Έστω ότι θέλετε να δημιουργήσετε ένα ηλεκτρονικό βιβλιοπωλείο το οποίο πουλάει δύο διαφορετικά είδη προϊόντων, βιβλία και CD. Κάθε ένα από αυτά περιέχει έναν (μη κενό) τίτλο και μια τιμή (μεγαλύτερη του μηδενός). Κάθε βιβλίο περιέχει και επιπλέον πληροφορίες, όπως τον συγγραφέα του βιβλίου, αν περιέχει μαλακό ή σκληρό εξώφυλλο και το έτος δημοσίευσής του. Όσoν αφορά τα CD, προσφέρεται η δυνατότητα της έκπτωσης σε κάποια επιλεγμένα κομμάτια.

Θέλουμε να δημιουργήσουμε μια εφαρμογή που να εμφανίζει στο χρήστη μια λίστα από τα διαθέσιμα βιβλία και CD του ηλεκτρονικού βιβλιοπωλείου, μαζί με τις λεπτομέρειές τους. Επιπλέον, πρέπει να δίνεται η δυνατότητα προσθήκης και διαγραφής των προϊόντων σε ένα καλάθι αγορών. Στη συνέχεια, ο χρήστης της εφαρμογής πρέπει να μπορεί να προβάλει τα προϊόντα τα οποία έχουν προστεθεί στο καλάθι μαζί με τις τιμές τους.

|  |  |
| --- | --- |
| **package** bookstoreAcceptable;  **public** **abstract** **class** Item {    **protected** String title;  **protected** **double** price;  **protected** **int** id;    **public** Item(){title="";price=-1.0;id=-1;}  **public** Item(String aTitle, **double** aPrice,**int** id){  title = aTitle; price=aPrice;**this**.id =id;  }    **public** **abstract** String showDetails();  **public** **abstract** String getDetails();  **public** **abstract** **double** getFinalPrice();  **public** String getTitle(){**return** title;}  **public** **int** getId(){**return** id;}  **public** **double** getOriginalPrice(){**return** price;}  } | **public** Book(String aTitle, String anAuthor, **int** aDate, **double**  aPrice,PackageType aPackageType, **int** id) {    **super**(aTitle, aPrice, id); //constructor of Item!!!  author = anAuthor;  packaging = aPackageType;  yearPublished = aDate;  }  @Override  **public** String showDetails() {  String packString = **this**.packaging.name().toLowerCase() +  " packaging";  String result = "\*\*\*Item id: " + id + "\*\*\*\n" +  title + "\t\t Price:" + price + "\n" +  " by " + author + " at " + yearPublished + "\n" +  "with " + packString + ".\n\n";  **return** result;  }    @Override  **public** String getDetails(){  String packString = **this**.packaging.name().toLowerCase() +  " packaging";  **return** (title + "\nby " + author + " at " + yearPublished  + "\nwith " + packString + "\n");  }  @Override  **public** **double** getFinalPrice() {  **return** price;  }  }//end Book |
| **package** bookstoreAcceptable;  **public** **final** **class** Book **extends** Item {  **public** **enum** PackageType{  ***SIMPLE***, ***HARD***;  }    **private** String author;  **private** PackageType packaging;  **private** **int** yearPublished; |

|  |  |
| --- | --- |
| **package** bookstoreAcceptable;  **public** **final** **class** CD **extends** Item {  **private** String artist;  **private** **double** discount;  **public** CD(String aTitle, **double** aPrice, String anArtist,  **double** aDiscount, **int** id) {  **super**(aTitle, aPrice, id);  artist = anArtist;  discount = aDiscount;  }    @Override  **public** **double** getFinalPrice() {  **return** (price - discount);  }  @Override  **public** String showDetails() {  String result = "\*\*\*Item id: " + id + "\*\*\* \n" +  title + "\t\t Price:" + price + "\n" +  "by " + artist + "\n" +  "final price: " + getFinalPrice()+ "\n\n";  **return** result;  }    @Override  **public** String getDetails(){  **return** (title + "\n by " + artist);  }  } | **package** bookstoreAcceptable;  **import** java.util.ArrayList;  **import** java.util.Iterator;  **import** java.util.List;  **public** **final** **class** ItemManager {  **private** List<Item> allItems;    **public** ItemManager(){  allItems = **new** ArrayList<Item>();  }    **public** **int** addItem(Item anItem){  allItems.add(anItem);  **return** allItems.size();  }    **public** **int** removeItem(**int** index){  **this**.allItems.remove(index);  **return** allItems.size();  }    **public** Item getItem(**int** index){  **return** **this**.allItems.get(index);  }    **public** List<Item> getAllItems(){  **return** **this**.allItems;  }    **public** String reportAllItems(){  String result ="";  **for**(Item item: **this**.allItems) {  result = result + item.showDetails();  }  result = result + "Total number of items: " +  **this**.allItems.size()+ "\n";  System.***out***.println(result);  **return** result;  }  } |
| **package** bookstoreAcceptable;  **import** java.util.ArrayList;  **import** java.util.List;  **public** **final** **class** ShoppingCart {  **private** List<Item> items;  **private** **float** totalCost;    **public** ShoppingCart(){  **this**.items = **new** ArrayList<Item>();  **this**.totalCost = 0;  }    **public** **int** addItem(Item item){  **this**.items.add(item);  **return** **this**.items.size();  }    **public** **int** removeItem(**int** anId){  **int** pos = -1;  **for**(**int** i = 0; i < **this**.items.size(); i++){  **if** (items.get(i).getId() == anId){  pos = i;  **break**;  }  }  **if** (pos>=0)  **this**.items.remove(pos);  **else**{  System.***out***.println("The id you have specified is not in the cart");  }  **return** **this**.items.size();  }    **private** **void** computeTotalCost(){  **float** cost = 0;  **for**(Item item: **this**.items){  cost += item.getFinalPrice();  }  **this**.totalCost = cost;  } | **public** **float** getTotalCost(){  **this**.computeTotalCost();  **return** **this**.totalCost;  }      **public** List<Item> getItems(){  **return** **this**.items;  }    **public** String showDetails(){  String intro = "------------------------\n" +  " CART ITEMS \n" +  "------------------------\n";    String itemsString = "";  **for**(**int** i = 0; i < **this**.items.size(); i++){  //System.out.println("\*\*\*Item id: " + i + "\*\*\*");  itemsString = itemsString + items.get(i).showDetails();  }  String costString = "Total cost: " + **this**.getTotalCost() + "\n";    String result = intro + itemsString + costString;  System.***out***.println(result);    **return** result;  }  } | |

