



T.C. İSTANBUL AREL ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

Akademik Yıl	Sınav			Ders		
	Tür	Veri	Saat	Kod	İsim	Dersi Veren
Güz 2025	Ara Sınav	17.11.2025	14:30-16:00	LBLM203	Nesne Tabanlı Programlama	Tuğberk Kocatekin

BİLGİLER

1. Cep telefonu kullanımı yasaktır.
 2. İlk 15 ve son 5 dakika sınavdan çıkışlamaz.
 3. Sınavda 15 dakikadan geç kalanlar alınmaz.
 4. Listede adı olmayan öğrenciler sınava giremez.
 5. Tükenmez kalem değil, kurşun kalem kullanılmalıdır.
 6. Sınav bittığında sınıfta en az 2 öğrenci olmalıdır.
 7. Hesap makinesi kullanılamaz.

Question:	1	2	3	4	5	6	7	Total
Points:	18	10	12	24	6	10	20	100
Score:								

Ad: _____

ID: _____

İmza: _____

Questions

1. (18 points) (DÖÇ 1-2-3-4) **Aşağıdaki soruları yanıtlayın. Her sorunun tek bir cevabı var. Her soru 2 puan.**

- (a) Aşağıdakilerden hangisi bir sınıfın (class) nesne oluşturulduğunda otomatik olarak çalışan özel metodtur?
- A. finalize()
 - B. build()
 - C. constructor
 - D. creator
- (b) Aşağıdakilerden hangisi encapsulation ile ilgili **doğru** bir ifadedir?
- A. Tüm değişkenlerin public yapılması
 - B. Veriyi gizleyip kontrol için getter/setter kullanılması
 - C. Sınıfların birden fazla interface implement etmesi
 - D. Aynı isimli metodların farklı parametrelerle yazılması
- (c) Aşağıdaki anahtar kelimelerden hangisi kalıtım (inheritance) için kullanılır?
- A. implements
 - B. extends
 - C. inherits
 - D. override
- (d) Bir interface'i implement eden sınıf hakkında aşağıdakilerden hangisi doğrudur?
- A. Interface'in constructor'ını çağırır
 - B. Interface içindeki tüm metodları override etmek zorundadır
 - C. Interface'teki private metodları override edebilir
 - D. Bir sınıf sadece tek bir interface implement edebilir
- (e) Aşağıdaki ilişki türlerinden hangisi composition (bileşim) için doğrudur?
- A. is-a
 - B. has-a
 - C. overrides
 - D. loads-a
- (f) Aşağıdakilerden hangisi run-time polymorphism örneğidir?
- A. Method overloading
 - B. Farklı ve ilişkisiz sınıflarda aynı isimli değişken kullanma
 - C. Subclass'in parent metodunu override etmesi
 - D. Statik metod kullanımı
- (g) Aşağıdakilerden hangisi abstract class ile ilgili **doğru** bir yorumdur?
- (h) Static değişkenler hakkında hangi ifade **doğrudur**?
- A. Her nesne için ayrı kopyası vardır
 - B. Sınıfa aittir, tüm nesneler tarafından paylaşıılır
 - C. Yalnızca interface'lerde bulunur
 - D. Sadece private olabilir
- (i) Aşağıdaki dosya okuma yöntemlerinden hangisi satır satır okuma yapar?
- A. Math.random()
 - B. BufferedReader
 - C. Files.readAllLines
 - D. System.in
2. (10 points) (DÖÇ 1-2-4) **Aşağıdaki sorulara kısa yanıtlar verin. Her boşluk 1 puan değerinde.**
- (a) Interface'eki tüm değişkenler otomatik olarak _____, _____ ve _____ olur.
- (b) Bir sınıf başka bir sınıfı içeriyorsa buna _____ denir.
- (c) Constructor içinde sınıfın kendi değişkenlerine erişirken kullandığı anahtar kelime: _____
- (d) Aynı isimli metodların farklı parametrelerle yazılmasına _____ denir.
- (e) Doğrudan nesnesi oluşturulamayan sınıf: _____
- (f) Bir subclass'ın üst sınıftaki metodу yeniden yazmasına _____ denir.
- (g) Verinin gizlenmesi ve kontrollü erişim sağlanması kavramı: _____
- (h) Java maskotunun adı: _____

