

Athlete Card Game

Berkay Yasin ifti
190202056@kocaeli.edu.tr

Ferhat Aıkalin
180202065@kocaeli.edu.tr

Projenin zeti:

Programlama Laboratuvarı I Projesi olarak bizden ‘‘Sporcu Kart Oyunu’’ adındaki bir kart oyunu olan ve bir masaüstü arayüz tasarımı içeren bir uygulama yapmamız beklenmektedir.

Biz proje için Java programlama dilini ve Netbeans geliştirme ortamını seçtik.

Proje dokümanında sınıflara göre deęişen isterleri bulunmaktadır. Tüm sınıflar için ise 3 adet ortak özellik kullanmamızı istediler..

Projede biz Java programlama dilinde bulunan ‘‘LibGDX’’ adlı tasarım kütüphanesinden yararlandık. Bu kütüphane sayesinde oyunu ve oyuncuların ellerinde bulunan kartların görülebileceęi ve takip edileceęi bir arayüz tasarımı gerçekleştirilmiştir.

İlk olarak classları tanımladık. Classlar için gerekli olan özellikleri , parametrelili ve parametresiz olmak üzere her class için en az iki adet constructor tanımladık. Bazı classların için gerekli olan kalıtımı yaptık. FutbolcuSınıfı ve BasketbolcuSınıfı için SporcuSınıfından kalıtım yapıldı. BilgisayarSınıfı ve KullanıcıSınıfı için OyuncuSınıfından kalıtım yapılmıştır.

Projede sporcuPuaniGoster() methodu ile kartSec() methodu gerekli classlara override edilmiştir.

Projemizde Encapsulation, Inheritance, Polymorphism, Abstraction yapılarından gerekli olanlar kullanılmıştır.

1.GİRİŞ

Proje için Java programlama dili ve Netbeans geliştirme ortamını kullandık.

Java programlama dili; açık kodlu, nesneye yönelik, zeminden bağımsız, yüksek verimli, çok işlevli, yüksek seviye, adım adım işlenen (yorumlanan- interpreted) bir dildir.

Netbeans platformu; Oracle tarafından geliştirilen bir java geliştirme ortamıdır (IDE) ve ücretsiz olarak dağıtılmaktadır. Özellikle kullanıcı arayüzü tasarımında sağladığı kolaylıklardan dolayı tercih edilmektedir. Bizim de Netbeans’i tercih etme sebeplerimizin başında kullanıcıya sağladığı kolaylıklar yatmaktadır.

2. TEMEL

BİLGİLER Yöntem :

Projemizde BasketbolcuSinifi.java , BilgisayarSinifi.java , FutbolcuSinifi.java , Kart.java , KullaniciSinifi.java , TestSinifi.java , OyuncuSinifi.java,SporcuSinifi.java,DesktopLauncher.java olarak toplam dokuz adet java sınıfından oluşmaktadır. Kart.java , Test.java , DesktopLauncher.java arayüz tasarımı ve futbolcu , basketbolcu isimlerinin ve özelliklerinin atanması için kullanılmıştır.

Classları oluşturduktan sonra işlevleri hakkında bilgi sahibi olmak istersek;

SporcuSinifi.java:

Bu Sınıfta sporcular için gerekli ad ve tip gibi deęişkenler olup dięer isterler bulunmaktadır.

String tipinde sporcuAdi ve sporcuTipi değişkenlerimiz bulunmaktadır . Bu değişkenler sporcularımızın adlarını ve tiplerini (futbolcu ya da basketbolcu) olduğunu göstermek için yazılmıştır. İki adet constuctorımız bulunuyor , ilki parametresiz diğeri parametre alıyor. Bu parametreler String Tipinde olup isimleri sporcuAdi ve sporcutipi gibidir. Kullanıcıdan alınan parametreleri classımızda ilk başta tanımladığımız değişkenlere atanmaktadır. sporcuPuaniGoster() methodumuz bulunmaktadır. Bu methodumuz gerçekleşen pozisyon sonrasında bilgisayar ve kullanıcının puanları göstermek için kullanılıyor . Classımızın ilk satırlarında tanımladığımız sporcuAdi ve sporcuTipi değişkenlerin set ve get methodları bulunmaktadır . Set methodları parametre almaktadır. Aldıkları parametreleri classımızdaki ilk satırlarda tanımladığımız değişkenlere atanmaktadır..

