

Book.java

```
package bookmap;
```

```
import javax.persistence.DiscriminatorColumn;
```

```
import javax.persistence.DiscriminatorType;
```

```
import javax.persistence.DiscriminatorValue;
```

```
import javax.persistence.Entity;
```

```
import javax.persistence.GeneratedValue;
```

```
import javax.persistence.Id;
```

```
import javax.persistence.Inheritance;
```

```
import javax.persistence.InheritanceType;
```

```
/*@Entity
```

```
@Inheritance(strategy=InheritanceType.JOINED)
```

```
@DiscriminatorColumn(name="BOOK_DETAILS", discriminatorType = DiscriminatorType.STRING)
```

```
@DiscriminatorValue("book")*/
```

```
public class Book {
```

```
//@Id
```

```
//@GeneratedValue
```

```
    private int id;
```

```
    private String name;
```

```
    private String author;
```

```
    private double price;
```

```
    private int isbnNo;
```

```
    public Book() {
```

```
        super();
```

```
        // TODO Auto-generated constructor stub
```

```
    }
```

```
    public Book(String name, String author, double price, int isbnNo) {
```

```
        super();
```

```
        this.name = name;
```

```
        this.author = author;
```

```
        this.price = price;
```

```
        this.isbnNo = isbnNo;
```

```
    }
```

```
    public int getId() {
```

```
        return id;
```

```
    }
```

```
    public void setId(int id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getAuthor() {
        return author;
    }

    public void setAuthor(String author) {
        this.author = author;
    }

    public double getPrice() {
        return price;
    }

    public void setPrice(double price) {
        this.price = price;
    }

    public int getIsbnNo() {
        return isbnNo;
    }

    public void setIsbnNo(int isbnNo) {
        this.isbnNo = isbnNo;
    }
}
```

E-Book.java

```
package bookmap;
```

```
import javax.persistence.DiscriminatorValue;
```

```
import javax.persistence.Entity;
```

```
//@Entity
```

```
//@DiscriminatorValue("ebook")
```

```
public class EBook extends Book {
```

```
    public EBook() {  
        super();  
        // TODO Auto-generated constructor stub  
    }
```

```
    private String downloadUrl;  
    private double sizeInMB;
```

```
    public String getDownloadUrl() {  
        return downloadUrl;  
    }
```

```
    public void setDownloadUrl(String downloadUrl) {  
        this.downloadUrl = downloadUrl;  
    }
```

```
    public double getSizeInMB() {  
        return sizeInMB;  
    }
```

```
    public void setSizeInMB(double sizeInMB) {  
        this.sizeInMB = sizeInMB;  
    }
```

```
    public EBook(String name, String author, double price, int isbnNo, String downloadUrl, double  
sizeInMB) {  
        super(name, author, price, isbnNo);  
        this.downloadUrl = downloadUrl;  
        this.sizeInMB = sizeInMB;  
    }  
}
```

```

Paperbook.java
package bookmap;

import javax.persistence.DiscriminatorValue;
import javax.persistence.Entity;

@Entity
@DiscriminatorValue("pbook")
public class PaperBook extends Book {
    private double shippingWeight;
    private int inStock;

    public PaperBook(String name, String author, double price, int isbnNo, double shippingWeight,
int inStock) {
        super(name, author, price, isbnNo);
        this.shippingWeight = shippingWeight;
        this.inStock = inStock;
    }

    public double getShippingWeight() {
        return shippingWeight;
    }

    public void setShippingWeight(double shippingWeight) {
        this.shippingWeight = shippingWeight;
    }

    public int getInStock() {
        return inStock;
    }

    public void setInStock(int inStock) {
        this.inStock = inStock;
    }

    public PaperBook(){
        super();
        // TODO Auto-generated constructor stub
    }
}

```

Bookmain.java

package org.isbn.bk;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import org.grocery.stock.Item;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.cfg.Configuration;

import bookmap.Book;

import bookmap.EBook;

import bookmap.PaperBook;

public class BookMain {

    public static void main(String[] args) throws IOException {

        SessionFactory sf = new Configuration().configure().buildSessionFactory();

        Session session = sf.openSession();

        BufferedReader bf = new BufferedReader(new InputStreamReader(System.in));

