

PROGRAMLAMA LABORATUVARI 1

2. PROJE

Fatma Betül İNAN
Bilgisayar Mühendisliği Bölümü
Kocaeli Üniversitesi
190202115@kocaeli.edu.tr

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.

Oyunu ve oyuncuların ellerinde bulunan kartların görülebileceği ve takip edilebileceği bir arayüz tasarlamamız beklenmektedir.

* Destede 16 adet sporcu kartı bulunmalıdır.

* Oyuncu desteden 8 (4 futbolcu, 4 basketbolcu) rastgele kart alır.

*Bilgisayar desteden geri kalan diğer 8 kartı alır.

*Hamle sırasında bilgisayar elinde bulunan 8 karttan birini rastgele seçerken kullanıcı kendi elindeki kartlardan istediği birini seçebilir. Burada kullanıcı bilgisayarın hangi kartı seçtiğini bilmemeli ve bilgisayarın elindeki kartlar kullanıcıya kapalı görünmelidir. Futbolcu kartı seçilmesi gerekirken basketbolcu kartının seçilmesine ya da tersine izin verilmemelidir. Bilgisayar kullanıcısı da aynı hatayı yapmamalıdır.

* Daha sonra ortaya konan kartlar açık bir şekilde gösterilir.

* Kartın seçilen pozisyon puanları aynıysa kartlar geri alınıp yeniden sıradaki sporcu için seçim yapılmalıdır.

* Kazanılan bir hamle içerisinde oyuna sürülen kartlar tekrar kullanılamayacaktır. Ancak son kartlarda üstünlük sağlanamaması durumunda yeni değerlendirme yapılmalıdır.

* Oyun tüm kartlar bitene kadar devam edecektir.

* Tüm kartlar bittikten sonra skoru yüksek olan oyuncu oyunu kazanır.

Amaç: Proje gerçekleştirimi ile beraber öğrencilerin nesneye yönelik programlama yapısını anlamasını ve çözüm sağlayabilmesini amaçlamaktadır.

1.1. Oluşturulması Gereken Sınıflar

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

❖ Sporcu Sınıfı

Bir sporcu sınıfı oluşturun. Sınıf tanımı şunları içermelidir:

- * Yapıcı (*constuctor*) metotları (parametrelili ve parametresiz olarak en az iki) yazılacak. Parametreler *sporcuIsim* ve *sporcuTakim* olmalı.
- * Sporcuların kart puanını göstermek için *sporcuPuaniGoster()* metodu yazılmalıdır.

❖ Futbolcu Sınıfı

Futbolcuların her biri birer sınıf olarak tanımlanacaktır. Sınıflar şu tanımlamaları içermelidir:

- * Sporcu sınıfından kalıtım alacaktır.

*Yapıcı (*constuctor*) metotları (parametrelili ve parametresiz olarak en az iki) yazılacak. Futbolcu sınıfında bulunan *futbolcuAdi* ve *futbolcuTakim* özelliklerine atama yapmak için *super()* kullanılacaktır.

*Bu sınıfın *penalti*, *serbestAtis* ve *kaleciKarsiKarsiya* özellikleri olmalıdır.

**sporcuPuaniGoster()* metodu *override* edilerek her bir futbolcu kartı için özelleştirilecektir.

**boolean* veri tipinde *kartKullanildiMi* bilgisi tutulmalıdır.

❖ Basketbolcu Sınıfı

Basketbolcuların her biri birer sınıf olarak tanımlanacaktır. Sınıflar şu tanımlamaları içermelidir:

- * Sporcu sınıfından kalıtım alacaktır.

*Yapıcı (*constuctor*) metotları (parametrelili ve parametresiz olarak en az iki) yazılacak. Futbolcu sınıfında bulunan *basketbolcuAdi* ve *basketbolcuTakim* özelliklerine atama yapmak için *super()* kullanılacaktır.

* Bu sınıfın ikilik, ucluk ve serbestAtis özellikleri olmalıdır.

*sporcuPuaniGoster() metodu override edilerek her bir futbolcu kartı için özelleştirilecektir.

*Boolean veri tipinde kartKullanildiMi bilgisi tutulmalıdır.

❖ Oyuncu Sınıfı:

Bilgisayar ve kullanıcı olmak üzere oyunu oynayan iki oyuncu olacaktır. Bu iki oyuncunun farklı ve aynı özellikleri olacaktır. Aynı özelliklerini temsil etmek için Oyuncu temel sınıfı oluşturulacaktır. Bu sınıfta bulunması gereken özellikler ve fonksiyonlar:

*oyuncuID, oyuncuAdi ve Skor özellikleri olmalı.

*Yapıcı (constuctor) metodları (parametrelili ve parametresiz olarak en az iki) yazılacak. Parametreler oyuncuID, oyuncuAdi ve Skor olmalı.

*kartListesi özelliği ile oyuncuların elinde bulunan kartlar listede tutulacaktır.

*SkorGoster() fonksiyonu ile oyuncuların skorları gösterilecektir.

