/\*\*

\* Stores books, cds and movies

\*/

import java.util.ArrayList;

public class Warehouse

{

private ArrayList<Book> books = new ArrayList<Book>();

private ArrayList<CD> cds = new ArrayList<CD>();

private ArrayList<Movie> movies = new ArrayList<Movie>();

/\*\*

\* Constructor for objects of class Warehouse

\*/

public Warehouse()

{

Book book1 = new Book("James Greene", "The fault in our stars", 15, 10);

addBook(book1);

Book book2 = new Book("Edna O'Brien", "A pagan place", 23, 10);

addBook(book2);

Book book3 = new Book("Steven Pinker", "Sense of style", 17, 10);

addBook(book3);

CD cd1 = new CD("Darude", "Sandstorm", 20, 10, "mp3");

addCD(cd1);

CD cd2 = new CD("Scandal", "Harukaze", 23, 10, "mp3");

addCD(cd2);

CD cd3 = new CD("One Ok Rock", "Zankyo Reference", 30, 10, "mp3");

addCD(cd3);

Movie mv1 = new Movie("Joss Whedon", "Kevin Feige", "Avengers Assemble", 30, 10, "avi");

addMovie(mv1);

Movie mv2 = new Movie("Hayao Miyazaki", "Toru Hara", "My Neighbor Totoro", 50, 10, "avi");

addMovie(mv2);

Movie mv3 = new Movie("Isao Takahata", "Toru Hara", "Grave of The Fireflies", 45, 10, "avi");

addMovie(mv3);

}

public void addBook(Book pbook)

{

books.add(pbook);

}

public void addCD(CD pcd)

{

cds.add(pcd);

}

public void addMovie(Movie pmovie)

{

movies.add(pmovie);

}

public final ArrayList<Book> getBooks()

{

return books;

}

public final ArrayList<CD> getCDs()

{

return cds;

}

public final ArrayList<Movie> getMovies()

{

return movies;

}

}

/\*\*

\* Gets books/cds/movies from the warehouse and sells them

\*/

import java.util.List;

import java.util.ArrayList;

public class Shop

{

// instance variables - replace the example below with your own

private List<Book> books;

private List<CD> cds;

private List<Movie> movies;

Warehouse warehouse = null;

private int sellableBooks = 0;

private int sellableCDs = 0;

private int sellableMovies = 0;

/\*\*

\* Constructor for objects of class Shop

\*/

public Shop()

{

warehouse = new Warehouse();

books = warehouse.getBooks();

cds = warehouse.getCDs();

movies = warehouse.getMovies();

executeSellable();

}

/\*\*

\* calculates total amount of items that are available

\*/

public void executeSellable()

{

for(int i = 0; i < books.size(); ++i){

if(books.get(i).getAvailable() > 0){

++sellableBooks;

}

}

for(int i = 0; i < movies.size(); ++i){

if(movies.get(i).getAvailable() > 0){

++sellableMovies;

}

}

for(int i = 0; i < cds.size(); ++i){

if(cds.get(i).getAvailable() > 0){

++sellableCDs;

}

}

}

/\*\*

\* @return (how many books in the warehouse)

\*/

public final int getSellableBooks()

{

return sellableBooks;

}

/\*\*

\* @return (how many cds in the warehouse)

\*/

public final int getSellableCDs()

{

return sellableCDs;

}

/\*\*

\* @return (how many movies in the warehouse)

\*/

public final int getSellableMovies()

{

return sellableMovies;

}

public final List<Book> getBooks()

{

return books;

}

public final List<CD> getCDs()

{

return cds;

}

public final List<Movie> getMovies()

{

return movies;

}

/\*\*

\* shows what's for sale

\* @param (Iselection)

\*/

public String displaySalesItems(int pselection)

{

String text = "";

int j = 1;

// we are looking at books.

// Yes, the user wants to buy books.

if(pselection == 1){

System.out.println("Type the number to choose a book\n");

books = warehouse.getBooks();

for(int i = 0; i < books.size(); ++i){

if(books.get(i).getAvailable() > 0){

books.get(i).setID(j++);

text += books.get(i).getID() + " " + books.get(i).toString();

}

}

return text;

}

if(pselection == 2){

System.out.println("Type the number to choose a CD\n");

cds = warehouse.getCDs();

for(int i = 0; i < cds.size(); ++i){

if(cds.get(i).getAvailable() > 0){

cds.get(i).setID(j++);

text += cds.get(i).getID() + " " + cds.get(i).toString();

}

}

return text;

