

# BLM19103

## Bilgisayar Programlama 1

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

# Soru 2

- id, ad, ulke özelliklerine sahip, **Uretici** isminde sınıf oluşturunuz.
- id, model, uretici (**Uretici** sınıfı tipinde değişken), uretim\_tarihi, ay, gun, yıl özelliklerine sahip, **Araba** isminde sınıf oluşturunuz.
- id, isim, arabaListesi[] (**Araba** sınıfı tipinde nesneleri kabul eder ve eleman sayısı **Fabrika** sınıfının yapıcı metodu ile belirlensin) özelliklerine sahip, **Fabrika** isminde sınıf oluşturunuz.
- **Uretici**, **Araba** ve **Fabrika** sınıflarının tüm özellikler sınıf dışı erişime kapalı olmalıdır, (hiçbir özelliğe sınıf dışı erişim doğrudan gerçekleşmemelidir, sadece set-get metodları ile erişilebilir)
- Bu fabrikada arabaların üretim tarihi **gün/ay/yıl** veya **gün-ay-yıl** şeklinde **»uretim\_tarihi«** ismindeki değişkende saklanmaktadır.
- Yeni alınan kararlar ile arabaların üretildiği gün, ay ve yıl değerleri ayrı ayrı değişkenlerde saklanmak isteniyor.
- Bunu gerçekleştirecek metotları yazmak istiyoruz.

# Soru 2 - Devam

- Araba sınıfı metot:
  - **public void bilgiVer():** Arabaya ait id, model, üretici adı, gun, ay ve yıl bilgilerini ekrana yazdırmalıdır.
- Fabrika sınıfı metotları:
  - **public void arabaEkle(Araba urun):** Araba sınıfından türetilen nesneyi arabaListesindeki boş alana ekler.
  - **public int[] split(String str):** Verilen stringi '/' karakterine göre parçalar ve geriye 3 elemanlı int dizi döndürür.
    - Örnek: `split(21/12/2020)` -> `[21, 12, 2020]`
  - **public int[] split(String str, char chr):** Verilen stringi gönderilen karaktere göre parçalar ve geriye 3 elemanlı int dizi döndürür.
    - Örnek: `split(21-12-2020, '-')` -> `[21, 12, 2020]`

# Soru 2 – Örnek Senaryo

```
Fabrika fbrk = new Fabrika(12345, "A Fabrika"); // id: 12345, isim: A Fabrika
Uretici u1 = new Uretici(321, "Tofaş", "Türkiye"); // id: 321, ad: Tofaş, ulke: Türkiye
Araba a1 = new Araba(123, "Şahin", "10/06/1990"); // id: 123, model: Şahin, uretim_tarihi: 10/06/1990
a1.setUretici(u1);
fbrk.arabaEkle(a1);
// Düzenleme Metotlarımızı Çağırıyoruz.
for(int i = 0; i < fbrk.getArabaListesi().length(); i++){
    Araba araba = fbrk.getArabaListesi()[i]; // Araba listesindeki i. araba objesini al
    int[] gun_ay_yil = fbrk.split(araba.getUretimTarihi()); // [10, 06, 1990]
    araba.setGun(gun_ay_yil[0]);
    araba.setAy(gun_ay_yil[1]);
    araba.setYil(gun_ay_yil[2]);
    araba.bilgiVer() // Çıktı: "Id: 123, model: Şahin, üretici adı: Tofaş, gün: 10, ay: 06, yıl: 1990"
}
```

