AbstractRentalItem.java

package spring\_2020;

/\*\*

\* **@author** Owen A. Riegle

\*

\*/

public abstract class AbstractRentalItem {

String title;

int yearOfRelease;

int slot;

int quantity;

/\*\*

\* **@param** title name of the rental item

\* **@param** yearOfRelease year rental item was released

\* **@param** slot position within rental machine

\* **@param** quantity number of rental item in rental machine

\*/

public AbstractRentalItem(String title, int yearOfRelease, int slot, int quantity) {

this.title = title;

this.yearOfRelease = yearOfRelease;

this.slot = slot;

this.quantity = quantity;

}

/\*\*

\* **@return** the title

\*/

public String getTitle() {

return title;

}

/\*\*

\* **@param** title the title to set

\*/

public void setTitle(String title) {

this.title = title;

}

/\*\*

\* **@return** the yearOfRelease

\*/

public int getYearOfRelease() {

return yearOfRelease;

}

/\*\*

\* **@param** yearOfRelease the yearOfRelease to set

\*/

public void setYearOfRelease(int yearOfRelease) {

this.yearOfRelease = yearOfRelease;

}

/\*\*

\* **@return** the slot

\*/

public int getSlot() {

return slot;

}

/\*\*

\* **@param** slot the slot to set

\*/

public void setSlot(int slot) {

this.slot = slot;

}

/\*\*

\* **@return** the quantity

\*/

public int getQuantity() {

return quantity;

}

/\*\*

\* **@param** quantity the quantity to set

\*/

public void setQuantity(int quantity) {

this.quantity = quantity;

}

/\*\*

\* **@return** values of data members

\*/

*@Override*

public String toString() {

return "AbstractRentalItem [title=" + title + ", yearOfRelease=" + yearOfRelease + ", slot=" + slot

+ ", quantity=" + quantity + "]";

}

*@Override*

public boolean equals(Object obj) {

if (this == obj)

return true;

if (obj == null)

return false;

if (getClass() != obj.getClass())

return false;

AbstractRentalItem other = (AbstractRentalItem) obj;

if (quantity != other.quantity)

return false;

if (slot != other.slot)

return false;

if (title == null) {

if (other.title != null)

return false;

} else if (!title.equals(other.title))

return false;

if (yearOfRelease != other.yearOfRelease)

return false;

return true;

}

/\*\*

\* **@param** rentalFee

\* **@return** daily rental fee

\*/

public abstract double dailyRentalFee(double rentalFee);

/\*\*

\* **@throws** IllegalStateException if quantity of item retrieved is less than or

\* equal to zero

\* **@param** quantity subtracts one from quantity for item retrieved

\*/

public int itemQuantity() {

if (quantity <= 0)

throw new IllegalStateException();

else

quantity -= 1;

return quantity;

Movie.java

package spring\_2020;

/\*\*

\* **@author** Owen A. Riegle

\*

\*/

public class Movie extends AbstractRentalItem {

public String diskType;

/\*\*

\* **@param** title

\* **@param** yearOfRelease

\* **@param** slot

\* **@param** quantity

\* **@param** diskType

\*/

public Movie(String title, int yearOfRelease, int slot, int quantity, String diskType) {

super(title, yearOfRelease, slot, quantity);

this.diskType = diskType;

}

/\*\*

\* **@return** the diskType

\*/

public String getDiskType() {

return diskType;

}

/\*\*

\* **@param** diskType the diskType to set

\*/

public void setDiskType(String diskType) {

this.diskType = diskType;

}

/\*\*

\* **@return** values of data members

\*/

*@Override*

public String toString() {

return quantity + " copies of " + title;

}

*@Override*

public boolean equals(Object obj) {

if (this == obj)

return true;

if (!super.equals(obj))

return false;

if (getClass() != obj.getClass())

return false;

Movie other = (Movie) obj;

if (diskType == null) {

if (other.diskType != null)

return false;

} else if (!diskType.equals(other.diskType))

return false;

return true;

}

*@Override*

public double dailyRentalFee(double rentalFee) {

if (diskType.equalsIgnoreCase("DVD"))

return 1.80;

if (diskType.equalsIgnoreCase("BluRay"))

return 2.00;

else

return 0;

VideoGame.java

package spring\_2020;

/\*\*

\* **@author** Owen A. Riegle

\*

\*/

public class VideoGame extends AbstractRentalItem {

public String console;

/\*\*

\* **@param** title

\* **@param** yearOfRelease

\* **@param** slot

\* **@param** quantity

\* **@param** console

\*/

public VideoGame(String title, int yearOfRelease, int slot, int quantity, String console) {

super(title, yearOfRelease, slot, quantity);

this.console = console;

}

/\*\*

\* **@return** the console

\*/

public String getConsole() {

return console;

