

# BLM19103

## Bilgisayar Programlama 1

Dr. Öğr. Üyesi Musa Aydın  
Arş. Gör. Muhammet Alkan  
Arş. Gör. Zeki Kuş

# Soru 1

- Şekil isminde abstract sınıf oluşturunuz.
- Şekil sınıfı ad ve renk özelliğine sahiptir.
- Aynı zamanda Şekil sınıfı bilgileriYazdir() isminde gövdesiz, abstract bir metoda da sahiptir.
- Şekil sınıfından kalıtım alan Daire ve Dikdörtgen sınıflarını oluşturunuz.
- Daire sınıfı renk ve yarıçap özelliklerine sahiptir.
- Dikdörtgen sınıfı renk, kısıkenar ve uzunkenar olmak üzere 3 farklı özelliğe sahiptir.
- Bu sınıflar için gerekli olan bilgileriYazdir() metotlarını doldurunuz.

# Soru 1 - Devam

- IAlanHesaplayici ve ICevreHesaplayici isimlerinde iki tane Interface oluşturunuz.
- Bu arayüzlere aşağıdaki gövdesiz metotları ekleyiniz:
  - IAlanHesaplayici: double alanHesapla()
  - ICevreHesaplayici: double cevreHesapla()
- Dikdortgen ve Daire sınıflarına bu iki arayüzü implements anahtar kelimesi ile entegre ediniz.
- İlgili sınıflar için doğru bir şekilde alanHesapla ve cevreHesapla metotlarının içini doldurunuz.
- Test sınıfında sekilListesi isminde Sekil sınıfından tureyen nesneleri Kabul eden bir object array oluşturunuz.
- Bu listedeki her bir seklin **bilgileriYazdir()** metodunu çağırınız.
- Yazılan tüm metotların test edildiği bir Test sınıfı yazınız.

# Soru 1 - Devam

- **SinirKontrol** isminde **Exception** sınıfından kalıtım alan bir sınıf oluşturunuz.
- **alanHesapla** veya **cevreHesapla** metotlarında bu işlemler için kullanılan parametrelerin (yaricap, kısa\_kenar, uzun\_kenar) iki basamaklı olması istenmiyor. İki basamaktan büyük bir değer girildiğinde **SinirKontrol** sınıfında yazılan hatayı fırlatınız ve ilgili özellik veya özellikleri tavan değerine (9) çekiniz
- **NullElemanKontrol** isminde **Exception** sınıfından kalıtım alan bir sınıf oluşturunuz.
- Test sınıfında **bilgileriYazdir** metodu çağırıldığında sekilListesinde null eleman varsa **NullElemanKontrol** sınıfında yazılan hatayı fırlatınız ve dizinin bu elemanına **bilgileriYazdir** metodunun sorunsuz bir şekilde çalışması için gereken parametreleri (ad, renk, **yaricap, kısa\_kenar, uzun\_kenar (Sekle göre belirlenecek)**) rastgele bir şekilde atayınız ve tekrar **bilgileriYazdir()** metodunu çalıştırınız.

# HATIRLATMA

- Bu hafta yapılacak örneklerde ArrayList **kullanılmayacak**.
- Bunun yerine object arrayler kullanılacak.
- Kullanılan object arraylerin boyutu minimum 5, maksimum 10 olmalıdır.

# Soru 2

1921221007  
200221007

FURKAN SELİM  
MELİKE ÖZTÜRK

SALİHOĞLU

- Personel isminde abstract sınıf oluşturunuz.
- Personel sınıfı id, adSoyad, maas, calismaYili, unvan özelliklerine sahiptir.
- Personel sınıfı **bilgileriYazdir()** isminde gövdesiz, abstract bir metoda da sahiptir.
- Personel sınıfından kalıtım alan **Muhendis**, **Isci** ve **Stajyer** sınıflarını oluşturunuz.
- Diğer sınıflardan farklı olarak **Muhendis** sınıfı **ekipListesi** isminde (Personel sınıfı tipinde nesneleri kabul eder) object arraye sahiptir. Bu listeye Muhendis sınıfı tipinde nesne eklenemez. En fazla 5 adet personel eklenebilir.
- **Isci** ve **Stajyer** sınıflarında **bilgileriYazdir()** metodu çağrıldığında ilgili personelin tüm bilgileri ekrana yazdırılır. **Muhendis** sınıfından ise bu bilgilere ek olarak ekipListesinde olan kişilerin bilgileri ve **ekipListesinde** kaç adet **Isci** ve **Stajyer** olduğu da ekrana yazdırılır.