}

if(pselection == 3){

System.out.println("Type the number to choose a Movie\n");

movies = warehouse.getMovies();

for(int i = 0; i < movies.size(); ++i){

if(movies.get(i).getAvailable() > 0){

movies.get(i).setID(j++);

text += movies.get(i).getID() + " " + movies.get(i).toString();

}

}

return text;

}

return text;

}

/\*\*

\* sell a book, returns a receipt

\* @param (Iselection)

\* @return (Dprice)

\*/

public double sellBook(int pselection)

{

for(int i = 0; i < books.size(); ++i){ //loop through the entire list

if(pselection == books.get(i).getID()){ //and now we've found the specified item

books.get(i).sellCopies(1);

return books.get(i).getPrice();

}

}

return 0;

}

/\*\*

\* rates the book

\* @param (Iselection, Irating)

\*/

public void rateBook(int pselection, int prating)

{

for(int i = 0; i < books.size(); ++i){ //loop through the entire list

if(pselection == books.get(i).getID()){ //and now we've found the specified item

books.get(i).updateRating(prating);

}

}

}

/\*\*

\* sells a cd, returns a receipt

\* @param (Iselection)

\* @return (Dprice)

\*/

public double sellCD(int pselection)

{

for(int i = 0; i < cds.size(); ++i){ //loop through the entire list

if(pselection == cds.get(i).getID()){ //and now we've found the specified item

cds.get(i).sellCopies(1);

return cds.get(i).getPrice();

}

}

return 0;

}

/\*\*

\* @param (Iselection, Irating)

\*/

public void rateCD(int pselection, int prating)

{

for(int i = 0; i < cds.size(); ++i){ //loop through the entire list

if(pselection == cds.get(i).getID()){ //and now we've found the specified item

cds.get(i).updateRating(prating);

}

}

}

/\*\*

\* sells a movie, returns a receipt

\* @param (Iselection)

\* @return (Dprice)

\*/

public double sellMovie(int pselection)

{

for(int i = 0; i < movies.size(); ++i){ //loop through the entire list

if(pselection == movies.get(i).getID()){ //and now we've found the specified item

movies.get(i).sellCopies(1);

return movies.get(i).getPrice();

}

}

return 0;

}

/\*\*

\* @param (Iselection, Irating)

\*/

public void rateMovie(int pselection, int prating)

{

for(int i = 0; i < movies.size(); ++i){ //loop through the entire list

if(pselection == movies.get(i).getID()){ //and now we've found the specified item

movies.get(i).updateRating(prating);

}

}

}

/\*\*

\* displays a string which says ‘Sold x books, y CDs and z movies, total revenue $xxx’.

\*/

public String show()

{

String text = "Sold ";

int bookSold = 0, cdSold = 0, movieSold = 0;

double revenue = 0;

for(int i = 0; i < books.size(); ++i){ //get total of books sold

bookSold += books.get(i).getSold();

//find the price, multiply with the amount sold

revenue += books.get(i).getPrice() \* books.get(i).getSold();

}

for(int i = 0; i < cds.size(); ++i){ //get total cds sold

cdSold += cds.get(i).getSold();

revenue += cds.get(i).getPrice() \* cds.get(i).getSold();

}

for(int i = 0; i < movies.size(); ++i){ //get total movies sold

movieSold += movies.get(i).getSold();

revenue += cds.get(i).getPrice() \* cds.get(i).getSold();

}

text += bookSold + " books, " + cdSold + " CDs, and " + movieSold + " movies, total revenue $" + revenue + ".";

return text;

}

/\*\*

\* A test harness, demonstrates how the real show would work when storing to files is enabled.

\* displays a string which says ‘Sold x books, y CDs and z movies, total revenue $xxx’.

\*/

public String testShow()

{

String text = "Sold ";

int bookSold = 0, cdSold = 0, movieSold = 0;

double revenue = 0;

//feed application with dummy data

for(int i = 0; i < books.size(); ++i){

books.get(i).sellCopies(10);

}

for(int i = 0; i < cds.size(); ++i){

cds.get(i).sellCopies(10);

}

for(int i = 0; i < movies.size(); ++i){

movies.get(i).sellCopies(10);

}

for(int i = 0; i < books.size(); ++i){ //get total of books sold

bookSold += books.get(i).getSold();

//find the price, multiply with the amount sold

revenue += books.get(i).getPrice() \* books.get(i).getSold();

