

Bire-bir Sahiplik İlişkisi ile İlgili Sorular:

Soru 1: Bir füzenin kilometre cinsinden menzili ve kilogram cinsinden ağırlığı mevcuttur. Bir füze ya kara hedefleri ya da hava hedefler için tasarlanır. Füzelerin bu özellikleri daha sonradan değiştirilemez. Füze sınıfının (Missile) Java kodunu yazınız.

Soru 2: Her uçağın bir boş ağırlığı ve bir maksimum kalkış ağırlığı vardır. Haddinden fazla yüklenen uçak uçamaz. Bir uçak ancak uçarken füze ateşleyebilir. Üzerinde pilot olmayan uçaklara UAV denir. Bazı UAV'ler füze taşıyamaz, taşıyabilenler ise tek türden ve sınırlı sayıda füze taşır. Verilen tüm bu bilgilere göre UAV sınıfının kaynak kodunu yazınız.

Soru 3: Önceki sorularda yazdığınız sınıfları test etmek üzere main metodu içeren bir sınıfın kaynak kodunu yazınız. Main metodunda kullanıcıdan en azından bir tamsayı bilgi girişi almaya çalışınız. Her sınıftan bir nesne oluşturup bir füze ateşlemek yeterlidir.

Soru 4: Şimdiye kadar kodladığınız sınıfları gösteren ayrıntılı bir UML şeması çiziniz.

Soru 5: Füze ateşlemeye yarayan metodun UML sıralama şemasını çiziniz.

Yanıtlar (kısmi, farklı çözümler de bulunabilir):

Question 1:

```
public class Missile {
    private final int range, weight;
    private final boolean forAirTargets;

    public Missile(int range, int weight, boolean forAirTargets) {
        this.range = range;
        this.weight = weight;
        this.forAirTargets = forAirTargets;
    }
    public int getRange() { return range; }
    public int getWeight() { return weight; }
    public boolean isForAirTargets() { return forAirTargets; }
}
```

Question 2:

```
public class UAV {
    private int emptyWeight, maxWeight;
    private Missile missile;
    private int missileCount, maxMissile;
    private boolean flying;

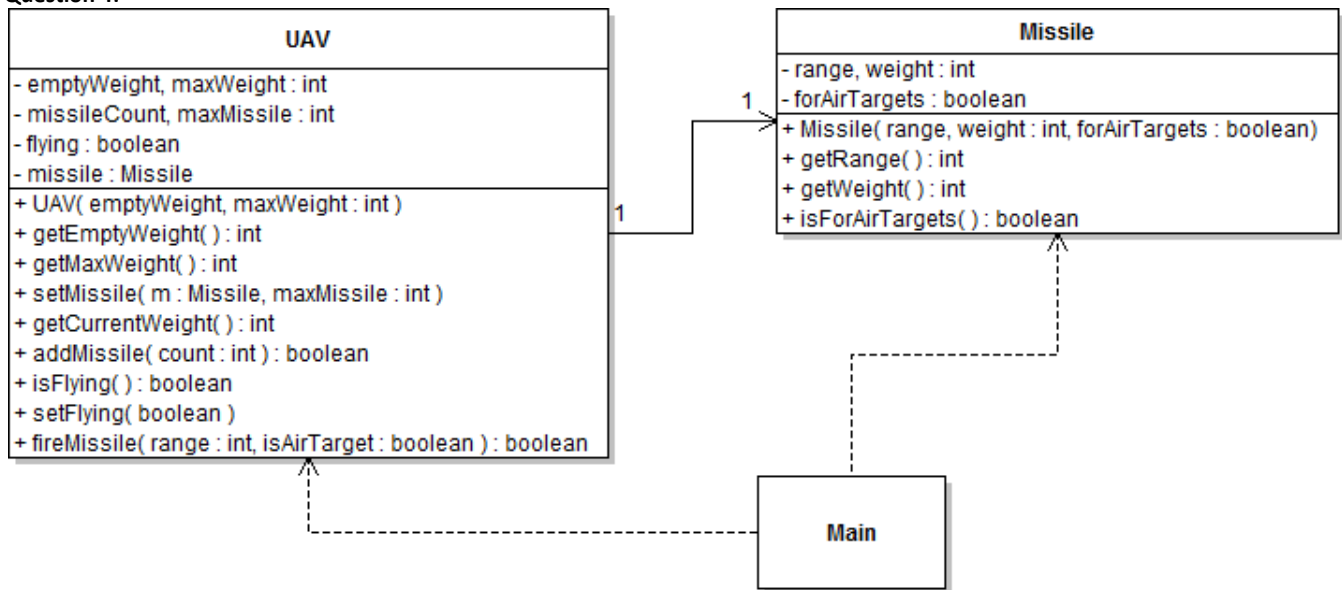
    public UAV(int emptyWeight, int maxWeight) {
        this.emptyWeight = emptyWeight;
        this.maxWeight = maxWeight;
    }
    public int getEmptyWeight() { return emptyWeight; }
    public int getMaxWeight() { return maxWeight; }
    public boolean isFlying() { return flying; }
    public void setFlying(boolean flying) { this.flying = flying; }

    public void setMissile(Missile missile, int maxMissile) {
        this.missile = missile;
        this.maxMissile = maxMissile;
        missileCount = 0;
    }
    public int getCurrentWeight( ) {
        int result = emptyWeight;
        if( missile != null )
            result += missileCount * missile.getWeight();
        return result;
    }
    public boolean addMissile( int count ) {
        if( getCurrentWeight() + count * missile.getWeight() < maxWeight
            && missileCount + count <= maxMissile ) {
            missileCount += count;
            return true;
        }
        return false;
    }
    public boolean fireMissile( int range, boolean isAirTarget ) {
        if( missile != null && missileCount > 0
            && missile.getRange() > range && isFlying() &&
            missile.isForAirTargets() == isAirTarget )
            return true;
        return false;
    }
}
```