# Soru 2 - Devam

- **IZamHesaplayici** ve **IYetki** isimlerinde iki tane Interface oluşturunuz.
- Bu arayüzlere aşağıdaki gövdesiz metotları ekleyiniz:
  - **IZamHesaplayici** : void zamHesapla()
  - **IYetki** : void personelEkle(Personel p)
- **Stajyer** ve **Isci** sınıflarına **IZamHesaplayici** arayüzünü implements anahtar kelimesi ile entegre ediniz.
- **Muhendis** sınıfında **zamHesapla** metodu çağırıldığında maaşa %20 zam yapılır, **Isci** sınıfında %15 ve **Stajyer** sınıfında hiç zam yapılmaz.
- **Muhendis** sınıfına **IZamHesaplayici** ve **IYetki** arayüzlerini implements anahtar kelimesi ile entegre ediniz.
- **Muhendis** sınıfında **personelEkle** metodu çağırıldığında parametre olarak gelen personel mühendis değil ise ve ekipListesinde boş yer varsa bu personel **ekipListesine** eklenir.
- Yazılan tüm metotların test edildiği bir Test sınıfı yazınız.

# Soru 2 - Devam

- **Fabrika** sınıfı oluşturunuz.
- **Fabrika** sınıfı id, ad, personelListesi( Personel sınıfı tipindeki nesneleri kabul eden object array, en fazla 10 adet personel Kabul eder)
- Fabrika sınıfı **personelEkle** metoduna sahiptir. Bu metot parametre olarak Personel sınıfından türeyen bir nesne alır ve dizide boş yer varsa personelListesine ekler.
- **Fabrika** sınıfı **yillikZamYap** metoduna sahiptir. Bu metot çağrıldığında personelListesindeki tüm personellerin **zamYap** metodu çağrılır.
- **Fabrika** sınıfı **bilgileriYazdir** metoduna sahiptir. Bu metot çağrıldığında personelListesindeki tüm personellerin **bilgileriYazdir** metodu çağrılır.
- Fabrika sınıfı **personelSay** metoduna sahiptir. Bu metot çağrıldığında personelListesinde kaç adet Mühendis, İşçi ve Stajyer olduğu bir arrayListe yazılıp geriye döndürülür (instanceof kullanınız).



# Soru 2 - Devam

- **BoyutTasmaHatasi** isminde **Exception** sınıfından kalıtım alan bir sınıf oluşturunuz.
- Diziye eleman ekleme işlemleri sırasında boyut taşma işlemi gerçekleşirse **BoyutTasmaHatasi** sınıfında yazılan hatayı fırlatınız (throws anahtar kelimesi ile) ve hatayı yakaladığınız yerde dizinin boyutunu iki katına çıkarıp, dizinin mevcut elemanlarını yeni diziye taşıyıp, orijinal diziyi yeni oluşturulan ile değiştiriniz.
- **NullElemanKontrol** isminde **Exception** sınıfından kalıtım alan bir sınıf oluşturunuz.
- Fabrika sınıfı **bilgileriYazdir** metodu çağırıldığında personelListesinde null eleman varsa **NullElemanKontrol** sınıfında yazılan hatayı fırlatınız ve dizinin bir sonraki elemanına geçiniz.
- **SinirKontrol** isminde **Exception** sınıfından kalıtım alan bir sınıf oluşturunuz.
- **Personel** sınıfında **zamYap** metodu çağırıldığında **personelListesindeki** personellerden herhangi birinin maaşı 20 bin TL'yi geçerse **SinirKontrol** sınıfını fırlatınız ve bu kişinin maaşını 20 bin TL'ye eşitleyiniz.

