));
oyKart2.setText(oybasarraylist.get(1));
oyKart3.setText(oybasarraylist.get(2));
oyKart4.setText(oybasarraylist.get(3));}

else if(oybasarraylist.size()==3){

oyKart1.setText(oybasarraylist.get(0));
oyKart2.setText(oybasarraylist.get(1));
oyKart3.setText(oybasarraylist.get(2));}

else if(oybasarraylist.size()==2){

yukarıdaki kod parçacıklarını uyarla;}

else if(oybasarraylist.size()==1){

yukarıdaki kod parçacıklarını uyarla;}

if(){oyuncu.bList size boyutu ne kadarsa o
kadar indekse kadar kartı görünür yap;}

ArrayList<String> pcbasarraylist = new
ArrayList<String>();

for(var i : bilgisayar.bList){

String kartlpcbas =i.getsporcodegerleri;

pcbasarraylist.add(kartlpcbas);}

if(pcbasarraylist size'ı kadar kartlara
pcbasarraylist.get(indeks) ile text ekle;}

if(){bilgisayar.bList size boyutu ne
kadarsa o kadar indekse kadar kartı görünür
yap;}

say++;}

else{

yukarıdaki futbolcu kartları için yazılan
kodların aynısını basketbol kartları için
uyarla;

say++;}

pcPuan.setText("Bilgisayar
Puani:"+bilgisayar.getSkor());

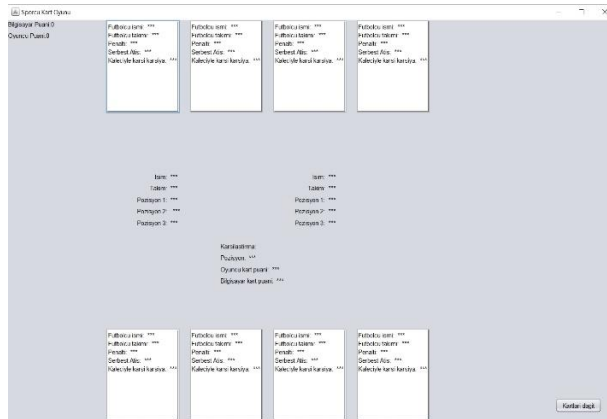
```

```
oyuncuPuan.setText("Oyuncu  
Puanı:"+oyuncu.getSkor());
```

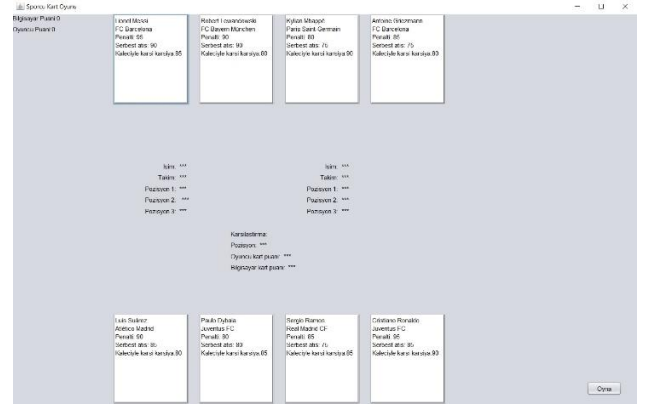
```
if(oyuncu.blist ve oyuncu.flist=empty){  
kazanantext.setVisible(true);  
  
if(oyuncu skor>pc skor){  
kazanantext.setText("KAZANDINIZ!");}  
  
else if(oyuncu skor==pc skor){  
kazanantext.setText("BERABERE");}  
  
else{  
kazanantext.setText("KAYBETTINIZ");} } }
```

```
oyanananKartMouseTiklandi(){  
  
tıklanan=tıklanan kart indeksi;  
oyananankart.setVisible(false);  
  
setText ile ekran ortasındaki önceki karta  
ait değerleri sil;  
  
if(say%2==0){  
oyananankartdegerleri.setText(oyuncu.flist.g  
et().sporcuDeğerleri);}  
  
else{  
oyananankartdegerleri.setText(oyuncu.blist.g  
et().sporcuDeğerleri);}
```

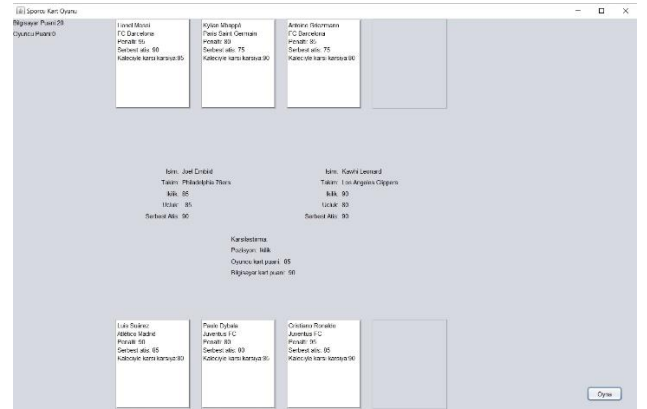
Deneysel Sonuçlar:



Oyunun açılış ekranı



Kartların dağıtılmış hali

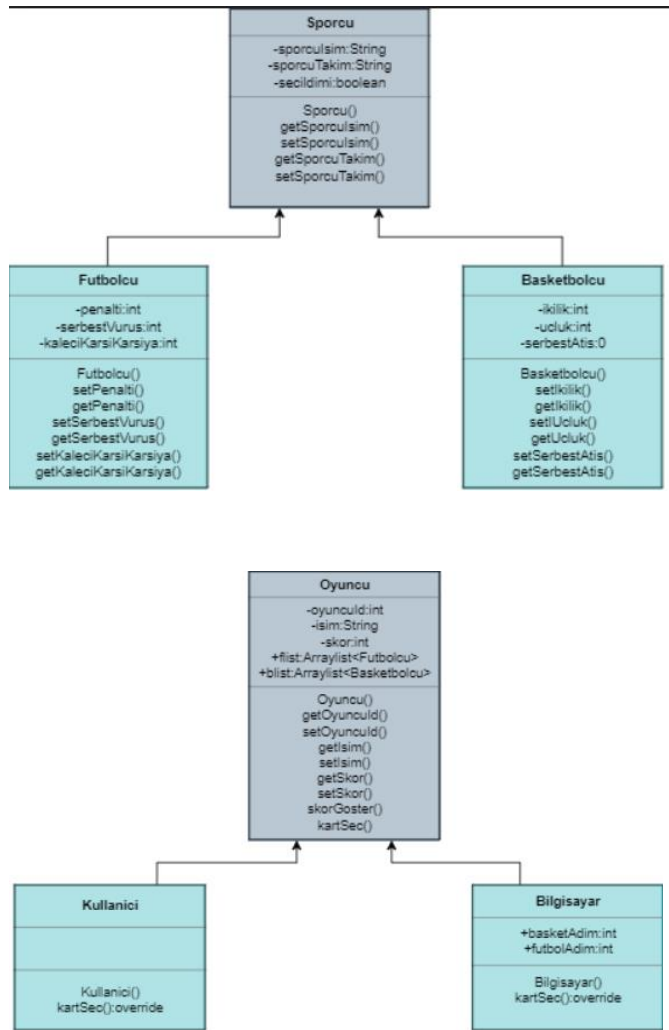


Kartların oynanmış hali



Oyunun bitmiş hali

UML Diyagramı:



Kaynakça:

<http://edestek2.kocaeli.edu.tr/>

<https://www.udemy.com/course/sifirdan-ileri-seviyeye-komple-java-gelistirici-kursu>