FutbolcuSinifi.java:

Bu sınıfta Futbolcular için açılmış olup onların özelliklerini ve gerekli istekleri bulunmaktadır. FutbolcuSinifi classımız SporcuSinifinden özellikleri miras almamız için extends yazılarak miras alınmış olur . Ve SporcuSinifin özelliklerine ve methodlarına erişim imkanımız oluşmuştur. Penaltı, serbestAtis,kaleciKarsiKarsiya adlı int tipinde değişkenlerimiz ve String tipinde futbolcuAdi , FutbolcuTakim değişkenlerimiz bulunmaktadır . Üç adet constuctorımız bulunuyor bunlardan sadece biri parametre almamaktadır. Parametrelili olan constuctorlardan ilki String türünde olan sporcuAdi ve sporcuTipi değişkenleri süper yapıyla (Miras özelliği) parametre olarak bunları sırasıyla futbolcuAdi ve futbolcuTakimi adlı String değişkenlerine aktarmaktadır . Son constuctor ise penaltı , serbestAtis, kaleciKarsiKarsiya adlarında int türünde değişkenleri olarak bunları classımızda ilk başta tanıttığımız değişkenlere atamaktadır.sporcuPuaniGoster() methodu SporcuSinifinden override edilmiştir. Boolean tipinde kartKullanildiMi() adlı methodumuz bulunmaktadır bu method basketbolcu kartlarının kullanılıp kullanılmadığını kontrol etmektedir eğer kullanılmadıysa true kullanıldıysa false döndürmektedir. Classımızın ilk satırlarında tanımladığımız İkilik, ucluk, serbestAtis , basketbolcuAdi, basketbolcuTakim değişkenlerin set ve get methodları bulunmaktadır. Set methodları parametre almaktadır ve aldıkları parametreleri classımızdaki ilk satırlarda

kullanılmadıysa true kullanıldıysa false döndürmektedir. Classımızın ilk satırlarında tanımladığımız penaltı , serbestAtis , kaleciKarsiKarsiya , futbolcuAdi , futbolcuTakimi değişkenlerin set ve get methodları bulunmaktadır. Set methodları parametre almaktadır ve aldıkları parametreleri classımızdaki ilk satırlarda tanımladığımız uygun değişkenlere atanmaktadır.

BasketbolcuSinifi.java:

Bu sınıfta Basketbolcular için açılmış olup onların özelliklerini ve gerekli istekleri bulunmaktadır.

BasketbolcuSinifi classımız SporcuSinifinden özellikleri miras almamız için extends yazılarak miras alınmış olur . Ve SporcuSinifin özelliklerine ve methodlarına erişim imkanımız oluşmuştur.. İkilik , ucluk , serbestAtis adlı int tipinde değişkenlerimiz ve String tipinde basketbolcuAdi , basketbolcuTakim değişkenlerimiz bulunmaktadır. . Üç adet constuctorımız bulunuyor bunlardan sadece biri parametre almamaktadır. Parametrelili olan constuctorlardan ilki String türünde olan sporcuAdi ve sporcuTipi değişkenleri süper yapıyla (Miras özelliği) parametre olarak bunları sırasıyla basketbolcuAdi ve basketbolcuTakim adlı String değişkenlerine aktarmaktadır . Son constuctor ise İkilik, ucluk, serbestAtis adlarında int türünde değişkenleri olarak bunları classımızda ilk başta tanıttığımız değişkenlere atamaktadır. sporcuPuaniGoster() methodu SporcuSinifinden override edilmiştir. Boolean tipinde kartKullanildiMi() adlı methodumuz bulunmaktadır bu method basketbolcu kartlarının kullanılıp kullanılmadığını kontrol etmektedir eğer kullanılmadıysa true kullanıldıysa false döndürmektedir. Classımızın ilk satırlarında tanımladığımız İkilik, ucluk, serbestAtis , basketbolcuAdi, basketbolcuTakim değişkenlerin set ve get methodları bulunmaktadır. Set methodları parametre almaktadır ve aldıkları parametreleri classımızdaki ilk satırlarda

tanımladığımız uygun değişkenlere atanmaktadır.

OyuncuSinifi.java:

Bu sınıfta kullanıcı ve bilgisayar için aynı olan özellikler ve isterler için açılmıştır.

oyuncuID , skor adında int türünde , String türünde adı oyuncuAdi olan değişkenlerimiz mevcuttur.Bu değişkenler oyuncuların adlarını , skorlarını ve id lerini tutmak için yazılmıştır. Constuctorlarımız bulunuyor , birincisi parametresiz diğeri parametrelili olmak üzere iki adet constuctorımız bulunuyor. Parametrelili olan constuctorımız int tipinde oyuncu id ve skor özellikleri yanında oyuncunun adını da alarak oyuncuSinifi classındaki değişkenlere atanmaktadır. kartSec() ve SkorGoster() methodlarımız bulunmaktadır.kartsec() fonksiyonu bilgisayar ve kullanıcı için farklı yazılmıştır. Ve bu fonksiyon bilgisayar ve kullanıcı classlarına override edilmiştir. kartListesi () fonksiyonu ile oyuncuların ellerindeki kartlar tutulmuştur. SkorGoster() methodu ise kullanıcı ve bilgisayarın skorlarını göstermektedir . Classımızın başında tanımladığımız değişkenler için get ve set methodlarını da classımızın sonuna eklenmiştir.

BilgisayarSinifi.java:

Bilgisayar için gerekli isterler ve özellikler için açılmıştır.

BilgisayarSinifi classımız OyuncuSinifinden özellikleri miras almamız için extends yazılarak miras alınmış olur . Ve OyuncuSinifin özelliklerine ve methodlarına erişim imkanımız oluşmuştur.futbolcularım ve basketbolcularım adlı tipi FutbolcuSinif ve BasketbolcuSinif olan arraylistler tanımlanır. Adı secilenFutbolcu olan tipi FutbolcuSinif nesnesi olan değişken ve adı secilenBasketbolcu olan BasketbolcuSinif nesnesi tipinde değişkenler tanımlanmıştır. İki adet constuctorımız vardır. Biri parametrelili diğeri parametre almamaktadır. Parametrelili olan constuctorın parametreleri oyuncunun id ve skor olan int tipinde ve String tipinde oyuncuAdi değişkenleri

süper yapısıyla (Miras Özelliği) parametre alınmıştır. kartsec()methodu override edilmiştir. kartSec() methodu içinde random bir şekilde kartları bilgisayara dağıtılmaktadır.

KullaniciSinifi.java:

Kullanıcı için gerekli isterler ve özellikler için kullanıcıSinifi açılmıştır.

kullaniciSinifi classımız OyuncuSinifinden özellikleri miras almamız için extends yazılarak miras alınmış olur . Ve OyuncuSinifin özelliklerine ve methodlarına erişim imkanımız oluşmuştur. İki adet constuctorımız vardır. Biri parametrelili diğeri parametre almamaktadır. Parametrelili olan constuctorın parametreleri oyuncunun id ve skor olan int tipinde ve String tipinde oyuncuAdi değişkenleri süper yapısıyla (Miras Özelliği) parametre alınmıştır. kartsec()methodu override edilmiştir. kartSec() methodu içinde kullanıcının istediği bir şekilde kartları kullanıcıya dağıtılmaktadır.

TestSinifi.java:

Futbolcu ve basketbolcuların isimleri ve özellikleri burada tanımlanmıştır. Pozisyon bilgisi içeren method bu classımızdadır. Kartlar seçildikten sonra hangi pozisyonun seçilmesi (rastgele)için yazılmış classtır.

kartlaridagit() adlı method içinde futbolcularımızın sırasıyla tipi , adi , kaleciKarsiKarsiya , penaltı , serbestAtis değerleri 2 Boyutlu String dizisine aktarılmıştır. Yine aynı şekilde basketbolcularımız için aynı şekilde sırasıyla tipi , adi , ikilik , üçlük , serbestAtis özelliklerinin değerlerini 2 boyutlu String dizisinin içine atılmıştır.Oluşturulan dizilerdeki değerlerden nesne oluşturmak için futbolcu ve basketbolcular için ayrı ayrı 2 for içinde set ve get methodlarını kullanarak sporcu Tipi ,SporcuAdi , penaltı , ikilik , kaleciKarsiKarsiya , SerbestAtis ,üçlük gibi özellikleri oluşturulan nesnelere eklenmesi sağlandı . Kullanıcının kartları seçmesi için iki adet for yazıldı birinci for

da kartlar array liste atandıktan sonra kullanıcı için kartlar seçildikten sonra geriye kalanlar bilgisayara dağıtıldı. Ayrıca bu class oyunun ana döngüsün çalıştığı yerdir. LibGDX paketinden extend edilen "create()" metodu ilk başta çalışır ve arayüz için gerekli hazırlıkları yapar.

"render()" metodu oyun kapatılana kadar sürekli çalışır bu döngü içerisinde hamleler, kullanıcıların kart seçip seçmediği kontrol edilir. Yine bu metot içerisinde puanlama yapılır. En sonunda oyunun bitip bitmediği burada belirlenir.

Kart.java:

Grafik kütüphanesini kullanarak oyununun ara yüzüne oyuncuların isimlerinin gerekli değişkenlere atanması için yazılmış classtır. Parametre olarak ekrana çizileceği konumu, çizileceği kartın referansını ve ait olduğu kullanıcının referansını alır. Oyun sırasında kartın konumunu, seçilebilirliğini ve seçildiğinde ise tekrar seçilmesini engellemeyi sağlar.

DesktopLauncher.java:

Bu classımız oyunun açılacağı pencerenin özelliklerinin bulunması için yazılmış classtır . LibGDX kütüphanesi farklı platformalarda çalışabildiği için her platform için ayrı ayarların yapılması gerekiyor.

3. Yalancı Kod :

1)LibGDX paketinde bulunan ApplicationAdapter sınıfından extend edilen TestSinifi Sınıfında ilk olarak override edilen "create()" metodu çalışır.

2)"BilgisayarSinifi" ve "KullaniciSinifi" sınıflarından oluşturulan "bilgisayar" ve "kullanici" nesnelere gerekli özellikler tanımlanır.

3)Arayüz için gerekli olan nesneler oluşturulur.

4)"BasketbolcuSinifi" ve "FutbolcuSinif" tipindeki "futbolcular" , "basketbolcular" ,"kullanici_futbolcular", "kullanici_basketbolcular" ArrayListi oluşturulur.

5)Oyun durumunu belirleyen state değişkenleri oluşturulur.

6)"kartlaridagit()" metodu çalıştırılır.

7)kartlaridagit() metodu futbolcu ve basketbolcu kartlarını rastgele olarak bilgisayar oyuncusu için "futbolcular" ve "basketbolcular", kullanıcı için "kullanici_futbolcular", "kullanici_basketbolcular" ArrayList lerine dağıtılır.

8) Kartları ekrana eklemek için "stage" nesnesi oluşturulur.

9)"ekranaCiz()" metodu çağrılır. Herbir kart için "Kart" sınıfından nesne oluşturulur. Bu nesne ekrana çizileceği

koordinatı,çizileceği kartın referansını, ait

olduğu kullanıcının referansını parametre

olarak alır. Sonra bu nesne "stage" içine

eklenir."Kart" sınıfı parametre aldığı

değerlere göre constructora sahiptir. "Kart"

sınıfında "draw()" metodu kartı ekrana

çizdirir, bilgilerini gösterir. "act()" metodu

kart oyuna sürülünce ekrandaki konumunu

değiştirir ya da seçilebilirliğini kontrol eder.

Seçme sırası oyuncudaysa ve seçilmesi

gerekten kart türü kendi türüyle aynıysa

seçilebilir yapar. "draw()" metodu kart

kullanıldıysa rengini değiştirir ve seçilemez

yapar.

10) "create()" metodundan sonra "render()"

metodu çalışır. Bu metot program bitene

kadar sürekli çalışır. İçerisinde oyun

döngüsü gerçekleştirilir.

11)"render()" içerisinde "skorciz()" metodu

ekrana skorları çizer.

12)"pozisyonCiz()" ekrana karşılaştırma

yapılan pozisyonu yazdırır.

13)"gameOverCheck()" oyunun sonun

gelip gelemediğini kontrol eder. Oyun sona

erdiyse ekrana kazanın kim olduğunu

yazdırır.

