

NESNEYE YÖNELİK PROGRAMLAMA

Ders İşleyişi
ve

Tarihsel ve mantıksal bağlamda nesneye yönelik programlama

Emir Öztürk

[illegible]

DEĞERLENDİRME

- Genel Ortalama
 - Vize: %30
 - Proje: %20
 - Final: %50



KAYNAKLAR

- Java How to Program Ninth Edition, Harvey M. Deitel & Paul J. Deitel, Prentice-Hall.
- UML Distilled, Martin Fowler, Addison-Wesley.

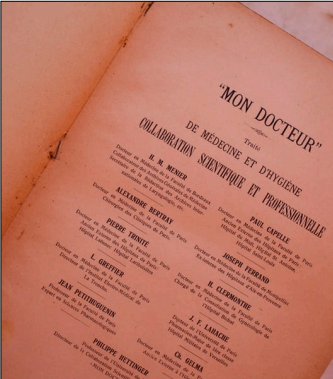
4



DERSİN AMACI

- Temel kavramlar
 - Programlama Dilleri
 - Nesneye Yönelik Programlama
 - Soyutlama
 - Kapsülleme
 - Çok biçimlilik
 - Kalıtım
- + Java

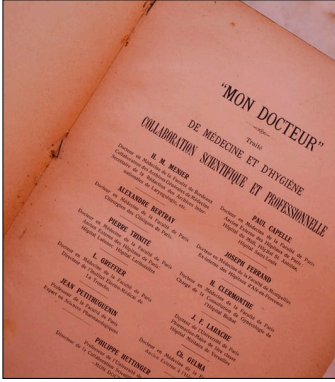
5



DERSİN KONULARI

- Programlama Paradigmaları
- Java dilinin genel tanıtımı
- Sınıflar ve nesneler
- Farklı dillerde sınıf yazımı ve karşılaştırma
- Yapıcılar ve yıkıcılar
- Sarmalama
- Arasınay

6



DERSİN KONULARI (DEVAM)

- Çok biçimlilik
- Kalıtım
- Soyutlama
- UML ile analiz ve tasarım
- SOLID ve Design Patterns
- Proje Sunumları
- Proje Sunumları

7



PROJENİN UYGULANMASI

- Kişi sayısı
 - Numara - Ad Soyad (1 Hafta)
- Proje Seçimi (1 Hafta)
 - Proje Listesi
 - Kendi Projeniz
 - Bir proje üç grup
- OTOMASYON**
- UNITY*****

8

1) En fazla x'er kişilik proje grupları oluşturulmalıdır.

2) Bir hafta içinde proje grupları üyelerini belirlemeli ve Numara - Ad Soyad formatında bildirmelidir.

3) Gruplar, konuların belirlendiği haftadan sonraki bir haftalık süre içerisinde seçmiş oldukları konuyu belirlemeli bildirmelidir.

4) Bir proje konusunu en fazla 3 grup seçebilir. Aynı konuyu seçen grup sayısının 3'ten fazla olması durumunda kura çekilecektir.

5) İsteyen grup kendi belirlediği proje konusu ile gelebilir. Ancak belirledikleri proje konusunu, amaç ve hedefleri detaylandırılmış bir şekilde sunmalıdırlar.

6) Proje konusu olarak herhangi bir "OTOMASYON" veya "STOK TAKİBİ" yapan uygulama projeleri kabul edilmeyecektir.



