Nesne Yönelimli Programlama

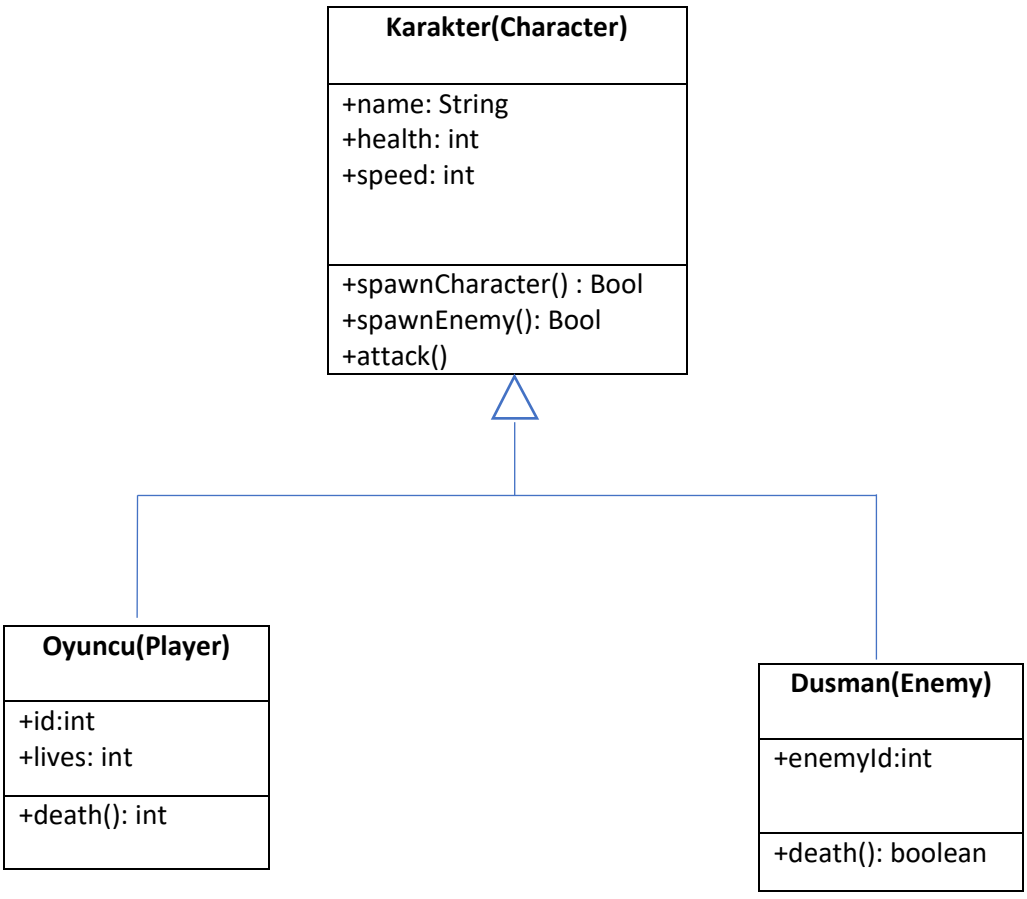
Celil ÖZTÜRK

UML Sınıf Diyagramları

Örnek 1.

Karakter sınıfı oyunda bulunan Enemy ve Player karakterlerinin kalıtıldığı(özelliklerini ve davranışlarını miras aldığı) Base(Parent) sınıftır. Oyun karakterlerinde ortak bulunan health, speed özellikleri Karakter sınıfında tanımlanmıştır. Ayrıca Oyun içinde bulunan tüm karakterleri yaratmak için kullanılan Spawn() Metotları ve ateş etmek için kullanılan Attack() metodu da Karakter sınıfında bulunmaktadır.

Oyuncu(Player) sınıfı Derived(Child) sınıftır. Karakter sınıfından türetilmiştir(inherited). Bu sınıfta kullanılan, oyuncuyu temsil eden Id değeri ve can adetini tutan lives özellikleri sadece Oyuncu sınıfında kullanıldığı için burada tanımlanmıştır. Ancak, Oyuncunun da sahip olduğu name, health, speed özellikleri de, Oyuncu sınıfı Karakter sınıfından türetildiği için kullanılabilmektedir. Ayrıca, Karakter sınıfında tanımlanan attack() metodu da Oyuncu(Player) tarafından kullanılmaktadır.

