

Pokemon Kart Oyunu Projesi

Pokemon Card Game Project

Cumali TOPRAK

Mühendislik Fakültesi, Bilgisayar Mühendisliği
Kocaeli Üniversitesi
Kocaeli, Türkiye
cumalitoprak@gmail.com

Berkay Efe ÖZCAN

Mühendislik Fakültesi, Bilgisayar Mühendisliği
Kocaeli Üniversitesi
Kocaeli, Türkiye
berkayefozcan@gmail.com

Özetçe—Günümüzde, neredeyse yapılmakta olan neredeyse tüm yazılım projeleri nesne tabanlı mimariye sahip olan, nesne yönelimli dillerle yapılmaktadır. Bizde gerçekleştirmiş olduğumuz “pokemon kart oyunu” projesinde Türkiye ve dünyada çok büyük bir kullanım oranı olan java nesne yönelimli programlama dilini kullandık. Ayrıca projemizde nesne yönelimli dillerin bize sunduğu kapsülleme, kalıtım, çok biçimlilik gibi yapıların doğru yerde doğru şekilde kullanımını gerçekleştirmeye çalıştık. Arayüz platformu olarak swing arayüz kütüphanesini gerekli ölçüde kullandık. Programı test ettiğimiz zaman gördük ki, yapmış olduğumuz proje gerekli şartlara ve kriterlere göre başarılı bir sonuç vermektedir. Eğer gerekli şartlar sağlandığı taktirde program geliştirilmeye ve güncellenmeye açık bir haldedir.

Anahtar Kelimeler—Nesne Yönelimli Diller, Java Programlama, Pokemon Oyunu.

Abstract — Today, almost all software projects that are currently being implemented are made in object-oriented languages with object-oriented architecture. We have made us "Pokemon card game" projects in Turkey and a very large utilization rate in the world, we use the java object-oriented programming language. In addition, in our project, we have tried to make use of structures such as encapsulation, inheritance, polymorphism, provided by object oriented languages in the right place and in the right way. We have used the swing interface library as necessary as the interface platform. When we tested the program, we found that our project was successful according to the necessary conditions and criteria. If the necessary conditions are met, the program is open to development and updating.

Keywords— Object Oriented Languages, Java Programming, Pokemon Game..

I. GİRİŞ

Almış olduğumuz programlama laboratuvarı dersi kapsamında proje ödevi olarak verilen “Pokemon Kart Oyunu” projesini tamamlamış bulunuyoruz. Giriş kısmında öncelikle projemiz hakkında genel bilgiler vermeye çalışacağız. Öncelikle projemizi yaparken gerekli yapıları kullanma kriterine bağlı olarak projemizi gerçekleştirdik. Bizden nesne yönelimli programlama dillerinden olan Java veya C++ dillerinden birini seçip bunlardan biri ile programı yazmamız istenmektedir. Biz projemize daha uygun