*kartSec() fonksiyonu yazılmalı fakat bu sınıf bilgisayar ve kullanıcı için farklı durumlarda çalışacağı unutulmamalıdır.

❖ Bilgisayar Sınıfı

*Oyuncu sınıfından kalıtım alacaktır.

*Yapıcı (constuctor) metodları (parametrelili ve parametresiz olarak en az iki) yazılacak. Oyuncu sınıfında bulunan oyuncuID, oyuncuAdi ve Skor özelliklerine atama yapmak için super() kullanılacaktır.

*Oyuncu sınıfında bulunan kartSec() metodu override edilecektir. Bilgisayar random olarak aldığı kartlar arasından yine random kart seçerek ortaya koyacaktır.

❖ Kullanıcı Sınıfı

*Oyuncu sınıfından kalıtım alacaktır.

*Yapıcı (constuctor) metodları (parametrelili ve parametresiz olarak en az iki) yazılacak. Oyuncu sınıfında bulunan oyuncuID, oyuncuAdi ve Skor özelliklerine atama yapmak için super() kullanılacaktır.

*Oyuncu sınıfında bulunan kartSec() metodu override edilecektir. Kullanıcı random olarak aldığı kartlar arasından kendi istediği kartı seçerek ortaya koyacaktır.

❖ Test Sınıfı

*Futbolcu ve basketbolcuların isimleri ve özellikleri burada tanımlanmalıdır. Pozisyon bilgisi için bir metod içermelidir. Futbolcu seçimi yapılacağı zaman penaltı, serbest vuruş ve kaleciyle karşı karşıya pozisyonlarından; basketbolcu seçimi yapılacağı zaman üçlük, ikilik ve serbest atış

pozisyonlarından rastgele birinin seçimi oyuncuların kart seçimi yapıldıktan sonra burada yapılmalıdır.

*Futbolcu ve basketbolcuların özelliklerin ataması için getter ve setter metodları da, Constructor'da kullanılabilir. Oyunun ilerleyişi buradan takip edilmelidir.

2.Araştırmalar ve Yöntem

Projeyi geliştirmeden önce nereden başlayacağıma karar verdim. İlk olarak Test sınıfını oluşturarak işe koyuldum çünkü projemin Main fonksiyonu orada, proje oradan build olmaya başlayacak. Ardından sınıflarımı oluşturmaya başladım, önce mother(ana) sınıflarımı oluşturdum yani Oyuncu ve Sporcu.. Oyuncu sınıfıma istenen özellikleri tanımladım, Sporcu sınıfı türünde bir dizi tanımlayarak her oyuncunun elindeki kartları burada sakladım. Projedeki tüm sınıflarda encapsulation gibi nesneye yönelim özelliklerini kullandım. Oyuncu sınıfında kullandığım kartSec alt sınıflarda override edildi. Her oyuncunun elinde bir kart vardır ve kart 'ın numarasına erişmem gerekiyordu. Sonuçta Bilgisayar bu kartı random çekecekti ve Test classtan bu karta erişmem lazımdı bu yüzden ek olarak Oyuncu sınıfına kartNo özelliği tanımladım. Sporcu sınıfını da istenene uygun şekilde(override dahil olmak üzere) tanımlayıp bitirdim. Projemde takıldığım kısımlardan biri, Test class içerisinde kartları nasıl türetilip nasıl kullanacağımdı. Bunun için 8 kartlık basketbol ve 8 kartlık futbolcu String dizileri tanımladım. Bu dizi içerisinde kartları alıp Bilgisayar ve Kullanıcıya oyun başında dağıttım. Bunun için kartEkle methodunu kullandım. Ardından oyundaki kart sayısı 2'den büyük olduğu sürece oyunu döndüren bir while yazdım. Yine takıldığım kısımlardan biri sıra kimdeydi ve hangi kart sırasına geldiğimizdi. Bunun için sıraKimde ve hangiKart değişkenleri tanımladım, bu değişkenler futbolcu kartı mı yoksa basketbolcu kartı sırası mı sorusuna ek olarak, sıra Bilgisayar da mı? Yoksa Kullanıcı da mı? Sorusuna da cevap veriyordu. Bu şekilde projemi hızlandırıp iyi bir zemin üstüne oturarak bitirdim. Proje nesneye yönelim felsefesine uygun olarak hazırlandı, tek eksiğim zamanım yetmediği için görsel tarafını (gui ya da swing) yetiştiremedim. Bu yüzden kullanıcı kart seçerken console' kart no almak zorunda kalıyorum

Geliştirme Ortamı

Windows üzerinde Netbeans 8.2 kullanılarak hazırlanmıştır.

1. Kod Bilgisi

4.1. UML Diyagramı

Son sayfaya eklenmiştir.

