

Tek Sınıf Kullanılarak Çözölebilecek Sorular:

Soru 1:

- Komut satırına * işaretlerinden içi dolu (daha kolay) veya içi boş (daha zor) olarak dikdörtgen çizen bir sınıf tasarlayın. Bu sınıf dikdörtgenin kenarlarını nextInt komutu kullanmadan belirlemeye yarayacak en azından bir metod içermelidir. Tasarımınızı bir UML sınıf şeması ile gösteriniz.
- Tasarımınızın Java kaynak kodunu yazınız.
- Sınıfınıza kullanıcının dikdörtgen çizmesine yarayacak bir main metodu ekleyin. Kenar uzunluklarını kullanıcıdan almak için nextInt komutları kullanınız.
- Main metodunun sıralama şemasını çizin.

Soru 2:

Farklı bilgilerin tutarlılığını denetlemek üzere bir hizmet sınıfı yazmak istiyorsunuz. Sınıfınızı aşağıdaki işlevleri sağlayacak şekilde kodlayınız:

- Bir e-posta adresinin geçerli olması için içerisinde bir @ karakteri, onun başında en az 1 harf sonunda ise en az bir nokta dahil 5 karakter bulunmalıdır. Ör. a@b.com
- Bir IP adresi aralarında . olan 4 adet 0-255 arası sayıdan oluşmalıdır. Ör. 123.45.67.8

İpuçları:

- String sınıfının şu metotlarının javadoc'una bakabilirsiniz:
 - public char[]toCharArray()
 - public char charAt(int index)
 - public static String valueOf(char c)
 - public int indexOf(String ch)
 - public int indexOf(String ch, int fromIndex)
 - public int lastIndexOf(String ch)
 - public String substring(int beginIndex, int endIndex)
- Integer sınıfının şu metotlarının javadoc'una bakabilirsiniz
 - public static Integer valueOf(String s)
- Tüm bu değinilen metotların dışında metotlarla da bu işi yapanlar çıkabilir mi?

Soru 1:

```
package onekler1A;

import java.util.*;

public class Dikdortgen {
    private int kenarA, kenarB;

    public Dikdortgen(int kenarA, int kenarB) {
        this.kenarA = kenarA;
        this.kenarB = kenarB;
    }
    public void cizDolu() {
        for( int j=0; j<kenarA; j++ ) {
            for( int i=0; i<kenarB; i++ ) {
                System.out.print("* ");
            }
            System.out.println();
        }
    }
    public void cizBos() {
        for( int i=0; i<kenarA; i++ ) {
            for( int j=0; j<kenarB; j++ ) {
                if( i == 0 || i == kenarA-1 )
                    System.out.print("* ");
                else if( j == 0 || j == kenarB-1 )
                    System.out.print("* ");
                else
                    System.out.print("  ");
            }
            System.out.println();
        }
    }
    public static void main( String args[] ) {
        Scanner in = new Scanner( System.in );
        System.out.print("Enter side A: ");
        int a = in.nextInt();
        System.out.print("Enter side B: ");
        int b = in.nextInt();
        Dikdortgen d = new Dikdortgen(a,b);
        System.out.println("İçi boş dikdörtgen:");
        d.cizBos();
        System.out.println("İçi dolu dikdörtgen:");
        d.cizDolu();
        in.close();
    }
}
```

