

Nesneye Dayalı Yazılım Geliştirme

Kodlama

Sınıf tanımları yazılım sınıflarının diyagramlarından yararlanılarak oluşturulur. Karmaşık veri tiplerine (örneğin sınıf) sahip üyeler referans ya da işaretçi olarak yaratılmalıdır.

```

public class SalesLineItem // Java
{
    private int quantity;
    private ProductSpecification description;
    public SalesLineItem(ProductSpecification spec, int qty) {...}
    public Money getSubtotal() {...}
}

```

Basit nitelik

Referans tipi nitelik

ProductSpecification

SalesLineItem

quantity : Integer

description : Text

price : Money

itemID : ItemID

addSaleLineItem(spec, qty)

getSubtotal() : Money

getDescription() : Text

getPrice() : Money

getItemID() : ItemID

www.buzluca.info@dyg

©2007-2008 Dr. Pinar BUZLUCA

6.1

Nesneye Dayalı Yazılım Geliştirme

Sınıfları Gerçekleme Sırası

Sınıflar gerçekleştirirken (kodlama ve sinama) en az bağımlı sınıftan başlanır ve diğerlerine (en çok bağımlı sınıfa) doğru gidilir. Yani 6.5'te örnek sınıfların olası gerçekleşme sıraları gösterilmiştir.

Birim Sinama (Unit Test): Her sınıf bir bütün olduğundan ve tek başına bir anlam taşıdığından ayrı bir birim olarak test edilir.

Sınıfların kodlarını yazmadan önce bu sınıfların sinamasını yapacak program parçalarının yazılması uygun bir yöntemdir (*Test-first programming*).

Sinamayı yapan program parçası, sinaması yapılacak olan sınıftan nesneler yaratır, bu nesnelere mesajlar gönderip sonuçlar alır ve sınıfın gönderilen mesajlara doğru yanıtlar verip vermediği kontrol edilir.

Sinama programını önce yazmanın yararları:

- Sinama programlarının göz ardı edilmesi önlenir.
- Sinama programı yazılırken kodlanacak olan sınıfın arayüzü hakkında daha ayrıntılı düşünülüş olur.
- Kodlarda değişiklik yapıldığında yapılan değişikliğin bir hataya neden olup olmadığını belirleyecek hazır sinama programları el altında bulunur.

www.buzluca.info@dyg

©2007-2008 Dr. Pinar BUZLUCA

6.4

Nesneye Dayalı Yazılım Geliştirme

Metotlar etkileşim diyagramlarından yararlanılarak oluşturulur.

```

public class Register
{
    private ProductCatalog catalog;
    private Sale currentSale;
    public Register(ProductCatalog pc) {...}
    public void endSale() {...}
    public void enterItem(ItemID id, int qty) {...}
    public void makeNewSale() {...}
    public void makePayment(Money cashTendered) {...}
}

```

ProductCatalog

Register

catalog

getSpecification(...)

endSale()

enterItem(id, ItemID, qty : Integer)

makeNewSale()

makePayment(cashTendered : Money)

Sale

date : Date

isComplete : Boolean

time : Time

becomeComplete()

makeLineItem(...)

makePayment(...)

getTotal()

currentSale

Payment

amount : Money

payment

www.buzluca.info@dyg

©2007-2008 Dr. Pinar BUZLUCA

6.2

Nesneye Dayalı Yazılım Geliştirme

Sınıfların Gerçeklenme Sıraları

Store

address : Address

name : Text

addSale(...)

ProductCatalog

descriptions (Map)

getSpecification(...)

ProductSpecification

description : Text

price : Money

itemID : ItemID

SalesLineItem

quantity : Integer

getSubtotal()

Sale

date : Date

isComplete : Boolean

time : Time

becomeComplete()

makeLineItem(...)

makePayment(...)

getTotal()

Register

endSale()

enterItem(...)

makeNewSale()

makePayment(...)

Payment

amount : Money

payment

completedSales (ordered)

currentSale

catalog

descriptions

description

quantity

getSubtotal

becomeComplete

makeLineItem

makePayment

getTotal

enterItem

makeNewSale

endSale

addSale

www.buzluca.info@dyg

©2007-2008 Dr. Pinar BUZLUCA

6.5

Nesneye Dayalı Yazılım Geliştirme

Şekildeki 1 ve 2 numaralı mesajlar Register sınıfının enterItem metodunda birer satır oluştururlar.

```

{
    ProductSpecification spec = catalog.getSpecification(id);
    ...
    currentSale.makeLineItem(spec, qty);
}

```

enterItem(id, qty)

1: spec := getSpecification(id)

2: makeLineItem(spec, qty)

ProductCatalog

Register

Sale

www.buzluca.info@dyg

©2007-2008 Dr. Pinar BUZLUCA

6.3

Nesneye Dayalı Yazılım Geliştirme

Örnek Kodlama

Bu bölümde örnek POS sistemine ilişkin yazılım sınıflarının bir kısmı örnek olarak Java dilinde kodlanmıştır.

```

public class Register
{
    private ProductCatalog catalog;
    private Sale currentSale;

    public Register( ProductCatalog catalog )
    {
        this.catalog = catalog;
    }

    public void makeNewSale()
    {
        currentSale = new Sale();
    }

    public void enterItem( ItemID id, int quantity )
    {
        if (currentSale != NULL){
            ProductSpecification spec = catalog.getSpecification(id);
            currentSale.makeLineItem(spec, quantity);
        }
        else ... // exception handler
    }

    public void endSale()
    {
        currentSale.becomeComplete();
    }

    public void makePayment( Money cashTendered )
    {
        currentSale.makePayment (cashTendered);
    }
} // Register sınıfının sonu

```

www.buzluca.info@dyg

©2007-2008 Dr. Pinar BUZLUCA

6.6

```

public class ProductSpecification
{
    private ItemID id;
    private Money price;
    private String description;

    public ProductSpecification( ItemID id, Money price, String description )
    {
        this.id = id;
        this.price = price;
        this.description = description;
    }

    public ItemID getItemID() { return id; }
    public Money getPrice() { return price; }
    public String getDescription() { return description; }
}

```

```

public class Sale
{
    private List lineItems = new ArrayList();
    private Date date = new Date();
    private boolean isComplete = false;
    private Payment payment;

    public Money getBalance()
    {
        return payment.getAmount().minus(getTotal());
    }
    public void becomeComplete() { isComplete = true; }
    public boolean isComplete() { return isComplete; }
    public void makeLineItem( ProductSpecification spec, int quantity )
    {
        lineItems.add ( new SalesLineItem(spec, quantity) );
    }
}

```

```

// Sale sınıfının devamı
public Money getTotal()
{
    Money total = new Money();
    Iterator i = lineItems.iterator();
    while( i.hasNext() )
    {
        SalesLineItem sli = (SalesLineItem) i.next();
        total.add( sli.getSubtotal() );
    }
    return total;
}

public void MakePayment ( Money cashTendered )
{
    payment = new Payment (cashTendered);
}

// Sale sınıfının sonu

```