}

for(int i = 0; i < cds.size(); ++i){ //get total cds sold

cdSold += cds.get(i).getSold();

revenue += cds.get(i).getPrice() \* cds.get(i).getSold();

}

for(int i = 0; i < movies.size(); ++i){ //get total movies sold

movieSold += movies.get(i).getSold();

revenue += cds.get(i).getPrice() \* cds.get(i).getSold();

}

text += bookSold + " books, " + cdSold + " CDs, and " + movieSold + " movies, total revenue $" + revenue + ".";

return text;

}

/\*\*

\* test harness to test the pop\_front rating function

\* and also to get the average for a book

\*/

public String testRating()

{

String text = "1st 20: \n";

//feed application with dummy data

for(int i = 0; i < 100; ++i){

books.get(0).updateRating(10);

}

for(int i = 0; i < 20; ++i){

text += i+1 + ": " + books.get(0).getRating().getScoreByIndex(i) + "\n";

}

text += "\npushing rating #101\n";

text += "\nLast 20: \n" ;

//push another in, remove front, add to back

books.get(0).updateRating(5);

for(int i = 80; i < 100; ++i){

text += i+1 + ": " + books.get(0).getRating().getScoreByIndex(i) + "\n";

}

text += "\nThe average rating for: " + books.get(0).toString() + " is: " + books.get(0).getAverageRating();

return text;

}

}

/\*\*

\* Representation of a book

\*/

public class Book

{

// instance variables - replace the example below with your own

private String author = null;

private String title = null;

private int id = 0;

private int available = 0;

private double price = 0.0;

private Rating rating = new Rating();

private int sold = 0;

/\*\*

\* @param (author, title, price, available)

\*/

public Book(String pauthor, String ptitle, double pprice, int pavailable)

{

// initialise instance variables

author = pauthor;

title = ptitle;

price = pprice;

available = pavailable;

}

public void setID(int pid)

{

id = pid;

}

/\*\*

\* @return (id)

\*/

public final int getID()

{

return id;

}

/\*\*

\* @param (# of copies available)

\*/

public void setAvailable(int pavailable)

{

available = pavailable;

}

/\*\*

\* @return (available)

\*/

public final int getAvailable()

{

return available;

}

/\*\*

\* @param (the price as a double)

\*/

public void setPrice(double pprice)

{

price = pprice;

}

/\*\*

\* @return (Dprice)

\*/

public double getPrice()

{

return price;

}

/\*\*

\* @param (Isold)

\*/

public void setSold(int psold)

{

sold = psold;

}

/\*\*

\* @return (# sold)

\*/

public final int getSold()

{

return sold;

}

/\*\*

\* Sells the copies, updates sold

\* @return # of successful transactions

\* for now, only sell one copy per transaction

\*/

public int sellCopies(int howMany)

{

if(available > 0 && available >= howMany){

sold += howMany;

available -= howMany;

}

return available;

}

/\*\*

\* updates the rating

\*/

public void updateRating(int newRating)

{

rating.updateRating(newRating);

}

/\*\*

\* @return (rating)

\*/

public final Rating getRating()

{

return rating;

}

/\*\*

\* @return (IaverageRating)

\*/

public int getAverageRating()

{

return rating.getAverage();

}

/\*\*

\* @return (string representation)

\*/

public final String toString()

{

String text = "";

text = "'" + title + "' by " + author + ", $" + price + "\n";

return text;

}

}

/\*\*

\* Representation of a CD

\*/

public class CD

{

// instance variables - replace the example below with your own

private String performer = null;

private String title = null;

private int id = 0;

private int available = 0; //how many copies available

private double price = 0.0;

private Rating rating = new Rating();

private String fileType = null;

private int sold = 0; //amount sold

/\*\*

\* @param (Sperformer, Stitle, Dprice, Iavailability, Sfiletype)

\*/

public CD(String pperformer, String ptitle, double pprice, int pavailable, String pfileType)

{

performer = pperformer;

title = ptitle;

price = pprice;

available = pavailable;

fileType = pfileType;

}

public void setID(int pid)

{

id = pid;

}

public final int getID()

{

return id;

}

/\*\*

\* @param (Dprice)

\*/

public void setPrice(double pprice)

{

price = pprice;

}

/\*\*

\* @return (Dprice)

\*/

public double getPrice()

{

return price;

}

/\*\*

\* Sells the copies, updates sold

\* @return # of successful transactions

\* for now, only sell one copy per transaction

\*/

public int sellCopies(int howMany)

{

if(available > 0 && available >= howMany){

sold += howMany;

available -= howMany;

}

return available;