}

/\*\*

\* **@param** console the console to set

\*/

public void setConsole(String console) {

this.console = console;

}

/\*\*

\* **@return** values of data members

\*/

*@Override*

public String toString() {

return quantity + " copies of " + title;

}

*@Override*

public boolean equals(Object obj) {

if (this == obj)

return true;

if (!super.equals(obj))

return false;

if (getClass() != obj.getClass())

return false;

VideoGame other = (VideoGame) obj;

if (console == null) {

if (other.console != null)

return false;

} else if (!console.equals(other.console))

return false;

return true;

}

*@Override*

public double dailyRentalFee(double rentalFee) {

if (console.equalsIgnoreCase("XBOX"))

return 3.00;

if (console.equalsIgnoreCase("PS4"))

return 2.50;

else

return 0;

ItemRental.java

package spring\_2020;

import java.util.Scanner;

/\*\*

\* **@author** Owen A. Riegle

\*

\*/

public class ItemRental {

/\*\*

\* **@param** args

\*/

public static void main(String[] args) {

AbstractRentalItem[] itemList = new AbstractRentalItem[9];

Scanner keyboard = new Scanner(System.***in***);

int slot = 0;

int i = 0;

itemList[0] = (new VideoGame("Anthem", 2019, 1, 2, "XBOX"));

itemList[1] = (new VideoGame("Madden 20", 2020, 2, 6, "PS4"));

itemList[2] = (new VideoGame("Jumanji", 2019, 3, 9, "PS4"));

itemList[3] = (new VideoGame("Gears 5", 2019, 4, 2, "XBOX"));

itemList[4] = (new Movie("Joker", 2019, 5, 0, "BluRay"));

itemList[5] = (new Movie("Hustlers", 2019, 6, 1, "DVD"));

itemList[6] = (new Movie("Oceans 8", 2019, 7, 6, "DVD"));

itemList[7] = (new Movie("Venom", 2019, 8, 8, "BluRay"));

itemList[8] = (new Movie("Tag", 2019, 9, 3, "BluRay"));

System.***out***.println("Enter the slot number of an item to rent it. Enter 0 stop renting.");

slot = keyboard.nextInt();

while (slot > 0) {

try {

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

if (itemList[i].getSlot() == slot) {

if (itemList[i].itemQuantity() >= 0) {

System.***out***.println("Item " + itemList[i].getTitle() + " rented with a daily fee of $"

+ String.*format*("%.2f", itemList[i].dailyRentalFee(i)));

}

}

if (slot > 9 ) {

System.***out***.println("Slot " + slot + " does not exist");

System.***out***.println("Enter the slot number of an item to rent it. Enter 0 stop renting.");

slot = keyboard.nextInt();

}

}

System.***out***.println("Enter the slot number of an item to rent it. Enter 0 stop renting.");

slot = keyboard.nextInt();

} catch (IllegalStateException e) {

System.***out***.println(itemList[i].getTitle() + " is out of stock");

System.***out***.println("Enter the slot number of an item to rent it. Enter 0 stop renting.");

slot = keyboard.nextInt();

}

}

for (i = 0; i < itemList.length; i++)

System.***out***.println(itemList[i]);

keyboard.close();

}

}

}

Output

Enter the slot number of an item to rent it. Enter 0 stop renting.

1

Item Anthem rented with a daily fee of $3.00

Enter the slot number of an item to rent it. Enter 0 stop renting.

2

Item Madden 20 rented with a daily fee of $2.50

Enter the slot number of an item to rent it. Enter 0 stop renting.

3

Item Jumanji rented with a daily fee of $2.50

Enter the slot number of an item to rent it. Enter 0 stop renting.

4

Item Gears 5 rented with a daily fee of $3.00

Enter the slot number of an item to rent it. Enter 0 stop renting.

5

Joker is out of stock

Enter the slot number of an item to rent it. Enter 0 stop renting.

6

Item Hustlers rented with a daily fee of $1.80

Enter the slot number of an item to rent it. Enter 0 stop renting.

7

Item Oceans 8 rented with a daily fee of $1.80

Enter the slot number of an item to rent it. Enter 0 stop renting.

8

Item Venom rented with a daily fee of $2.00

Enter the slot number of an item to rent it. Enter 0 stop renting.

9

Item Tag rented with a daily fee of $2.00

Enter the slot number of an item to rent it. Enter 0 stop renting.

10

Slot 10 does not exist

Enter the slot number of an item to rent it. Enter 0 stop renting.

0

1 copies of Anthem

5 copies of Madden 20

8 copies of Jumanji

1 copies of Gears 5

0 copies of Joker

0 copies of Hustlers

5 copies of Oceans 8

7 copies of Venom

2 copies of Tag