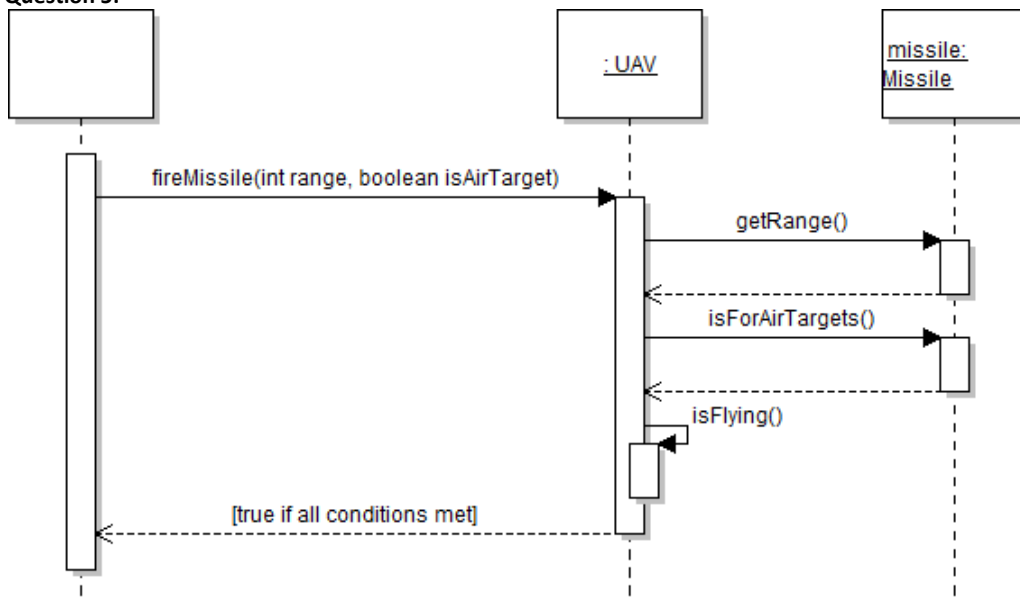
Question 3:

```
import java.util.*;
public class Main {
    public static void main(String[] args) {
        Missile sidewinder = new Missile(1200, 100, true);
        UAV predator = new UAV(5000, 20000);
        Scanner in = new Scanner( System.in );
        System.out.print("How many missiles? ");
        int count = in.nextInt();
        predator.setMissile(sidewinder, count);
        predator.addMissile(count);
        predator.setFlying(true);
        if(predator.fireMissile(100, true))
            System.out.println("Test is successful");
        else
            System.out.println("Test has failed");
        in.close();
    }
}
```

Question 4:

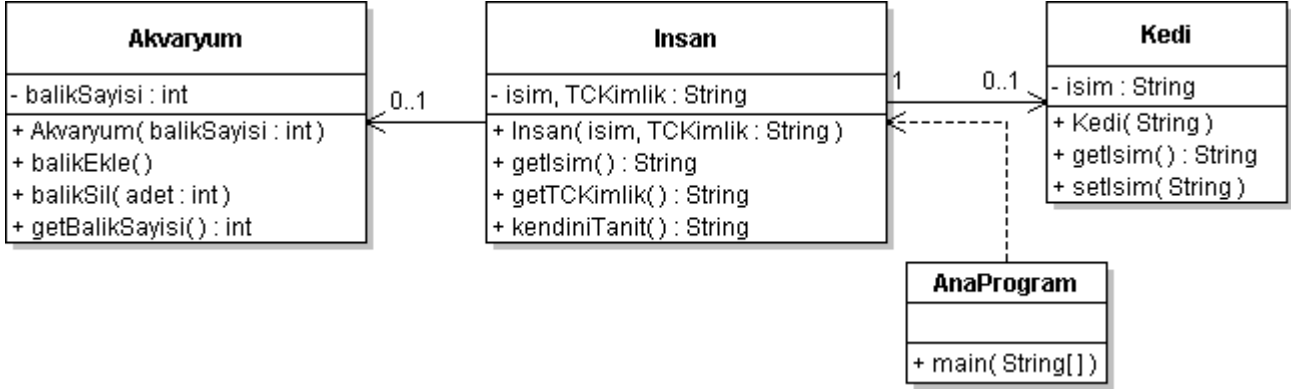


Question 5:



SORULAR

Sorular aşağıdaki UML sınıf şemasına göre yanıtlanmalıdır.



Soru 1: Kedi sınıfının kaynak kodunu yazınız.

Soru 2: Akvaryum sınıfının kaynak kodunu yazınız.

Soru 3: İnsan sınıfının kendiniTanıt metodunun kaynak kodunu yazınız. Bir insan kendini tanıtırken ismini söyler ve ardından evcil hayvanları ile ilgili bilgi verir.

Soru 4: İnsan sınıfının kendiniTanıt metodunun UML etkileşim şemasını çiziniz.

Soru 5: İnsan sınıfının UML sınıf şemasında gösterilmemiş eksik iki metodu vardır. Bu metotların neler olduğunu keşfediniz ve kaynak kodlarını yazınız.

Soru 6: AnaProgram sınıfının main metodunun kaynak kodunu yazınız.

Soru 1: (20) Kedi sınıfının kaynak kodunu yazınız.

```
public class Kedi {  
    private String isim;  
  
    public Kedi(String isim) { this.isim = isim; }  
  
    public String getIsim() { return isim; }  
  
    public void setIsim(String isim) { this.isim = isim; }  
}
```

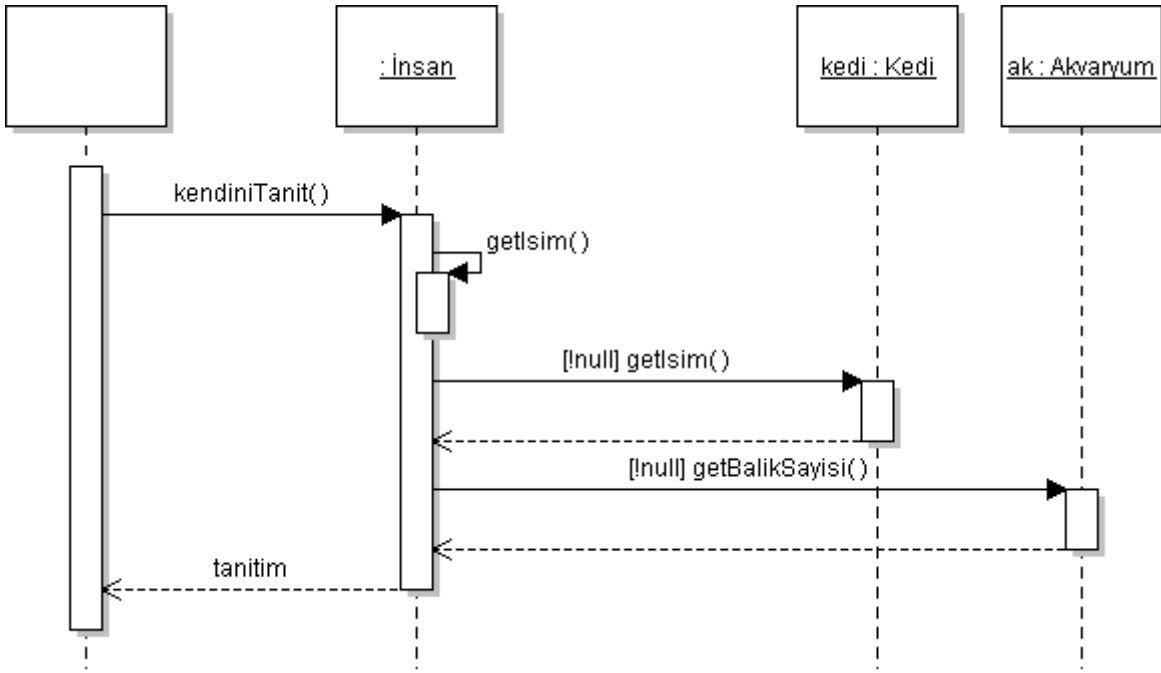
Soru 2: (20) Akvaryum sınıfının kaynak kodunu yazınız.

```
public class Akvaryum {  
    private int balikSayisi;  
  
    public Akvaryum(int balikSayisi) { this.balikSayisi = balikSayisi; }  
  
    public void balikEkle( ) { balikSayisi++; }  
  
    public void balikSil( int adet ) {  
        if( adet <= balikSayisi ) {  
            balikSayisi -= adet;  
        }  
        else  
            balikSayisi = 0;  
    }  
  
    public int getBalikSayisi() { return balikSayisi; }  
}
```

