**package** phaseendproject;

**import** java.util.\*;

**class** InsufficientBalanceException **extends** Exception {

**public** InsufficientBalanceException(String message) {

**super**(message);

}

}

**class** ExcessiveRentException **extends** Exception {

**public** ExcessiveRentException(String message) {

**super**(message);

}

}

**class** Camera {

**private** **int** id;

**private** String brand;

**private** String model;

**private** **double** rentalAmount;

**private** **boolean** available;

**public** Camera(**int** id, String brand, String model, **double** rentalAmount) {

**this**.id = id;

**this**.brand = brand;

**this**.model = model;

**this**.rentalAmount = rentalAmount;

**this**.available = **true**;

}

**public** **int** getId() {

**return** id;

}

**public** String getBrand() {

**return** brand;

}

**public** String getModel() {

**return** model;

}

**public** **double** getRentalAmount() {

**return** rentalAmount;

}

**public** **boolean** isAvailable() {

**return** available;

}

**public** **void** setAvailable(**boolean** available) {

**this**.available = available;

}

}

**class** CameraRentalApp {

**private** List<Camera> cameraList;

**private** **double** walletAmount;

**public** CameraRentalApp() {

cameraList = **new** ArrayList<>();

walletAmount = 1200.0;

}

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

cameraList.add(camera);

}

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

**boolean** found = **false**;

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

**if** (camera.getId() == id) {

cameraList.remove(camera);

found = **true**;

**break**;

}

}

**if** (!found) {

System.***out***.println("Camera not found.");

}

}

**public** **void** rentCamera(**int** id) **throws** InsufficientBalanceException, ExcessiveRentException {

Camera rentedCamera = **null**;

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

**if** (camera.getId() == id) {

rentedCamera = camera;

**break**;

}

}

**if** (rentedCamera == **null**) {

System.***out***.println("Camera not found.");

**return**;

}

**if** (!rentedCamera.isAvailable()) {

**throw** **new** ExcessiveRentException("Camera is already rented.");

}

**if** (rentedCamera.getRentalAmount() > walletAmount) {

**throw** **new** InsufficientBalanceException("Insufficient wallet balance.");

}

rentedCamera.setAvailable(**false**);

walletAmount -= rentedCamera.getRentalAmount();

}

**public** **void** viewAllCameras() {

**if** (cameraList.isEmpty()) {

System.***out***.println("No cameras available for rent.");

} **else** {

System.***out***.println("CAMERA ID\tBRAND\t\tMODEL\t\tRENTAL AMOUNT\tSTATUS");

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

System.***out***.printf("%-10d\t%-15s\t%-15s\t%.2f\t\t%s\n", camera.getId(), camera.getBrand(),

camera.getModel(), camera.getRentalAmount(), camera.isAvailable() ? "Available" : "Rented");

}

}

}

**public** **void** viewWalletBalance() {

System.***out***.println("Current wallet balance: " + walletAmount);

}

**public** **void** depositToWallet(**double** amount) {

walletAmount += amount;

System.***out***.println("Amount deposited successfully.");

}

}

**public** **class** Main {

**public** **static** **void** main(String[] args) {

System.***out***.println("\n\nWelcome to Camera Rental App!\n\n");

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

CameraRentalApp app = **new** CameraRentalApp();

**boolean** running = **true**;

**while** (running) {

System.***out***.println("1. Add a camera");

System.***out***.println("2. Remove a camera");

System.***out***.println("3. Rent a camera");

System.***out***.println("4. View all cameras");

System.***out***.println("5. View wallet balance");

System.***out***.println("6. Deposit to wallet");

System.***out***.println("7. Exit");

System.***out***.print("Enter your choice: ");

**int** choice = scanner.nextInt();

**switch** (choice) {

**case** 1:

// Add a camera

System.***out***.print("Enter the camera ID: ");

**int** id = scanner.nextInt();

System.***out***.print("Enter the brand: ");

String brand = scanner.next();

System.***out***.print("Enter the model: ");

String model = scanner.next();

System.***out***.print("Enter the rental amount: ");

**double** rentalAmount = scanner.nextDouble();

Camera camera = **new** Camera(id, brand, model, rentalAmount);

app.addCamera(camera);

System.***out***.println("Camera added successfully.");

**break**;

**case** 2:

// Remove a camera

System.***out***.print("Enter the camera ID to remove: ");

**int** removeId = scanner.nextInt();

app.removeCamera(removeId);

System.***out***.println("Camera removed successfully.");

**break**;

**case** 3:

// Rent a camera

System.***out***.print("Enter the camera ID to rent: ");

**int** rentId = scanner.nextInt();

**try** {

app.rentCamera(rentId);

System.***out***.println("Camera rented successfully.");

} **catch** (InsufficientBalanceException e) {

System.***out***.println("Failed to rent camera: " + e.getMessage());

} **catch** (ExcessiveRentException e) {

System.***out***.println("Failed to rent camera: " + e.getMessage());

}

**break**;

**case** 4:

// View all cameras

app.viewAllCameras();

**break**;

**case** 5:

// View wallet balance

app.viewWalletBalance();

**break**;

**case** 6:

// Deposit to wallet

System.***out***.print("Enter the amount to deposit: ");

**double** amount = scanner.nextDouble();

app.depositToWallet(amount);

**break**;

**case** 7:

// Exit

running = **false**;

System.***out***.println("Exiting the program.");

**break**;

**default**:

System.***out***.println("Invalid choice. Please try again.");

}

}

scanner.close();}}