

PROGRAMLAMA LABORATUVARI KART OYUNU

İrem ÇELİKKANAT
Kocaeli Üniversitesi
Mühendislik Fakültesi
Bilgisayar Mühendisliği(İÖ)
190202124@kocaeli.edu.tr

Eliz KURTULUŞ
Kocaeli Üniversitesi
Mühendislik Fakültesi
Bilgisayar Mühendisliği(İÖ)
190202015@kocaeli.edu.tr

Özet- Bu projede nesneye yönelik programlama ve arayüz programlama ile basit bir kart oyunu gerçekleştirdik. Oyun, basketbolcu ve tenisçi kartlarıyla oynanan basit bir oyundur. Oyunu; çeşitli veri yapılarını ve kalıtım, kapsülleme gibi nesneye yönelik programlamanın temel kavramlarını kullanarak tamamladık.

Anahtar kelimeler-Nesne,sınıf,kalıtım,java,swing

I. GİRİŞ

Proje, mantık açısından nesneye yönelik programlama temellerine daha çok uymaktadır. Bu bakımdan projeyi oluşturmak için Java Programlama Dili belirlenmiştir. Java; Java geliştiricileri, mimarları ve gönüllülerden oluşan özel bir topluluk tarafından test edilmiştir, iyileştirilmiştir, genişletilmiştir ve onaylanmıştır. Java, bilgisayar platformları için mümkün olan en geniş kapsamda taşınabilir, yüksek performanslı uygulamaların geliştirilmesine olanak sağlayacak şekilde tasarlanmıştır. Projeyi gerçekleştirebilmemiz için de bir arayüze ihtiyacımız vardı. Arayüz ortamını; Java'nın dahili grafik kütüphanelerinden olan swing seçerek belirledik. Javax.Swing paketi ;134 sınıfı olan, 25 arayüz algılayan büyük bir pakettir. Java ile Grafikselsel Arayüzler (GUI) hazırlamak için gerekli bütün araçlara sahiptir.

Kart oyunu projesi kullanıcı tarafından çalıştırılmaktadır. Oluşturulan "start" butonuna basıldığında oyun ekranı karşımıza gelmektedir. Oyun içerisindeki farklı nesneler için çeşitli sınıflar tanımlanmıştır. Oluşturulan sınıfların yapısı ve özellikleri bizlere daha önceden belirtilmiştir. İstenen methodlar düzenlenerek ve ekleme yapılarak projenin yapısı oluşturulmuştur. Oyundaki kartların ArrayList yapısında tutulmaktadır. Kullanırken sağladığı kolaylıklar sebebiyle ArrayList yapısı tercih edilmiştir. Oyun, bilgisayar ve kullanıcı arasında oynanmaktadır. Her iki tarafa da kartların dağıtılmasıyla oyun başlamaktadır. İlk olarak kullanıcı istediği bir kartı atar ve

bilgisayardan gelen aynı kategoriden bir kartla karşılaştırılır. Dağıtılan kartlar bittiğinde oyun da son bulmaktadır. En yüksek puana sahip oyuncu kazanmış kabul edilmektedir.

II. YÖNTEMLER ve TASARIM

Bu kısımda kart oyununu oluşturmak için kullandığımız yöntemler üzerinde durulacaktır. Oyunun tasarımıyla ilgili detaylı bilgi verilecektir.

A. Arayüz Tasarımı

Projede oyunu ekranını tasarlamak için Java2'nın grafik kütüphanelerinden swing(GUI)'yi tercih ettik. Swing,Java programları için grafik kullanıcı arabirimini sağlayan bir API olan Oracle'ın Java Temel Sınıflarının bir parçasıdır. Oyun oluşturulurken bir adet JFrame nesnesi içerisine JPanel nesnesini extend eden Test sınıfını ekleyerek temel pencereyi oluşturduk. Test sınıfı, oyunun tüm işleyişini gerçekleştirdiği gibi bütün çizim fonksiyonlarını da içerisinde barındırıyor. Kartlarımızı birer buton halinde tasarladık. Kartların oluşturulması için swing kütüphanesindeki JBUTTON 'u kalıtım alan kartT ve kartBasketbol sınıfları kullanıldı. Bu sınıflardan elde edilen kart nesneleri kartlarB ve kartlarT ArrayListlerine eklenerek oyuna dahil edildi. Aynı zamanda bu kart nesnelerine ActionListener eklenerek kartlar oynanabilir hale getirildi.