Soru 3: (20) İnsan sınıfının kendiniTanit metodunun kaynak kodunu yazınız.

```
public String kendiniTanit( ) {  
    String tanitim = "Adım: " + isim;  
    if( akvaryum != null )  
        tanitim += ", Balık sayısı: " +  
            akvaryum.getBalikSayisi();  
    if( kedi != null )  
        tanitim += ", Kedimin adı: " +  
            kedi.getIsim( );  
    return tanitim;  
}
```

Soru 4: (15) İnsan sınıfının kendiniTanit metodunun UML etkileşim şemasını çiziniz.



Soru 5: (10) İnsan sınıfının UML sınıf şemasında gösterilmemiş eksik iki metodu vardır.

```
public void setAkvaryum(Akvaryum akvaryum) { this.akvaryum = akvaryum; }
```

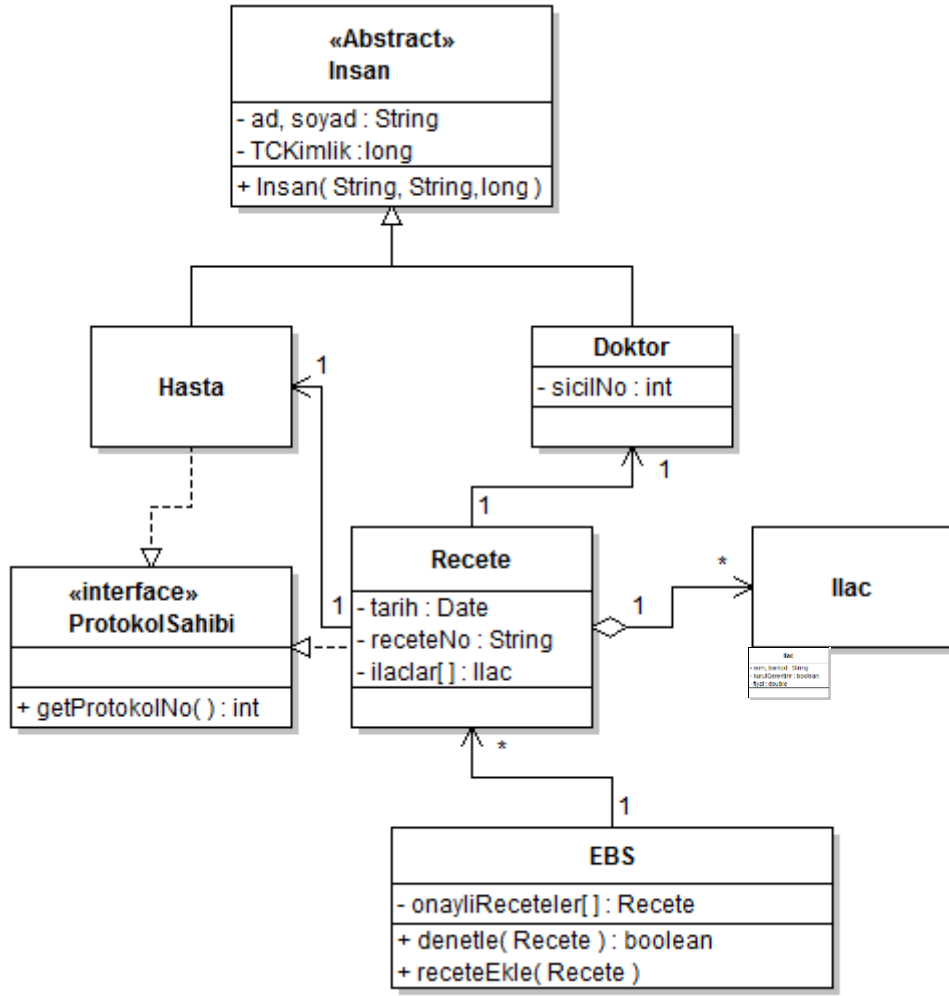
```
public void setKedi(Kedi kedi) { this.kedi = kedi; }
```