}

/\*\*

\* updates the rating

\*/

public void updateRating(int newRating)

{

rating.updateRating(newRating);

}

/\*\*

\* @return (rating)

\*/

public final Rating getRating()

{

return rating;

}

/\*\*

\* @param (Iavailable)

\*/

public void setAvailable(int pavailable)

{

available = pavailable;

}

public int getAvailable()

{

return available;

}

/\*\*

\* @param (Isold)

\*/

public void setSold(int psold)

{

sold = psold;

}

public int getSold()

{

return sold;

}

/\*\*

\* @return string representaion

\*/

public final String toString()

{

String text = "";

text += "'" + title + "' by " + performer + ", $" + price + "\n";

return text;

}

}

/\*\*

\* Representation of a movie

\*/

public class Movie

{

// instance variables - replace the example below with your own\

private String director = null;

private String producer = null;

private String title = null;

private int id = 0;

private int available = 0;

private double price = 0.0;

private Rating rating = new Rating();

private String fileType = null;

private int sold = 0;

/\*\*

\* @param (Sdirector, Sproducer, Stitle, Dprice, Iavailable, SfileType)

\*/

public Movie(String pdirector, String pproducer, String ptitle, double pprice,int pavailable, String pfileType)

{

director = pdirector;

producer = pproducer;

title = ptitle;

price = pprice;

available = pavailable;

fileType = pfileType;

}

public void setID(int pid)

{

id = pid;

}

public final int getID()

{

return id;

}

/\*\*

\* @param (Dprice)

\*/

public void setPrice(double pprice)

{

price = pprice;

}

/\*\*

\* @return (Dprice)

\*/

public double getPrice()

{

return price;

}

/\*\*

\* Sells the copies, updates sold

\* @return # of successful transactions

\* for now, only sell one copy per transaction

\*/

public int sellCopies(int howMany)

{

if(available > 0 && available >= howMany){

sold += howMany;

available -= howMany;

}

return available;

}

/\*\*

\* updates the rating

\*/

public void updateRating(int newRating)

{

rating.updateRating(newRating);

}

/\*\*

\* @return (rating)

\*/

public final Rating getRating()

{

return rating;

}

/\*\*

\* @param (Iavailable)

\*/

public void setAvailable(int pavailable)

{

available = pavailable;

}

public final int getAvailable()

{

return available;

}

/\*\*

\* @param (Isold)

\*/

public void setSold(int psold)

{

sold = psold;

}

public final int getSold()

{

return sold;

}

/\*\*

\* Returns results in String

\*/

public final String toString()

{

String text = "";

text += "'" + title + "' directed by " + director + " produced by " + producer+ " $" + price + "\n";

return text;

}

}

/\*\*

\* Representation of a rating for a Book/CD/Movie

\*/

public class Rating

{

// instance variables - replace the example below with your own

private int[] scores = new int[100];

int numberOfRatings = 0;

/\*\*

\* Constructor for objects of class Rating

\*/

public Rating()

{

// initialise instance variables

}

public void updateRating(int prating)

{

if(numberOfRatings < 100){

scores[numberOfRatings] = prating;

++numberOfRatings;

return;

}

//simulate a pop\_front, but without the resizing of the array capacity

for(int i = 0; i < 99; ++i){

scores[i] = scores[i + 1]; //Oh boy, don't go out of bounds here

}

//don't forget to update the last value.

scores[99] = prating;

}

/\*\*

\* @return (the scores array)

\*/

public final int[] getScores()

{

return scores;

}

/\*\*

\* @return (a score by index

\*/

public final int getScoreByIndex(int pindex)

{

if(pindex < 100){

return scores[pindex];

}

return -1;

}

/\*\*

\* @return (Iaverage)

\*/

public final int getAverage()

{

int average = 0;

for(int i = 0; i < scores.length; ++i){

average += scores[i];

}

return (average / scores.length);

}

}

/\*\*

\* The Main class that starts up the program and maintains states for all the

\* other class Objects.

\* This class handles user inputs, then sends it down to the appropriate

\* classes.