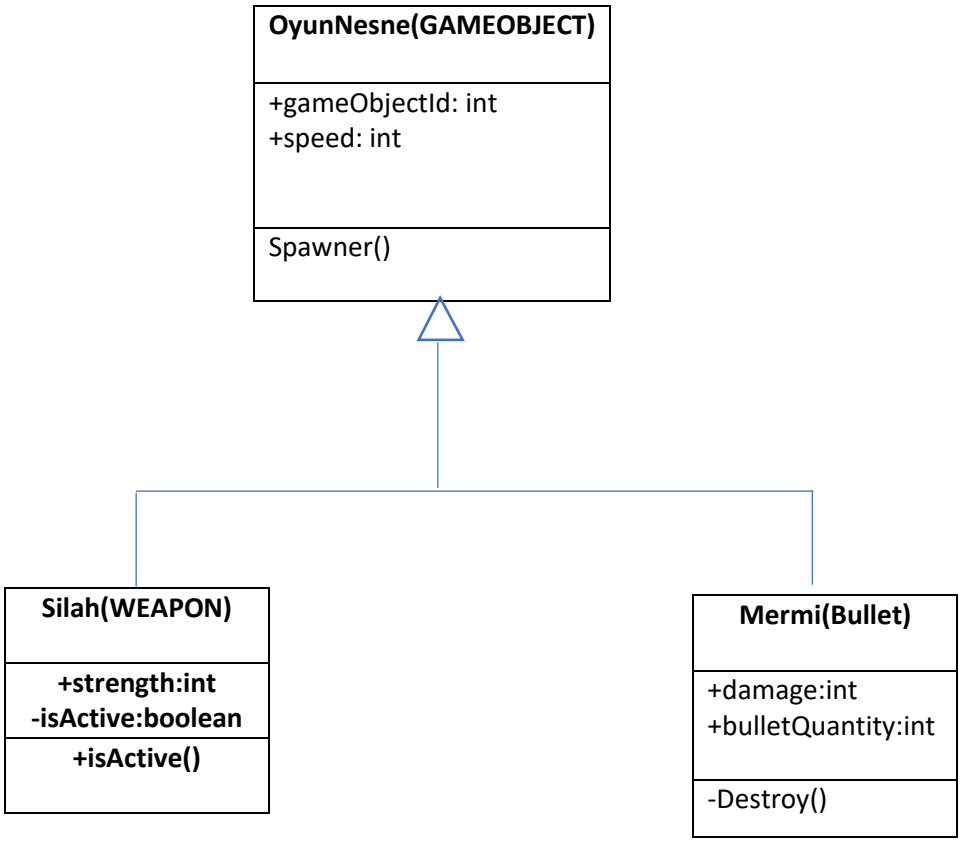


Dusman sınıfı Karakter sınıfından türetilmiş bir sınıftır. Dusman sınıfı Karakter sınıfında tanımlanan name, health, speed, spawnEnemy, attack gibi özellik ve davranışları kullanabilmektedir. Bunlara ek olarak sadece Dusman sınıfına ait olan enemyId özelliği ve death davranışı tanımlanmıştır.

Örnek 2.

