CAMERA RENTAL APP

Camera.java –

**package** assessmentProject;

**public** **class** Camera **implements** Comparable<Camera> {

**private** **int** cameraId;

**private** String brand;

**private** String model;

**private** **double** rent;

**private** String status;

**public** **int** getCameraId() {

**return** cameraId;

}

**public** **void** setCameraId(**int** cameraId) {

**this**.cameraId = cameraId;

}

**public** String getBrand() {

**return** brand;

}

**public** **void** setBrand(String brand) {

**this**.brand = brand;

}

**public** String getModel() {

**return** model;

}

**public** **void** setModel(String model) {

**this**.model = model;

}

**public** **double** getRent() {

**return** rent;

}

**public** **void** setRent(**double** rent) {

**this**.rent = rent;

}

**public** String getStatus() {

**return** status;

}

**public** **void** setStatus(String status) {

**this**.status = status;

}

@Override

**public** **int** compareTo(Camera cam) {

**if**(**this**.getCameraId() < cam.cameraId) {

**return** -1;

}

**if**(**this**.getCameraId() > cam.cameraId) {

**return** 1;

}

**return** 0;

}

}

User.java –

**package** assessmentProject;

**public** **class** User {

**private** String username;

**private** String password;

**private** **double** walletBalance;

**public** User() {

setWalletBalance(50000);

}

**public** String getUsername() {

**return** username;

}

**public** **void** setUsername(String username) {

**this**.username = username;

}

**public** String getPassword() {

**return** password;

}

**public** **void** setPassword(String password) {

**this**.password = password;

}

**public** **double** getWalletBalance() {

**return** walletBalance;

}

**public** **void** setWalletBalance(**double** walletBalance) {

**this**.walletBalance = walletBalance;

}

**public** **void** addToWallet(**double** amount)

{

**double** balance = getWalletBalance();

setWalletBalance(balance+amount);

}

}

CameraOperations.java –

**package** cam;

**import** java.util.ArrayList;

**import** java.util.Collections;

**public** **class** CameraOperations {

**static** ArrayList<Camera> *cameras* = **new** ArrayList<Camera>();

**public** **void** addCamera(Camera cm) {

*cameras*.add(cm);

Collections.*sort*(*cameras*);

System.***out***.println("YOUR CAMERA HAS BEEN SUCCESSFULLY ADDED TO THE LIST.");

}

**public** **void** removeCamera(**int** cameraId) {

Camera camera = **new** Camera();

**try** {

**int** index = findIndex(*cameras*, 0,*cameras*.size(), cameraId);

camera = *cameras*.get(index);

**if**(camera.getStatus().equals("Rented")) {

System.***out***.println("CAMERA CANNOT BE REMOVED AS IT IS RENTED.");

}

**else** {

*cameras*.remove(camera);

Collections.*sort*(*cameras*);

System.***out***.println("CAMERA SUCCESSFULLY REMOVED FROM THE LIST.");

}

} **catch** (IndexOutOfBoundsException e) {

System.***out***.println("INVALID CAMERA ID.");

}

}

**public** **int** findIndex(ArrayList<Camera> list, **int** low, **int** high, **int** cameraId) {

**if**(low<=high) {

**int** mid = low + (high-low)/2;

**if**(cameraId == list.get(mid).getCameraId()) {

**return** mid;

}

**if**(cameraId < list.get(mid).getCameraId()) {

**return** findIndex(list, low, mid-1, cameraId);

}

**return** findIndex(list, mid+1, high, cameraId);

}

**return** -1;

}

**public** **void** RentCamera(User user,**int** cameraId) {

Camera camera = **new** Camera();

**try** {

**int** index = findIndex(*cameras*, 0,*cameras*.size(), cameraId);

camera = *cameras*.get(index);

**if**(camera.getStatus().equals("Rented")) {

System.***out***.println("CAMERA IS ALREADY RENTED.");

}

**else** {

**double** balance = user.getWalletBalance();

**if**(balance >= camera.getRent()) {

camera.setStatus("Rented");

user.setWalletBalance(balance-camera.getRent());

System.***out***.println("YOUR TRANSACTION FOR CAMERA - "+camera.getBrand()+" "+camera.getModel()+" with rent INR."+camera.getRent()+" HAS SUCCESSFULLY COMPLETED.");

}

**else** {

System.***out***.println("ERROR: TRANSACTION FAILED DUE TO INSUFFICIENT WALLET BALANCE. PLEASE DEPOSIT THE AMOUNT TO YOUR WALLET.");

}

}

} **catch** (IndexOutOfBoundsException e) {

System.***out***.println("INVALID CAMERA ID.");

}

}

**public** **void** displayAllCameras() {

**if**(*cameras*.size() == 0) {

System.***out***.println("No data present at this moment.");

}

**else** {

System.***out***.println("====================================================================================================");

System.***out***.println("CAMERA ID\t\tBRAND\t\tMODEL\t\tPRICE(PER DAY)\t\tSTATUS");

System.***out***.println("====================================================================================================");

**for**(Camera camera:*cameras*) {

System.***out***.print(camera.getCameraId()+"\t\t\t"+camera.getBrand()+"\t\t"+camera.getModel()+"\t\t"+camera.getRent()+"\t\t\t"+camera.getStatus()+"\n");

}

System.***out***.println("====================================================================================================");

