

# SPORCU KART OYUNU

Mevlûde Sıla Kazan , Şule Vural

Bilgisayar Mühendisliği Bölümü, Kocaeli Üniversitesi, Umutepe Kampüsü, İzmit/KOCAELİ  
[silakazan26@gmail.com](mailto:silakazan26@gmail.com), [sulevural54@gmail.com](mailto:sulevural54@gmail.com)

## Özetçe

Programlama Laboratuvarı 1 – 2.Projesi olan *Sporcu Kart Oyunu* ile proje dökümanında bulunan ‘İsterler’ kısmında belirtilenlerden yola çıkılarak kurulması gereken oyun mantığının işlendiği kod ile oyunu ve oyuncuların ellerinde bulunan kartların görülebileceği ve takip edilebileceği bir arayüz tasarlanması beklenmiştir.

Bu proje için kullanılabilecek dil seçenekleri arasında Java Programlama Dili seçilmiştir. Görselleştirme için seçilen programlama diline uygun kütüphanenin kullanılması istenmiştir.

Bu proje için seçilen dil Java Programlama Dili olup geliştirme ortamı olarak “Apache NetBeans 12.1” ortamı seçilmiş ve arayüz için “GUI” kütüphanesinden yararlanılmıştır.Kütüphanenin kullanımı süre zarfında öğrenilmiş olup gerekli tasarımlar yapılmıştır.

İsterler kısmında belirtilen “UML SINIF DİYAGRAMI” araştırılmıştır ve raporda açıklanmış bir şekilde mevcuttur.

Kodun okunaklı olmasına özen gösterilmiştir ve gereken yerlerde düzeltmeler yapılarak daha kısa ve öz bir proje çıkarılmaya çalışılmıştır.

## 1.GİRİŞ

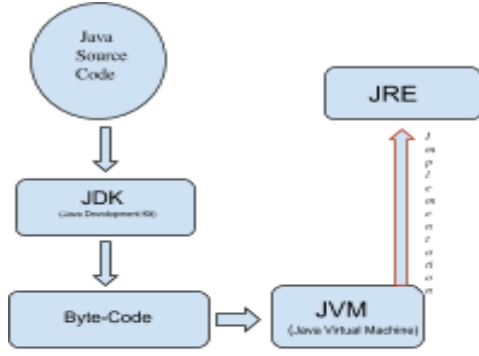
**Java Programlama Dili**, Sun Microsystems mühendislerinden James Gosling tarafından geliştirilmeye başlanmış açık kodlu, *nesneye yönelik (object-oriented)*, zeminden bağımsız, yüksek verimli, çok işlevli, yüksek seviye, adım adım işletilen yapısal bir dildir.

Java, herhangi bir bilgisayar mimarisine veya herhangi bir platforma bağlı olmadan çalışabilen bir programdır.

“JVM: Java Virtual Machine” olarak adlandırılan Java Sanal Makinesi bunu sağlamaktadır. İstenildiğinde farklı platformlara entegre edilebilen tipik *bytecode* kullanılır.

JVM’de çalıştırılabilen bu kod sistemi Java’nın kullandığı kodlardır. Farklı programlara entegre edilebilmesi sayesinde platform kısıtlaması gibi sorunlar yaratmaz, kullanıcılarına daha özgür bir kullanım alanı yaratır.

Java’nın mottosu *WORA (write once, run anywhere)* olarak anılır: bir kere yaz, her yerde kullan. Böylece Java’yı destekleyen her platformda yeniden derlemeye gerek kalmadan Java kodları kullanılabilir. Örneğin bir Java programını UNIX’te yazabilir ve derleyebilir, sonrasında hiçbir şeyi değiştirilmeden Microsoft Windows’ta veya Macintosh’ta kullanılabilir.