\*/

import java.util.Scanner;

import java.util.InputMismatchException;

public class Main

{

public static int selection = 0; //to store user input

public static Shop shop = null; //a Shop

public static Scanner sc = null; //to engage System.in

public static void main(String[] args)

{

sc = new Scanner(System.in);

shop = new Shop();

//Item type selection sequence

if(args[0].equals("sell")){

String text = "Welcome to the Media Shop\n\n\nChoose from the following:\n1. Books (enter '1')\n2. CDs (enter '2')\n3. Movies (enter '3')";

System.out.println(text + "\nChoice: ");

while(true){ //keep looping until user enters the correct selection

try{

selection = sc.nextInt(); //choose category

if(selection > 3 || selection < 1){

throw new InputMismatchException();

}

break;

}

catch(InputMismatchException e){

System.out.println("Input out of range or invalid." + "\nChoice: ");

continue;

}

}

if(selection == 1){ //Books

sellBook();

}

else if(selection == 2){ //CDs

sellCD();

}

else if(selection == 3){

sellMovie();

}

else{

System.out.println("Invalid Input");

}

}

else if(args[0].equals("show")){

System.out.println(shop.show());

}

else if(args[0].equals("testShow")){

System.out.println(shop.testShow());

}

else if(args[0].equals("testRating")){

System.out.println(shop.testRating());

}

else{

System.out.println("Invalid Input");

}

}

/\*\*

\* sells a book and asks for rating

\*/

public static void sellBook()

{

System.out.println(shop.displaySalesItems(selection) + "\nChoice: ");

while(true){ //keep looping until user enters the correct input

try{

selection = sc.nextInt(); //choose item

if(selection > shop.getSellableBooks() || selection < 1){

throw new InputMismatchException();

}

break;

}

catch(InputMismatchException e){

System.out.println("Input out of range or invalid." + "\nChoice: ");

continue;

}

}

double lprice = shop.sellBook(selection);

//sold item, now ask for rating

System.out.println("You have been charged: " + lprice + "\n\n");

System.out.println("Rate the book you bought on a scale of 1 to 10.\n 10 is excellent, 1 is absolutely uninteresting.");

int lrate = 0;

while(true){

try{

lrate = sc.nextInt();

if(lrate < 1 || lrate > 10){

throw new InputMismatchException();

}

break;

}

catch(InputMismatchException e){ //Movies

System.out.println("Invalid Input" + "\nChoice: ");

continue;

}

}

shop.rateBook(selection, lrate);

System.out.println("Your rating has been saved.Thank you for your feedback.\nGood bye.");

}

/\*\*

\* sells a movie and asks for rating

\*/

public static void sellMovie()

{

System.out.println(shop.displaySalesItems(selection) + "\nChoice: ");

while(true){ //keep looping until user enters the correct input

try{

selection = sc.nextInt(); //choose item

if(selection > shop.getSellableMovies() || selection < 1){

throw new InputMismatchException();

}

break;

}

catch(InputMismatchException e){

System.out.println("Input out of range or invalid." + "\nChoice: ");

continue;

}

}

double lprice = shop.sellMovie(selection);

//sold item, now ask for rating

System.out.println("You have been charged: " + lprice + "\n\n");

System.out.println("Rate the movie you bought on a scale of 1 to 10.\n 10 is excellent, 1 is absolutely uninteresting.");

int lrate = 0;

while(true){

try{

lrate = sc.nextInt();

if(lrate < 1 || lrate > 10){

throw new InputMismatchException();

}

break;

}

catch(InputMismatchException e){ //Movies

System.out.println("Invalid Input" + "\nChoice: ");

continue;

}

}

shop.rateMovie(selection, lrate);

System.out.println("Your rating has been saved.Thank you for your feedback.\nGood bye.");

}

/\*\*

\* sells a cd and asks for rating

\*/

public static void sellCD()

{

System.out.println(shop.displaySalesItems(selection) + "\nChoice: ");

while(true){ //keep looping until user enters the correct input

try{

selection = sc.nextInt(); //choose item

if(selection > shop.getSellableCDs() || selection < 1){

throw new InputMismatchException();

}

break;

}

catch(InputMismatchException e){

System.out.println("Input out of range or invalid." + "\nChoice: ");

continue;

}

}

double lprice = shop.sellCD(selection);

//sold item, now ask for rating

System.out.println("You have been charged: " + lprice + "\n\n");

System.out.println("Rate the CD you bought on a scale of 1 to 10.\n 10 is excellent, 1 is absolutely uninteresting.");

int lrate = 0;

while(true){

try{

lrate = sc.nextInt();

if(lrate < 1 || lrate > 10){

throw new InputMismatchException();

}

break;

}

catch(InputMismatchException e){ //Movies

System.out.println("Invalid Input" + "\nChoice: ");

continue;

}

}

shop.rateCD(selection, lrate);

System.out.println("Your rating has been saved.Thank you for your feedback.\nGood bye.");

}

}