PROJENİN UYGULANMASI

- > 🚩🚩🚩🚩 NESNEYE YÖNELİK 🚩🚩🚩
- > Proje %20
- > UML / Sınıf diyagramı
- > Kod teslimi
- > Sunum
- > Değerlendirme
 - > NYP
 - > Sunum
 - > Dosyalar

9

- 7) Bütün projeler nesneye yönelik modellemeye ait kavramları uygulamaya yönelik çalışmalar olmalıdır.
- 8) Projeler başarı notunun hesaplanmasında %20'lik paya sahiptir (vize %30, final %50).
- 9) Projelerin analiz ve tasarım aşamasında projeye ait belirtilen UML diyagramları hazırlanarak teslim edilmelidir. Proje notu değerlendirmesinde dikkate alınacaktır.
- 10) Projeler, dönem sonunda teslim edilmelidir. Proje bitiminde her grubun (grup üyelerinin tümü) projesine ait sunumu önceden belirlenen gün ve saatte sunmaları zorunludur. Sunumu yapılmayan projeler değerlendirilmeyecektir.
- 11) Projelerini teslim etmeyen ve sunum yapmayan grupların projeleri değerlendirmeye alınmayacaktır.
- 12) Değerlendirme sırasıyla 20 30 10 + 20 ve proje büyüklüğü / kişi sayısı olarak kalan 20



PROGRAMLAMA DİLİ

- > Programlama Dili
 - > Yazım kuralları
- > Imperatif Programlama
- > Prosedürel Programlama
- > Deklaratif Programlama
- > Fonksiyonel Programlama
- > Nesneye Yönelik Programlama

10

- Programlama dili, programlanabilir herhangi bir ortama, hangi veri üzerinde işlem yapılacağı, verinin nasıl depolanıp iletileceği, hangi koşullarda hangi işlemlerin hangi sıra ile yapılacağı gibi konularda talimat vermeyi sağlayan ve belirli bir yazım kuralı (sentaks) olan dildir.
- İmperatif programlama, bir programın yapacağı işlemlerin sırasının bilgisayarın anlayacağı kod karşılıkları ile ifade edilmesi için kullanılan bir modeldir. İmperatif programlama modelinde bir programın adım adım nasıl çalışacağını ve her adımda ne yapılacağını belirtmesi gerekmektedir.
- Bildirimsel programlamada yazılan kod, bir programın ne iş yaptığını (mantığını),

işlem adımlarını belirtmeden tanımlamayı sağlar.

İmparatif programlamanın aksine bir programın nasıl çalışacağından çok ne yapacağını ifade etmeyi hedefler.

Bunun amacı programcıya programın mantığı üzerinde odaklanma fırsatı sunmaktır.

Fonksiyonel programlama, matematiksel fonksiyonların kullanılması ve verilerin değiştirilmemesi prensibine dayanmaktadır.

Verilerin değiştirilmesine izin verilmemesi ile programın her anında bir girdi verisi için bir fonksiyonun hep aynı sonucu üretmesi ve bu sayede hesaplamada hız



```
function hexSearch() {
  var hex = $('#hexSearch').val().replace(/[\r\n]/g, '');
  if (hex.length == 0) {
    hex.length == 0 ? hex.length == 0 : hex.length == 0;
    var rx = /([0-9a-f]{4})/gi;
    var rx2 = /([0-9a-f]{2})/gi;
    if (hex.test(hex) || rx.test(hex) || rx2.test(hex)) {
      if (hex.length == 0) {
        if (hex.length == 0) {
          hex = hex.toUpperCase();
          hex = hex[0] + hex[1] + hex[2] + hex[3] + hex[4] + hex[5] + hex[6] + hex[7];
        } else {
          $('#notificationText').html('The thing you typed into the input is not a hex value');
          $('#notificationText').css('display', 'block');
          return;
        }
      } else {
        $('#notificationText').html('The thing you typed into the input is not a hex value');
        $('#notificationText').css('display', 'block');
        return;
      }
    }
  }
  var colors = ['#7B2323', '#A020F0', '#FFD700', '#FF00FF', '#0000FF', '#000000'];
  var colors = colors;
  colors = colors;
  colors = colors;
}
```

NYP'NİN TEMEL KAVRAMLARI

- > Soyutlama (Abstraction)
- > Sarmalama (Encapsulation)
- > Çok Biçimlilik (Polymorphism)
- > Kalıtım (Inheritance)

11



SOYUTLAMA

- > Gereksiz detayların elenmesi
 - > Veri bazlı
 - > İşleyiş bazlı

12

Soyutlamanın amacı bir işin nasıl yapıldığından çok ne yaptığının bilinmesidir. Veri bazlı soyutlamada istenilen işlemler gerçekleşirken manipüle edilen verinin takibinin yapılmasına gerek yoktur. İşlev bazlı manipülasyonda ise yapılan işlemlerin nasıl yaptığı bilinmez fakat ne yaptığı bilinmektedir.