Bu projede faydalanılan IDE olan *NetBeans*, Oracle tarafından geliştirilen bir Java geliştirme ortamıdır (IDE) ve ücretsiz olarak dağıtılmaktadır. Özellikle kullanıcı arayüzü tasarımında sağladığı kolaylıklardan dolayı tercih edilmektedir.

## 2.YÖNTEM

Bir kullanıcının bilgisayarla oynayabileceği basit bir kart oyunu tasarlanmak istenmiştir. Tasarlanacak oyunda 16 adet oyun kartı olması ve iki takımdan oluşması istenmiştir.

Bu projede spor dalı olarak Basketbol ve Voleybol dalları seçilmiştir ve istenen takımlar ve gereken özellikler için “*BasketbolTakimi*” ve “*VoleybolTakimi*” adlarında iki “*class*” ve bu iki takım için ortak olan özellikler ve bazı methodları barındırması istenen “*Sporcu*” classı açılmıştır. İki takım classı *Sporcu* classından “*miras*” almalıdır.

“*Test Class*”nın içinde sekiz adet basketbolcu ve sekiz adet voleybolcu nesnesi tanımlanmıştır.

Nesnelerin “*İsterler*” kısmında gerekli görülen özellikleri ve methodları

tanımlanmıştır ve oyunun ilerleyen kısımlarında çağrılarak kullanılmıştır.

Tasarlanacak oyunda kullanıcıya rastgele sekiz sporcu kartı -*dördü basketbolcu ve dördü voleybolcu olmak koşuluyla*- verilir ve destede geriye kalan diğer sekiz kartı bilgisayar alır.

Hamleler bir basketbolcu bir voleybolcu sırasında olmalıdır. Basketbolcu kartı seçilecek iken voleybolcu kartı seçilememelidir ve aynı şekilde voleybolcu kartı seçilecek iken basketbolcu kartına tıklanamamalıdır.

Hamle sırası geldiğinde bilgisayar o an hangi takımdan bir kart seçecekse elindeki kartlardan rastgele seçmeli kullanıcı ise takımına göre elindeki kartlardan istediğini seçmelidir. Bunun için bu projede bilgisayarın kendi kartlarının (nesneler) bulunduğu bir *liste* oluşturulmuş, rastgele seçebilmesi için random değerlerin atanması ile gelecek elemanın seçilmesi sağlanmıştır.

Kullanıcı içinde aynı şekilde kendi kartlarının (nesneler) bulunduğu bir *liste* tanımlanmış ve kullanıcının kendi istediği elemanı seçmesi sağlanmıştır.

Bu iki listede olan nesneler yukarıda anlatıldığı gibi seçildiklerinde ekrana seçilen nesnenin (kartın) görseli gelmelidir. Bunun için her nesnenin kendi görseline sahip olması amacıyla “*Constructor*” yapısı ile çalışılmış ve “*Encapsulation*”dan yararlanılmıştır.

Bu bölümde “*İsterler*” kısmında oyun başlayıp kartlar dağıtıldığında bilgisayarın elindeki kartların ön yüzünün görünmesi fakat kullanıcının elindeki kartların ön yüzünün görünmesi ve kartlar seçildikten sonra karşılaştırılmaları için karşı karşıya geldiklerinde iki kartında ön yüzünün görünmesi istenmiştir.

Projede bunun için öncelikle bilgisayar için seçilen her kart bilgisayarın *listesinde* tutulmuştur ve ekrana kartların arka yüzleri basılmıştır.

Bilgisayar her kart seçtiğinde arayüz için kullanılan kütüphane ile etkileşimli olarak seçilen kart ekranda istenen konuma getirilerek ön yüzü kullanıcının da görmesi için ekrana basılmıştır.

Benzer şekilde *ister* olarak belirtilen şekilde ön yüzleri görünen kullanıcı kartlarından seçilen kart, arayüz için kullanılan kütüphane ile etkileşimli olarak ekranda istenen konuma getirilmiştir.