# Soru 3

1921221009

FEYZA KURUÇAY

1921221048

MUHAMMED ENSAR

KANYILMAZ

- id, isim, adres özelliklerine sahip, **Firma** isminde sınıf oluşturunuz.
- id, ad, açıklama, kategori ve firma (Firma sınıfı tipinde değişken) özelliklerine sahip, **Urun** isminde sınıf oluşturunuz.
- Ad, urunListesi[] (Urun sınıfı tipinde nesneleri kabul eder ve eleman sayısı Magaza sınıfının yapıcı metodu ile belirlensin) özelliklerine sahip, **Magaza** isminde sınıf oluşturunuz.
- **Firma**, **Urun** ve **Magaza** sınıflarının tüm özellikler sınıf dışı erişime kapalı olmalıdır, (hiçbir özelliğe sınıf dışı erişim doğrudan gerçekleşmemelidir, sadece set-get metodları ile erişilebilir)
- Bu mağazanın kategori ve id isimlendirmelerinde sorun olduğunu varsayalım. Her bir ürünün id ve kategori değerleri ayrı değişkenlerde tutulmak yerine birden fazla stringin birleştirilmesi şeklinde tutulmuştur.
- Örneğin: **ad** değişkeninde ürünün sadece adı tutulması gerekirken hem **adı** hem de **id** değeri tutulmuştur (Ürün Adı: telefon\_12345).
- Aynı şekilde **açıklama** değişkeninde sadece ürüne ait açıklama tutulması gerekirken hem **açıklama** hem **kategori** değeri tutulmuştur (Ürün açıklama: a markasına ait akıllı telefon\_elektronik)

# Soru 3 - Devam

- Yazacağımız substring metodları ile bu hataları düzeltmeye çalışacağız.
- Urun sınıfı metot:
  - **public void bilgiVer():** Urune ait id, ad, açıklama ve kategori bilgilerini ekrana yazdırmalıdır.
- Mağaza sınıfı metotları:
  - **public void urunEkle(Urun urun):** Urun sınıfından türetilen nesneyi urunListesindeki boş alana ekler.
  - **public int indeksBul(String str, char chr):** str içerisinde, gönderilen karakterin kaçınca indekste olduğunu bulup geriye bu indeksi döndürür.
  - **public String substring(String str, int indeks):** str üzerinde verilen indeksten strnin sonuna kadar olan elemanları geriye döndürür.
    - Örnek: `substring(blm_19103, 3) -> 19103`
  - **Public String substring(String str, int indeks, int adet):** str üzerinde verilen indeksten itibaren “adet” kadar elemanı alır ve geriye döner
    - Örnek: `substring(blm_19103, 3, 4) -> 1910`

# Soru 3 – Örnek Senaryo

```
Magaza mgz = new Magaza("A Mağazası");
Urun u1 = new Urun("MyPhone_15735", "Akıllı Telefon_Elektronik");
System.out.println("Ürün 1 Ad: " + u1.getAd()) // Çıktı: MyPhone_15735
System.out.println("Ürün 1 Id: " + u1.getID()) // Çıktı: 0
System.out.println("Ürün 1 Açıklama: " + u1.getAciklama()) // Çıktı: Akıllı Telefon_Elektronik
System.out.println("Ürün 1 Kategori: " + u1.getKategori()) // Çıktı: null
Mgz.urunEkle(u1);
// Düzenleme Metotlarımızı Çağırıyoruz.
for(int i = 0; i < mgz.getUrunListesi().length(); i++){
    Urun urn = mgz.getUrunListesi()[i]; // Urun listesindeki i. ürünü objesini al
    int indeks_1 = mgz.indeksBul(urn.getAd(), '_'); // 7
    int indeks_2 = mgz.indeksBul(urn.getAciklama(), '_'); // 15
    String yeni_ad = mgz.substring(urn.getAd(), -1, indeks_1 - 1); // MyPhone
    String yeni_id = mgz.substring(urn.getAd(), indeks_1); // 15735
    String yeni_aciklama = mgz.substring(urn.getAciklama(), -1, indeks_2 - 1); // Akıllı Telefon
    String yeni_kategori = mgz.substring(urn.getAciklama(), indeks_2); // Elektronik
    // Set metotları ile "i. indeksteki" ürünün ad, id, açıklama ve kategori bilgileri yenileri ile değiştirilir.
    urn.bilgiVer() // Çıktı: "Ad: MyPhone, Id: 15375, Açıklama: Akıllı Telefon, Kategori: Elektronik"
}
```