Soru 2:

```
package onekler1A;
```

```
public class Denetlemeler {
    public static boolean denetleEMail( String eMail ) {
        if( eMail.length() < 7 )
            return false;
        int atKonum = eMail.indexOf("@");
        if( atKonum == -1 || atKonum != eMail.lastIndexOf("@") )
            return false;
        String sol = eMail.substring(0, atKonum);
        String sag = eMail.substring(atKonum+1, eMail.length());
        if( sol.length() < 1 || sag.length() < 5 || sag.indexOf(".") == -1 )
            return false;
        return true;
    }
    public static boolean denetleIP( String adres ) {
        int sayi, konum; String parca;
        adres += ".";
        for( int i = 0; i < 4; i++ ) {
            konum = adres.indexOf(".");
            if( konum == -1 )
                return false;
            parca = adres.substring(0, konum );
            adres = adres.substring(konum+1,adres.length() );
            sayi = Integer.valueOf(parca);
            if( sayi < 0 || sayi > 255 )
                return false;
        }
        return true;
    }
    public static void main( String args[] ) {
        System.out.println(denetleEMail("yunus@gmail.com"));
        System.out.println(denetleEMail("a@b.com"));
        System.out.println(denetleEMail("a@b.c"));
        System.out.println(denetleIP("123.45.67.8"));
        System.out.println(denetleIP("123.45.678.9"));
    }
}
```

Soru 1: Fahrenheit cinsinden verilen sıcaklığı Kelvin'e çeviren bir program yazınız. Aşağıdaki dönüşüm formülünü kullanabilirsiniz:

$$\text{Kelvin} = \frac{\text{Fahrenheit} - 32}{1.8} + 273.15$$

Soru 2: Girilen n adet sayının harmonik ortalamasını bulan bir program yazınız. Aşağıdaki dönüşüm formülünü kullanabilirsiniz:

$$\text{Harmonik} = \frac{n}{\text{say}_1 + \text{say}_2 + \dots + \text{say}_n}$$

Soru 3: Bir öğrencinin çift anadal programına (ÇAP) başvurabilmesi için not ortalamasının en az 2,75 olması gerekir. Ayrıca öğrenci 3. ila 5. dönemleri arasında başvurması gerekmektedir. Bir öğrencinin ÇAP yapıp yapamayacağını belirleyen bir program yazınız.

Soru 4: 4K+1 şeklinde yazılabilen sayılara Hilbert sayısı denir. Örneğin ilk birkaç Hilbert sayısı şunlardır: 5, 9, 13, 17, ... Buna göre bir sayının Hilbert sayısı olup olmadığını belirleyen bir program yazınız.

Soru 5: Kullanıcının girdiği sayının tam bölenlerini listeleyen bir program yazınız. Örneğin kullanıcı 36 girerse çıktı şöyle olmalıdır: 1 2 3 4 6 9 12 18

Soru 6: Bir sayının iki tamsayının kareleri toplamı şeklinde yazılıp yazılamayacağını gösteren bir program yazınız. Örneğin kullanıcı 100 girerse çıktı şöyle olmalıdır:

100 sayısının kare toplamı açılımları:

$$0*0 + 10*10$$

$$6*6 + 8*8$$

$$8*8 + 6*6$$

$$10*10 + 0*0$$

Soru 7: İkinci dereceden bir denklemin köklerini bulan bir program yazın. $Ax^2+Bx+C=0$ denklemini çözmek için delta şöyle hesaplanır: $\Delta=B^2 - 4AC$. Sonra Δ değerine göre kökler şöyle hesaplanır:

$$\Delta > 0 \Rightarrow x_{1,2} = \frac{-b \pm \sqrt{\Delta}}{2a}$$

İki kök x_1, x_2 mevcuttur

$$\Delta = 0 \Rightarrow x_1 = \frac{-b \pm \sqrt{\Delta}}{2a}$$

Tek kök x_1 mevcuttur

$$\Delta < 0 \Rightarrow$$

Gerçek kök yoktur.

Soru 8: Parametre olarak bir dizi alan ve dizideki asal sayıları listeleyen bir metod yazınız ve main metodu ile deneyiniz.

Soru 1:

```
package ornekler1B;
import java.util.*;
public class Fahrenheit2Kelvin {
    public static void main(String args[]) {
        Scanner giris = new Scanner(System.in);
        System.out.print("Fahrenheit cinsinden bir deęer giriniz: ");
        double fahrenheit = giris.nextDouble();
        double kelvin = ((fahrenheit-32)/1.8)+273.15;
        System.out.println("Kelvin karřılıęı: " + kelvin);
        giris.close();
    }
}
```

Soru 2:

```
package ornekler1B;
import java.util.*;
public class HarmonikOrta {
    public static void main(String args[]) {
        Scanner giris = new Scanner(System.in);
        System.out.print("Ka adet sayının harmonik ortalamasını bulacaksınız? ");
        int n = giris.nextInt();
        int toplam = 0;
        for( int i = 0; i < n; i++ ) {
            System.out.print("Tamsayı giriniz: ");
            toplam += giris.nextInt();
        }
        double ortalama = n / toplam;
        System.out.println("Harmonik ortalama: " + ortalama);
        giris.close();
    }
}
```

Soru 3:

```
package ornekler1B;
import java.util.*;
public class CiftAnaDal {
    public static void main(String args[]) {
        Scanner giris = new Scanner(System.in);
        System.out.print("Öęrencinin not ortalamasını girin: ");
        double notOrt = giris.nextDouble();
        if( notOrt >= 2.75 ) {
            System.out.print("Öęrencinin dönemini girin: ");
            int donem = giris.nextInt();
            if( donem >= 3 && donem <= 5 ) {
                System.out.println("Öęrenci AP yapabilir.");
            }
            else {
                System.out.println("Öęrenci AP yapamaz.");
                System.out.println("Dönem gereksinimi karřılanmıyor.");
            }
        }
        else {
            System.out.println("Öęrenci AP yapamaz.");
            System.out.println("Not gereksinimi karřılanmıyor.");
        }
        giris.close();
    }
}
```

```

package ornekler1B;
import java.util.*;
public class CiftAnaDalV2 {
    public static void main(String args[]) {
        Scanner giris = new Scanner(System.in);
        System.out.print("Öğrencinin not ortalamasını girin: ");
        double notOrt = giris.nextDouble();
        System.out.print("Öğrencinin dönemini girin: ");
        int donem = giris.nextInt();
        if( notOrt >= 2.75 && donem >= 3 && donem <= 5 ) {
            System.out.println("Öğrenci ÇAP yapabilir.");
        }
        else {
            System.out.println("Öğrenci ÇAP yapamaz.");
            System.out.println("En az bir gereksinim karşılanmıyor.");
        }
        giris.close();
    }
}

```

Birinci çözüm daha kullanıcı dostudur (daha açıklayıcı olması ve sağlanmayan gereksinim anlaşılınca daha fazla soru sormadığı nedenleri ile). Başka çözümler de düşünülebilir.

Soru 4:

```

package ornekler1B;
import java.util.*;
public class Hilbert {
    public static void main(String args[]) {
        Scanner giris = new Scanner(System.in);
        System.out.print("Bir tamsayı giriniz: ");
        int sayi = giris.nextInt();
        if( sayi % 4 == 1 ) {
            System.out.println("Bu bir Hilbert sayısıdır.");
        }
        else {
            System.out.println("Bu bir Hilbert sayısı değildir.");
        }
        giris.close();
    }
}

```

Soru 5:

```

package ornekler1B;
import java.util.*;
public class TamBolenler {
    public static void main(String args[]) {
        Scanner giris = new Scanner(System.in);
        System.out.print("Bir tamsayı giriniz: ");
        int sayi = giris.nextInt();
        for( int i=1; i<=sayi/2; i++ )
            if( sayi % i == 0 )
                System.out.print( i + " " );
        giris.close();
    }
}

```

Soru 6:

```
package ornekler1B;
import java.util.*;
public class KareToplamlar {
    public static void main(String args[]) {
        Scanner giris = new Scanner(System.in);
        System.out.print("Bir tamsayı giriniz: ");
        int sayi = giris.nextInt();
        System.out.println(sayi+" sayısının kare toplamı açılımları:");
        for( int i=0; i<=sayi/2; i++ )
            for( int j=0; j<=sayi/2; j++ )
                if( i*i + j*j == sayi )
                    System.out.println( i+"*"+i+"+"+j+"*"+j );
        giris.close();
    }
}
```