3. (12 points) (DÖÇ 1-4) Aşağıda bir **Ogrenci** ve bir de **Main** sınıfı var. Main sınıfı içerisinde **Barney** adında bir öğrenci yaratın. Adı Barney olacak bu öğrencinin obje adı **o1** olsun. Öğrenci sınıfının içinde ise, öğrenci sayısını tutan ve her öğrenci yaratıldığında otomatik olarak artan bir değişken olacak şekilde içerisinde doldurun. Bu değişkene **ogrenciSayisi** ismi verin. Getter ve setter metotlarını yazmayın.

```
class Ogrenci { (5p)

    _____;

    _____;

    public Ogrenci(String name) { (5p)

        _____;

        _____;

    }

    public static void main(String[] args) { (2p)

        _____;

    }
}
```

4. (DÖÇ 1-4) Aşağıda hem bir Kitap hem de Kütüphane sınıfları için kodlar bulunmaktadır. Bu Kitap sınıfında **yazar** (String) ve **yazarTarihi** (String) bulunması gerekiyor. Kütüphane ise, içinde kitapları tutacak.

- (a) (6 points) Kitap sınıfındaki tüm boşlukları doldurun.

```
public class Kitap { (2p)

    _____;

    _____;

    public Kitap(String a, String b) { (2p)

        _____;

        _____;

    }

    public String getYazar() { (2p)

        _____;
}
```

- (b) (18 points) (DÖÇ 1-2) Verilen bilgilere göre, Kütüphane sınıfı içerisinde kitapları tutan, **kitaplar** adında bir Liste olması gerekiyor. Bununla birlikte, Kütüphane sınıfı istediği takdirde kendi içindeki listeye bir kitap ekleyebilir.

```
public class Kutuphane { (5p)  
-----;  
  
public Kutuphane() { (3p)  
  
    this.kitaplar = -----;  
}  
  
public ----- kitapEkle(-----) { (10p)  
  
    kitaplar.add(-----);  
}  
}
```

5. (6 points) Aşağıda Executable isimli bir interface tanımlanmıştır. Bu interface'i implement eden bir **Uygulama** sınıfı yaratın, baslat metodunu override edin. Ekrana AREL yazdırın.

```
public interface Calistirabilir {  
    void baslat();  
}  
  
public class -----{ (3p)  
  
    @Override  
    void baslat() {  
  
        -----; (3p)  
    }  
}
```

6. (10 points) (DÖÇ 1-2-4) Sistemimizde Personel adında bir sınıf mevcuttur. Bu sınıf, *isim* ve *departman* (ikisi de string) bilgilerini tutar. **Ayrıca, bu iki değişkeni de aldığı bir constructor'u vardır.** Bir de Akademisyen sınıfı var. Bu sınıf, Personel sınıfında ne varsa hepsine sahip olacak, üstüne bir de dersSayısı isminde bir değişkene daha sahip olacak.

```
class Akademisyen -----{ (3p)  
  
    int dersSayisi;  
  
    Akademisyen(String isim, String department, int d) {  
  
        -----; (5p)  
        -----; (2p)  
    }  
}
```

7. (20 points) (DÖÇ 1-4) Aşağıda bir Ürün sınıfı ile Main sınıfı verilmiştir. Main sınıfına bakarak, Urun sınıfının içerisindeğini doldurun.

```
public static void main(String[] args){  
    Urun u1 = new Urun();  
    Urun u2 = new Urun("Laptop");  
    Urun u3 = new Urun("Mouse", 250);  
}
```

(a) Ürün sınıfının içinde **isim** ve **fiyat** değişkenlerini de tanımlayın.

```
public class Urun {  
    -----; (1p)  
    -----; (1p)  
    //aşağıdaki boşluğa constructorları ekleyin (12p)
```

```
public ----- getIsim() { (1p)  
    -----; (1p)  
}  
  
public ----- setFiyat(-----) { (3p)  
    -----; (1p)  
}  
}
```