# Soru 4

1921221010

YUNUS SÜMER

1921221049

İRFAN VURAL

- id, icerik, kullanıcı (**Kullanıcı** sınıfı tipinde değişken) özelliklerine sahip, **Yorum** isminde sınıf oluşturunuz.
- Id, kullanıcı\_adi, yorum\_listesi (**Yorum** sınıfı tipinde object array), özelliklerine sahip, **Kullanıcı** isminde sınıf oluşturunuz.
- url, isim, kullanıcıListesi[] (**Kullanıcı** sınıfı tipinde nesneleri kabul eder ve eleman sayısı **WebSite** sınıfının yapıcı metodu ile belirlensin) özelliklerine sahip, **WebSite** isminde sınıf oluşturunuz.
- **Kullanıcı**, **Yorum** ve **WebSite** sınıflarının tüm özellikler sınıf dışı erişime kapalı olmalıdır, (hiçbir özelliğe sınıf dışı erişim doğrudan gerçekleşmemelidir, sadece set-get metodları ile erişilebilir)
- Bu web sitesinde daha önceden, kullanıcılar yorum yaparken herhangi bir denetim gerçekleştirilmemektedir. Biz bu aşamadan sonra kullanıcıların yorumları içerisinde istemediğimiz karakterleri silmek istiyoruz.
- Yorumlar içerisinde istenmeyen karakterleri silip, bu yorumları düzenleyen metotları yazmak istiyoruz.



# Soru 4 - Devam

- Urun sınıfı metot:
  - **public void yorumYap(Yorum yorum):** Yorum sınıfından türetilen nesneyi yorumListesindeki boş alana ekler.
  - **public void yorumListele():** yorumListesindeki tüm yorumların içerik bilgisini ekrana yazdırır.
- WebSite sınıfı metotları:
  - **public void kullanıcıEkle(Kullanici kullanıcı):** Kullanici sınıfından türetilen nesneyi kullanıcıListesindeki boş alana ekler.
  - **public String delete(String str, char yasakKarakter):** str içerisinden gönderilen karakteri siler ve yeni stringi geriye döndürür.
    - Örnek: delete(Bunu\_sevdim , '\_' ) -> Bunusevdim
  - **public String delete(String str, char[] karakterListesi):** str içerisinden gönderilen **karakterleri** siler ve yeni stringi geriye döndürür.
    - Örnek: delete(Bunu ?sevdim\_, ['\_', '?']) -> Bunu sevdim

# Soru 4 – Örnek Senaryo

```
WebSite site = new WebSite("A Sitesi"); // isim: A Sitesi
Kullanici k1 = new Kullanici(12345, "blm_deneme"); // id: 12345, kullanıcı_adi: blm_deneme
Yorum y1 = new Yorum("<Tehlikeli Yorum>");
Yorum y2 = new Yorum("Yasak K4r4kt£r");
k1.yorumYap(y1); // Kullanıcının yorumListesine bu yorum eklendi
k1.yorumYap(y2); // Kullanıcının yorumListesine bu yorum eklendi
site.kullaniciEkle(k1); // Sitenin kullanıcıListesine bu kullanıcı eklendi
// Düzenleme Metotlarımızı Çağırıyoruz.
for(int i = 0; i < site.getKullaniciListesi().length(); i++){
    Kullanici kullanici = site.getKullaniciListesi()[i]; // i. kullanıcıyı getir
    for(int j = 0; j < kullanici.getYorumListesi().length; j++){
        Yorum yorum = kullanici.getYorumListesi()[j]; // Kullanıcının j. yorumunu getir
        String duzenlenmis_icerik = site.delete(yorum.getIcerik(), ['<', '>', '4', '£']); // Tehlikeli Yorum
        // Set metodu ile "j. indeksteki" yorumun icerik bilgisi yenisi ile değiştirilir.
        yorum.setIcerik(duzenlenmis_icerik);
    }
    kullanici.yorumListele() // Tehlikeli Yorum
    // Yasak Krktr
}
```

