

Öğrenci No:	Adı, Soyadı:						İmza:
Puanlar:	1	2	3	4	5	6	
	30	10	20	30	30		

### SORULAR

**Soru 1:** Tarih verisi için kullanılmak üzere gün-ay-yıl verisi içerecek şekilde bir sınıf kodlayınız. Şu gereksinimleri karşılamayı unutmayınız:

- Bir tarih nesnesi kendisinin bir başka tarihten önce mi olduğunu geri döndürebilmelidir.
- Bir tarih nesnesi kendisinin bir başka tarih ile eşit olup olmadığını geri döndürebilmelidir.
- Tarih tutarlılığı denetimini ihmal etmeyiniz. Geçersiz tarih girilmişse tarih nesnesi sistemin o anki tarihini göstermelidir. Geçerli bir tarih değerinde:
  - Ay değeri 1-12 arasında olmalıdır.
  - Ocak, Mart, Mayıs, Temmuz, Ağustos, Ekim, Aralık ayları 31 gün çeker. Nisan, Haziran, Eylül, Kasım ayları 30 gün çeker.
  - Artık yıllarda Şubat ayı 29 çeker. Artık yıllar 4 rakamının katı olan yıllardır ancak 100'ün katı olan yıllardan sadece 400'e kalansız olarak bölünebilenler artık yıldır.

**Soru 2:** Çalışan adlı bir sınıf kodlayınız. Bir çalışanın adı, soyadı, işe giriş ve çıkış tarihleri ile sicil no'su saklanmalıdır.

**Soru 3:** Görev adlı bir sınıf kodlayınız. Bir görevin metinsel bir tanımı, Çalışan türünden bir sorumlusu, başlangıç ve bitiş tarihleri ile ID'si tanımlanmalıdır. Bir görevin tamamlanıp tamamlanmadığı da ayrı bir alan olarak tanımlanmalıdır, çünkü bitiş tarihi geçtiği halde tamamlanmamış görevler olabilir.

Şirket
- isim : String
- çalışanlar : Çalışan[]
- görevler : Görev[]
- çalışanSayısı, görevSayısı : int
+ Şirket( isim : String )
+ çalışanEkle( kişi : Çalışan ) : boolean
+ çalışanAra( sicilNo : int ) : Çalışan
+ görevEkle( görev : Görev ) : boolean
+ görevAra( görevID : int ) : Görev
- çalışanlarDizisiniGenişlet( artım : int )
- görevlerDizisiniGenişlet( artım : int )
+ görevTamamla( görevID : int ) : boolean
+ gecikenGörevleriListele()

**Soru 4:** Çalışanlar ve görevler içerecek Şirket isimli bir sınıfın şeması solda verilmiştir. Buna göre Şirket sınıfının sadece aşağıda listelenen metotlarını kodlayınız:

- gecikenGörevleriListele: Bitiş tarihi bugünün tarihinden önce olmasına rağmen tamamlanmamış olan görevler, sorumlusu ile birlikte listelenmelidir.
- görevEkle: çalışanlar dizisine yeni bir görev ekler. Bir şirkette aynı ID'ye sahip birden fazla görev bulunmamalıdır. Eklenecek görevin sorumlusu da şirketin çalışanları içinde yer almalıdır. Eğer dizide yer kalmamışsa dizi genişletilir. Metot başarılı olduğu takdirde true döndürülür.

aynca ilkel üye alanlar için getter metotları

**Soru 5:** Şimdiye dek yazdığınız tüm sınıfları içeren UML sınıf şemasını çizin. Şirket sınıfının üyelerini göstermeyiniz.

**İPUCU:**Sistemin şu anki tarihini java.util paketindeki Calendar sınıfından yararlanarak öğrenebilirsiniz:

```
Calendar simdi = Calendar.getInstance();
int gun = simdi.get(Calendar.DAY_OF_MONTH); //1-31 arası değer döner
int ay = simdi.get(Calendar.MONTH); //0-11 arası değer döner
int yıl = simdi.get(Calendar.YEAR);
```