# Soru 3

1921221013  
2021221505

NESLİHAN YILDIRIM  
ABDULMELİK

BEKMEZ

- Kullanıcı isminde abstract sınıfı oluşturunuz.
- Kullanıcı sınıfı adSoyad, yaş, tcNo, bakiye özelliklerine sahiptir.
- Kullanıcı sınıfı **bilgileriYazdir()** isminde gövdesiz, abstract bir metoda da sahiptir.
- Kullanıcı sınıfından kalıtım alan **Ogrenci** ve **Emekli** sınıflarını oluşturunuz.
- Kullanıcı sınıfı seansListesi (**Seans** sınıfı tipindeki nesneleri kabul eden object array) özelliğine sahiptir.
- Ogrenci ve Emekli sınıflarında **bilgileriYazdir()** metodu çalıştırıldığında kullanıcıya ait tüm bilgiler ekrana yazdırılmalıdır. Emekli sınıfında ek olarak, eklenen seansların kaçının kredi kartı ile kaçının nakit bir şekilde ödendiği bilgisi de ekrana yazdırılmalıdır.
- **IDemelslemleri** ve **IBiletIslemleri** isimlerinde iki tane Interface oluşturunuz.
- Bu arayüzlere aşağıdaki gövdesiz metotları ekleyiniz:
  - **IDemelslemleri** : void kartOdeme(Seans s),  
void nakitOdeme(Seans s)
  - **IBiletIslemleri** : void satinAl(Seans p)

# Soru 3 - Devam

- **Ogrenci** ve **Emekli** sınıflarına **IBiletIslemleri** ve **IOdemelslemleri** arayüzleri implements anahtar kelimesi ile entegre ediniz.
- **Ogrenci** ve **Emekli** sınıfında **satinAl** metodu override edilir. Her iki sınıfta da Seans sınıfı tipinde bir nesne alır ve onu seansListesine ekler (Kullanıcı yeterli bakiyeye sahipse). **Oğrenci** için bilet fiyatına %20 indirim uygulanır, bu fiyattan bakiyeden düşülür ve "X filmi eklendi" şeklinde bir mesaj yazdırılır. **Emekli** sınıfında ise %15 indirim uygulanır, bu fiyattan bakiyeden düşülür ve "X filmi eklendi" şeklinde bir mesaj yazdırılır.
- **Ogrenci** sınıfında **kartOdeme** metodu çağrıldığında öğrencinin yaşı kontrol edilir. Eğer öğrencinin yaşı 18'den küçük ise ödeme işlemine izin verilmez. Aksi durumda ödeme işlemi gerçekleştirilir, kullanıcının **satinAl** metodu çalıştırılır (Kullanıcının yeterli bakiyesi varsa). **nakitOdeme** metodu çağrıldığında herhangi bir kontrol gerçekleştirilmeden seans, kullanıcının **satinAl** metodu çalıştırılır (Bakiye kontrolü gerçekleştirilir).
- Emekli sınıfında **kartOdeme** ve **nakitOdeme** metotları çağırıldığında bakiye kontrolü dışında bir kontrol gerçekleştirilmeden kullanıcının **satinAl** metodu çalıştırılır. **nakitOdeme** metodu çalıştırılırsa kullanıcının bakiyesine film ücretinin %1'i kadar bakiye eklenir.

# Soru 3 - Devam

- **Film** sınıfı oluşturunuz.
- Film sınıfı id, ad, tur, ucret özelliklerine sahiptir.
- **Seans** sınıfı oluşturunuz.
- **Seans** sınıfı id, film (**Film** sınıfı tipinde nesne), seans\_saati (integer olabilir) özelliklerine sahiptir.
- **Sinema** sınıfı oluşturunuz.
- Bu sınıf ad ve seansListesi (**Seans** sınıfı tipindeki nesneleri kabul eden object array)
- **Sinema** sınıfı **seansEkle** metoduna sahiptir. Bu metot parametre olarak **Seans** sınıfı tipinde bir nesne alır ve onu seansListesine ekler.
- **Sinema** sınıfı **seansSil** metoduna sahiptir. Bu metot parametre olarak **Seans** sınıfı tipinde bir nesne alır ve onu seansListesinden siler. İlgili indeksi null yapar.
- Sinema sınıfı **biletSat** metoduna sahiptir. Parametre olarak **Kullanici** ve **Seans** olmak üzere iki parametre alır. Bu metot çağırıldığında parametre olarak verilen kullanıcının **kartOdeme** veya **nakitOdeme** metotlarından herhangi biri rastgele bir şekilde çalıştırılır. Rastgele sayı üretiniz; 1 gelirse kartOdeme, 0 gelirse nakitOdeme metodu çağırılsın.
- Test sınıfı oluşturunuz. Yazılan her bir metodu test ediniz.

