南开大学

JAVA语言与应用

控制台版CD出租销售店实验报告

姓 名：冯朝芃

学 号：2012039

年 级： 2020 级

学 院： 计算机学院

专 业 ：计算机科学与技术

授课教师：刘嘉欣

完成日期：2021年 11月 7日

一、概述：

本作业为控制台版Java语言CD出租销售店。本作业实现的功能有：输入用户、删除用户、进货、汇报存货、出租、销售、帮助等功能。本代码使用了MVC架构进行开发。本程序充分利用Java的面向对象编程思想，一定程度上具有可扩展性高，代码逻辑框架清晰。代码复用性强、事件处理流程明确等特点。

二、运行展示：

运行效果截图：

帮助：

电脑萤幕的截图

描述已自动生成

添加用户：

电脑萤幕的截图

描述已自动生成

删除用户：

电脑萤幕的截图

描述已自动生成

添加CD和汇报

电脑萤幕的截图

描述已自动生成

出借

电脑萤幕的截图

描述已自动生成

运行结果（购买）

Welcome CDStore! Enter h for Help!

au

ADDING USERS

Enter User's Name:

Wang

Enter User's Id:

1

Enter User's Money:

15

Finish

Do you wanna add again?[y/n]

y

ADDING USERS

Enter User's Name:

Feng

Enter User's Id:

2

Enter User's Money:

100

Finish

Do you wanna add again?[y/n]

n

ac

ADDING DISKS

Enter Disk's Name:

Matrix

Enter Disk's Quantity:

10

Enter Disk's Money:

20

Finish

Do you wanna add again?[y/n]

y

ADDING DISKS

Enter Disk's Name:

Tomorrow

Enter Disk's Quantity:

10

Enter Disk's Money:

5

Finish

Do you wanna add again?[y/n]

n

buy

BUY

Enter User's Name:

Wang

Enter User's Id:

1

Enter the Disk to Buy:

Matrix

Enter the Number of Disks:

100

No Money or No Enough Quantity!

Finish

Do you wanna add again?[y/n]

y

BUY

Enter User's Name:

Feng

Enter User's Id:

2

Enter the Disk to Buy:

Matrix

Enter the Number of Disks:

5

Finish

Do you wanna add again?[y/n]

n

ru

[User [name=Wang, id=1, borrowed=[], money=15.0], User [name=Feng, id=2, borrowed=[], money=0.0]]

rd

[Disk [name=Matrix, money=20.0, num=5], Disk [name=Tomorrow, money=5.0, num=10]]

附录：完整代码

＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝

『Disk.java』

＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝

package Bean;

import java.util.Objects;

public class Disk {

private String name;

private double money;

private int num;

public String getName() {

return name;

}

public double getMoney() {

return money;

}

public int getNum() {

return num;

}

public void setName(String name) {

this.name = name;

}

public void setMoney(double money) {

this.money = money;

}

public void setNum(int num) {

this.num = num;

}

@Override

public String toString() {

return "Disk [name=" + name + ", money=" + money + ", num=" + num + "]";

}

@Override

public int hashCode() {

final int prime = 31;

int result = 1;

long temp;

temp = Double.doubleToLongBits(money);

result = prime \* result + (int) (temp ^ (temp >>> 32));

result = prime \* result + ((name == null) ? 0 : name.hashCode());

result = prime \* result + num;

return result;

}

@Override

public boolean equals(Object obj) {

if (this == obj)

return true;

if (obj == null)

return false;

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

return false;

Disk other = (Disk) obj;

if (name == null) {

if (other.name != null)

return false;

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

return false;

return true;

}

}

＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝

『storeDisk.java』

＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝

package Bean;

import java.util.LinkedList;

public class storeDisk extends LinkedList<Disk>{

}

＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝

『storeUser.java』

＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝

package Bean;

import java.util.LinkedList;

public class storeUser extends LinkedList<User>{

}

＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝

『User.java』

＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝

package Bean;

import java.util.LinkedList;

import java.util.Objects;

public class User {

private String name;

private int id;

private storeDisk borrowed;

private double money;

public User() {

borrowed=new storeDisk();

}

public String getName() {

return name;

}

public int getId() {

return id;

}

public double getMoney() {

return money;

}

public storeDisk getBorrowed() {

return borrowed;

}

public void setName(String name) {

this.name = name;

}

public void setId(int id) {

this.id = id;

}

public void setMoney(double money) {

this.money = money;

}

public void setBorrowed(Disk borrowed) {

this.borrowed.push(borrowed);