Soru 7:

```
package ornekler1B;
import java.util.*;
public class DenklemCozum {
    public static void main(String args[]) {
        Scanner giris = new Scanner(System.in);
        System.out.println("İkinci derece denklem çözücü.");
        System.out.print("a katsayısını girin: ");
        int a = giris.nextInt( );
        System.out.print("b katsayısını girin: ");
        int b = giris.nextInt( );
        System.out.print("c katsayısını girin: ");
        int c= giris.nextInt( );
        double delta = b*b - 4*a*c;
        if( delta > 0 ) {
            double x1 = (-b+Math.sqrt(delta))/(2*a);
            double x2 = (-b-Math.sqrt(delta))/(2*a);
            System.out.println("İki kök mevcut: " + x1 + " ve " + x2);
        }
        else if( delta == 0 ) {
            double x = (-b+Math.sqrt(delta))/(2*a);
            System.out.println("Bir kök mevcut: " + x);
        }
        else
            System.out.println("Gerçek sayı kök yoktur.");
        giris.close();
    }
}
```

Soru 8:

```
package ornekler1B;
public class AsalDizi {
    public static void asalListele( int[] sayilar ) {
        for( int i=0; i<sayilar.length; i++ ) {
            boolean isPrime = true;
            for( int j=2; j<=sayilar[i]/2; j++ ) {
                if( sayilar[i] % j == 0 ) {
                    isPrime = false; break;
                }
            }
            if( isPrime )
                System.out.print(sayilar[i]+" ");
        }
    }
    public static void main(String[] args) {
        int dizi[] = { 4, 19, 45, 17, 60 };
        asalListele( dizi );
    }
}

package ornekler1B;
public class AsalDiziV2 {
    public static void asalListele( int[] sayilar ) {
        for( int i=0; i<sayilar.length; i++ ) {
            if( asalMi(sayilar[i]) )
                System.out.print(sayilar[i]+" ");
        }
    }
    public static boolean asalMi( int sayi ) {
        for( int i=2; i<=sayi/2; i++ ) {
            if( sayi % i == 0 ) {
                return false;
            }
        }
        return true;
    }
    public static void main(String[] args) {
        int dizi[] = { 4, 19, 45, 17, 60 };
        asalListele( dizi );
    }
}
```

İlk şekli her işi tek metotta yaparken ikinci şekli ek bir metotla daha iyi görev bölümlendirme yapmıştır.

Soru 1. 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;

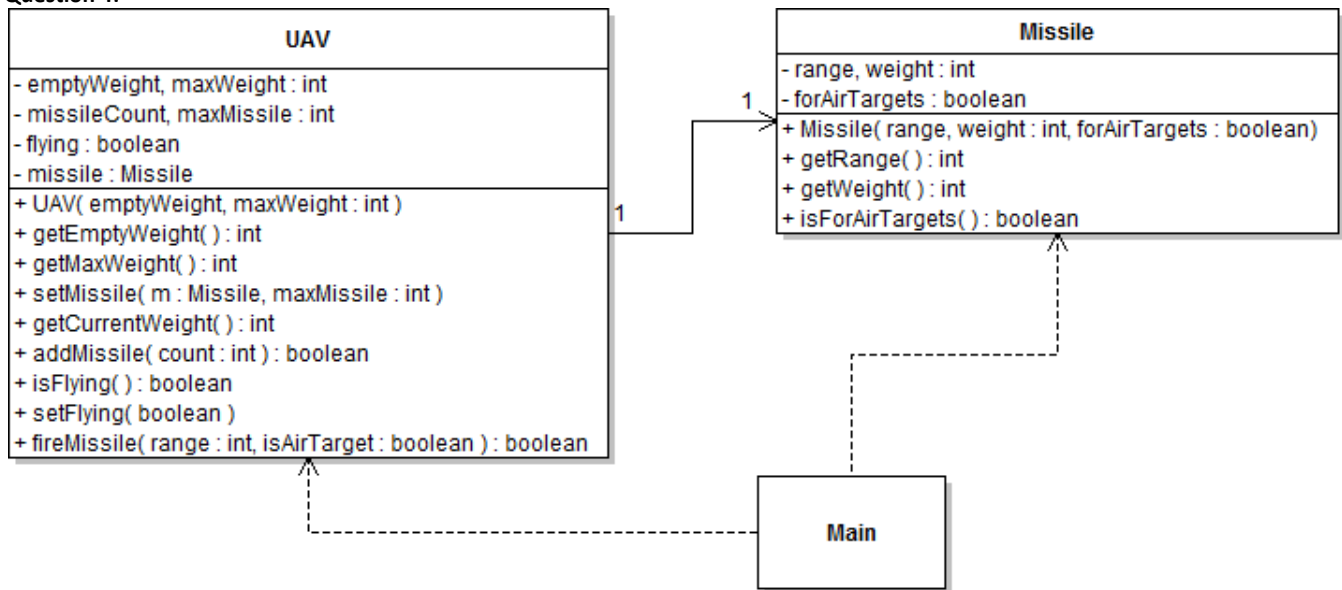
    public UAV(int emptyWeight, int maxWeight) {
        this.emptyWeight = emptyWeight;
        this.maxWeight = maxWeight;
    }
    public int getEmptyWeight() { return emptyWeight; }
    public int getMaxWeight() { return maxWeight; }
    public boolean isFlying() { return flying; }
    public void setFlying(boolean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

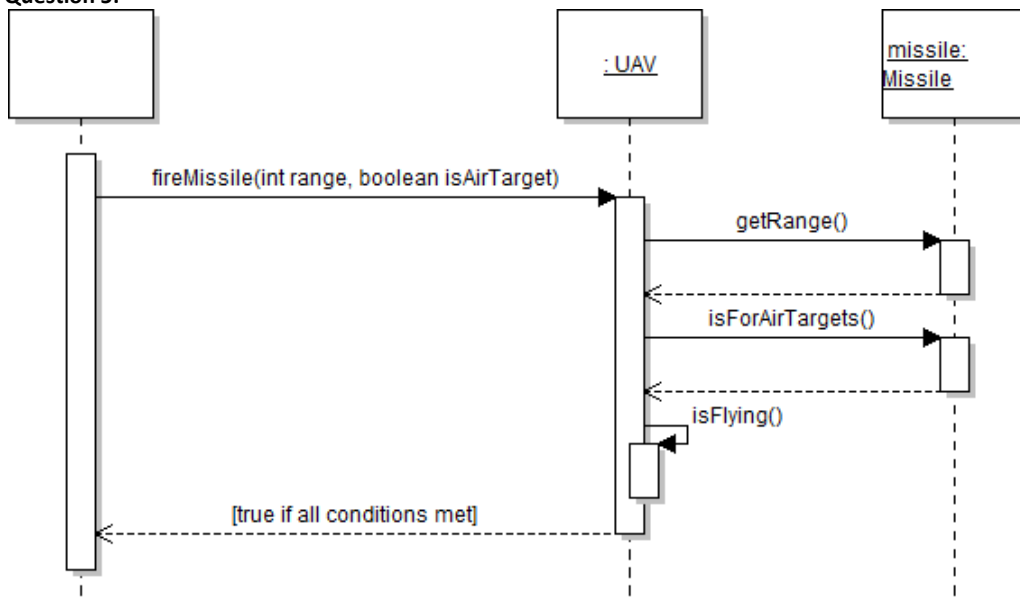
Question 3:

```
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:

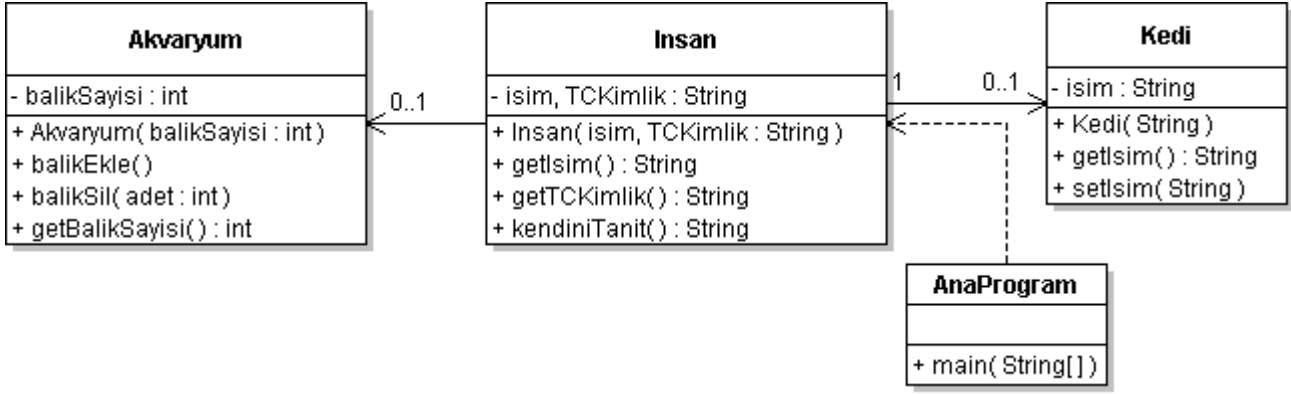


Question 5:



SORU II. Bire-bir Sahiplik İlişkisi ile İlgili 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 `kendiniTanit` metodunun kaynak kodunu yazınız. Bir insan kendini tanıtırken ismini söyler ve ardından evcil hayvanları ile ilgili bilgi verir.

Soru 4: İnsan sınıfının `kendiniTanit` 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; }  
}
```