# Soru 3 - Devam

- **BoyutTasmaHatasi** isminde **Exception** sınıfından kalıtım alan bir sınıf oluşturunuz.
- Diziye eleman ekleme işlemleri sırasında boyut taşma işlemi gerçekleşirse **BoyutTasmaHatasi** sınıfında yazılan hatayı fırlatınız (throws anahtar kelimesi ile) ve hatayı yakaladığınız yerde dizinin boyutunu iki katına çıkarıp, dizinin mevcut elemanlarını yeni diziye taşıyıp, orijinal diziyi yeni oluşturulan ile değiştiriniz.
- **SaatKontrol** isminde **Exception** sınıfından kalıtım alan bir sınıf oluşturunuz.
- Sinema sınıfında **biletSat** metodu çağırıldığında seans saati bilgisi 10'dan küçük 23'den büyük ise **SaatKontrol** sınıfında yazılan hatayı fırlatınız ve bu seansı **Sinema** sınıfındaki **seansListesinden** siliniz.
- **NesneKontrol** isminde **Exception** sınıfından kalıtım alan bir sınıf oluşturunuz.
- **Sinema** sınıfında **seansSil** metodu çağırıldığında parametre olarak gönderilen seans, sinema sınıfının **seansListesinde** yoksa **NesneKontrol** sınıfında yazılan hatayı fırlatınız ve bu seansı seansListesine ekleyiniz.

# Soru 4

1921221021  
2021221056

ASİYE İLAYDA  
YUNUS EMRE

SAĞKOL  
ERDEM

- Kullanıcı isminde abstract sınıf oluşturunuz.
- Kullanıcı sınıfı adSoyad, kullanıcı\_adi, gazete (**Gazete** sınıfı tipinde nesne) özelliklerine sahiptir.
- Kullanıcı sınıfı **bilgileriYazdir()** isminde gövdesiz, abstract bir metoda sahiptir.
- **Kullanıcı** sınıfından kalıtım alan **Admin** ve **Yazar** sınıflarını oluşturunuz.
- **Admin** ve **Yazar** sınıflarında **bilgileriYazdir()** metodu çağırıldığında kullanıcıya ait bilgiler ekrana yazdırılmalıdır. Ekstra olarak **Admin** sınıfında kaç haberin silindiği ve güncellediği bilgileri de ekrana yazdırılmalıdır.
- **IYetkilslemleri** ve **I HaberIslemleri** isminde iki Interface oluşturunuz.
- Bu arayüzlere aşağıdaki gövdesiz metotları ekleyiniz:
  - **IYetkilslemleri** : void haberSil(Haber h),  
void haberGuncelle(Haber h, String icerik)
  - **I HaberIslemleri** : void haberEkle(Haber h)

# Soru 4 - Devam

- haberNo, baslik, aciklama ve Yazar (**Yazar** sınıfı tipinde nesne) özelliklerine sahip **Haber** sınıfı oluşturunuz.
- **Admin** ve **Yazar** sınıflarına **IHaberIslemleri** arayüzünü implements anahtar kelimesi ile entegre ediniz.
- **Admin** ve **Yazar** sınıflarında **haberEkle** metodunun gövdesi şu şekilde doldurulur:
  - **Admin:** Parametre olarak gelen haber nesnesini haberin yazarı varsa **Gazete** sınıfındaki **haberListesine** ekler.
  - **Yazar:** Parametre olarak gelen haber nesnesinin yazar değişkenini (**Yazar** sınıfı tipinde nesne) bu yazar olarak set eder.
- Admin sınıfına ekstra olarak **IYetkilslemleri** arayüzünü de implements anahtar kelimesi ile entegre ediniz.
- Admin sınıfında **haberSil** metodu çağırıldığında parametre olarak gönderilen **Haber** gazetenin **haberListesinden** silinir.
- Admin sınıfında **haberGuncelle** metodu çağırıldığında parametre olarak gönderilen **Haberin açıklama** bilgisi parametre olarak gönderilen değer ile değiştirilir.