B. Kullanılan Sınıf ve Nesne Yapısı

Program, Java programlama dilinde gerçekleştirilmiş olup geliştirme ortamı olarak Eclipse kullanılmıştır. Java, nesneye yönelik programlama dili olduğundan bu bölümde projede kullanılan class ve objelerden detaylı olarak bahsedilecektir. Öncelikle oyunun ana paneli olan ve birçok farklı temel fonksiyonların kullanıldığı bir Test sınıfı

bulunmaktadır. Test sınıfında bu sınıftan bir obje oluşturulmamıştır. Test sınıfı, diğer sınıflardan oluşturulan nesneleri ArrayListlerde tutmaktadır. Görevi ve projedeki diğer sınıflardan farkı, oyunu oluşturan nesneleri içerisinde tutarak oyunun işleyişini sağlamasıdır. Bu sınıf main methodunu da kapsamaktadır. Bunun yanı sıra oyuna giriş sınıfından da başlanılabilir. giriş sınıfında JFrame'den extends edilerek panel oluşturulmuştur. Bu panel üzerinde oyuncudan ismi istenmektedir. Oyuncu adını JTextField'a yazarak "start" butonuna basmasıyla oyun ekranı görünür hale gelmektedir.



Oyun giriş menüsü

Projede kullanılan temel sınıflardan birisi de Sporcu sınıfıdır. Sporcunun ismini, takımını ve sporcuid'sini tutmaktadır. Basketbolcu ve Tenisci sınıfları Sporcu sınıfından kalıtım alınarak oluşturulmuşlardır. Basketbolcu sınıfı; ikilik, üçlük ve serbestAtis özelliklerini barındırmaktadır. Tenisci sınıfı ise forhand, backhand ve servis özelliklerini tutmaktadır. Oluşturulan bir diğer önemli sınıf ise Oyuncu sınıfıdır. Oyuncu sınıfı bir abstract class'dır. Oyunculd, oyuncuAdi ve Skor özelliklerini tutmaktadır. Bu sınıftan extends edilen Kullanici ve Bilgisayar sınıfları oluşturulmuştur. JBUTTON 'dan extends edilen kartT ve kartBasketbol sınıfları bilgisayar kartlarının ekranda karşımıza gelebilmesi için kullanıldı. kartBasketbol sınıfı ad, ikilik, ucluk ve serbestAtis özelliklerini tutmaktadır. kartT sınıfı ise forhand, backhand ve servis özelliklerini barındırmaktadır. Projemizde oluşturulan sınıfların tamamı şunlardır: Test, Sporcu, Basketbolcu, Tenisci, Oyuncu, Kullanici, Bilgisayar, giriş, kartBasketbol, kartT sınıflarıdır. Sınıflara ait 'UML Sınıf Diyagramı' aşağıda verilmiştir.

C. Proje Mantığı ve Kullanılan Methodlar

Projemizde swing elemanlarını kullanarak arayüz ile oyun işleyişini sağladık. İlk olarak Tenisci ve Basketbolcu sınıflarından tanımladığımız kartları teniscideste ve basketcideste ArrayListlerin içine ekledik. Random sayı kullanarak while() döngü fonksiyonu ile oyuncuT, oyuncuB ve pcT,pcB arraylistlerine dağıttık. Oluşturduğumuz JBUTTON'dan kalıtım alan kartT ve kartBasketbol sınıflarından arraylistler tanımladık. Bilgisayar kartlarını bu sınıflardan oluşturarak arraylistlerine ekledik. Bu kartların setBounds() methoduyla ekrandaki yerlerini belirledik. Böylece eklediğimiz eventlerle kartları oyuna dahil ettik. Oyuncu kartlarını ise JLabel eventi ile oluşturarak oyun paneline ekledik.