}

@Override

public int hashCode() {

final int prime = 31;

int result = 1;

result = prime \* result + ((borrowed == null) ? 0 : borrowed.hashCode());

result = prime \* result + id;

result = prime \* result + ((name == null) ? 0 : name.hashCode());

return result;

}

@Override

public boolean equals(Object obj) {

if (this == obj)

return true;

if (obj == null)

return false;

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

return false;

User other = (User) obj;

if (id != other.id)

return false;

if (name == null) {

if (other.name != null)

return false;

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

return false;

return true;

}

@Override

public String toString() {

return "User [name=" + name + ", id=" + id + ", borrowed=" + borrowed + ", money=" + money + "]";

}

}

＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝

『Control.java』

＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝

package Controller;

import Viewer.View;

import java.util.Scanner;

import Mapper.\*;

public class Control {

private Mapper.Map st;

private View vwView;

private Scanner cScanner=new Scanner(System.in);

public Control() {

st=new Mapper.Map();

vwView=new View();

}

private void starthere() {

vwView.welcome();

while(parser(vwView.getMessage()));

}

public static void main(String[] args) {

Control boss=new Control();

boss.starthere();

}

private void addUser() {

while (true) {

st.addUser(vwView.addUser());

System.out.println("Finish");

System.out.println("Do you wanna add again?[y/n]");

if(cScanner.next().equals("n")) {

break;

}

}

}

private void delUser() {

while (true) {

st.delUser(vwView.delUser());

System.out.println("Finish");

System.out.println("Do you wanna add again?[y/n]");

if(cScanner.next().equals("n")) {

break;

}

}

}

private void addCD() {

while (true) {

st.addDisk(vwView.addDisk());

System.out.println("Finish");

System.out.println("Do you wanna add again?[y/n]");

if(cScanner.next().equals("n")) {

break;

}

}

}

private void reportStore() {

st.reportStore();

}

private void reportUser() {

st.reportUser();

}

private void rentCD() {

while (true) {

st.rent(vwView.rent());

System.out.println("Finish");

System.out.println("Do you wanna add again?[y/n]");

if(cScanner.next().equals("n")) {

break;

}

}

}

private void buyCD() {

while (true) {

st.buy(vwView.buy());

System.out.println("Finish");

System.out.println("Do you wanna add again?[y/n]");

if(cScanner.next().equals("n")) {

break;

}

}

}

private boolean parser(String s) {

switch (s) {

case "h":

case "help":

help();

break;

case "b":

case "buy":

buyCD();

break;

case "r":

case "rent":

rentCD();

break;

case "ru":

case "reportUser":

reportUser();

break;

case "rd":

case "reportDisk":

reportStore();

break;

case "ac":

case "addCD":

addCD();

break;

case "au":

case "addUser":

addUser();

break;

case "du":

case "deleteUser":

delUser();

break;

case "exit":

return false;

default:

System.out.println("Unknown Command");

break;

}

return true;

}

private void help() {

System.out.println("This is Help: ");

System.out.println("Commands: "

+ "buy disks:buy/b "

+ "rent disks:rent/r "

+ "add user:addUser/au ");

System.out.println( "add disk:addCD/ac "

+ "delete user:deleteUser/du "

+ "check all users:reportUser/ru "

+ "check all disks:reportDisk/rd "

+ "exit:exit");

}

}

＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝

『Map.java』

＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝

package Mapper;

import java.util.Iterator;

import Bean.Disk;

import Bean.User;

import Bean.storeDisk;

import Bean.storeUser;

public class Map {

private storeDisk sDisk;

private storeUser sUser;

public Map() {

sDisk=new storeDisk();

sUser=new storeUser();

}

public storeDisk getsDisk() {

return sDisk;

}

public storeUser getsUser() {

return sUser;

}

public void addDisk(Disk d) {

for (Iterator iterator =sDisk.iterator(); iterator.hasNext();) {

Disk disk = (Disk) iterator.next();

if(d.equals(disk)) {

disk.setMoney(d.getMoney());

disk.setNum(d.getNum());

return;

}

}

sDisk.add(d);

}

public void addUser(User u) {

for (Iterator iterator =sUser.iterator(); iterator.hasNext();) {

User user = (User) iterator.next();

if(u.equals(user)) {System.out.println("User Deplicated");

return;

}

}

sUser.add(u);

}

public void delUser(User u) {

if(sUser.remove(u)==false) {

System.out.println("Delete Failed");

}

}

private void delDisk() {

}

public void reportStore() {

System.out.println(sDisk.toString());

