

FIRAT ÜNİVERSİTESİ

TEKNOLOJİ FAKÜLTESİ

YAZILIM MÜHENDİSLİĞİ BÖLÜMÜ

BİLGİSAYAR BİLİMLERİNE GİRİŞ DERSİ

LABORATUVAR FÖYÜ

HAFTA: 8

KONU: VERİ YAPILARININ TEMELLERİ

Doç. Dr. MUHAMMET BAYKARA

Arş. Gör. Z. BEYZA METİN

Laboratuvar Föyü: Veri Yapılarının Temelleri

BÖLÜM 1: AMAÇ VE TEMEL KAVRAMLAR

Amaç: Bu laboratuvarın amacı, bir yazılım mühendisinin, gerçek dünyadaki bir problemi bilgisayar ortamına aktarmak için nasıl "Veri Modelleri" kullandığını anlamak ve bu modellerin bellekte nasıl organize edildiğini (Veri Yapıları) öğrenmektir.

Temel Kavram (Veri Modeli): Gerçek dünyadaki bir varlığı (Öğrenci, Bina, Araba) ve bu varlığın özelliklerini (Adı, Rengi, Numarası) bilgisayarın anlayacağı şekilde soyutlama sanatıdır.

BÖLÜM 2: VERİ MODELLEME (VERİTABANI TASARIMI)

Senaryo: Bir Öğrenci Bilgi Sistemi (OBS) tasarlıyoruz. Sistemimizin en önemli tablosu **ogrenciler** tablosudur.

Görev: Bu tablodaki verileri en verimli şekilde saklamak için her "Sütun" için doğru "Veri Tipi"ni seçin. (Örn: **Integer**, **String**, **Date-Timestamp**, **Double-Float**)

Sütun Adı	Veri Tipi
ogrenciNo	
adsoyad	
kayitTarihi	
notOrtalamasi	
sinif	

FIRAT ÜNİVERSİTESİ – YAZILIM MÜHENDİSLİĞİ
BİLGİSAYAR BİLİMLERİNE GİRİŞ LABORATUVARI (8.HAFTA)

BÖLÜM 3: KOD MODELLEME (UML SINIF DİYAGRAMI)

Veritabanı veriyi saklar; Yazılım (Kod) ise bu veriyi işler. Yazılım mühendisleri, bu kodun mimarisini ve sınıflarını tasarlamak için **UML (Unified Modeling Language)** diyagramları kullanır.

Görev: Öğrenciler tablosu için belirlediğiniz veri sütunları, kodlama dünyasında bir Öğrenci Sınıfı'nın "**Nitelikleri**" (**Properties/Attributes**) olur. Bu sınıfın ayrıca "Not Ortalamasını Hesapla" gibi "**Eylemleri**" (**Methods/Functions**) de vardır.

Şimdi, Ekle > Şekiller menüsünü kullanarak oluşturduğunuz aşağıdaki örneği olan **boş, 3 bölmeli dikdörtgeni** (UML Sınıf Diyagramı) aşağıdaki kurallara göre doldurun:

- **En Üst Bölme (1. Bölme):** Sınıfın Adı yazılır. Buraya **Oğrenci** yazın.
- **Orta Bölme (2. Bölme):** Sınıfın "Nitelikleri" (yani verileri) yazılır. Görev 1'deki **ogrenciNo** ve **adSoyad** alanlarını buraya yerleştirin.
- **En Alt Bölme (3. Bölme):** Sınıfın "Eylemleri" (yani yapabildiği işler) yazılır. Buraya **notOrtalamasiniHesapla()** ve **dersKaydiniYap()** eylemlerini yerleştirin.

SınıfAdı
Veriler
Eylemler

FIRAT ÜNİVERSİTESİ – YAZILIM MÜHENDİSLİĞİ
BİLGİSAYAR BİLİMLERİNE GİRİŞ LABORATUVARI (8.HAFTA)

Soru 1: `adSoyad` (bir veri/nitelik) ile `dersKaydiniYap()` (bir eylem/metot) arasındaki temel fark nedir? UML şemasında neden farklı bölmelere yazdık?

Cevap:

Soru 2 (İpucu: + herkese açık (public), - ise özel (private) demektir): `ogrenciNo` gibi önemli bir verinin başına (yani 2. bölmedeyken) `+` mı, yoksa `-` mi koymak daha güvenli bir kod tasarımı olur? Neden?

Cevap:

BÖLÜM 4: GÖREV 3 - İLERİ DÜZEY VERİ MODELLERİ

Giriş: Verileri `Ogrenciler` gibi basit bir listede tutmak kolaydır. Ancak bazı problemler daha karmaşık modeller gerektirir.

Görev: Aşağıdaki 2 senaryo için hangi veri modelinin daha mantıklı olacağını nedenleriyle açıklayın.

Soru 1 (Ağaç Modeli): Üniversitenin Rektör'den başlayıp Dekanlara, oradan Bölüm Başkanlarına inen **hiyerarşik** yapısını saklamak için basit bir liste mi, yoksa Ağaç modeli mi daha mantıklıdır?

Cevap:

Nedeni:

Soru 2 (Graf Modeli): Kampüsteki binalar arası **yolları ve bağlantıları** (örn: Mühendislik'ten Fen Fakültesi'ne 2 farklı yol var, ikisi de 300 metre) saklamak için Graf modeli mi daha uygundur?