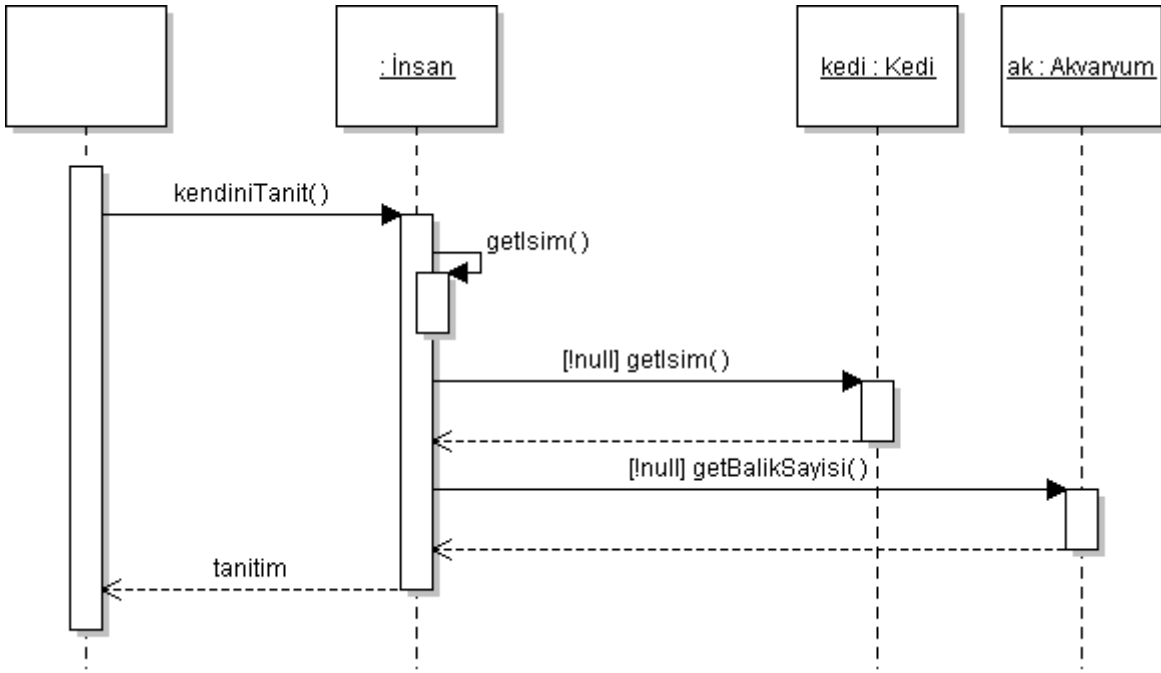
Soru 2: (20) Akvaryum sınıfının kaynak kodunu yazınız.

```
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; }
```