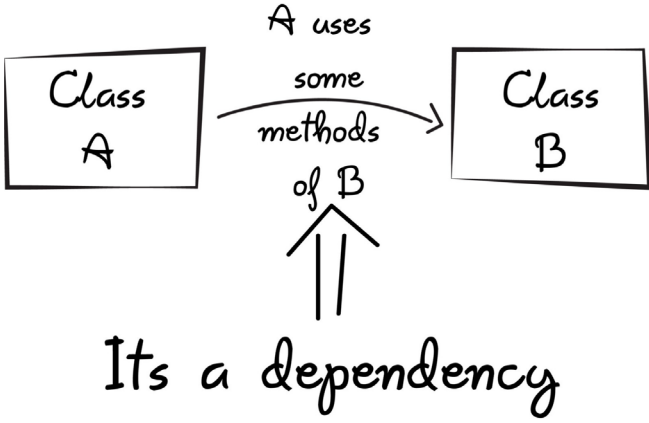
olduğunu düşündüğümüz Java programlama dilini kullandık. Arayüz platformu olarak ise swing kütüphanesini projeye entegre ettik. Projede kullanılması zorunlu yapılar olan inheritance, polymorphism, encapsulation gibi yapıları efektif bir şekilde kullanmaya çalıştık. Projemizin konusuna ve işleyişine gelecek olursak; bilgisayar tabanlı, nesne yönelimli dillerin sağladığı imkanları kullanarak basit bir kart oyunu yapmaktır. Öncelikle oyunumuz iki aşamadan oluşmaktadır. Kullanıcı(user) bilgisayara karşı bu oyunu oynadığı gibi, bilgisayar bilgisayara karşı da oynayabilmektedir. Tasarladığımız arayüz ile bu seçim kolayca yapılabilmektedir. Bu seçim yapıldıktan sonra oyun başlamış olur. Farz edelim ki, bilgisayara karşı oynuyoruz; Toplamda on kart bulunmaktadır ve bu kartlar bitince oyun da bitmektedir. Öncelikle hem bilgisayara hem de bize sistem tarafından random olarak üçer tane kart dağıtılır. Ve sırayla kartlar ortaya atılmaya başlar. Atılan kartlardan hasar puanı yüksek olan kartın sahibinin skor tabelasına artı beş puan eklenir. Eksilen kartların yerine sistem tarafından her iki oyuncuya da kalan dört karttan random olarak birer tane daha verilir. Ta ki her iki oyuncuda da hiç kart kalmayana kadar devam eder. Oyun sonunda skoru daha fazla olan oyuncu oyunu kazanmış olur. Oyun genel hatlarıyla bu şekildedir. Programlarken kurduğumuz bağımsız kodlar sayesinde bu yazılım gerekli şartlar oluştuğu taktirde geliştirilebilir, güncellenebilir durumdadır. İlerleyen bölümlerde sırası ile kullanılan yöntemler bölümünde kullandığımız tüm yöntemler açıklanacak, sonrasında deneysel sonuçlar açıklanacak, sonuç ve kaynakça bölümü ile bitirilecektir.

II. YÖNTEM

A. Dependency Injection(Bağımlılık Azaltma)

Projemizde bulunan kodların birbirlerine olan bağımlılığını azaltmak adına ve projeye yapılacak eklemelere ve değişikliklere karşın programın göstereceği direnci azaltmak adına dependency injection yöntemini kullandık. Bu sayede projeye yapacağımız herhangi değişiklik veya güncelleme esnasında var olan kodlara herhangi bir değişiklik yapılması gerekmez. Günümüz yazılımlarında elzem olan bu tekniği biz de projemizde kullanmayı uygun

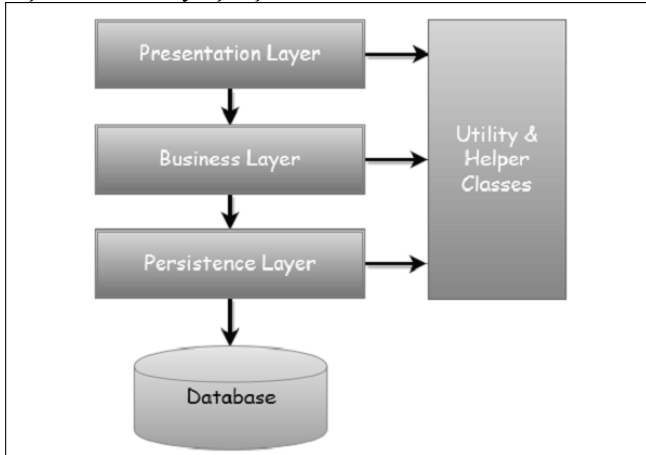
gördük.



Şekil-1: Kodda bağımlılığı gösteren şekil.

