

Nesne Yönelimli Programlama

Meltem Yıldırım

11 Mart 2009

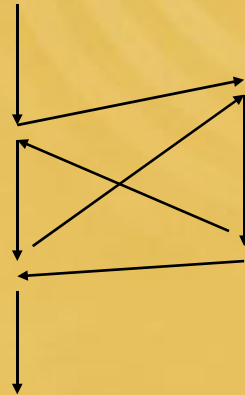
Nesne Yönelimli Programlama

- Genel kavramlar
- Soyutlama - Abstraction
- Kapsülleme - Encapsulation
- Kalıtım – Inheritance
- Çok biçimlilik - Polymorphism

Programlama Dili

- İşlem yönelimli model
 - Kod büyüklüğü
 - Karmaşıklık

main program procedure



Nesne Yönelimli Programlama

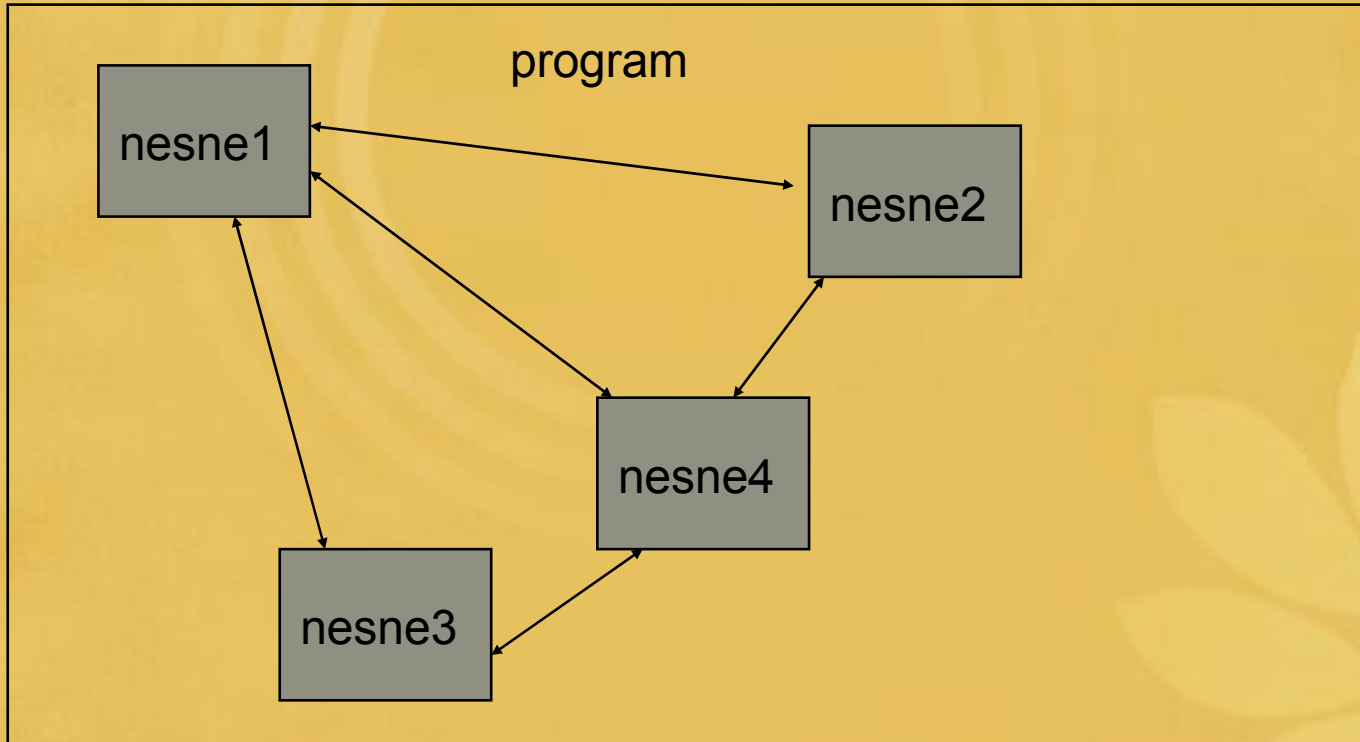
- 90'lara kadar ana akım değil
- Birçok programlama dili NYP destekliyor
- Java, C++, Python, Rubby, Smaltalk, C#, PHP, Visual Basic

Nesne Yönelimli Programlama

- Nesne ve onların etkileşimini kullanan bir programlama paradigması
- Program
 - nesne
 - iyi tanımlanmış arabirim çerçevesinde şekillenme

Nesne Yönelimli Programlama

Mesaj göndererek etkileşimde bulunma





Sınıf

- Sınıf bir nesne için şablon
- Nesne sınıfın örneği

```
class sınıfAdı
{
    tip değişken1;
    tip değişken2,
    tip metotismi(parametre)
    {
        .....
    }
}
```