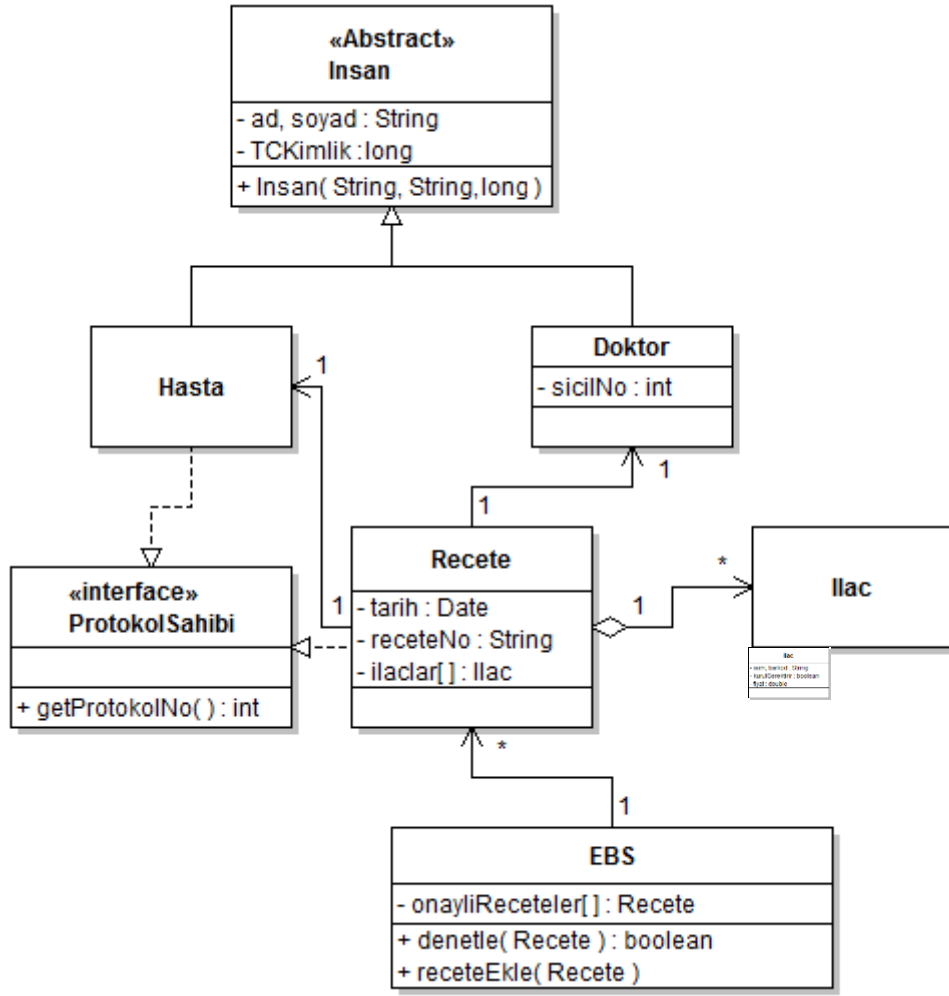
```
public void setKedi(Kedi kedi) { this.kedi = kedi; }
```

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());
}
```

Soru III. Bire-Çoklu Sahiplik ve Kalıtım İlişkileri 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.

```

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; }
}

public interface ProtokolSahibi {
    public int getProtokolNo();
}

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;
    }
}

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++;  
}  
}
```