Soru 6: (15) AnaProgram sınıfının main metodunun kaynak kodunu yazınız.

```
public static void main( String[] args ) {
    İnsan yunus = new İnsan("Yunus Emre Selçuk", "01234567890");
    Akvaryum ak = new Akvaryum(5);
    yunus.setAkvaryum(ak);
    Kedi torpil = new Kedi("Torpil");
    yunus.setKedi(torpil);
    System.out.println(yunus.kendiniTanit());
}
```

Daha Karmaşık İlişkiler ile İlgili Sorular:

Sorular aşağıdaki UML sınıf şemasına göre yanıtlanmalıdır.



Soru 1: İnsan sınıfının kaynak kodunu yazınız.

Soru 2: ProtokolSahibi arayüzünün kaynak kodunu yazınız.

Soru 3: Hasta sınıfının kaynak kodunu yazınız.

Soru 4: EBS sınıfının kaynak kodunu yazınız.

Soru 5: EBS sınıfına şu metotları ekleyiniz:

- `receteAra(String) : Recete`
 - Hatta Soru 4'te `denetle` metodundan önce `receteAra` kodlayın ve `denetle` metodunda `receteAra` metodunu kullanın.
- Diğerleri henüz kararlaştırılmadı, ders sırasında doğaçlama yapılacaktır.

```

package ornekler1C.EBS;
public abstract class Insan {
    private String ad, soyad;
    private long Tckimlik;
    public Insan(String ad, String soyad, long tCkimlik) {
        this.ad = ad; this.soyad = soyad;
        Tckimlik = tCkimlik;
    }
    public String getAd() { return ad; }
    public String getSoyad() { return soyad; }
    public long getTckimlik() { return Tckimlik; }
}

```

```

package ornekler1C.EBS;
public interface ProtokolSahibi {
    public int getProtokolNo();
}

```

```

package ornekler1C.EBS;
public class Hasta extends Insan implements ProtokolSahibi {
    private int protokolNo;

    public Hasta(String ad, String soyad, long Tckimlik, int protNo) {
        super(ad, soyad, Tckimlik);
        protokolNo = protNo;
    }
    public int getProtokolNo() {
        return protokolNo;
    }
}

```

```

package ornekler1C.EBS;
public class EBS {
    private Recete onayliReceteler[];
    private int receteSayisi, maxRecete;

    public EBS( ) {
        maxRecete = 100;
        onayliReceteler = new Recete[maxRecete];
    }
    private void diziGenislet() {
        Recete yeniReceteler[] = new Recete[maxRecete*2];
        for( int i=0; i<maxRecete; i++ )
            yeniReceteler[i] = onayliReceteler[i];
        onayliReceteler = yeniReceteler;
        maxRecete *= 2;
    }
    public Recete receteAra( String receteNo ) {
        for( Recete recete : onayliReceteler )
            if( recete.getReceteNo().equalsIgnoreCase(receteNo) )
                return recete;
        return null;
    }
}

```



```
public boolean denetle( Recete recete ) {
    if( receteAra( recete.getReceteNo() ) == recete )
        return true;
    return false;
}
public void receteEkle( Recete recete ) {
    if( receteSayisi == maxRecete )
        diziGenislet();
    if( denetle(recete) )
        return;
    onayliReceteler[receteSayisi] = recete;
    receteSayisi++;
}
}
```