}

}

**public** **void** displayAllCameras(String status) {

**if**(*cameras*.size() == 0) {

System.***out***.println("No data present at this moment.");

}

**else** {

System.***out***.println("====================================================================================================");

System.***out***.println("CAMERA ID\t\tBRAND\t\tMODEL\t\tPRICE(PER DAY)\t\tSTATUS");

System.***out***.println("====================================================================================================");

**for**(Camera camera:*cameras*) {

**if**(camera.getStatus().equals(status)) {

System.***out***.print(camera.getCameraId()+"\t\t\t"+camera.getBrand()+"\t\t"+camera.getModel()+"\t\t"+camera.getRent()+"\t\t\t"+camera.getStatus()+"\n");

}

}

System.***out***.println("====================================================================================================");

}

}

}

Main.java –

package cam;

import java.util.InputMismatchException;

import java.util.Scanner;

public class Main {

public static CameraOperations obj = new CameraOperations();

public static Scanner sc = new Scanner(System.in);

public static String options[] = { "1. MY CAMERA", "2. RENT A CAMERA", "3. VIEW ALL CAMERAS", "4. MY WALLET", "5. EXIT" };

public static String subOptions[] = { "1. ADD", "2. REMOVE", "3. VIEW MY CAMERAS", "4. GO TO PREVIOUS MENU" };

public static void printOptions(String arr[]) {

for (String str : arr) {

System.out.println(str);

}

}

public static void run(User user) {

printOptions(options);

int choice = 0;

try {

choice = sc.nextInt();

switch (choice) {

case 1:

printOptions(subOptions);

int subChoice = sc.nextInt();

switch (subChoice) {

case 1:

System.out.print("ENTER THE CAMERA BRAND - ");

String brand = sc.next();

System.out.print("ENTER THE MODEL - ");

String model = sc.next();

boolean validRent = false;

do {

System.out.print("ENTER THE PER DAY PRICE (INR) - ");

double rent = sc.nextDouble();

if(rent <= 0) {

System.out.println("RENT MUST BE GREATER THAN ZERO.");

}

else {

validRent = true;

int cameraId = CameraOperations.cameras.size() + 1;

Camera newCamera = new Camera();

newCamera.setCameraId(cameraId);

newCamera.setBrand(brand);

newCamera.setModel(model);

newCamera.setRent(rent);

newCamera.setStatus("Available");

obj.addCamera(newCamera);

}

}

while(!validRent);

run(user);

break;

case 2:

obj.displayAllCameras("Available");

if(CameraOperations.cameras.size() > 0) {

System.out.print("ENTER THE CAMERA ID TO REMOVE - ");

int id = sc.nextInt();

obj.removeCamera(id);

}

run(user);

break;

case 3:

obj.displayAllCameras();

run(user);

break;

case 4:

run(user);

break;

default:

System.out.println("INVALID CHOICE");

run(user);

break;

}

break;

case 2:

if (CameraOperations.cameras.size() == 0) {

System.out.println("No data present at this moment.");

}

else {

System.out.println("FOLLOWING IS THE LIST OF AVAILABLE CAMERAS");

obj.displayAllCameras("Available");

System.out.print("ENTER THE CAMERA ID YOU WANT TO RENT - ");

int cameraId = sc.nextInt();

obj.RentCamera(user, cameraId);

}

run(user);

break;

case 3:

obj.displayAllCameras();

run(user);

break;

case 4:

System.out.print("YOUR CURRENT WALLET BALANCE IS - INR." + user.getWalletBalance() + "\n");

System.out.print("DO YOU WANT TO DEPOSIT MORE AMOUNT TO YOUR WALLET?(1.YES 2. NO) - ");

int option = sc.nextInt();

if (option == 1) {

System.out.print("ENTER THE AMOUNT (INR) - ");

double amount = sc.nextDouble();

if (amount <= 0) {

System.out.println("AMOUNT MUST BE GREATER THAN 0");

}

else {

user.addToWallet(amount);

System.out.println("YOUR WALLET BALANCE UPDATED SUCCESSFULLY. CURRENT WALLET BALANCE - INR."+user.getWalletBalance());

}

run(user);

}

else if (option == 2) {

run(user);

}

else {

System.out.println("INVALID CHOICE");

run(user);

}

break;

case 5:

System.out.println("Closing the application....\nThank you!");

break;

default:

System.out.println("INVALID CHOICE");

run(user);

break;

}

} catch (InputMismatchException e) {

System.out.println("INVALID INPUT");

}

}

public static void main(String[] args) {

User user = new User();

user.setUsername("camera");

user.setPassword("camrent");

boolean login = false;

System.out.println("+---------------------------------------+");

System.out.println("|\tWELCOME TO CAMERA RENTAL APP\t|");

System.out.println("+---------------------------------------+");

while (!login) {

System.out.println("PLEASE LOGIN TO CONTINUE - ");

System.out.print("USERNAME - ");

try {

String username = sc.next();

System.out.print("PASSWORD - ");

String password = sc.next();

if (!username.equals(user.getUsername()) || !password.equals(user.getPassword())) {

System.out.println("INCORRECT CREDENTIALS");

}

else {

login = true;

}

} catch (InputMismatchException e) {

System.out.println("INVALID INPUT");

}

}

run(user);

sc.close();

}

}