

<b>mmmmFIRAT ÜNİVERSİTESİ BİLGİSAYAR MÜHENDİSLİĞİ</b> <b>Bilgisayar Müh. Tasarım</b> <b>VİZE SINAVI – 26.11.2019</b>	<b>No:</b>  <b>Ad Soyadı:</b>
--	-------------------------------------

### SORU 1) (30p)

- A. “Arayüze programlama (programming to an interface)” ve “uygulama programlama (programming to an implementation)” kavramlarını tasarım kalıplarının temel kurallarını hatırlayarak açıklayınız.

Tasarım kalıplarının temel kurallarına göre uygulamaya programlama yazmak yerine ara yüze programlama yazılmalıdır.

Uygulama programlamada, somut bir sınıf değişkeninden yine somut bir sınıf üretiyoruz.

Arayüze programlamada, süper sınıf ya da arayüzden ilgili sınıfı üretiyoruz.

Arayüze yazılım yaparak (özellikle değişkenlik gösteren davranış ve metotlar arayüzün arkasına toplanarak) run-time sırasında ilgili davranışların değiştirilmesi veya geliştirilmesine olanak sağlıyoruz.

Not: Gang of Four Tasarım Kalıpları kitabında ara yüz kullanmayı aşağıdaki avantajlarla anlatmıştır:

1. Nesneler, müşterilerin beklediği ara yüzlere bağımlı kaldığı sürece, kullandıkları nesne türlerinin farkında olmazlar.
2. Müşteriler bu nesneleri uygulayan sınıfların farkında olmamalıdır. Müşteriler yalnızca arayüzü tanımlayan soyut sınıfları bilecektir.

- B. Tasarım sürecinin iki çıktısını işaretleyiniz?

- Kod uygulaması
- Tasarım planı
- Kavramsal tasarım
- Teknik tasarım

- C. Kapsülleme (encapsulation) prensibini üç önemli fikrini hatırlayarak (paketleme, açığa çıkarma ve kısıtlama) açıklayınız.

Paketleme (bundle) özelliği: Değerleri (ya da verilerin) ve davranışları (ya da fonksiyonların) kendi kendine yetebilen bir nesneye dönüştürebilen (self-contained object) paketleme yeteneğidir.

Açığa çıkarma (expose) özelliği: Bir nesnenin belirli bir verisini ya da fonksiyonunu genellikle bir arayüz (interface) ile diğer nesnelerden erişilebilecek şekilde “açığa çıkarma” yeteneğidir.

Kısıtlama (restrict) özelliği: Sadece nesne içindeki belirli fonksiyonlara ve verilere kısıtlı erişme yeteneğidir.

### SORU 2) (20p)

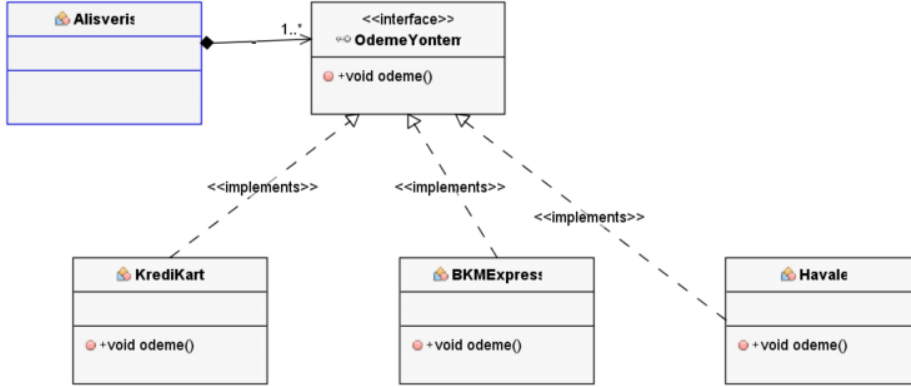
Bir ATM için bir kart okuyucu önemlidir. Kart okuyucu, ATM ile kartı bağlayan arayüzdür. Bir kart takıldığında ATM'ye bildirilmeli ve karttaki bilgileri okuyabilmelidir. Kartı çıkarmak ve kartı tutmak aynı zamanda kart okuyucunun temel işlevleridir. Bu senaryo göz önünde bulundurularak bir kart okuyucu sınıfının CRC kartını tasarlayınız.

KartOkuyucu Sınıfı	
Responsibilities	Collaborators
Kart yerleştirildiğinde Atm'ye bildir	ATM
Karttan bilgiyi oku	KART
Kartı çıkar	
Kartı tut	

### SORU 3) (20p)

Bir alışveriş sitesi için, müşterinin istediği ürünleri alışveriş sepetine koyup ödemeyi üç farklı şekilde gerçekleştirebilmesini sağlayan bir ödeme mekanizması geliştirmek istenmektedir. Aşağıdaki adımları takip ederek bu problemi tasarlayınız ve UML diyagramını çizin. Bu problemin tasarımına uygun hangi tasarım kalıbı takip edildiğini ve sebebini açıklayınız.

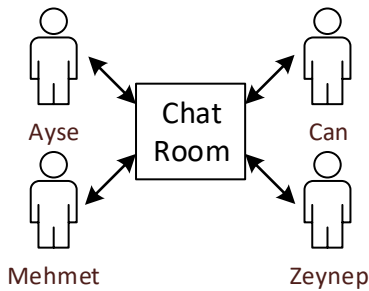
1. Üç farklı ödeme tipi vardır: KrediKarti (Visa ya da MasterCard), BKM Express ve Havale
2. Ödeme yöntemi müşterinin seçimine göre çalışma zamanında (run time) değiştirilmelidir.
3. Davranışsal değişiklikleri dinamik hale getirebilmek için arayüz (interface) kullanılmalıdır.



**Strateji Tasarım Kalıbı.**

### SORU 4) (15)

Bir forum sitesi için canlı mesajlaşma uygulaması (chat room) gerçekleştirmeniz isteniyor. Kullanıcılar bu uygulamaya katılabilir ve mesajlarını paylaşabilirler. Ahmet bir mesaj gönderdiği zaman gönderdiği mesajı uygulamaya katılan tüm kullanıcılar görebilecektir. Bu uygulamayı gerçekleştirmek için kullanacağınız uygun tasarım kalıbı ne olmalıdır, nedenini kısaca açıklayınız.



**Medyator Tasarım Kalıbı. Çoklu nesnelerin iletişimi.**

**Soru 5 (15p).** Tasarım kalıplarının (design patterns) temel kuralları nelerdir? Yalnızlık-tekil (singleton) tasarım kalıbını bir örnek ile kısaca açıklayınız.

#### Temel Kurallar:

- ✓ Uygulamaya (implements) değil de arayüze (interface) program yazılmalı
- ✓ Nesneler arasında esnek bağ oluşturulmalı, nesneler kesinlikle birbirlerine somut bağ oluşturmamalı
- ✓ Değişken ve gelişmeye açık olan bölümler tespit edilmeli ve bir çatı altında tutulmalı
- ✓ Kalıtım (inheritance) yerine kompozisyon (composition) kullanılmalı
- ✓ Tasarım geliştirmeye açık olmalı fakat değiştirilmeye kapalı olmalı
- ✓ Somut sınıflar kullanılmamalı. Somut sınıflar yerine daha çok soyut sınıflar (abstract veya interface) kullanılmalı. Yalnızlık-tekil tasarım kalıbı, bir nesnenin sadece ve sadece bir kez yaratılmasını sağlar, örnek veritabanına bağlantı.

Not: Sınav süresi 70 dk. Başarılar Dilerim.

Dr. Öğr. Üyesi Betül AY