Cevap:

Nedeni:

BÖLÜM 5: DİZİYE (ARRAY) GİRİŞ

Problem: Tebrikler, Sayfa 1'de bir Öğrenci'yi ve Bina'yı modellediniz. Peki 1000 adet öğrencinin bilgisini veya 100 adet sınav notunu bellekte (RAM) nasıl organize edip tutarız? Bunun en temel yolu bir **Dizi (Array)** kullanmaktır.

Dizi, **aynı türdeki** verilerin (örn: hepsi `tamSayı` veya hepsi `Öğrenci` modeli) bilgisayar belleğinde **arka arkaya (bitişik)** bir şekilde saklandığı en temel veri yapısıdır.

Dizideki bir elemanın "adresi" veya "sıra numarası"dır. Dizinin ilk elemanına ulaşmak için 0 numaralı indeks kullanılır.

Örnek:

```
int[] vizeNotlari = [45, 80, 72]
```

- (Bu dizide `vizeNotlari[0]`'ın değeri 45'tir.)
- (Bu dizide `vizeNotlari[1]`'in değeri 80'dir.)
- (Bu dizide `vizeNotlari[2]`'nin değeri 72'dir.)

Senaryo: Bir dersin vize notları aşağıdaki gibi bir dizide tutulmaktadır. Bu diziye kullanarak aşağıdaki **iki problemi** çözen algoritmaların **sözde kodunu (pseudocode)** yazın.

```
int[] vizeNotlari = [45, 80, 72, 50, 95, 60]
```

Ortalama Hesaplama: Bu dizideki notların ortalamasını bulan algoritmanın sözde kodunu aşağıdaki boşluğa yazın. (*İpucu: Önce tüm notları toplamak için bir döngü kurun, sonra toplamı eleman sayısına (6) bölün.*)

Cevap:

FIRAT ÜNİVERSİTESİ – YAZILIM MÜHENDİSLİĞİ
BİLGİSAYAR BİLİMLERİNE GİRİŞ LABORATUVARI (8.HAFTA)

Koşullu Sayım: Bu dizide 'Geçme Notu' olan 60'dan *düşük* kaç not olduğunu (kalanları) sayan algoritmanın sözde kodunu aşağıdaki boşluğa yazın.

(İpucu: Bir sayaç değişkeni (*kalanSayisi* = 0) tanımlayın. Döngü içinde bir 'Eğer (If)' koşulu kullanarak notu kontrol edin ve koşul sağlanıyorsa sayacı 1 artırın.)

Cevap:

BÖLÜM 6: GÖREV 5 - DİZİ ANALİZİ (EKLEME / SİLME)

Giriş: Diziler, yukarıda gördüğünüz gibi verilere **erişmek** (indeks ile, örn: `notlar[2]`) ve diziyi **gezmek** (döngü ile) için çok hızlı ve verimlidir. Peki, dizinin *ortasına* eleman eklemek veya *ortasından* eleman silmek de bu kadar hızlı mı?

Araçlar: Aşağıdaki senaryoyu analiz etmek için **iki yöntemden birini** seçebilirsiniz:

1. Yöntem (Simülasyon): visualgo.net sitesine gidin, "Array" modülünü seçin ve senaryoyu canlı olarak izleyin.

2. Yöntem (Kâğıt üzerinde): Senaryoyu mantıksal olarak kâğıt üzerinde kendiniz çizin ve analiz edin.

Senaryo: [10, 20, 30, 40] elemanlarından oluşan bir dizimiz var. Bu dizinin **2. indeksine** (yani 30'un olduğu yere) 15 sayısını eklemek istiyoruz.

Bu eklemeyi yapabilmek için (15'e yer açmak) 30 ve 40 sayılarına ne olması gerekir?

FIRAT ÜNİVERSİTESİ – YAZILIM MÜHENDİSLİĞİ
BİLGİSAYAR BİLİMLERİNE GİRİŞ LABORATUVARI (8.HAFTA)

Gözlem: 15'e yer açmak için 30 ve 40 sayılarına ne olması gerekir? (Simülasyonda ne gördünüz veya kâğıtta ne çizdiniz?)

Cevap:

Analiz: Peki, bu dizide 1.000.000 (bir milyon) eleman olsaydı ve biz en başa (0. indekse) yeni bir eleman ekleysek, bu "kaydırma" işlemi neden felaket bir performans sorununa (yavaşlığa) yol açardı?

Cevap:

Senaryo 2: Dizisinin **1. indeksindeki** 20 sayısını sildiğimizi varsayalım.

Gözlem: 30 ve 40 sayılarının, oluşan 'boşluğu' kapatmak için ne yapması gerekir?

Cevap:

Analiz: Dizide 1.000.000 eleman olsaydı ve biz en baştaki (0. indeksteki) elemanı silseydik, bu "sola kaydırma" işlemi neden **verimsiz** olurdu?

Cevap:

FIRAT ÜNİVERSİTESİ – YAZILIM MÜHENDİSLİĞİ
BİLGİSAYAR BİLİMLERİNE GİRİŞ LABORATUVARI (8.HAFTA)

BÖLÜM 8: TESLİMAT

Bu laboratuvar çalışmasını tamamladıktan sonra, **doldurmuş olduğunuz bu Word dosyasını** teslim etmeniz gerekmektedir.

Dosya Adlandırma: Doldurulmuş olan bu Word dosyasını .docx veya .pdf formatında, aşağıdaki gibi adlandırarak Classroom'a yükleyiniz:

OgrenciNo_AdSoyad_Lab6.doc

Arş. Gör. Z. Beyza Metin