14)Kullanıcı kart seçmediyse

"kartSecUyari()" metodu çağrılır.

15)Kullanıcı kart seçtiyse ve bilgisayar kart

seçmediyse, "bilgisayar.kartSec()" ile

bilgisayar için rastgele oynanacak kart

seçilir.

16)Bilgisayar ve kullanıcı kart seçtiyse

hesaplama işlemine geçilir. Sıra futbolcu

kartındaysa"futbolcuVfutbolcu()",

basketbolcudaysa

"basketbolcuVbasketbolcu()" metodu çağrılır. Bu metodların her ikisinde rastgele bir pozisyona göre kullanıcının ve bilgisayarın seçtiği kartları karşılaştırır. Hangisinin değeri büyükse ona +10 puan verir.

17)Oyunun sonu olup olmadığı kontrol edilir. Oyun sonuysa ekrana "Oyun Bitti" ve kazanan oyuncu yazılır.

18)Oyun sonuysa ve beraberlik durumu varsa son kart tekrar farklı bir pozisyona göre eşleştirilir ve tekrar

"futbolcuVfutbolcu()",

"basketbolcuVbasketbolcu()"

metotlarından gerekli olan çağrılır. Tekrar puan verilir.

19)Oyun sonu değilse 10. adıma tekrar dönerek oyun döngüsü devam ettirilir.

4.Deneysel Sonuçlar :



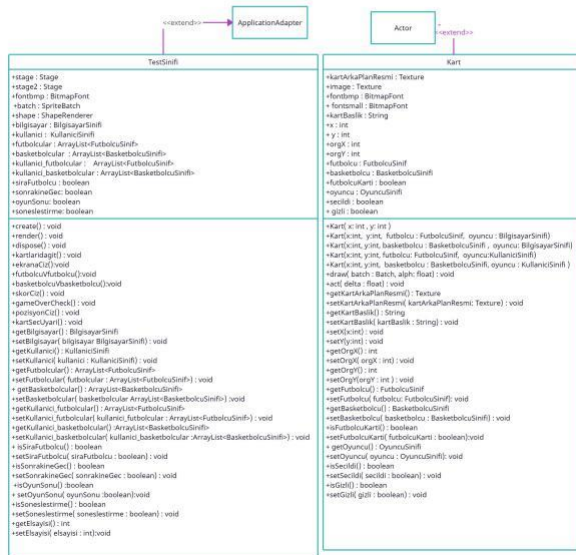
5.Karşılaşılan Sorunlar :

Arayüzü hangi kütüphaneyi kullanarak tasarlayacağımız hakkında karar veremedik . En son Java programlama dilinde bulunan “LibGDX” adlı tasarım kütüphanesinden yararlandık. Inheritance (kalıtım) konusunda pek bilgi sahibi değildik yaptığımız araştırmalar ve video eğitimleri sonucunda projemizin kalıtım için çok uygun olduğunu öğrendik. Nesneler arası kart seçme durumlarında ve bunu görsel kütüphaneyi kullanarak aktarma konusunda sıkıntı yaşadık.

6.Sonuç :

Bu proje sayesinde java programlama dilini kullanarak nesneye yönelik programlama mantığını (kalıtım , classlar , nesneler , methodlar)ve grafik kütüphanesi kullanımını konusunda bol bol bol bilgi sahibi olduk , en sonunda da bir oyunun ne kadar zorluklarla yapıldığını bu projede benimsemiş olduk. Teşekkür Ederiz.

7.UML Sınıf Diyagramı :



7. Kaynakça :

<https://medium.com/gokhanyavas/java-oop-miras-alma-inheritance-6-95b0958f7dec>

<https://yazdoldur.com/programlama/java/java-kalitim-inheritance-nedir/>

<https://www.mobilhanem.com/java-inheritance/?cfchljschltk=3316fbbaf150e856322a7be4b44cafa9b61c068->

16073711570AeJsTnggNeNt3jZp2OkQ87GE6Dde8Y
QQXDfFSjOZqW6aEGregchnFbY1cHhJlBESNrLz
JROqQpVe3OSX 6buVMWwlPpLutdL2wW_SFm
ury5U1jLFxWxfiysLSNqlZONiaMbV9Xi

<https://github.com/libgdx/libgdx/wiki>