B. Katmanlı Mimari(Layered Architecture)

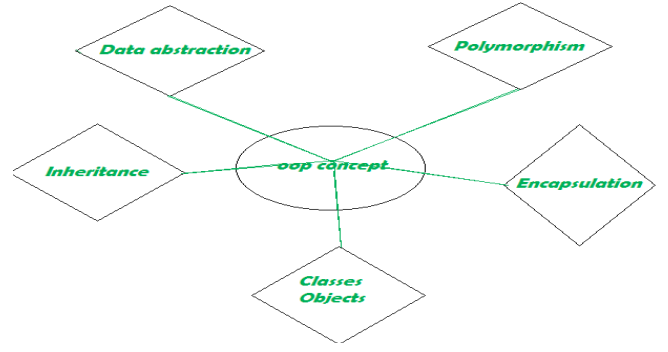
Yazılım projeleri gerçekleştirirken belli nedenlerden dolayı katmanlı mimari dediğimiz yapılar kullanılır. Katmanlı mimari sayesinde yazdığımız kodlar arasında belli bir hiyerarşi, düzen sağlanır. Ve bu sayede kod karmaşıklığı ve dosya karmaşıklığı büyük bir oranda azaltılmış olur. Biz de bu sebepten ötürü projemizde katmanlı mimariyi belli bir ölçüde kullanmaya çalıştık.



Şekil-2: Katmanlı mimari yapısı gösteren şekil.

C. Nesnel Programlama Teknikleri(Objective Programming Techniques)

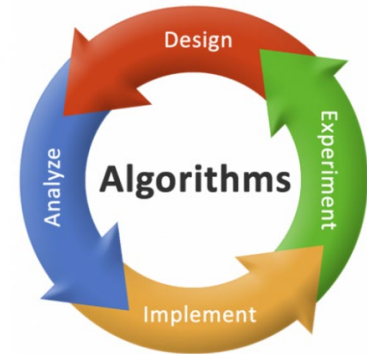
Giriş bölümünde de belirttiğimiz üzere projemizde nesnel programlama tekniklerinin bir çoğunu etkili bir şekilde kullanmaya çalıştık. Özellikle kart nesneleri arasında inheritance kullanımı gerçekleştirdik. Soyut(Abstract) class yapısını kullandık. Getter ve setter methodlarını encapsulation yapısı ile birlikte kullandık. Sonuç olarak diyebiliriz ki programımız nesnel programlama yaklaşımı ile örtüşmektedir.



Şekil-3: Nesnel programlama yaklaşımını gösteren şekil.

D. Algoritmik Yaklaşım(Algorithmic Approach)

Kodlarımızı yazarken projeye algoritmik yöntemler uygulamaya çalıştık. Programcılar arasında yaygın olan “çalışıyorsa elleme” söyleminden tamamı ile sırt çevirdik. Kodun sürdürülebilirliği açısından da oldukça önemli olan algoritmik yaklaşımı mümkün olduğunca kullanmaya çalıştık. Var olan ve projeye en iyi verimi sağlayacak algoritmaları seçmeye çalıştık.



Şekil-4: Algoritmik döngüyü gösteren şekil.

III. DENEYSEL SONUÇLAR

Yapmış olduğumuz projeyi birçok bilgisayarda ve işletim sisteminde ayrı ayrı denediğimizde hem java programlarının taşınabilir(portable) olduğunu bir kez daha görmüş, hem de projemizin sorunsuz bir şekilde çalıştığını teyit etmiş olduk. Ayrıca projelerin istenilen koşulları sağladığı da görülmüştür. Farklı işler sırası ile denenmiş ve başarıya ulaşılmıştır. Proje temel yapıdadır ve istenildiği zaman değişime açık bir projedir. SOLİD prensiplerini kullanmış ve bunda başarıya ulaşılmıştır.