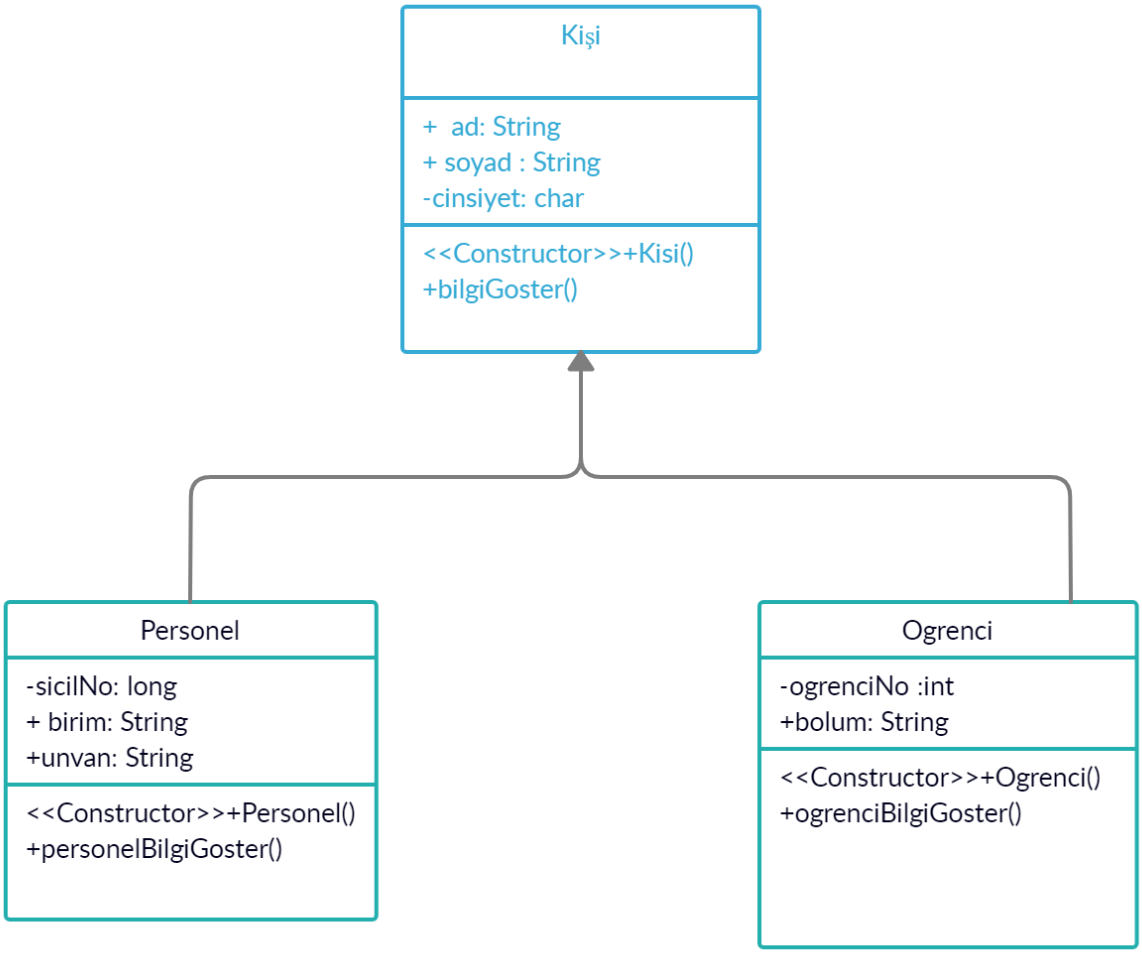
OyunNesne sınıfı bir dijital oyun tasarımında kullanılan, oyun içi nesnelerin ortak özelliklerini bulunduran bir sınıftır. Bu sınıf bir Üst sınıf(Super Class)'tır.

Silah ve **Mermi** sınıfları ise, *OyunNesne* sınıfından türetilen sınıflardır ve OyunNesne sınıfının özelliklerini ve davranışlarını *kalıtım* ile alırlar.

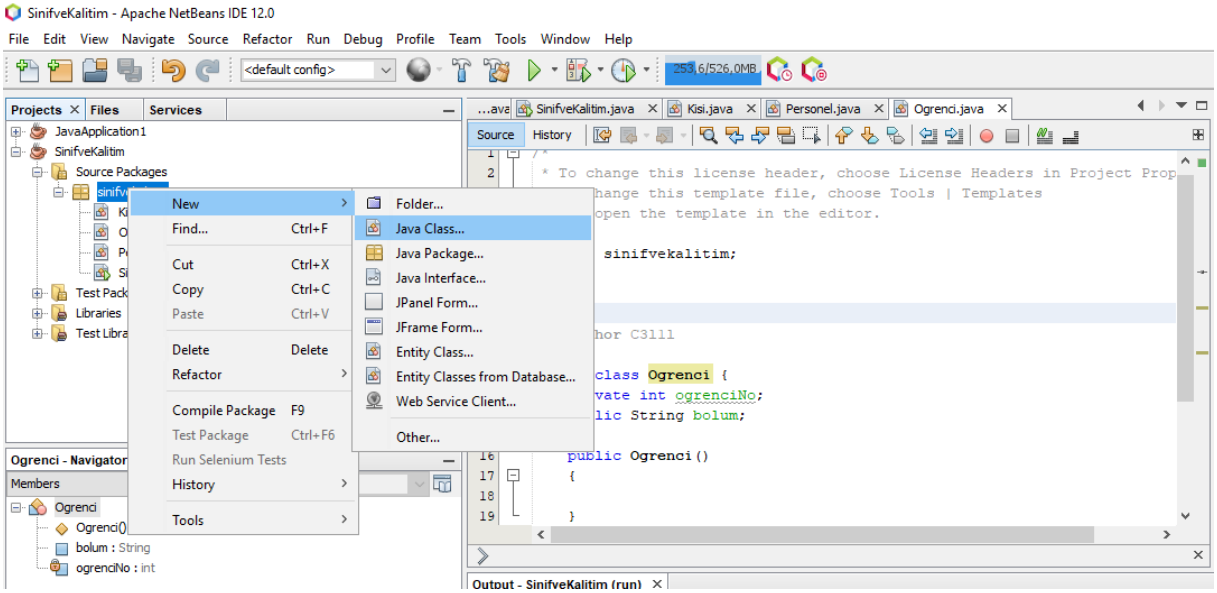


Örnek 3.