# Soru 5

1921221011

ŞAMİL BİLAL

ÖZAYDIN

1921221350

MHD EOBAL

AYOUB

- id, baslik, aciklama, kullanıcı (**Kullanıcı** sınıfı tipinde değişken) özelliklerine sahip, **Konu** isminde sınıf oluşturunuz.
- Id, kullanıcı\_adi, konuListesi (**Konu** sınıfı tipinde object array), özelliklerine sahip, **Kullanıcı** isminde sınıf oluşturunuz.
- kullanıcıListesi[] (**Kullanıcı** sınıfı tipinde nesneleri kabul eder ve eleman sayısı **Duzenleyici** sınıfının yapıcı metodu ile belirlensin) özelliklerine sahip, **Duzenleyici** isminde sınıf oluşturunuz.
- **Kullanıcı**, **Konu** ve **Duzenleyici** sınıflarının tüm özellikler sınıf dışı erişime kapalı olmalıdır, (hiçbir özelliğe sınıf dışı erişim doğrudan gerçekleşmemelidir, sadece set-get metodları ile erişilebilir)
- Bu uygulamada daha önceden, kullanıcılar konu açarken herhangi bir denetim gerçekleştirilmemektedir. Biz bu aşamadan sonra kullanıcıların açıklamaları içerisinde istemediğimiz karakterleri istenen karakter ile değiştiriyoruz.
- Konular içerisinde istenmeyen karakterleri bulup, bunları değiştiren metotları yazmak istiyoruz.

# Soru 5 - Devam

- Kullanici sınıfı metot:
  - **public void konuEkle(Konu yorum):** Konu sınıfından türetilen nesneyi konuListesindeki boş alana ekler.
  - **public void konuListele():** konuListesindeki tüm konuların başlık ve açıklama bilgisini ekrana yazdırır.
- Duzenleyici sınıfı metotları:
  - **public void kullaniciEkle(Kullanici kullanici):** Kullanici sınıfından türetilen nesneyi kullaniciListesindeki boş alana ekler.
  - **public String replace(String str, char aranan, char degistir):** str içerisinde aranan karakteri bulur ve yeni karakter ile değiştirip geriye döndürür.
    - Örnek: `replace(Bu_aciklama_bitti, '_', ' ')` -> Bu aciklama bitti
  - **public String replace(String str, char[] aranan, char degistir):** str içerisinde aranan **karakterleri** bulur ve yeni karakter ile değiştirip geriye döndürür.
    - Örnek: `replace(<İstEnmEyEn_k4r4kt£rl£r', ['<', '£', '4'], ")` -> `istnmyn_krktrlr`

# Soru 5 – Örnek Senaryo

```
Duzenleyici dz= new Duzenleyici();
Kullanici k1 = new Kullanici(12345, "blm_deneme"); // id: 12345, kullanıcı_adi: blm_deneme
Konu ko1 = new Konu("Başlık 1", "<Tehlikeli Yorum>"); // baslik: Başlık 1, aciklama: <Tehlikeli Yorum>
Konu ko2 = new Konu(("Başlık 2", "Yasak K4r4ktër"); // baslik: Başlık 2, aciklama: Yasak K4r4ktër
k1.konuEkle(ko1); // Kullanıcının konuListesine bu konu eklendi
k1.konuEkle(ko2); // Kullanıcının konuListesine bu konu eklendi
dz.kullaniciEkle(k1); // Duzenleyicinin kullaniciListesine bu kullanıcı eklendi
// Düzenleme Metotlarımızı Çağırıyoruz.
for(int i = 0; i < dz.getKullaniciListesi().length(); i++){
    Kullanici kullanici = dz.getKullaniciListesi()[i]; // i. kullanıcıyı getir
    for(int j = 0; j < kullanici.getKonuListesi().length; j++){
        Konu ko = kullanici.getKullaniciListesi()[j]; // Kullanıcının j. konusunu getir
        String yeni_aciklama = dz.replace(ko.getAciklama(), ['<', '>', '4', '£'], '*'); // *Tehlikeli Yorum*
        // Set metodu ile "j. indeksteki" yorumun icerik bilgisi yenisi ile değiştirilir.
        ko.setAciklama(duzenlenmis_icerik);
    }
    kullanici.konuListele() // *Tehlikeli Yorum*
                          // Yasak K*r*kt*r
}
```