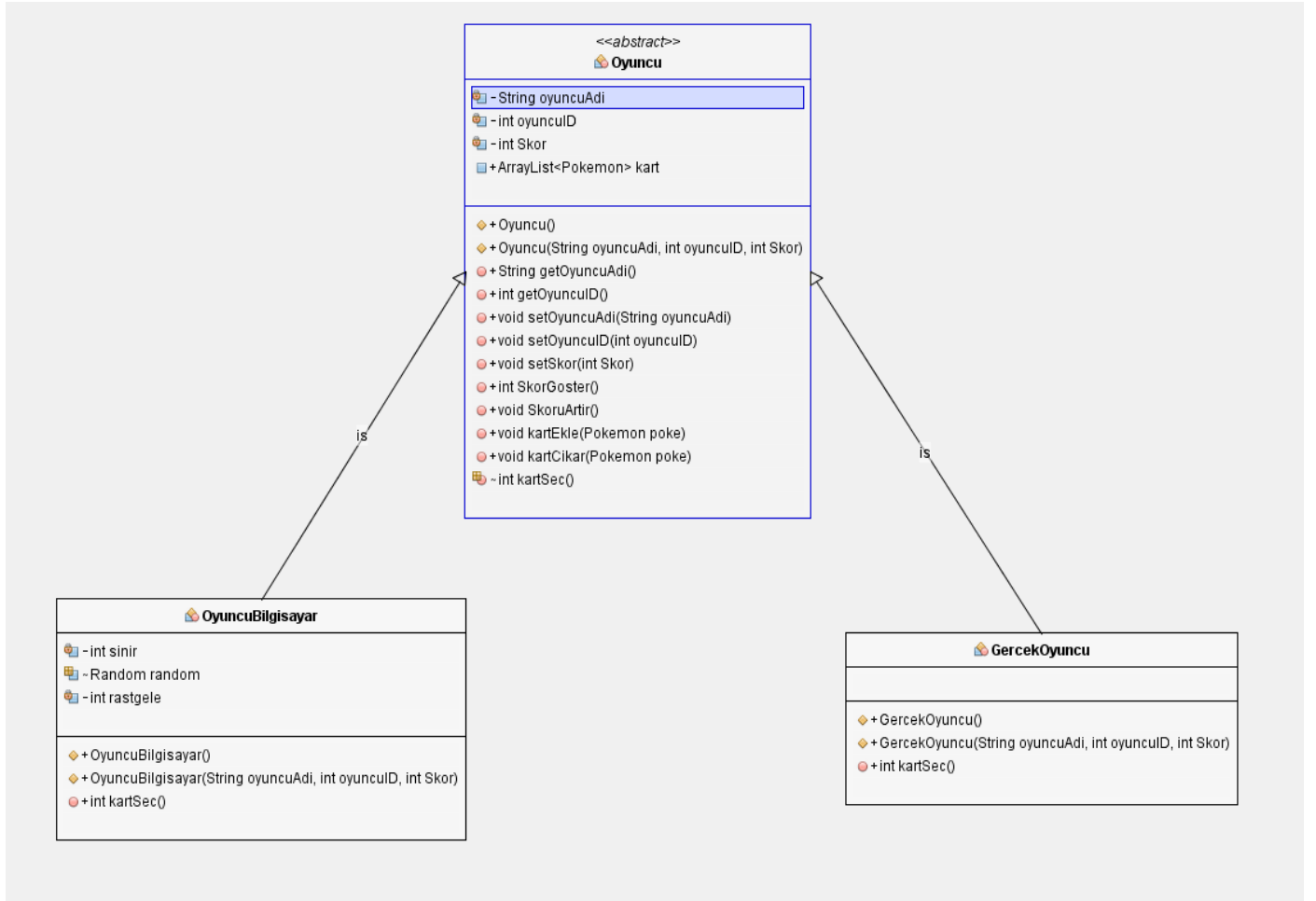
IV. ÖZET VE SONUÇLAR

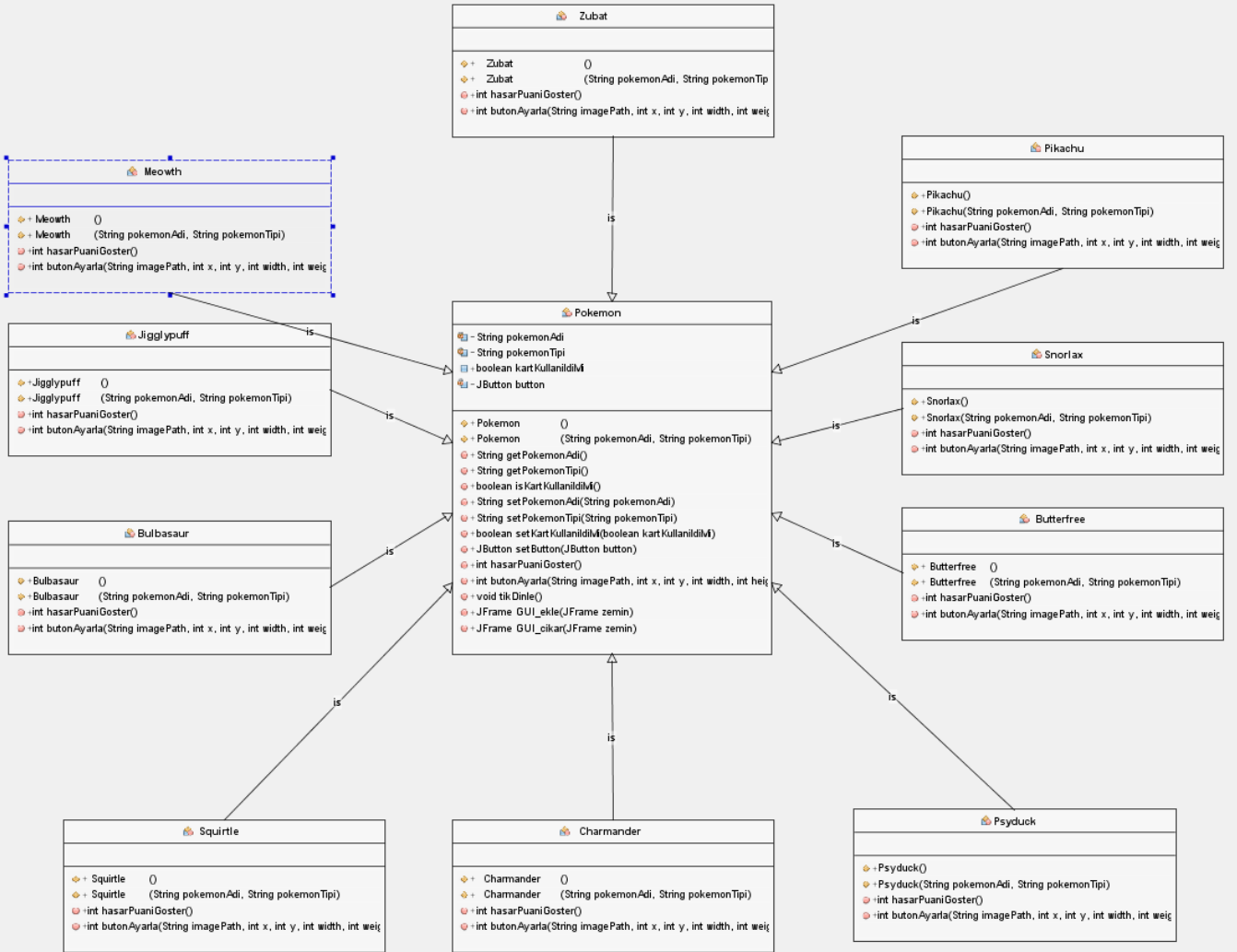
Bu projemizde basit bir pokemon kart oyunu tasarlanmıştır. Projede java programlama dili kullanılmış ve nesnel programlama teknikleri kullanılmıştır. Proje istenilen koşullara göre başarı ile oluşturulmuştur. Deneyleri yapılmış, test edilmiştir. Proje vasıtası ile nesnel programlama tekniklerini gayet iyi bir şekilde öğrenmiş bulunmaktayız.

KAYNAKÇA

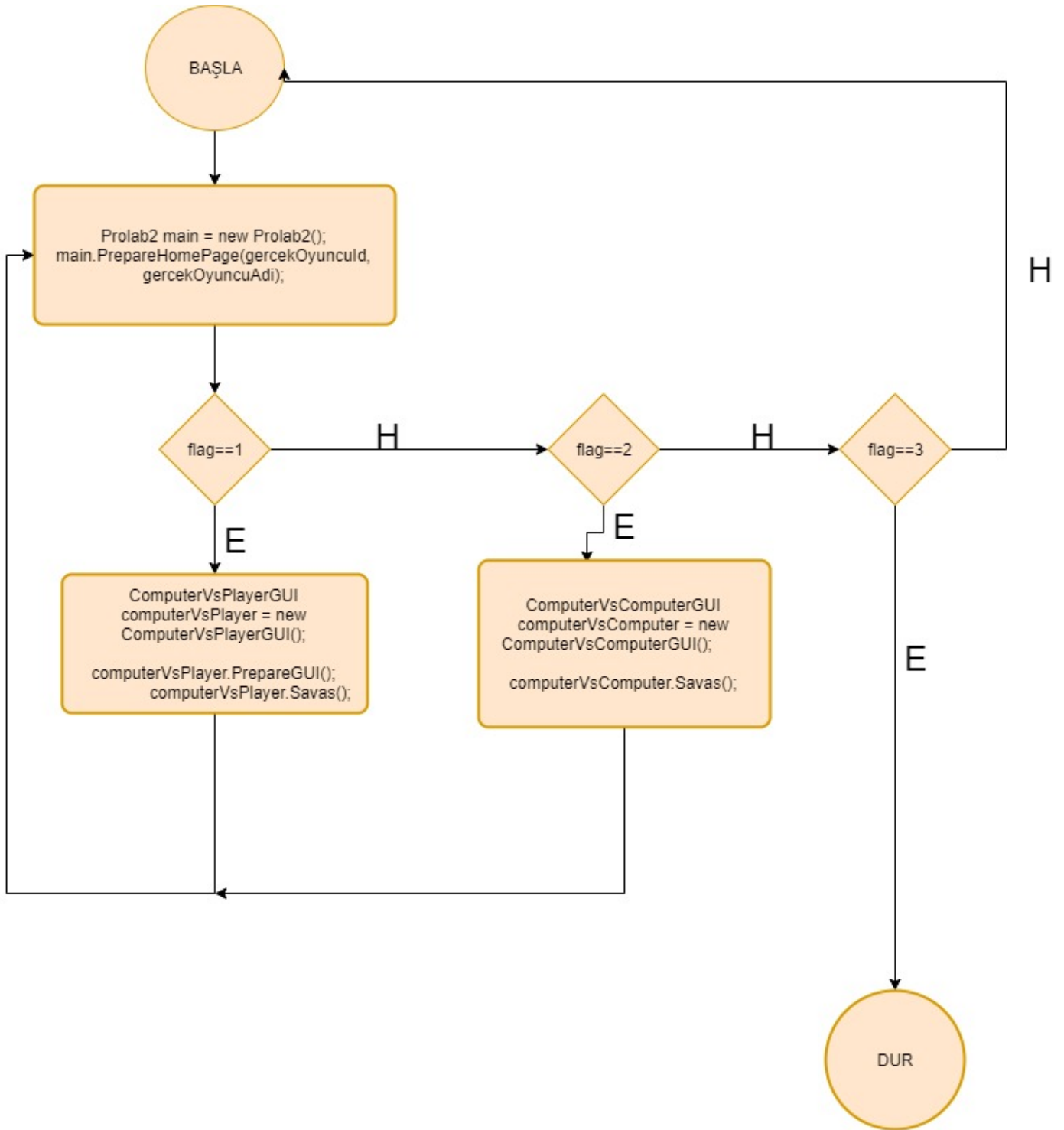
- [1] <https://www.javatpoint.com>
- [2] <https://www.java.com/tr>
- [3] <https://scotch.io/bar-talk/s-o-l-i-d-the-first-five-principles-of-object-oriented-design>
- [4] <https://www.youtube.com/watch?v=Xgj15AMkcvA>
- [5] <https://www.w3schools.com/java>
- [6] <https://medium.com/gokhanyavas/java-oop-kavram%C4%B1-2-1e4daca08a5>
- [7] <https://stackoverflow.com>
- [8] <https://www.tutorialspoint.com/>
- [9] <https://www.geeksforgeeks.org/>
- [10] <https://beginnersbook.com/>