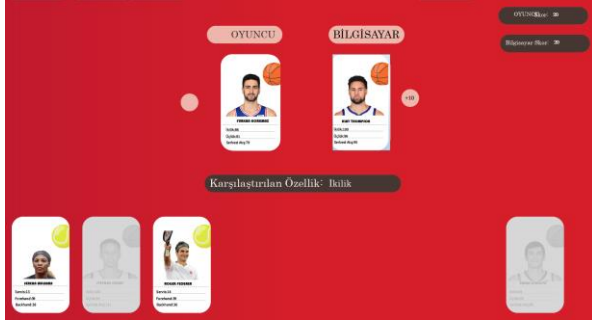


Oyun başlangıcı

Oyuncunun seçtiği kart ile bilgisayardan atanan kartı, random belirlenen özelliğe bakılarak if else yapılarıyla kıyasladık. Kimin puan alacağını belirleyerek puanları panele yazdırdık. Oyun esnasında, örneğin basketbolcu kartlarından sonra tenisçi kartlarının oyunu devam ettirilebilmesi için setVisible() methoduyla karşılaştırılan kartların sahip olduğu sınıftan olan kartlar ekranda kapalı görüntülenmektedir.



Ekran çıktısında da olduğu gibi kullanıcı kart oynamak istediği zaman kartların üzerine gelip tıklaması yeterlidir. Kullanıcı kartını oynadığı zaman otomatik olarak bilgisayar da kendi kartını oynamaktadır. Puanlar, her açılan kartın beraberinde sağ üst köşede gösterilmektedir.



Oyun sunum ekranı

Oyuncu, oyundan çıkmak istediğinde “EXIT” butonuna tıklayarak oyundan çıkış yapabilmektedir.

D. Kullanılan Temel Veri Yapıları

Programda farklı sınıflardan türetilen kartlarla, program süresince çeşitli işlemler yapılması gerektiğinden kullanılacak en uygun yapının arraylist olduğuna karar verdik. Arraylist, bize projede birçok açıdan kolaylık sağladı. Kartların dağıtılmasında arraylistlerin add() ve remove(), clean() gibi özelliklerinden faydalandık. Oyunda bir desteden diğer bir desteye kart aktarılmasında bu methodlar kullanıldı. Collections sınıfının shuffle metodu ile ArrayListi rastgele bir şekilde karıştırdık. Ayrıca arraylistin sahip olduğu dinamik yapı kartların karşılaştırılmasında bize yardımcı oldu.

III. DENEYSEL SONUÇLAR

Gerçekleştirdiğimiz proje sayesinde nesneye yönelik programlamaya dair bilgilerimizi geliştirdik. Java Swing kütüphanesini öğrenirken arayüz kullanımı hakkında bildi sahibi olduk. Java event sistemlerinden yardım alarak bu eventleri kullandık. Sporcu Oyuncu sınıflarını extends eden sınıflardan nesneler oluşturup kullandık. Projede sıklıkla kalıtım ve kapsülleme

yapılarını kullanarak uyguladık. Oyunu uygularken de arraylist veri yapısını ve çeşitli methodlarını kullanma fırsatı bulduk.

IV. YAKLAŞIMLAR

A. Abstraction(Soyutlama)

Soyutlama, nesneyi bazı karakteristik özellikleri olan ve bazı fonksiyonları gerçekleştirebilen bir veri tipi olarak genelleştirmektedir. Class yapısı ile gerçekleştirilir. Soyut sınıflardan nesne oluşturulamaz. Soyut yonteme sahip bir sınıfın kendisi de otomatik olarak soyuttur. Static,final,private olarak tanımlı yontemler soyut olarak tanımlanamazlar.

B. Encapsulation(Paketleme)

Kapsülleme, bir nesnenin iç yapısını, özelliklerini bozulmalara ve hatalara karşı korumaktır. Bununla birlikte başka sınıflar tarafından korunan özelliklere erişilmemesi demektir. Bu işlem erişim belirleyiciler sayesinde gerçekleşmektedir. Bir sınıfın özelliklerini dışardan erişime kapatabilmek için sınıf private olarak tanımlanır. Tanımlanan bu sınıfın özelliklerine doğrudan ulaşmak mümkün değildir. Erişebilmek için getter ve setter metotlarını ve Constructorlar kullanılır.

C. Inheritance(Kalıtım alma)

Bir sınıfın kendisine ait özelliklerini ve methodlarını bir başka sınıfa aynen aktarması ya da diğer sınıflarda kullanılması kalıtım almadır. Üst sınıf özelliklerini alt sınıflarda kullanmak için extends kullanılır. Bir alt sınıf üst sınıfa erişmek istediğinde süper() kullanmalıdır. Süper(), üst sınıftaki öğelere erişmek ve üst sınıfa ait nesne yaratmak için kullanılır.

D. Polymorphism(Çok biçimlilik)

Farklı şekilde çalışan nesnelere aynı şekilde erişmek şeklinde açıklanabilir.

Projede kullanılan yöntemler: Inheritance, Encapsulation ve abstract yöntemleridir.

Inheritance;

Basketbolcu() ve Futbolcu() classlarının Sporcu() classından özellik ve method alması için kullanılmıştır.

Aynı zamanda Bilgisayar() ve Kullanici() classlarının Oyuncu() classından özellik ve method almasında da kullanılmıştır.

Encapsulation ;

Oyuncu(),Sporcu() classlarındaki private değişkenlere sonrasında getter() ve setter() methodları ile projenin gerekli yerlerinde erişim için kullanılmıştır.

Abstraction;

Oyuncu() classı abstract bir classtır.

İçindeki methodlar farklı sınıflarda farklı mantıkla kullanılabilir.

V. SONUÇ

Projede Frame ve panel yapılarını, eventleri öğrenmiş olduk. Nesneye yönelik programlamanın sahip olduğu sınıf, nesne, obje hiyerarşilerini daha iyi kavrayıp onları en iyi şekilde kullanmayı öğrendik. Sonuç olarak ortaya bir ürün çıkartmış olduk. Sporcu kartlarıyla basit uygulamaya sahip olan bir uygulama gerçekleştirdik.

VI. KAYNAKÇA

- (1) <https://www.java.com/tr/about/>
- (2) <http://mail.baskent.edu.tr/~tkaracay/etudio/ders/prg/java/ch27/JavaSwing.pdf>
- (3) [https://en.wikipedia.org/wiki/Swing_\(Java\)](https://en.wikipedia.org/wiki/Swing_(Java))