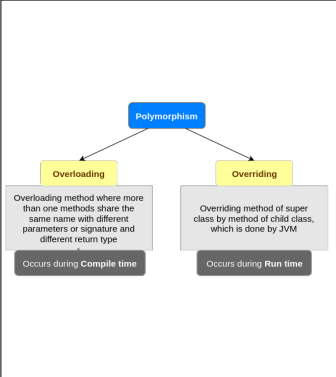


SARMALAMA

- Veri bazlı sarmalama
- Fonksiyon bazlı sarmalama

13

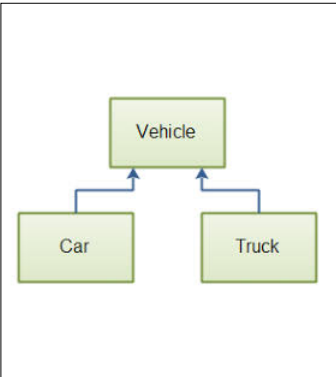
Bir nesnenin iç yapısını diğerlerinden saklar. Detaylar gözardı edilir. Örneğin bir araçta durmayı, kalkmayı ve yön değiştirmeyi biliriz fakat benzini nasıl yaktığı tekerleklerinin nasıl döndüğü gibi detaylardan habirimiz olmasına gerek yoktur. Amaç implementasyon detayından çok neyin yapıldığının bilinmesidir.



ÇOK BİÇİMLİLİK

- Veri türüne göre birden fazla implementasyon
- İki tamsayı
- İki ondalıklı sayı
- Bir tamsayı bir ondalıklı sayı
- Farklı bir alt sınıfta yeni bir implementasyon
- Parametreler aynı
- Parametreler farklı

14



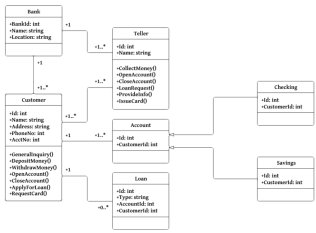
KALITIM

- Hiyerarşik sınıflandırma
- Nesneler arası ilişki

15

Kalıtım ile bir sınıfın diğerinin atası olduğu kabul edilebilir. Alt sınıf ebeveyn sınıfın özelliklerini ve davranışlarını kalıtım yolu ile alır. Örneğin bir hayvan sınıfına dahil, hayvan sınıfının tüm ortak özelliklerini içeren özel kedi, köpek gibi sınıflar oluşturulabilir. Bu sınıflar ortak özelliklerini üst sınıftan alırken, kendine ait özelliklerini ise ayrı olarak tanımlayabilirler.

SINIF VE NESNE KAVRAMLARI



Sınıf	Nesne
Plan	Örnek
Mantıksal	Fiziksel
Bellek Kullanmaz	Bellek Kullanır
Tekil	Çoklu

16

Sınıf nesneler için bir plan görevi görür. Nesneler sınıfların bir örneğidir (instance).

Sınıflar mantıksal tanımlanır. Nesneler sınıfların fiziksel oluşturulmuş halleridir. Sınıflar mantıksal oldukları için bellek kullanmaz, nesneler ise tanımlandıkları yerde bellekte yer kaplarlar.

Sınıflar bir kere tanımlanır. Nesneler ise birden fazla yerde ihtiyaç duyulduğu an tanımlanabilirler.