# Soru 4 - Devam

- ad, kurulus\_yil, kullanıcıListesi (**Kullanıcı** sınıfı tipindeki nesneleri kabul eden object array), haberListesi (**Haber** sınıfı tipindeki nesneleri kabul eden object array) özelliklerine sahip **Gazete** adında bir sınıf oluşturunuz.
- **Gazete** sınıfı **yazarEkle** metoduna sahiptir. Bu metot parametre olarak gelen kullanıcı nesnesini kullanıcıListesine ekler.
- **Gazete** sınıfı **yazarListele** metoduna sahiptir. Bu metot **kullanıcıListesindeki** kullanıcıların tüm bilgilerini listeler.
- **Gazete** sınıfı **haberListele** metoduna sahiptir. Bu metot **haberListesindeki** haberlerin tüm bilgilerini listeler.
- Test sınıfı yazınız ve tüm metotları test ediniz (Admin ve Yazar sınıfındaki tüm metotlar dahil test edilmeli).



# Soru 4 - Devam

- **BoyutTasmaHatasi** isminde **Exception** sınıfından kalıtım alan bir sınıf oluşturunuz.
- Diziye eleman ekleme işlemleri sırasında boyut taşma işlemi gerçekleşirse **BoyutTasmaHatasi** sınıfında yazılan hatayı fırlatınız (throws anahtar kelimesi ile) ve hatayı yakaladığınız yerde dizinin boyutunu iki katına çıkarıp, dizinin mevcut elemanlarını yeni diziye taşıyıp, orijinal diziyi yeni oluşturulan ile değiştiriniz.
- **NullKontrol** isminde **Exception** sınıfından kalıtım alan bir sınıf oluşturunuz.
- Gazete sınıfında **yazarListele** metodu çağırıldığında kullanıcıListesinde **gazete** özelliği null eleman varsa **NullKontrol** sınıfında yazılan hatayı fırlatınız ve bu kullanıcının gazete özelliğini metodu çağıran Gazete sınıfı nesnesi ile değiştiriniz.
- **NesneKontrol** isminde **Exception** sınıfından kalıtım alan bir sınıf oluşturunuz.
- **Admin** sınıfında **haberGuncelle** metodu çağırıldığında parametre olarak gönderilen haber, Gazete sınıfının **haberListesinde** yoksa **NesneKontrol** sınıfında yazılan hatayı fırlatınız ve bu haberi haberListesine ekleyiniz.

# Soru 5

1921221023  
2021221057

MUSTAFA DEVECİ  
ERDİNÇ KUVVET

- Kullanıcı isminde sınıf oluşturunuz.
- Kullanıcı sınıfı adSoyad, aracListesi (**Otomobil** sınıfı tipindeki nesneleri kabul eden object array) özelliklerine sahiptir.
- Otomobil isminde abstract sınıf oluşturunuz.
- Otomobil sınıfı marka, model, beygirGucu, satisFiyati ve kullanici (Kullanıcı sınıfı tipinde nesne) özelliklerine sahiptir.
- Otomobil sınıfı **bilgileriYazdir()** isminde isminde gövdesiz, abstract bir metoda sahiptir.
- Otomobil sınıfından kalıtım alan **Suv** ve **Sedan** sınıflarını oluşturunuz.
- **Sedan** sınıfı, **Otomobil** ile aynı özelliklere sahipken **Sedan** sınıfı ekstra olarak **ozellikListesi** isminde String nesneleri kabul eden bir object arraye de sahiptir.
- **Suv** ve **Sedan** sınıfında **bilgileriYazdir()** metodu çağırıldığında otomobile ait bilgiler ekrana yazdırılır. **Sedan** sınıfında bu bilgilere ek olarak **ozellikListesindeki** özellikler de ekrana bastırılır.
- **ISatisIslemleri** ve **IModifiyeIslemleri** isminde iki Interface oluşturunuz.
- Bu arayüzlere aşağıdaki gövdesiz metotları ekleyiniz:
  - **ISatisIslemleri** : void satisFiyatiHesapla()
  - **IModifiyeIslemleri** : void navigasyonEkle()  
void parkSensoruEkle()

