Java Code of the NextGen POS

The following code is taken from the book (section 20.11)
Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, Third Edition
By Craig Larman

"Description" is replaced by "Specification"

This code defines a simple case; it is not meant to illustrate a robust, fully developed Java program with synchronization, exception handling, and so on.

Class Payment

```
// all classes are probably in a package named
// something like:
package com.foo.nextgen.domain;


public class Payment
{
   private Money amount;

   public Payment( Money cashTendered ){ amount = cashTendered; }
   public Money getAmount() { return amount; }
}
```

Class ProductCatalog

```
public class ProductCatalog
{
   private Map<ItemID, ProductSpecification>
         specifications = new HashMap()<ItemID, ProductSpecification>;

   public ProductCatalog()
   {
      // sample data
      ItemID id1 = new ItemID( 100 );
      ItemID id2 = new ItemID( 200 );
      Money price = new Money( 3 );

      ProductSpecification spec;
      spec = new ProductSpecification( id1, price, "product 1" );
      specifications.put( id1, spec );
      spec = new ProductSpecification( id2, price, "product 2" );
      specifications.put( id2, spec );
   }

   public ProductSpecification getProductSpecification( ItemID id )
```

```
   {
      return specifications.get( id );
   }
}
```

Class Register

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

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

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

   public void enterItem( ItemID id, int quantity )
   {
      ProductSpecification spec = catalog.getProductSpecification( id );

      currentSale.makeLineItem( spec, quantity );
   }

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

   public void makePayment( Money cashTendered )
   {
      currentSale.makePayment( cashTendered );
   }

}
```

Class ProductSpecification

```
public class ProductSpecification
{
   private ItemID id;
   private Money price;
   private String specification;

   public ProductSpecification
      ( ItemID id, Money price, String specification )
   {
      this.id = id;
```

```
      this.price = price;
      this.specification = specification;
   }

   public ItemID getItemID() { return id;   }

   public Money getPrice() { return price; }
   public String getSpecification() { return specification; }
}
```

Class Sale

```
public class Sale
{
   private List<SalesLineItem> lineItems =
                          new ArrayList()<SalesLineItem>;
   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 ) );
   }

   public Money getTotal()
   {
      Money total = new Money();
      Money subtotal = null;

      for ( SalesLineItem lineItem : lineItems )
      {
         subtotal = lineItem.getSubtotal();
         total.add( subtotal );
      }
   return total;
   }

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

Class SalesLineItem

```
public class SalesLineItem
{
   private int    quantity;
   private    ProductSpecification    specification;

   public SalesLineItem (ProductSpecification spec, int quantity )
   {
      this.specification = spec;
      this.quantity = quantity;
   }
   public Money getSubtotal()
   {
      return specification.getPrice().times( quantity );
   }
}
```

Class Store

```
public class Store
{
   private ProductCatalog catalog = new ProductCatalog();
   private Register register = new Register( catalog );

   public Register getRegister() { return register; }
}
```