Nesne Yaratılması


- Sınıfın nesnesini elde etme
 - Sınıf tipinde bir değişken bildirimini
 - Nesnenin fiziksel kopyasını elde etme

`sınıfAdı yeni = new sınıfAdı();`



Mesaj Geçirme


- `Class b1 = new Box();`
- `b1.setWidth(5);`
- `b1` nesnesine gönderilen mesaj
- `setWidth` metodunu 5 argümanı ile uygula



Arayüz- Interface

- Gerçekleştirmesi olmadan sınıf gösterimi
- Sınıfın ne yapması gerektiği belirtilir

```
interface person
{
    string getSurname(string name);
    int getAge();
}
class teacher implements person
{
    public getAge()
    {
        .....
    }
}
```




Soyutlama- Abstraction

- NYP'nın gerekli unsurlarından
- Karmaşıklığın çözümü
- Örnek : Araba
 - On bin parçadan oluşma
 - Tek bir nesne
- Hiyerarşik sınıflandırma
 - Semantik katmanlara ayırma
 - Daha yönetilebilir parçalar

Kapsülleme- Encapsulation

- Dış kaynaklı karıştırma ve yanlış kullanımdan koruma
- Erişim - arayüz tarafından kontrol edilebilme
- Temeli sınıf
- Gerçekleşme ayrıntılarından bağımsız kullanabilme

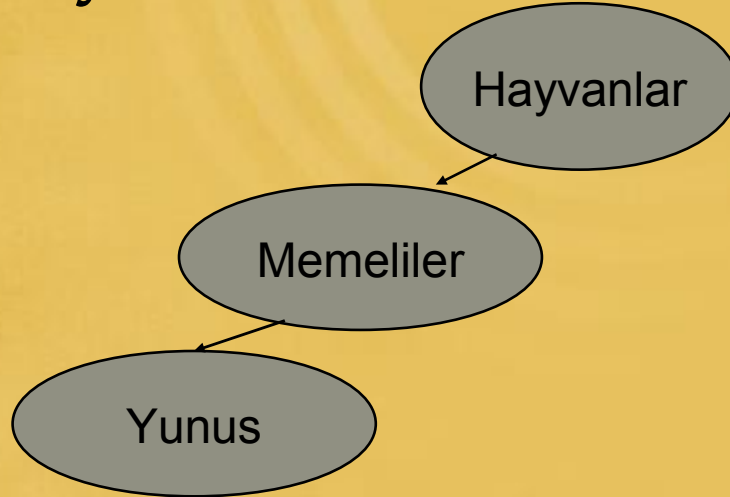


Kapsülleme

- Sınıf içerisinde deęişken ve metotlara erişim
 - Genel – tüm sınıflar
 - Özel – tanımlandığı sınıf
 -

Kalıtım- Inheritance

- Bir nesnenin diğerk bir nesnenin özelliklerini kazanması işlemi
- Alt sınıf sınıfın daha uzmanlaşmış versiyonu



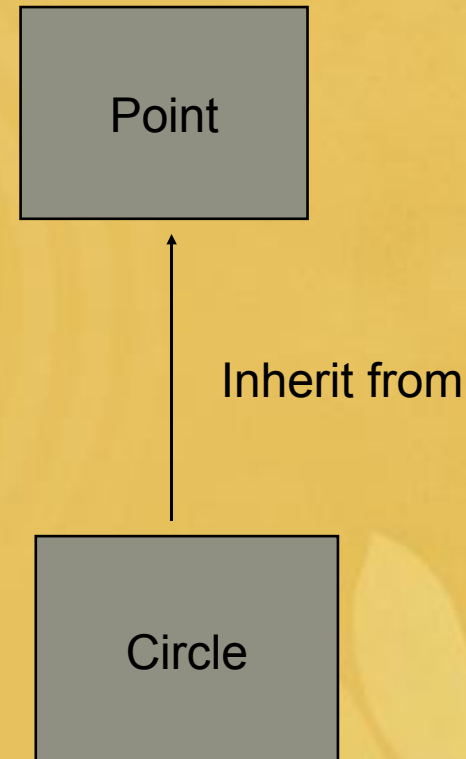
Kalıtım

- Daire bir noktadır

```
class Circle inherits Point
```

```
{  
    int radius;  
    setRadius(int newRadius);  
    getRadius();  
}
```

```
Circle c1;  
c1.setX(1);
```



Çok Biçimlilik - Polymorphism

- Aynı isimli metotlardan veri tipine uygun olanın yanıt vermesi
- Add(double d1, double d2);
- Add(int d1,int d2);
- Overload
- + : int, float, string

Özellikler

- Yeniden kullanım
- Geliştirilebilir
- Güvenilir
- Dayanaklı



References

- Java 2, Herbert Schildt
- C++ Strategies and Tactics, Robert B. Muray
- Introduction to OOP Using C++, Peter Müller
- <http://en.wikipedia.org>

SORULAR