        // session.beginTransaction();

        while (true) {

            System.out.println("Enter your choice:");

            int ch = Integer.valueOf(bf.readLine());

            switch (ch) {

                case 1:

                    session.beginTransaction();

                    System.out.print("enter Book Name:");

                    String name = bf.readLine();

                    System.out.print("enter Author Name:");

                    String author = bf.readLine();

                    System.out.print("enter Price:");

                    double price = Double.valueOf(bf.readLine());

```

        System.out.print("enter isbn no:");
        int isbnNo = Integer.valueOf(bf.readLine());
        Book book = new Book(name, author, price, isbnNo);

        book.setName(name);
        book.setAuthor(author);
        book.setPrice(price);
        book.setIsbnNo(isbnNo);

        System.out.println("Enter the details of e-book:");
        System.out.println("enter the url:");
        String url = bf.readLine();
        System.out.println("Enter the size in MB");
        double mb = Double.valueOf(bf.readLine());

        EBook eb = new EBook(name, author, price, isbnNo, url, mb);
        eb.setDownloadUrl(url);
        eb.setSizeInMB(mb);

        System.out.println("Enter the details of paper-book:");
        System.out.println("Enter the shipping weight:");
        double weight = Double.valueOf(bf.readLine());
        System.out.println("Enter the instock:");
        int inStock = Integer.valueOf(bf.readLine());

        PaperBook pb = new PaperBook(name, author, price, isbnNo, weight,
inStock);

        pb.setShippingWeight(weight);
        pb.setInStock(inStock);
        session.save(book);
        session.save(eb);
        session.save(pb);
        // session.getTransaction().commit();

        break;
case 2:
        session.beginTransaction();
        System.out.println("Enter a valid id to update:");
        int vid = Integer.valueOf(bf.readLine());
        Book book1 = new Book();
        book1.setIsbnNo(14526);
        /*

```

```

        * EBook eb1 = new EBook(); eb1.setSizeInMB(50);
        */
        session.update(book1);
        // session.update(eb1);
        // session.getTransaction().commit();

        break;
case 3:
    session.beginTransaction();
    System.out.println("Enter a valid id to delete:");
    int did = Integer.valueOf(bf.readLine());
    Book b = new Book();
    b.setId(did);
    session.delete(b);
    // session.getTransaction().commit();

    break;
case 4:
    session.beginTransaction();
    System.out.println("Enter a valid id to search");
    int id = Integer.valueOf(bf.readLine());
    Book bk = session.get(Book.class, id);
    System.out.println("Book Name=" + bk.getName());
    System.out.println("Author Name=" + bk.getAuthor());
    System.out.println("Price=" + bk.getPrice());
    System.out.println("Isbn No=" + bk.getIsbnNo());
    // session.getTransaction().commit();

    break;
case 5:

    System.out.println("Exit");
    System.exit(0);
    break;
default:
    System.exit(0);
    break;
}

session.getTransaction().commit();

sf.close();

```

```

        session.close();
    }

}
}

```

## Book.hbm.xml

```

<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE hibernate-mapping PUBLIC
    "-//Hibernate/Hibernate Mapping DTD//EN"
    "http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">

<hibernate-mapping>
    <class name="bookmap.Book" table="BOOK_DETAILS" discriminator-value="BOOK">
        <id name="id" type="int" column="BOOK_ID">
            <generator class="native"/>
        </id>
        <discriminator column="DISCRIMINATOR" type="string"/>

        <property name="name" column="BOOK_NAME" type="string"/>
        <property name="author" column="AUTHOR_NAME" type="string"/>
        <property name="price" column="PRICE" type="double"/>
        <property name="isbnNo" column="ISBN_NO" type="int"/>

        <subclass name="bookmap.EBook" extends="Book" discriminator-value="EB">
            <property name="downloadUrl" column="download_URL "
type="string" />
            <property name="sizeInMB" type="double"
column="size_in_MB" />
        </subclass>
        <subclass name="bookmap.PaperBook" extends="Book" discriminator-
value="PB">
            <property name="shippingWeight" column="shipping_weight"
type="double" />
            <property name="inStock" type="int" column="in_stock" />
        </subclass>

    </class>
</hibernate-mapping>

```