}

public void reportUser() {

System.out.println(sUser.toString());

}

public boolean rent(User rent) {

for (Iterator iteratorUser =sUser.iterator(); iteratorUser.hasNext();) {

User user = (User) iteratorUser.next();

if(user.equals(rent)) {

for (Iterator iteratorDisk =sDisk.iterator(); iteratorDisk.hasNext();) {

Disk disk = (Disk) iteratorDisk.next();

if(rent.getBorrowed().getLast().equals(disk)) {

Disk torentDisk=rent.getBorrowed().getLast();

torentDisk.setMoney(disk.getMoney());

if(user.getMoney()>=torentDisk.getMoney()\*torentDisk.getNum()&&torentDisk.getNum()<=disk.getNum()) {

user.setBorrowed(torentDisk);

user.setMoney(user.getMoney()-torentDisk.getMoney()\*torentDisk.getNum());

disk.setNum(disk.getNum()-torentDisk.getNum());

return true;

}else {

System.out.println("No Money or No Enough Quantity!");

return false;

}

}

}

System.out.println("No this disk!");

return false;

}

}

System.out.println("User not Found!");

return false;

}

public boolean buy(User buy) {

for (Iterator iteratorUser =sUser.iterator(); iteratorUser.hasNext();) {

User user = (User) iteratorUser.next();

if(user.equals(buy)) {

for (Iterator iteratorDisk =sDisk.iterator(); iteratorDisk.hasNext();) {

Disk disk = (Disk) iteratorDisk.next();

if(buy.getBorrowed().getLast().equals(disk)) {

Disk tobuyDisk=buy.getBorrowed().getLast();

tobuyDisk.setMoney(disk.getMoney());

if(user.getMoney()>=tobuyDisk.getMoney()\*tobuyDisk.getNum()&&tobuyDisk.getNum()<=disk.getNum()) {

user.setMoney(user.getMoney()-tobuyDisk.getMoney()\*tobuyDisk.getNum());

disk.setNum(disk.getNum()-tobuyDisk.getNum());

return true;

}else {

System.out.println("No Money or No Enough Quantity!");

return false;

}

}

}

System.out.println("No this disk!");

return false;

}

}

System.out.println("User not Found!");

return false;

}

}

＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝

『View.java』

＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝

package Viewer;

import java.util.Scanner;

import Bean.Disk;

import Bean.User;

public class View {

private Scanner inScanner=new Scanner(System.in);

public void welcome() {

System.out.println("Welcome CDStore! Enter h for Help!");

}

public String getMessage() {

String tmpString=inScanner.next();

return tmpString;

}

public User addUser() {

User user=new User();

System.out.println("ADDING USERS");

System.out.println("Enter User's Name:");

user.setName(getMessage());

System.out.println("Enter User's Id:");

user.setId(inScanner.nextInt());

System.out.println("Enter User's Money:");

user.setMoney(inScanner.nextDouble());

return user;

}

public User delUser() {

User user=new User();

System.out.println("DELETE USERS");

System.out.println("Enter User's Name:");

user.setName(getMessage());

System.out.println("Enter User's Id:");

user.setId(inScanner.nextInt());

return user;

}

public Disk addDisk() {

Disk newDisk= new Disk();

System.out.println("ADDING DISKS");

System.out.println("Enter Disk's Name:");

newDisk.setName(getMessage());

System.out.println("Enter Disk's Quantity:");

newDisk.setNum(inScanner.nextInt());

System.out.println("Enter Disk's Money:");

newDisk.setMoney(inScanner.nextDouble());

return newDisk;

}

public User rent() {

User user=new User();

System.out.println("RENT");

System.out.println("Enter User's Name:");

user.setName(getMessage());

System.out.println("Enter User's Id:");

user.setId(inScanner.nextInt());

Disk disk=new Disk();

System.out.println("Enter the Disk to Rent:");

disk.setName(getMessage());

System.out.println("Enter the Number of Disks:");

disk.setNum(inScanner.nextInt());

user.getBorrowed().add(disk);

return user;

}

public User buy() {

User user=new User();

System.out.println("BUY");

System.out.println("Enter User's Name:");

user.setName(getMessage());

System.out.println("Enter User's Id:");

user.setId(inScanner.nextInt());

Disk disk=new Disk();

System.out.println("Enter the Disk to Buy:");

disk.setName(getMessage());

System.out.println("Enter the Number of Disks:");

disk.setNum(inScanner.nextInt());

user.getBorrowed().add(disk);

return user;

}

}