# Soru 5 - Devam

- **Sedan** ve **Suv** sınıflarına **ISatisIslemleri** arayüzünü implements anahtar kelimesi ile entegre ediniz.
- **Suv** ve **Sedan** sınıflarında bulunan **satisFiyatiHesapla** metodlarını  $\text{fiyat} * \text{vergi\_bedeli} + (4 * \text{beygirGucu})$  hesaplamasını yapıp ilgili sınıfın **satisFiyatını** güncelleyecek şekilde kodlayınız. (Suv sınıfı için  $\text{vergi\_bedeli} = 0.20$ , Sedan sınıfı için  $\text{vergi\_bedeli} = 0.15$ )
- **Sedan** sınıfına **IModifiyeIslemleri** arayüzünü implements anahtar kelimesi ile entegre ediniz.
- **Sedan** sınıfında **navigasyonEkle** metodu çağırıldığında arabanın özellikListesine “Navigasyon” Stringi eklenir ve arabanın fiyatı 2500 TL arttırılır.
- **Sedan** sınıfında **parkSensoruEkle** metodu çağırıldığında arabanın özellikListesine “Park Sensörü” Stringi eklenir ve arabanın fiyatı 1200 TL arttırılır.
- id, kurulusYili, otoListesi (**Otomobil** sınıfı tipindeki nesneleri kabul eden object list) özelliklerine sahip **Fabrika** sınıfını oluşturunuz.

# Soru 5 - Devam

- **Fabrika** sınıfında **otoUret ()** metodu vardır ve bu metot çağırıldığında yeni bir **Otomobil** nesnesi (marka ve model bilgilerini parametre olarak alan bir yapılandırıcıya sahip olacak) oluşturulup **Fabrika** sınıfındaki **otoListesi** isimli listeye eklenir (Üretilcek otomobil Suv veya Sedan olabilir). Yeni bir otomobil nesnesi üretilirken Otomobil sınıfının yapılandırıcısında **satisFiyatıHesapla()** metodu çalıştırılmalıdır. Bu sayede arabanın fabrika çıkış fiyatı belirlenir.
- **Fabrika** sınıfında **otoSat(Kullanıcı, Otomobil)** metodu vardır ve bu metot çağırıldığında **otoListesi'**nden, parametre olarak verilen **Otomobil** silinir ve parametre olarak gönderilen Kullanıcının **aracListesine** bu otomobil eklenir.
- **Fabrika** sınıfında **modifiyeEt(Otomobil)** metodu vardır ve gönderilen araba tipi **Sedan** ise bu aracın **navigasyonEkle** veya **parkSensoruEkle** metodu çalıştırılmalıdır. **modifiyeEt** metodu içerisinde rastgele sayı üretiniz. Üretilen sayı 0 ise **navigasyonEkle**, 1 ise **parkSensoruEkle**, 3 ise her iki metod da çalıştırılmalıdır.
- Test sınıfı yazınız ve tüm metotları test ediniz.

# Soru 5 - Devam

- **BoyutTasmaHatasi** isminde **Exception** sınıfından kalıtım alan bir sınıf oluşturunuz.
- Diziye eleman ekleme işlemleri sırasında boyut taşma işlemi gerçekleşirse **BoyutTasmaHatasi** sınıfında yazılan hatayı fırlatınız (throws anahtar kelimesi ile) ve hatayı yakaladığınız yerde dizinin boyutunu iki katına çıkarıp, dizinin mevcut elemanlarını yeni diziye taşıyıp, orijinal diziyi yeni oluşturulan ile değiştiriniz.
- **TekrarKontrol** isminde **Exception** sınıfından kalıtım alan bir sınıf oluşturunuz.
- Fabrika sınıfında **otoSat** metodu çağırıldığında parametre olarak gönderilen Kullanıcı'nın **aracListesinde** parametre olarak gönderilen Otomobil varsa **TekrarKontrol** sınıfında yazılan hatayı fırlatınız.
- **FiyatKontrol** isminde **Exception** sınıfından kalıtım alan bir sınıf oluşturunuz.
- **Otomobil** sınıfında **satisFiyatıHesapla** metodu çağırıldığında bu sınıfın beygirGucu özelliği 0 ise **FiyatKontrol** sınıfında yazılan hatayı fırlatınız ve bu otomobilin beygirGucunu 1500 olarak güncelleyiniz.