UML SINIF DİYAGRAMLARI



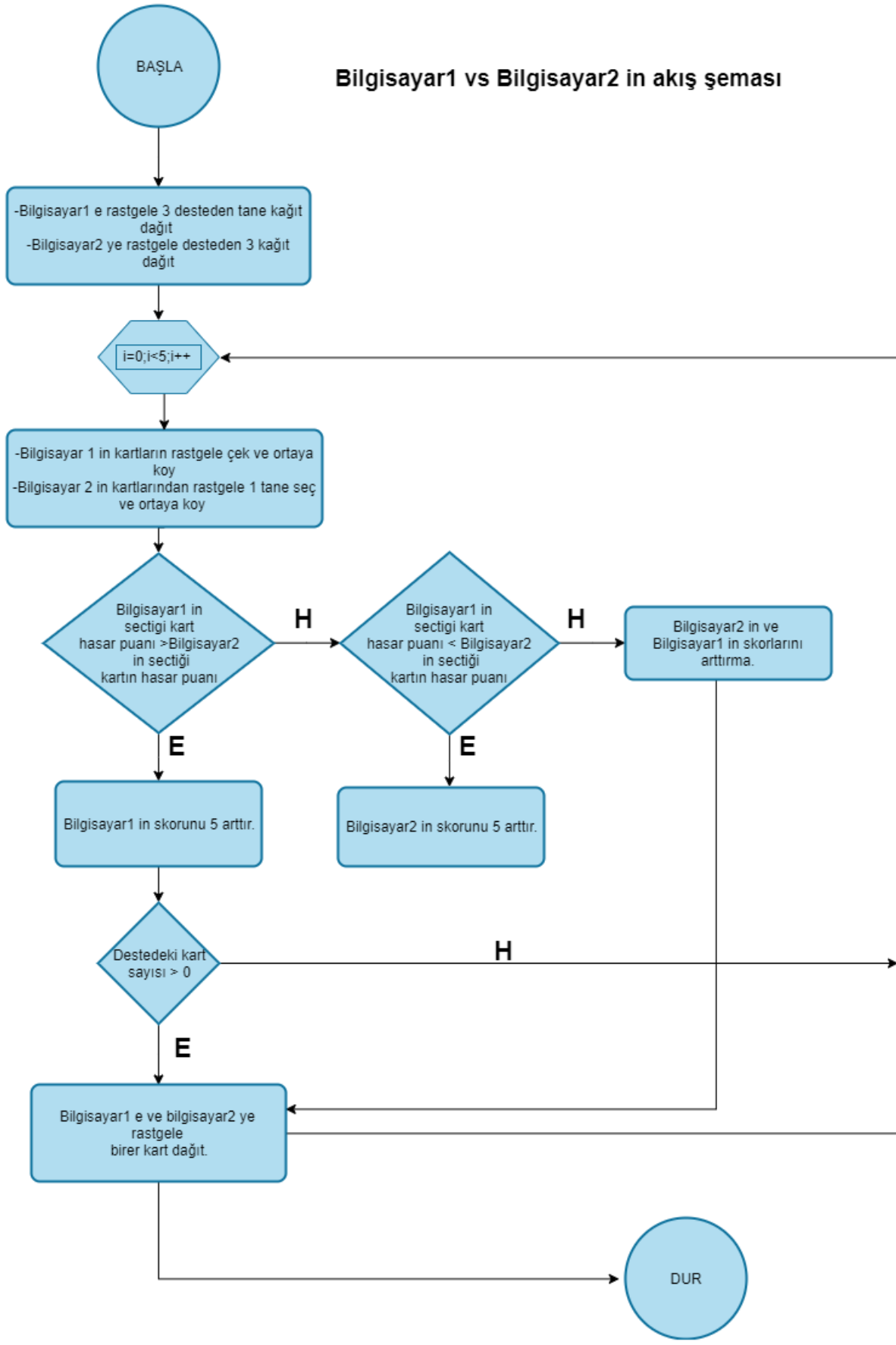


POKEMON CLASSININ UML DİYAGRAMI



Ana sınıfın akış şeması

Bilgisayar1 vs Bilgisayar2 in akış şeması



Bilgisayar Vs Gerçek Oyuncu Sınıfının Akış Şeması