|  |  |  |
| --- | --- | --- |
| package bookstoreacceptable;  import java.util.Scanner;  public class **SimpleBookstoreApplication** {  private Scanner reader;    public SimpleBookstoreApplication(){  reader = new Scanner(System.in);  }    public int printMenu(){  int answerOperation = 0;  while( answerOperation > 5 || answerOperation <= 0){  System.out.println("Choose(1-4)\n 1. Show items\n 2. Add item to cart\n 3. Show cart\n 4. Remove item from cart\n 5. Exit");  answerOperation = reader.nextInt();  if(answerOperation > 5 || answerOperation <= 0)  System.out.println("Wrong answer! Try again...");  }    return answerOperation;  }    public Scanner getReader(){ return reader; }    public static void main(String args[]){  SimpleBookstoreApplication app = new SimpleBookstoreApplication();    ItemManager itemManager = new ItemManager();    Item item = new Book("Discours de la methode","Rene Descartes",  1637, 50.00, 0, 0);  itemManager.addItem(item);  item = new Book("The Meditations", "Marcus Aurelius", 180,30.00,  1,1);  itemManager.addItem(item);  item = new Book("The Bacchae","Euripides", -405,30.00, 1,2);  itemManager.addItem(item);  item = new Book("The Trojan Women","Euripides",-415,40.00, 0,3);  itemManager.addItem(item);  item = new CD("Piece of Mind", 10.0,"Iron Maiden",4.0,4);  itemManager.addItem(item); | | item = new CD("Matter of Life and Death", 12.0,"Iron Maiden",2.0,5);  itemManager.addItem(item);  item = new CD("Perfect Strangers", 12.00, "Deep Purple",1.0,6);  itemManager.addItem(item);    ShoppingCart cart = new ShoppingCart();    while(true){  int operation = app.printMenu();  if(operation == 1){  itemManager.reportAllItems();  }  else if(operation == 2){  System.out.println("Choose the id of the item that you"  + " want to add to your cart");  int id = app.getReader().nextInt();  System.out.println(id);  if(id > itemManager.getAllItems().size())  System.out.println("Error: there is no product"  + " with the specified id");  else  cart.addItem(itemManager.getItem(id));  }  else if(operation == 3){  cart.showDetails();  }  else if(operation == 4){  System.out.println("Choose the id of the item that you"  + " want to remove from your cart");  int id = app.getReader().nextInt();  System.out.println(id);  if(id > itemManager.getAllItems().size())  System.out.println("Error: there is no product"  + " with the specified id");  else  cart.removeItem(id);  }  else{  break;  }  }    } //endofmain()  } |
| **Output**  Choose(1-4)  1. Show items  2. Add item to cart  3. Show cart  4. Remove item from cart  5. Exit  **1**  \*\*\*Item id: 0\*\*\*  Discours de la methode Price:50.0  by Rene Descartes at 1637  with simple packaging.  \*\*\*Item id: 1\*\*\*  The Meditations Price:30.0  by Marcus Aurelius at 180  with hard box packaging.  \*\*\*Item id: 2\*\*\*  The Bacchae Price:30.0  by Euripides at -405  with hard box packaging.  \*\*\*Item id: 3\*\*\*  The Trojan Women Price:40.0  by Euripides at -415  with simple packaging.  \*\*\*Item id: 4\*\*\*  Piece of Mind Price:10.0  by Iron Maiden  final price: 6.0  \*\*\*Item id: 5\*\*\*  Matter of Life and Death Price:12.0  by Iron Maiden  final price: 10.0  \*\*\*Item id: 6\*\*\*  Perfect Strangers Price:12.0  by Deep Purple  final price: 11.0 | Choose(1-4)  1. Show items  2. Add item to cart  3. Show cart  4. Remove item from cart  5. Exit  **2**  Choose the id of the item that you want to add to your cart  **2**  Choose(1-4)  1. Show items  2. Add item to cart  3. Show cart  4. Remove item from cart  5. Exit  **3**  ------------------------  CART ITEMS  ------------------------  \*\*\*Item id: 2\*\*\*  The Bacchae Price:30.0  by Euripides at -405  with hard box packaging.  Total cost: 30.0 | |