# Soru 6

2021221375

HAYA

CHALYON

- Kullanıcı isminde abstract sınıf oluşturunuz.
- Kullanıcı sınıfı `adSoyad`, `urunListesi` (**Urun** sınıfı tipindeki nesneleri kabul eden object list) özelliklerine sahiptir.
- Kullanıcı sınıfı **`urunListele()`** metoduna sahiptir. Bu metot `urunListesindeki` ürünlere ait özellikleri ekrana yazdırır.
- Kullanıcı sınıfından kalıtım alan **PremiumKullanici** ve **StandartKullanici** sınıflarını oluşturunuz. **PremiumKullanici** sınıfı ekstra **`puanDegeri`** isminde bir değişkene daha sahiptir.
- Sepet sınıfı oluşturunuz.
- Sepet sınıfı `sepetSahibi` (**Kullanici** sınıfı tipinde nesne), `urunListesi` (**Urun** sınıfı tipindeki nesneleri kabul eden object list) özelliklerine sahiptir.
- *Sepet* sınıfı **`urunEkle()`** metoduna sahiptir. Bu metot çağırıldığında **Sepet** sınıfının **`urunListesine`** parametre olarak gönderilen **Urun** eklenir.
- *Sepet* sınıfı **`satinAl()`** metoduna sahiptir. Bu metot çağırıldığında **Sepet** sınıfının **`urunListesindeki`** ürünler, **`sepetSahibinin urunListesine`** eklenir ve Sepet sınıfındaki `urunListesi` boşaltılır. Sepetin sahibi **PremiumKullanici** ise işlem sırasında **`primeKullan(urunListesi.size())`** metodu da çalıştırılmalıdır.

# Soru 6 - Devam

- ad, kategori ve fiyat özelliklerine sahip **Urun** sınıfı oluşturunuz.
- **Urun** sınıfından kalıtım alan **Telefon** ve **Bilgisayar** sınıflarını oluşturunuz.
- **IndirimIslemleri** ve **IUyeFirsatIslemleri** isminde iki Interface oluşturunuz.
- Bu arayüzlere aşağıdaki gövdesiz metotları ekleyiniz:
  - **IndirimIslemleri** : void indirimUygula()
  - **IUyeFirsatIslemleri** : void primeKullan(int urunSayisi)
- **Telefon** ve **Bilgisayar** sınıflarına **IndirimIslemleri** arayüzünü implements anahtar kelimesi ile entegre ediniz.
- **Bilgisayar** ve **Telefon** sınıflarında **indirimUygula()** metodunun gövdesi şu şekilde doldurulur:
  - **Bilgisayar**: Ürünün fiyatına %20 indirim uygular ve urun fiyatını günceller.
  - **Telefon**: Ürünün fiyatına %10 indirim uygular. Telefonun hat sayısı 1'den fazla ise ekstra %5 indirim daha uygular ve urun fiyatı günceller.

# Soru 6 - Devam

- **PremiumKullanici** sınıfına **IUyeFirsatIslemleri** arayüzünü implements anahtar kelimesi ile entegre ediniz.
- **PremiumKullanici** sınıfında **primeKullan** metodu çağırıldığında parametre olarak gönderilen **urunSayisi** değerinin 5 katını **puan** olarak kullanıcının **puanDegeri** değişkenine eklemelidir.
- Test sınıfı yazınız ve tüm metotları test ediniz.



# Soru 6 - Devam

- **BoyutTasmaHatasi** isminde **Exception** sınıfından kalıtım alan bir sınıf oluşturunuz.
- Diziye eleman ekleme işlemleri sırasında boyut taşma işlemi gerçekleşirse **BoyutTasmaHatasi** sınıfında yazılan hatayı fırlatınız (throws anahtar kelimesi ile) ve hatayı yakaladığınız yerde dizinin boyutunu iki katına çıkarıp, dizinin mevcut elemanlarını yeni diziye taşıyıp, orijinal diziyi yeni oluşturulan ile değiştiriniz.
- **FiyatKontrol** isminde **Exception** sınıfından kalıtım alan bir sınıf oluşturunuz.
- **Bilgisayar** veya **Telefon** sınıfında **indirimUygula** metodu çağırıldığında bir bilgisayarın fiyatı 3500 TL'nin altına, bir telefonun fiyatı 2000 TL'nin altına düşmemelidir. Fiyat bu değerlerin altına düşerse **IndirimKontrol** sınıfında yazılan hatayı fırlatınız ve ürünün fiyatını taban değere eşitleyiniz.
- **UrunKontrol** isminde **Exception** sınıfından kalıtım alan bir sınıf oluşturunuz.
- Bir kullanıcı, aynı üründen iki adet alamaz. **Sepet** sınıfında **satınAl** metodu çalıştırıldığında bu şartı kontrol ediniz. Eğer şarta uymayan bir durum olursa **UrunKontrol** sınıfında yazılan hatayı fırlatınız ve aynı olan ürünlerden sadece birini **urunListesine** ekleyiniz.

Sorular?