PROJENİN KABA KODU

- Başla
- Basketbolcu ve Tenisci ArrayListi oluştur.
- Arraylistlere eleman ata.
- Random sayı üret
- Basketbolcu ve Tenisçi ArrayListindeki elemanları random şekilde oyuncu ve bilgisayar arraylistlerine ata.
- kartT ve KartBasketbol sınıflarına sırasıyla Tenisci ve Basketbolcu özellikleri tanımla
- Bilgisayar Kartları için 8 adet buton oluştur ve bu butonlara kartT ve kartBasketbol sınıfındaki özellikleri tanımla.
- Oyuncu Kartları için 8 adet Label oluştur.
- Button ve Labellara resimleri ata.
- Her bir oyuncu kartı için mouseClicked özelliği tanımla.
- oyuncuKart1'in mouseClicked özelliği aktif olduğunda oyuncuKart1 e random bir tenis oyuncusu ata.
- oyuncuKart1'in mouseClicked özelliği aktif olduğunda Random bir bilgisayar kartı ata.
- oyuncuKart1'in mouseClicked özelliği aktif olduğunda tüm tenis kartlarının setVisible özelliklerini false,tüm Basketbol kartlarının setVisible özelliklerini true yap.
- Karşılaştırma için random bir sayı üret.
- Sayı 0'a eşit ise Forehand,1 ise Backhand, 2 ise servis özelliklerini karşılaştır.
- Bilgisayar kartının forehand sayısı Oyuncu kartından büyük ise bilgisayar kartına +10 puan ekle.
- Oyuncu kartının forehand sayısı Bilgisayar kartından büyük ise Oyuncu kartına +10 puan ekle.
- Bilgisayar kartının forehand sayısı Oyuncu kartı ile eşitse iki kartada 0 puan ekle ve kartların konumlarını eski konumlarına ata.
- Bilgisayar kartının backhand sayısı Oyuncu kartından büyük ise bilgisayar kartına +10 puan ekle.
- Oyuncu kartının backhand sayısı Bilgisayar kartından büyük ise Oyuncu kartına +10 puan ekle.

- Bilgisayar kartının backhand sayısı Oyuncu kartı ile eşitse iki kartta 0 puan ekle ve kartların konumlarını eski konumlarına ata.

- Bilgisayar kartının servis sayısı Oyuncu kartından büyük ise bilgisayar kartına +10 puan ekle.

- Oyuncu kartının servis sayısı Bilgisayar kartından büyük ise Oyuncu kartına +10 puan ekle.

- Bilgisayar kartının servis sayısı Oyuncu kartı ile eşitse iki karta da 0 puan ekle ve kartların konumlarını eski konumlarına ata.

- oyuncuKart2'in mouseClicked özelliği aktif olduğunda oyuncuKart2'ye random bir basketbol oyuncusu ata.

- oyuncuKart2'in mouseClicked özelliği aktif olduğunda Random bir bilgisayar kartı ata.

- oyuncuKart1'in mouseClicked özelliği aktif olduğunda tüm Basketbol kartlarının setVisible özelliklerini false,tüm Tennis kartlarının setVisible özelliklerini

- true yap.

- Karşılaştırma için random bir sayı üret.

- Sayı 0'a eşit ise İkilik,1 ise Üçlük, 2 ise Serbest Atış özelliklerini karşılaştır.

- Bilgisayar kartının İkilik sayısı Oyuncu kartından büyük ise bilgisayar kartına +10 puan ekle.

- Oyuncu kartının İkilik sayısı Bilgisayar kartından büyük ise Oyuncu kartına +10 puan ekle.

- Bilgisayar kartının İkilik sayısı Oyuncu kartı ile eşitse iki karta da 0 puan ekle ve kartların konumlarını eski konumlarına ata.

- Bilgisayar kartının Üçlük sayısı Oyuncu kartından büyük ise bilgisayar kartına +10 puan ekle.

- Oyuncu kartının Üçlük sayısı Bilgisayar kartından büyük ise Oyuncu kartına +10 puan ekle.

- Bilgisayar kartının Üçlük sayısı Oyuncu kartı ile eşitse iki karta da 0 puan ekle ve kartların konumlarını eski konumlarına ata.

- Bilgisayar kartının Serbest Atış sayısı Oyuncu kartından büyük ise bilgisayar kartına +10 puan ekle.

- Oyuncu kartının Serbest Atış sayısı Bilgisayar kartından büyük ise Oyuncu kartına +10 puan ekle.

- Bilgisayar kartının Serbest Atış sayısı Oyuncu kartı ile eşitse iki karta da 0 puan ekle ve kartların konumlarını eski konumlarına ata.

- Bunu geri kalan 6 Tennis ve Basketbol kartı için yap.

- Bitir.

UML SINIF DIYAGRAMI