Aşağıdaki UML diyagramında Gösterilen sınıflar ile Kalıtım ilişkisi, ve sahip oldukları alanlar(field) ile Metotların Java dili ile Netbeans üzerinde gerçekleştirilmesi.



Netbeans üzerinde yeni bir sınıf oluşturma



Main Java Sınıfı(sinifvekalitim)

```
package sinifvekalitim;


public class SinifveKalitim {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {

        Kisi yenikisi=new Kisi();
        yenikisi.ad="Celil";
        System.out.println(yenikisi.ad);
        yenikisi.bilgiGoster();

        Personel yenipersonel=new Personel();
        yenipersonel.ad="Ahmet";
        System.out.println(yenipersonel.ad);
        yenipersonel.personelBilgiGoster();

    }
}
```

Kişi Sınıfı

```
package sinifvekalitim;


public class Kisi {
    public String ad;
    public String soyad;
    private char cinsiyet;

    public Kisi()
    {
        //Constructor Metot
    }
    public void bilgiGoster()
    {
        //Bilgi göster metodu !
        System.out.println("Bilgi Goster metodu çalıştı !");
    }
}
```

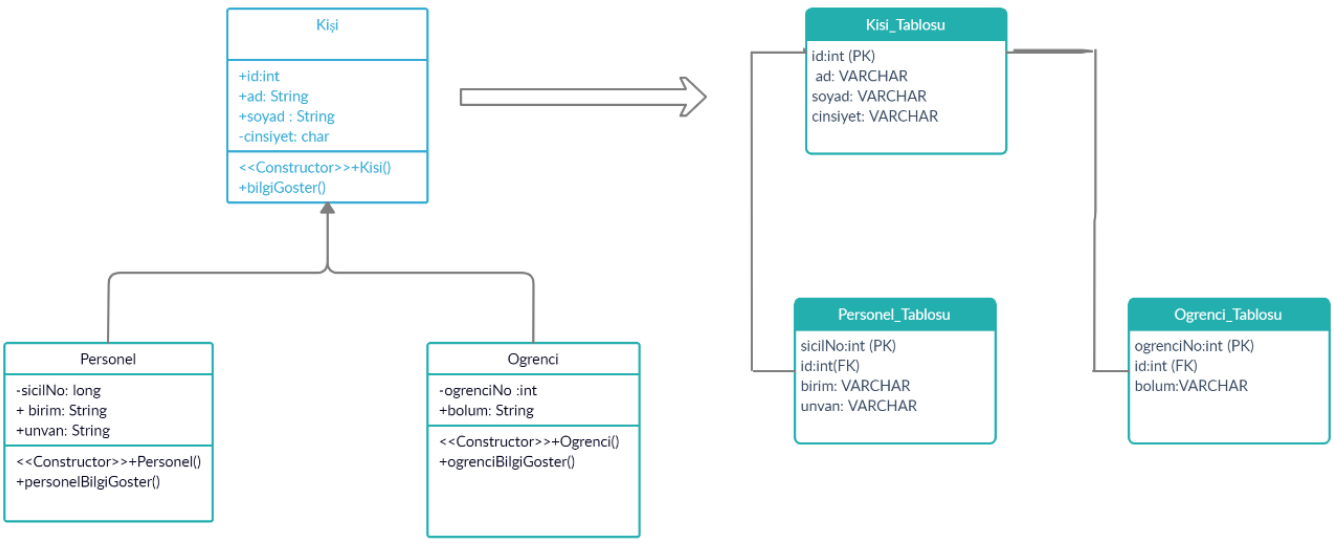
Öğrenci Sınıfı

```java
public class Ogrenci extends Kisi{
    private int ogrenciNo;
    public String bolum;

    public Ogrenci()
    {

    }
    public void ogrenciBilgiGoster()
    {
    System.out.println("Ogrenci Bilgi Goster Metodu çalıştı !");
    }
}
```

Personel Sınıfı

```java
package sinifvekalitim;

import sinifvekalitim.Kisi;

public class Personel extends Kisi {
    private long SicilNo;
    public String birim;
    public String unvan;

    public Personel()
    {

    }
    public void personelBilgiGoster()
    {
        System.out.println("Personel Bilgi Goster Metodu çalıştı !");
    }

}
```

***Örnek 3 uygulamasının Veritabanı tablolarının UML gösterimi aşağıdadır.



Çalışma !.

Aşağıda UML diyagramı verilen sınıfları ve sınıfların özelliklerini Erişim belirteçlerini dikkate alarak Java ile oluşturunuz.