Kartlar karşı karşıya gelip karşılaşma başladığı zaman karşılaşılabilecek özellik "*İsterler*" kısmında *pozisyon bilgisi* adı altında tutulan üç özellikten rastgele biri olmalıdır ve hangi sporcunun özellik değeri daha büyükse skor o kartın sahibine gitmelidir. Her iki oyuncunun ortaya koyduğu iki kart her hamlenin sonunda karşılaştırılacaktır.

Değerler aynıysa kartlar geri alınmalıdır. Bu durumda hamle sırası diğer sporcu özelliğinden (basketbolcuysa voleybolcuysa, voleybolcuysa basketbolcuysa) devam etmelidir.

Eğer oyuncuların ortaya koydukları ellerindeki son kartlarda pozisyon bilgisine göre aynı puandaysa rastgele farklı bir pozisyon olarak yeni karşılaştırma yaparak ilerlenir. Üç pozisyon bilgisi aynıysa eldeki kartlar ile oyun sonlandırılır.

Bunun için projede her nesneye pozisyon bilgisi atanmış ve koşul ifadelerinden yararlanılarak durumların kontrolü sağlanmıştır.

Projede gelen pozisyon bilgilerinin o koşulu sağlayan kod parçacığında kontrolü

sağlanarak o eli kazanan oyuncunun skorunun artması sağlanmıştır.

Oyun sonlandığında kazanma veya beraberlik durumları ekrana basılmaktadır.

### 3.DENEYSEL SONUÇLAR





#### 4.SONUÇ

Projede Java Programlama Dilini kullanarak nesneye yönelik programlamanın yapısı anlaşılmış ve problemlere çözüm sağlanabilmesi için “Encapsulation”, “Inheritance”, “Polymorphism”, “Abstraction” yapılarından ihtiyaç olanlar kullanılarak pratik yapılmıştır.

Nesneye yönelimli programlama ve yapısal programlama arasındaki farklar anlaşılmış ve nesneye yönelimli programlama ile çalışarak algoritmik düşünme yetisi geliştirilmiştir.

#### Kaba Kod:

1-Başla.

2-Kullanıcı ismini al ve oyun ekranında kullanıcı ismi için ayrılan kısma yazdır.

3-“*ListeOlustur1*” ve “*ListeOlustur2*” fonksiyonlarını kullanarak bir adet voleybolcu ve bir adet basketbolcu kartlarından oluşan listeleri iki adet listeyi oluştur.Bilgisayarın ve kullanıcının kartlarını “*KartDağıtım*” fonksiyonu ile rastgele dağıt.

4-Bilgisayarın kartlarının arka yüzlerini,kullanıcının kartlarının ön yüzlerini bastır.

5-Kullanıcı istediği bir kartı seçtiğinde seçilen kartı ekranda istenilen konuma getirirken eş zamanlı olarak bilgisayarın rastgele seçtiği kartı da karşılaşma için ekranda istenilen konuma getir.

6-Seçilen kartların karşılaştırılabilmesi için “*Karsilasma*” fonksiyonu ile rastgele bir pozisyon seç ve bu pozisyonu karşılaştır.

7-Eğer eşitlik durumu yoksa pozisyon değeri hangi kart için büyükse o karta sahip oyuncunun skoruna **10** puan ekle.

8-Eğer eşitlik durumu varsa ve son kart değilse diğer ele geçilmesini iste.

9-Eğer eşitlik varsa ve son kartsa “*Karsilasma\_sonkart*” fonksiyonu ile



#### Pozisyon puanı eşit ise :



#### Son kartta ilk seçilen pozisyon puanları eşit olursa:



tekrar rastgele bir pozisyon seç ve karşılaştı.

**10-**Oyundaki tüm kartlar oynandıktan sonra varsa skoru büyük olan oyuncunun kazandığını ekrana bastır yoksa beraberlik durumunun olduğunu ekrana bastır.