4.2. Algoritma

Öncelikle oyunu Test class'ı yönetiyor. Oyuna başlamadan önce basketbolcu futbolcu kartlarını tutan iki ayrı String dizi tanımlıyorum. Bu dizi sayesinde kullanıcılara oradan oyuncu seçip dağıtabiliyorum. Bunu yapmak için de temp(geçici) birer Futbolcu ve Basketbolcu nesnesi oluşturdum. Bu nesneler ile for kullanıp oyunculara kart dağıttım. Ardından oyun, kullanıcının kart girmesini isteyerek devam ediyor. Başlangıçta karışıklık olmaması için ilk kartı futbolcu kartı seçtim ve oyuna kullanıcı başlıyor. Bunun için consoleda istenen numaraları belirledim, kartListesi dizisinde o aralıkta kullanıcının futbolcu kartları bulunuyor. Buna uygun bir giriş isteyerek projeyi başlattım. Sağladığım en büyük kolaylık sıraKimde, hangiKart ve kartListesi değişiklikleri oldu. Bu sayede karışıklık yaşanmadan sırayı Bilgisayar'a geçirtip kart türünü de değiştirebiliyordum. Projemizin Test sınıfında static değişkenler ve fonksiyonlar mevcut. Bunun temel sebebi, java'da bir fonksiyon kullanabilmek için öncelikle o fonksiyonun ait olduğu sınıftan bir nesne oluşturmamız gerekiyor. Oysa static dersek, nesne oluşturmadan da kullanabiliyoruz. Test class benim main class'ım olduğundan global değişkenleri static tanımladım. Aynı şekilde bazı fonksiyonlar da static olarak tanımlandı. Projeye geri dönersek önce PozisyonBilgisi fonksiyonu ile random olarak bir pozisyon belirliyoruz. Bu pozisyon belirlenişi sonrası kartlar seçilip güçler karşılaştırılıyor. Tam bu noktada kullanıcıPozPower ve bilgisayarPozPower değişkenleri devreye giriyor. Bu iki değişken global tanımlanan, mevcut karşılaşmada tarafların gücünü tutan değişkendir. Örneğin futbolcu kartları seçilirken random pozisyon penalti geldiyse, kullanıcıPozPower o kartın penalti gücünü ve bilgisayarPozPower ise bilgisayar'ın seçtiği kartın penalti gücünü tutuyor. Bu sayede istediğim yerden erişip kullanabiliyorum

Test Class'ındaki Fonksiyonlar

Rand() fonksiyonu büyük kolaylık sağladı. Random sayı gereken her noktada bu fonksiyonu kullandım.

Karsilasma() oyunun her çarpışma/karşılaşma kısmında kullanıcı pozisyon gücü ve bilgisayar pozisyon gücünü kullanıyor. Karşılaşma fonksiyonunu kullanmadan önce elbette bir pozisyon belirlemek gerekiyor. Bu fonksiyon mevcut karşılaşmada kim daha güçlüyse ona 10

skor puanı verip bitiriliyor.

GameOver() çağrıldığı noktada ekrana kazananı yazıp oyunu bitirmemizi sağlıyor.

ikisiArasindaMi() girilen parametre istenen iki değer arasında ise true çeviriyor. Bu fonksiyonu kullanma sebebim swing kullanamamış olmam. Girilen sayı istenen iki değer arasında değilse false döndürüp tekrar sayı isteyecektir.

4.4 Kaba Kod

Projemizin kaba kodunun detayları:

- 1 Test Class'ında 3 global değişken oluşturarak başla
- 2 Sıranın kimde olduğunu ve hangi kart türü ile oyuna başlanacağını belirle
- 3 Futbolcu ve Basketbolcu kartları için diziler belirle
- 4 Bu dizilerden sporcu isimleri çekip kullanıcı ve bilgisayar'a ver
- 5 Kullanılmayan kart sayısı > 2 olduğu sürece projeyi çalıştıracak olan döngüyü başlat
- 6 Futbolcu sırası ise ve kullanıcı sırası ise, kullanıcının kart sayısı > 1 ise kullanıcıdan futbolcu kartı iste
- 7 Değilse, oyunu bitir (çünkü kartı yok)
- 8 Sıra bilgisayarda ise sınıfa erişip random bir futbolcu kartını ver
- 9 Seçilen kartlar için pozisyon bilgisi çalıştır.
- 10 Kart güçleri farklı ise Karsilasma() ile güçlü olan tarafa 10 puan ver
- 11 Eğer kart güçleri eşit ise ve son kartlar ise 3 pozisyon gücünü de dene
- 12 Eğer kart güçleri eşit ve son kart değillerse kartları geri ver
- 13 Kart sırasını basketbola verip aynı işlemleri baştan sona yinele
- 14 Kazananı yaz ve oyunu bitir.

Kaynakça

1) Nesne yönelimli programlamaya giriş

(Tubitak) Link :

https://web.cs.hacettepe.edu.tr/~bbm102/misc/java_notes_by_oa.pdf

2) Youtube Bilgisayar Kavramları sayfasından Java ile Nesne Yönelimli Programlamaya giriş serisi

3) Global değişken tanımlama ve static fonksiyonlar hakkında yardım aldığım forum sitesi: stackoverflow.com