## Soru 1:

```
import java.util.*;
public class Tarih {
    private int gun, ay, yil;

    public Tarih(int gun, int ay, int yil) {
        int maxGun = 30;
        if( ay == 2 && yil % 4 == 0 ) {
            maxGun = 29;
            if( yil % 100 == 0 && yil % 400 != 0 )
                maxGun = 28;
        }
        if( (ay <= 7 && ay % 2 == 1) || (ay >= 8 && ay % 2 == 0) )
            maxGun = 31;
        if( gun >= 1 && gun <= maxGun && ay >= 1 && ay <= 12 ) {
            this.gun = gun; this.ay = ay; this.yil = yil;
        }
        else {
            Calendar simdi = Calendar.getInstance();
            this.gun = simdi.get(Calendar.DAY_OF_MONTH);
            this.ay = simdi.get(Calendar.MONTH)+1;
            this.yil = simdi.get(Calendar.YEAR);
        }
    }

    public int getGun() { return gun; }
    public int getAy() { return ay; }
    public int getYil() { return yil; }
    public boolean onceMi( Tarih diger ) {
        if( yil < diger.yil )
            return true;
        if( yil == diger.yil && ay < diger.ay )
            return true;
        if( yil == diger.yil && ay == diger.ay && gun < diger.gun )
            return true;
        return false;
    }
    public boolean esitMi( Tarih diger ) {
        if( gun == diger.gun && ay == diger.ay && yil == diger.yil )
            return true;
        return false;
    }
    public String toString(){
        return gun + "/" + ay + "/" + yil;
    }
}
```

## Soru 2:

```
public class Calisan {
    private String ad, soyad;
    private int sicilNo;
    private Tarih iseGiris, istenCikis;

    public Calisan(String ad, String soyad, int sicilNo) {
        this.ad = ad; this.soyad = soyad; this.sicilNo = sicilNo;
    }
    public String getAd() { return ad; }
    public String getSoyad() { return soyad; }
    public int getSicilNo() { return sicilNo; }
    public Tarih getIseGiris() { return iseGiris; }
    public Tarih getIstenCikis() { return istenCikis; }

    public void setIseGiris(Tarih iseGiris) { this.iseGiris = iseGiris; }
    public void setIstenCikis(Tarih istenCikis) { this.istenCikis = istenCikis; }

    public String toString() {
        return ad + " " + soyad + " (Sicil no:" + sicilNo + ")";
    }
}
```

### Soru 3:

```
public class Gorev {
    private String tanim;
    private Calisan sorumlu;
    private Tarih baslangic, bitis;
    private int gorevID;
    private boolean tamam;
    public Gorev(String tanim, Calisan sorumlu, Tarih baslangic,
        Tarih bitis, int gorevID) {
        this.tanim = tanim; this.sorumlu = sorumlu;
        this.baslangic = baslangic;
        this.bitis = bitis; this.gorevID = gorevID;
    }
    public String getTanim() { return tanim; }
    public Calisan getSorumlu() { return sorumlu; }
    public Tarih getBaslangic() { return baslangic; }
    public Tarih getBitis() { return bitis; }
    public int getGorevID() { return gorevID; }
    public boolean tamamMi() { return tamam; }
    public void setTamam( boolean durum ) { tamam = durum; }
    public void setBitis( Tarih bitis ) {
        if( baslangic.onceMi(bitis) || baslangic.esitMi(bitis) )
            this.bitis = bitis;
        else
            bitis = new Tarih(0, 0, 0);
        //Tarih kurucusunu öyle kodladık ki 0,0,0 verince bugünü ayarlıyor.
    }
}
```

### Soru 4:

```
public boolean gorevEkle( Gorev gorev ) {
    if( gorevSayisi >= gorevler.length )
        gorevlerDizisiniGenislet(10);
    if( gorevAra(gorev.getGorevID()) != null )
        return false;
    gorevler[gorevSayisi] = gorev;
    gorevSayisi++;
    return true;
}
public void gecikenGorevleriListele( ) {
    Calendar simdi = Calendar.getInstance();
    Tarih bugun = new Tarih(simdi.get(Calendar.DAY_OF_MONTH),
        simdi.get(Calendar.MONTH)+1, simdi.get(Calendar.YEAR));
    System.out.println("Geciken görevler: ");
    for( int i=0; i<gorevSayisi; i++ ) {
        if( gorevler[i].tamamMi() == false &&
            gorevler[i].getBitis().onceMi(bugun) ) {
            System.out.println("Görev ID ve tanımı: "+ gorevler[i].getGorevID()
                +" "+gorevler[i].getTanim());
            System.out.println("Bitmesi gereken tarih: "+ gorevler[i].getBitis() );
            System.out.println("Sorumlusu: "+ gorevler[i].getSorumlu() );
        }
    }
}
```

### Soru 5:

