

Öğrenci Numarası:

Ad Soyadı:

Süre 60dk. Başarılar...

**1) (DÇ2) Interface ve absract kavramlarını temel özelliklerini ve kullanım amaçlarını vurgulayarak açıklayınız. (5P)**

Bir Java interface'i, bir veya daha fazla soyut metodu içeren ve bu metodların bir sınıf tarafından uygulanmasını zorunlu kılan bir yapıdır. Java'da bir sınıf sadece bir sınıfı genişletebilir (extends) ancak birden fazla interface'i uygulayabilir (implements). Interface'ler, bir sınıfa belli bir davranış kazandırmak veya bir grup benzer sınıfa ortak bir arayüz sağlamak için kullanılır.

Java'da abstract anahtar kelimesi, soyut sınıfları ve soyut metodları tanımlamak için kullanılır. Soyut sınıflar, kendileri bir nesne oluşturulamayan sınıflardır ve genellikle soyut metodları içerir. Soyut metodlar ise sadece imza (parametre ve dönüş tipi) tanımlar, ancak gerçek implementasyonu alt sınıflara bırakır.

2)

**Kitap (Book) Sınıfı:**

Kitabın adı (title), yazarı (author) ve yayın yılı (year) bilgilerini tutan değişkenlere sahiptir.

**Üye (Member) Sınıfı:**

Kütüphaneye üye olan kişileri temsil eder. Üyenin adı (name), üye numarası (memberID) ve ödünç aldığı kitapları takip etmek için bir liste gibi davranan bir değişken içerir.

**Kütüphane (Library) Sınıfı:**

Kitapları ve üyeleri yöneten ana sınıftır.

Kitapları saklamak için bir liste ve üyeleri saklamak için bir başka liste içerir.

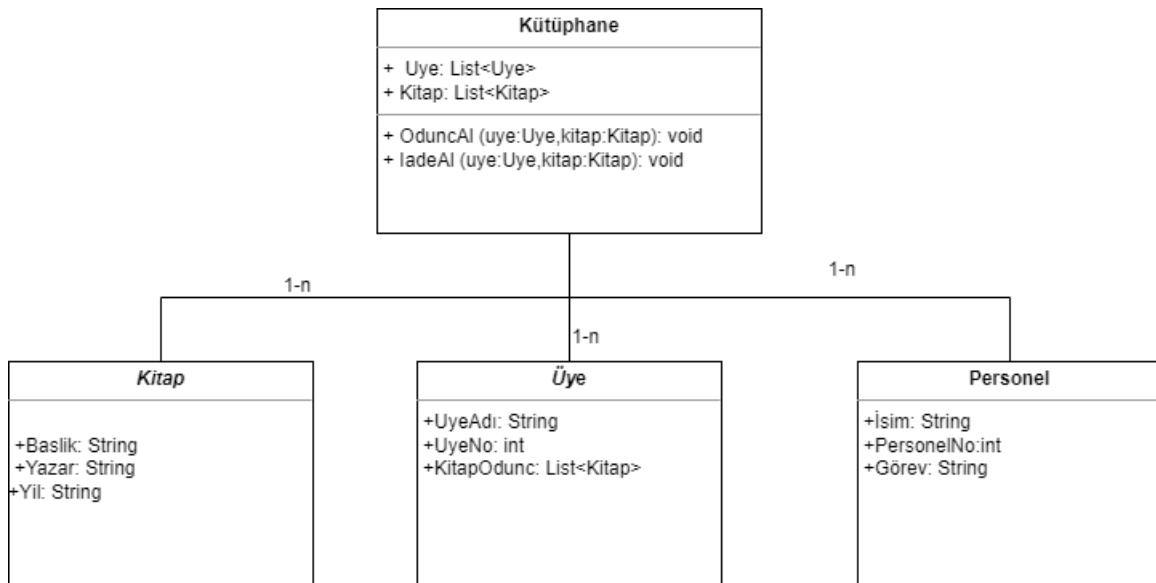
Üyelere kitap ödünç verme ve iade alma gibi işlemleri gerçekleştirebilen metodlara sahiptir.

**Personel (Staff) Sınıfı:**

Kütüphanede çalışan personeli temsil eder. Personelin adı (name), personel numarası (staffID) ve görev bilgilerini içeren değişkenlere sahiptir.

Not: Diyagramınızda sınıflar arasındaki ilişkileri (örneğin, aggregation, association) ve sınıfların içerdikleri değişkenleri ve metodları belirtmeye özen gösterin.

**(DÇ2) Bir kütüphane sistemi tasarlamamız istenmektedir. Bu sistemi temsil eden bir UML diyagramı çiziniz. Yukarıdaki gereksinimlere uygun olarak sınıfları, arayüzleri ve ilişkileri içeren bir diyagram oluşturunuz. (15P)**



Öğrenci Numarası:

Ad Soyadı:

Süre 60dk. Başarılar...

**3) (DÇ4) SOLID prensipleri nelerdir? Açıklamaları ile birlikte yazınız. (5P)**

**S** — Single-responsibility principle: Bir sınıfın sadece bir sorumluluğu olmalıdır. Yani, bir sınıfın değişiklik yapılması gerektiğinde sadece bir nedeni olmalıdır.

**O** — Open-closed principle: Bir yazılım bileşeni (sınıf, modül, fonksiyon, vb.) genişletmeye açık, ancak değişikliklere kapalı olmalıdır. Yani, mevcut kodu değiştirmeden yeni özellikler ekleyebilmelisiniz.

**L** — Liskov substitution principle: Alt sınıflar, üst sınıfların yerine kullanılabilir. Yani, bir üst sınıf nesnesinin, alt sınıfların nesneleriyle değiştirilebilmesi gerekir.

**I** — Interface segregation principle: Bir sınıf, ihtiyaç duymadığı metotlara bağımlı olmamalıdır. Yani, bir sınıf, kullandığı arayüzde sadece ihtiyaç duyduğu metotları içermelidir.

**D** — Dependency Inversion Principle: Yüksek seviyeli modüller, düşük seviyeli modüllere bağlı olmamalıdır. İki seviye, soyutlamalar aracılığıyla birbirine bağlanmalıdır. Yani, bağımlılıklar soyutlamalara (interface ve abstract class gibi) dayanmalıdır.

**4) (DÇ1) Java programlama dilinde, farklı geometrik şekillerin alanlarını hesaplayabilen bir program yazmanız istenmektedir. Aşağıda belirtilen iki geometrik şekil üzerinde alan hesaplaması yapacak bir Java programı oluşturunuz. (25P)**

**Dikdörtgen (Rectangle) Metodu:**

Rectangle metodu, dikdörtgenin uzunluğu ve genişliğini temsil eden iki değişkene sahip olmalıdır.

Dikdörtgenin alanını hesaplayan bir işlem içermelidir.

**Üçgen (Triangle) Metodu:**

Triangle metodu, üçgenin taban uzunluğu ve yüksekliğini temsil eden iki değişkene sahip olmalıdır.

Üçgenin alanını hesaplayan bir işlem içermelidir.

**Ana Uygulama (main metod):**

Main metodu içinde, kullanıcıdan bir dikdörtgenin uzunluğu ve genişliği ile bir üçgenin taban uzunluğu ve yüksekliğini alın.

Kullanıcının girdiği değerlerle gerekli parametreleri alan bir dikdörtgen ve bir üçgen nesnesi oluşturun.

Oluşturulan nesnelerin alanlarını hesaplayarak ekrana yazdırın.

Öğrenci Numarası:

Ad Soyadı:

Süre 60dk. Başarılar...

```
public class Main {  
    public static void main(String[] args) {  
  
        int rectLength=5;  
        int rectWidth=6;  
        int triBase=7;  
        int triHeight=8;  
  
        /*kullanıcıdan sayı almak için:  
        Scanner scanner = new Scanner(System.in);  
  
        int rectLength=scanner.nextInt();  
        int rectWidth=scanner.nextInt();  
        int triBase=scanner.nextInt();  
        int triHeight=scanner.nextInt();  
  
        */  
  
        // Dikdörtgen ve üçgen nesneleri oluşturuluyor  
        double rectArea = calculateRectangleArea(length: rectLength, width: rectWidth);  
        double triArea = calculateTriangleArea(base: triBase, height: triHeight);  
  
        // Alanlar hesaplanıp ekrana yazdırılıyor  
        System.out.println("Dikdörtgen Alanı: " + rectArea);  
        System.out.println("Üçgen Alanı: " + triArea);  
  
    }  
  
    // Dikdörtgen alanını hesaplayan metod  
    static double calculateRectangleArea(int length, int width) {  
        return length * width;  
    }  
  
    // Üçgen alanını hesaplayan metod  
    static double calculateTriangleArea(int base, int height) {  
        return 0.5 * base * height;  
    }  
}
```

5) (DÇ3) Singleton tasarım deseni nedir hangi durumlarda kullanılır? Örnek vererek kısaca açıklayınız. (5P)

Singleton tasarım deseni, bir sınıfın yalnızca bir örneğine sahip olmasını sağlayan bir creational (oluşturucu) tasarım desendir. Sistem bünyesinde bir sınıftan sadece bir nesne oluşturulması bu nesnenin kullanılması gerektiği durumlarda singleton tasarım şablonu kullanılır.

6) (DÇ1) Geleneksel yazılım modelleri ile çevik yazılım modeli arasındaki farkları açıklayınız. (5P)

#### Geleneksel Yöntemler

- Müşteriler ne istediğini iyi bilir.
- Geliştiriciler neyi, ne şekilde üreteceklerini iyi bilir.
- Bu yol boyunca hiçbir şey değişmeyecektir.

#### Çevik Yöntemler

- Müşteriler ne istediğini keşfeder.
- Geliştiriciler neyi nasıl üreteceğini keşfeder.
- Bu yol boyunca birçok değişiklik yapılabilir.