### UML DİYAGRAMI:

UML, bir sistemin tasarımını görselleştirmek için yazılım mühendisliği alanında genel amaçlı modelleme dilidir. Yazılı bir dil değildir. Farklı amaçlar için kategorilere ayrılmış olsa da, genel itibarıyla modelleme için kullanılır. 1995 yılında, yazılımlarda bir standart yaklaşım oluşturmak için geliştirilmiştir.

UML diyagramları ile önceden modellenilen bir yazılım projesini, modele uygun olacak şekilde herhangi bir dil ile geliştirmedir. Bu da yazılım mühendisleri arasında ortak bir dil oluşturmaktadır. İlk çıktığı zamandan beri sürekli geliştirme göstererek, birçok farklı dala ayrılmıştır.

Aşağıda UML diyagramlarının kategorilerini mevcuttur. Projede istenen Class Diagram, Structure Diagram altında bulunmaktadır. Structure Diagrams(Yapısal Diyagramlar),

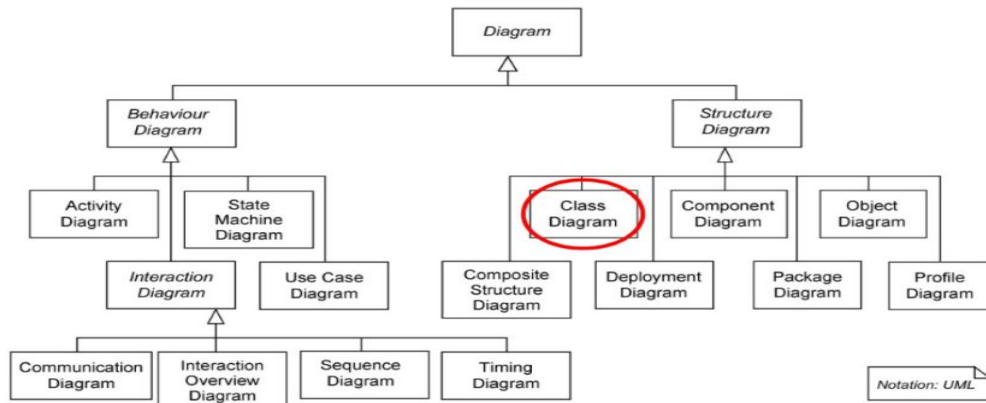
modellenmekte olan sistemde bulunması gerekenler için kullanılır.

### UML Sınıf Diyagramı

Class diyagramları, OOP(Nesne Tabanlı Programlama) temel alınarak tasarlanmıştır. Amaç yazılım içindeki sınıflar ve aralarındaki ilişkileri tanımlamaktır.

Class diyagramlarında sınıflar; sınıf adı en üstte, özellikler ortada ve methodlar en altta olmak üzere ” +,-,# , ~ “ işaretlerini erişim belirleyicileri tanımlamak için kullanılarak ifade edilir.