# Soru 6

1921221012  
1921221355

BEYZANURERDAĞI  
RAMI AYOUB

- id, isim, dosya\_adi, ogrenci (**Ogrenci** sınıfı tipinde değişken) özelliklerine sahip, **Odev** isminde sınıf oluşturunuz.
- id, ad, soyad, mail ve odevListesi[] (**Odev** sınıfı tipinde nesneleri kabul eder ve eleman sayısı **Ogrenci** sınıfının yapıcı metodu ile belirlensin) özelliklerine sahip, **Ogrenci** isminde sınıf oluşturunuz.
- id, ogrenciListesi[] (**Ogrenci** sınıfı tipinde nesneleri kabul eder ve eleman sayısı **Moodle** sınıfının yapıcı metodu ile belirlensin) özelliklerine sahip, **Moodle** isminde sınıf oluşturunuz.
- **Odev**, **Ogrenci** ve **Moodle** sınıflarının tüm özellikler sınıf dışı erişime kapalı olmalıdır, (hiçbir özelliğe sınıf dışı erişim doğrudan gerçekleşmemelidir, sadece set-get metodları ile erişilebilir)
- Moodle sistemi öğrenciler tarafından yüklenen ödevleri sadece adSoyad.pdf şeklinde isimlendirmektedir.
- Yeni yapılan güncelleme ile öğrencilerin ödevlerini **adSoyad\_id\_odevAdi.pdf** veya **id\_odevAdi\_adSoyad.pdf** şeklinde isimlendirmek istiyoruz.
- Bunu gerçekleştirecek metotlar yazılmalı.

# Soru 6 - Devam

- Öğrenci sınıfı metot:
  - **public void odevEkle(Odev yorum):** Odev sınıfından türetilen nesneyi odevListesindeki boş alana ekler.
  - **public void odevListele():** odevListesindeki tüm ödevlerin isim ve dosya adı bilgisini ekrana yazdırır.
- Moodle sınıfı metotları:
  - **public void ogrenciEkle(Ogrenci ogrenci):** Ogrenci sınıfından türetilen nesneyi ogrenciListesindeki boş alana ekler.
  - **public String insert(String str, int offset, String[] eklenecek):** str içerisine belirtilen indisten itibaren gönderilen stringleri ekler.
    - Örnek: insert("AhmetAk.pdf", 0, [123, "Odev1"]) ->123\_Odev1\_AhmetAk.pdf
  - **public String insert(String str, String[] eklenecek):** str içerisine gönderilen stringleri ekler.
    - Örnek: insert("AhmetAk.pdf", [123, "Odev1"]) ->AhmetAk\_ 123\_Odev1.pdf

# Soru 6 – Örnek Senaryo

```
Moodle mddle= new Moodle();
Ogrenci o1 = new Ogrenci(12345, "Ahmet", "Ak"); // id: 12345, ad: Ahmet, soyad: Ak
Odev od1 = new Odev(123, "Odev_1", o1) // id: 123, isim: Odev_1, ogrenci: o1
// Odeve ait öğrenci atanır atanmaz ödevin dosya adı öğrenci nesnesinin ad ve soyad bilgileri kullanılarak
oluşturulmalıdır.
o1.odevEkle(od1); // Öğrencinin odevListesine bu ödev eklendi
mddle.ogrenciEkle(o1); // Moodle'ın ogrenciListesine bu öğrenci eklendi
// Düzenleme Metotlarımızı Çağırıyoruz.
for(int i = 0; i < mddle.getOgrenciListesi().length(); i++){
    Ogrenci ogr = mddle. getOgrenciListesi()[i]; // i. öğrenciyi getir
    for(int j = 0; j < ogr.getOdevListesi().length; j++){
        Odev odev = ogr. getOdevListesi()[j]; // Öğrencinin j. ödevini getir
        String[] eklenecekBilgiler = [ogr.getId(), odev.getIdim()];
        mddle.insert(odev.getDosyaAdi(), eklenecekBilgiler); // AhmetAk_ 123_Odev1.pdf
    }
    ogr.odevListele() // "Ödev Ad: Odev_1, Dosya_Adı: AhmetAk_ 123_Odev1.pdf"
}
```



Sorular?