Öğrenci Numarası:

Ad Soyadı:

Süre 60dk. Başarılar...

7)

Ölçüm	Çevik Modelleme	Çağlayan Modeli
Planlama ölçeği	Kısa	Uzun
Müşteri ile geliştirici arasındaki mesafe	Kısa	Uzun
Özelleştirme ve uygulama arasındaki zaman	Kısa	Uzun
Sorunları keşfetmek için zaman	Kısa	Uzun
Proje tamamlanma riski	Düşük	Yüksek
Değişikliklere uyum yeteneği	Yüksek	Düşük

(DÇ1) Yukarıda verilen tabloyu çevik ve geleneksel model yöntemlerini göz önünde bulundurarak (kısa-uzun/ yüksek – düşük anahtar kelimeleri ile) doldurunuz. (5P)

8) (DÇ1) Agile modelinin diğer modellerden farkları nelerdir? Üzerinde durduğu temel noktalar nelerdir? (5P)

- **Esnek ve İteratif Gelişim:** Agile, projenin başından sonuna kadar esnek ve iteratif bir gelişim sürecini benimser. Küçük iterasyonlar halinde çalışarak, sürekli değişen gereksinimlere hızlı uyum sağlar.
- **Müşteri Katılımı ve Geri Bildirim:** Müşteri, geliştirme sürecinin her aşamasında aktif olarak dahil edilir. Sürekli geri bildirimle ürünün geliştirilmesine yönlendirilir ve değerli katkılarda bulunabilir.
- **Ekip iş birliği ve İletişim:** Küçük, kendini yöneten ekiplerin işbirliği ve iletişimi vurgulanır. Ekip üyeleri sürekli olarak birbirleriyle etkileşimde bulunur, sorunları birlikte çözer ve birbirlerinden öğrenirler.
- **Değer Odaklı ve Risk Yönetimi:** Agile, en yüksek değere sahip özelliklere öncelik verir ve riskleri erken tanımlamaya odaklanır. Değer üretmeye odaklanarak, sürekli olarak müşteri ihtiyaçlarına yanıt verir ve projedeki riskleri minimize eder.

9) (DÇ1) Scrum haricindeki diğer çevik yazılım geliştirme yöntemlerinden iki tanesini örnek vererek açıklayınız. (5P)

**Kanban:** Kanban, üretim süreçlerinden türetilmiş bir yönetim metodolojisidir. Çevik yazılım geliştirmede, bir tahtaya (kanban tahtası) yapılacak işlerin kartları yerleştirilir ve bu kartlar farklı sütunlarda ilerler (genellikle "Yapılacaklar," "Yapılıyor," "Tamamlandı" gibi). Görsel olarak işin akışını izlemek, ekip üyelerine iş yüklerini dengeleme ve süreçteki engelleri daha hızlı tanıma imkânı sağlar.

**Extreme Programming (XP):** Extreme Programming, yazılım geliştirme süreçlerini iyileştirmek ve yazılım kalitesini artırmak için kullanılan bir çevik metodolojidir. XP, müşteri geribildirim, küçük ve sık iterasyonlar, test odaklı geliştirme, birlikte programlama gibi pratiklere dayanır.

(Crystal Clear, DSDM, FDD, LSD yöntemleri de değerlendirmeye alınacaktır.)

10) (DÇ1) Veritabanı adında bir sınıftan **sadece bir tane nesne** üretilmek istenmektedir. Veritabanı ilk defa oluşturulduğunda “**Veritabanı Oluşturuldu**” çıktısı alınacaktır. Birden fazla nesne üretilmek istendiğinde “**Zaten bir veritabanı oluşturulmuş**” uyarısı vermelidir. Main sınıfından VeritabanıOluştur() static metodu ile nesne oluşturulmalıdır. Problemi Singleton Method tasarım deseni ile kodlayınız. (25 Puan)

<code>public class Veritabanı {</code>		3 Puan
--	--	--------

Fırat Üniversitesi, Mühendislik Fakültesi, Yazılım Mühendisliği Bölümü  
2023 – 2024 Güz Dönemi YMÜ-327 Yazılım Tasarımı ve Mimarisi Ara Sınav Soruları

Öğrenci Numarası:

Ad Soyadı:

Süre 60dk. Başarılar...

<code>private static Veritabani instance;</code>	class ile aynı türde static bir member mevcuttur	5 Puan
<code>private Veritabani(){ Else System.out.println("Veritabani Oluşturuldu"); }</code>	private constructor tanımlanmıştır.	5 Puan
<code>public static Veritabani Olustur(){</code>	static member'a ulaşmak için kullanılan static metot mevcuttur.	5 Puan
<code>if (instance == null){</code>		5 Puan
<code>instance = new Veritabani();</code>		2 Puan
<code>}</code>		
<code>Else System.out.println("Zaten bir veritabanı oluşturulmuş");</code>		3 Puan
<code>return instance;</code>		2 Puan
<code>}</code>		
<code>}</code>		