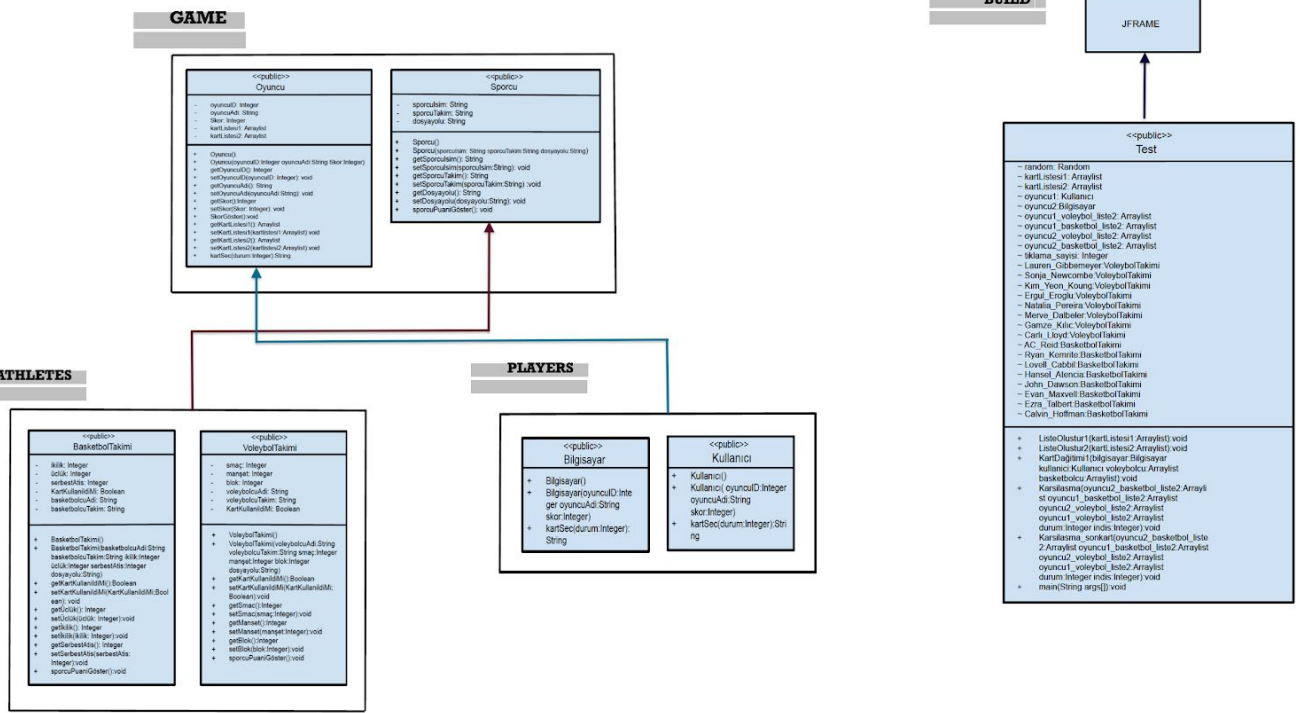
Kalıtım (inheritance),abstract class yapıları veya güçlü-zayıf bağ ilişkileri ile karşılaşıldığında a bunları okların kesiksiz veya kesikli olması veya ok ucunun içi boş eşkenar üçgen,içi dolu eşkenar üçgen, boş üçgen vb. şeklinde tanımlayarak diyagram çizilir.



görsel kaynak: wikipedia.org

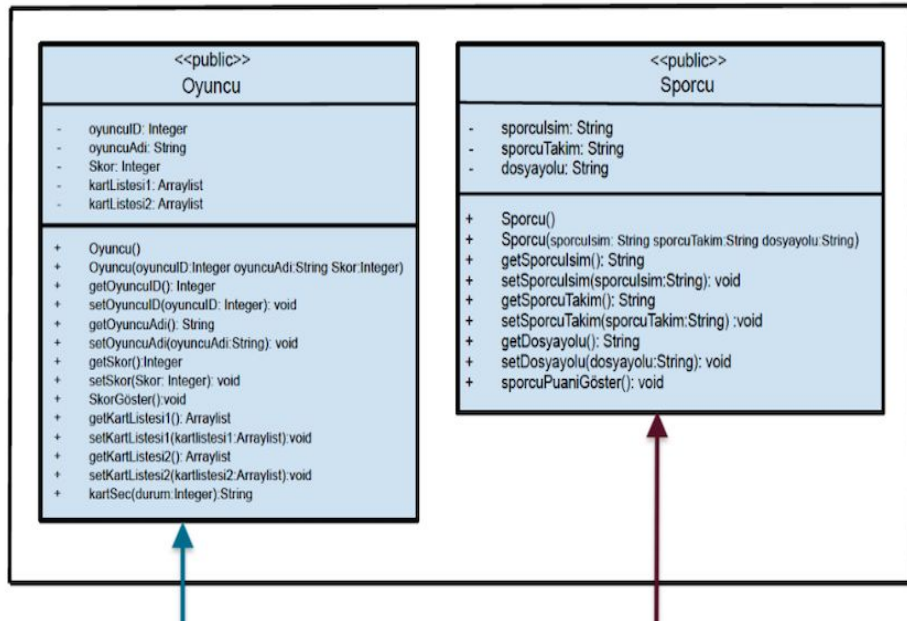
# PROJENİN UML SINIF DİYAGRAMI

## UML SINIF DİYAGRAMI



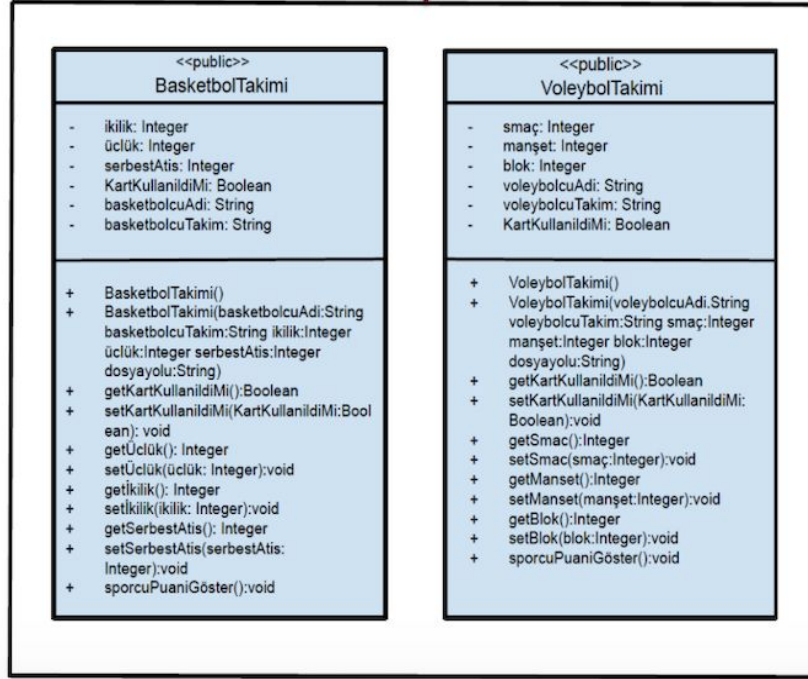
## BÜYÜK PARÇA 1:

## GAME



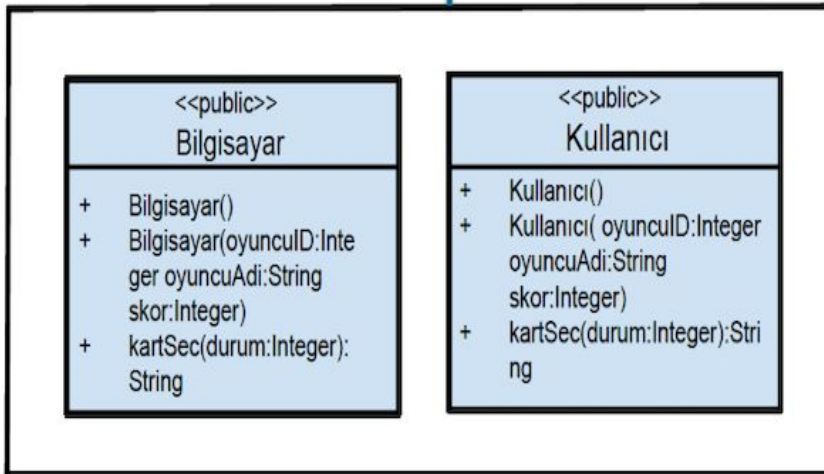
## BÜYÜK PARÇA 2:

### ATHLETES



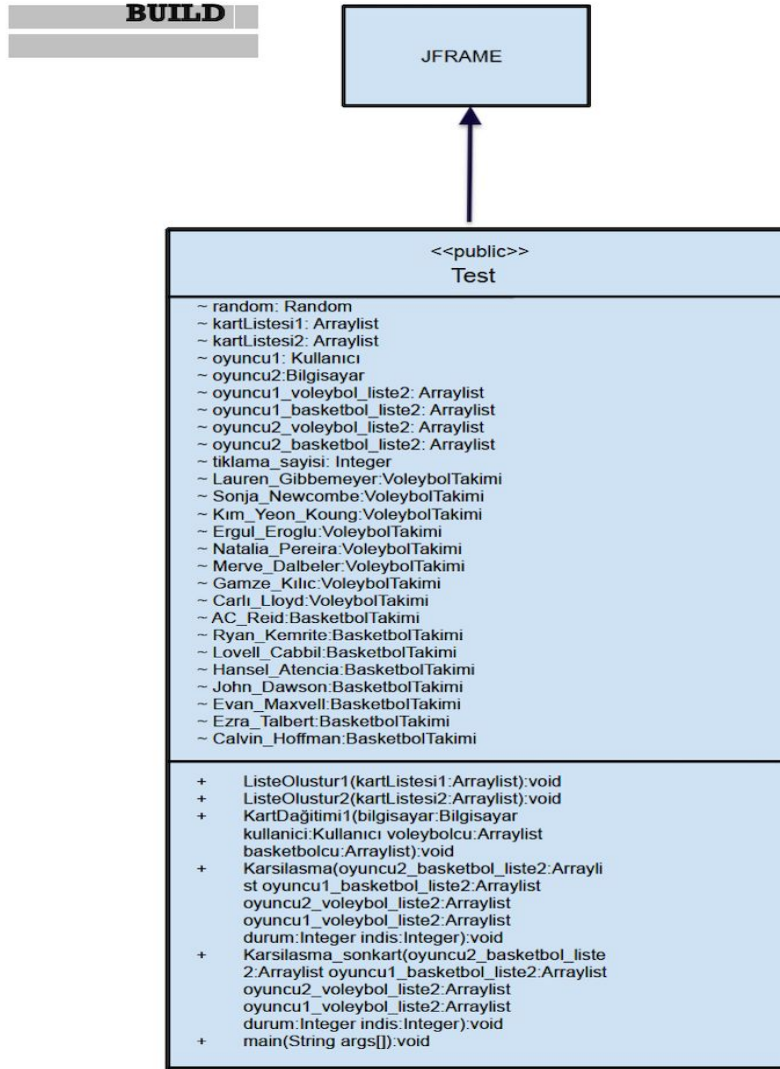
## BÜYÜK PARÇA 3:

### PLAYERS





#### BÜYÜK PARÇA 4:



#### 5.KAYNAKÇA

1- [Java Kart oyunu Bölüm1:Temel Sınıflar](#)

2-<https://docs.oracle.com/javase/tutorial/>

3-<https://www.tutorialspoint.com/java/index.htm>

4- <https://www.w3schools.com/java/>



5-<https://www.javatpoint.com/java-jframe>

6-<https://examples.javacodegeeks.com/desktop-java/swing/java-swing-form-example/>

7-[Java Swing - Bilgisayara Karşı Taş, Kâğıt, Makas Oyunu | Java Swing Dersleri #1](#)

8-<https://smartpro.com.tr/java-nedir-nerelerde-kullanilir/>

9-<https://medium.com/@tugrulbayrak/uml-class-diagramlari-4c3bb7e9cc4c>

10-<https://tr.wikipedia.org/wiki/UML>

11-[https://tr.wikipedia.org/wiki/Sınıf\\_diagramı](https://tr.wikipedia.org/wiki/Sınıf_diagramı)