**package** CameraRentalApplication;

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

**class** Camera {

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

**private** String brand;

**private** String model;

**private** **double** price;

**private** String status;

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

**this**.id = id;

**this**.brand = brand;

**this**.model = model;

**this**.price = d;

**this**.status = status;

}

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

**return** id;

}

**public** String getBrand() {

**return** brand;

}

**public** String getModel() {

**return** model;

}

**public** **double** getPrice() {

**return** price;

}

**public** String getStatus() {

**return** status;

}

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

**this**.status = status;

}

}

**public** **class** CameraRentalApp {

**private** **static** **final** String ***MyWallet*** = **null**;

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

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

List<Camera> cameras = **new** LinkedList<>();

// Add some default cameras

cameras.add(**new** Camera(1, "Samsung" , "DS123", 500.0, "Available"));

cameras.add(**new** Camera(2, "Sony" , "HD214", 5000.0, "Available"));

cameras.add(**new** Camera(3, "Canon " , "XC" , 500.0, "Available"));

cameras.add(**new** Camera(4, "Fujistu" , "J5" , 500.0, "Available"));

cameras.add(**new** Camera(5, "Sony ", "HD226", 500.0, "Available"));

cameras.add(**new** Camera(6, "Samsung" , "DS246", 500.0, "Available"));

cameras.add(**new** Camera(7, "LG" , "L123 ", 500.0, "Rented" ));

cameras.add(**new** Camera(8, "Canon", "XPL" ,500.0, "Available"));

cameras.add(**new** Camera(9, "Chroma", "CT", 500.0, "Available"));

cameras.add(**new** Camera(10, "Something", "Some", 200.0, "Rented"));

cameras.add(**new** Camera(11, "Canon", "Digital", 123.0, "Available"));

cameras.add(**new** Camera(12, "Some", "Another", 100.0, "Rented"));

cameras.add(**new** Camera(13, "Sony", "DSLR12", 5000.0, "Available"));

cameras.add(**new** Camera(14, "Samsung", "SM123", 200.0, "Rented"));

cameras.add(**new** Camera(15 , "NIKON", "DSLR-D7500", 200.0, "Rented"));

cameras.add(**new** Camera(16, "Sony", "SONY1234", 123.0, "Available"));

cameras.add(**new** Camera(17, "nikon", "NK567", 566.0, "Available"));

cameras.add(**new** Camera(18, "Sony", "SONY14", 133.0, "Available"));

cameras.add(**new** Camera(19, "Nikon", "NIK34", 123.0, "Available"));

cameras.add(**new** Camera(20, "Chroma", "Ch12", 123.0, "Available"));

cameras.add(**new** Camera(21, "panasonic", "pan4", 123.0, "Available"));

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

System.***out***.println("| Welcome to Camera Rental App |");

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

System.***out***.println("Please login to continue");

System.***out***.print("Username: ");

String username = scanner.nextLine();

System.***out***.print("Password: ");

String password = scanner.nextLine();

// Login logic can be implemented here

**boolean** exit = **false**;

**while** (!exit) {

System.***out***.println("1. My Camera");

System.***out***.println("2. Rent a Camera");

System.***out***.println("3. View All Cameras");

System.***out***.println("4. My Wallet");

System.***out***.println("5. Exit");

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

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

scanner.nextLine(); // Consume newline character

**switch** (choice) {

**case** 1:

*handleMyCamera*(cameras, scanner);

**break**;

**case** 2:

*handleRentCamera*(cameras, scanner);

**break**;

**case** 3:

*handleViewAllCameras*(cameras);

**break**;

**case** 4:

*handleMyWallet*(cameras, scanner);

**break**;

**case** 5:

exit = **true**;

System.***out***.println("thankyou ! good bye");

**break**;

**default**:

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

**break**;

}

}

}

**private** **static** **void** handleMyWallet(List<Camera> cameras, Scanner scanner) {

// **TODO** Auto-generated method stub

**double** walletBalance = 1723.5; // Initial wallet balance

System.***out***.println("Your current wallet balance is: INR " + walletBalance);

System.***out***.println("Do you want to deposit more amount to your wallet?");

System.***out***.println("1. Yes");

System.***out***.println("2. No");

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

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

scanner.nextLine(); // Consume newline character

**if** (choice == 1) {

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

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

scanner.nextLine(); // Consume newline character

walletBalance += depositAmount;

System.***out***.println("Your wallet balance has been updated successfully.");

} **else** **if** (choice == 2) {

System.***out***.println("No additional amount deposited. Wallet balance remains the same.");

} **else** {

System.***out***.println("Invalid choice. No changes made to the wallet balance.");

}

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

}

**private** **static** **void** handleRentCamera(List<Camera> cameras, Scanner scanner) {

// Rent camera logic can be implemented here

System.***out***.println("Available Cameras:");

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

System.***out***.printf("%-10s | %-10s | %-10s | %-10s | %-10s%n",

"ID", "Brand", "Model", "Price", "Status");

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

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

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

System.***out***.printf("%-10d | %-10s | %-10s | %-10.2f | %-10s%n",

camera.getId(), camera.getBrand(), camera.getModel(),

camera.getPrice(), camera.getStatus());

}

}

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

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

scanner.nextLine();

// Find the camera with the given ID

Camera rentedCamera = **null**;

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

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

rentedCamera = camera;

**break**;

}

}

**if** (rentedCamera != **null** && rentedCamera.getStatus().equals("Available")) {

**double** cameraPrice = rentedCamera.getPrice();

**double** walletBalance = 1723.5; // Current wallet balance

**if** (walletBalance >= cameraPrice) {

// Sufficient balance, deduct the camera price from the wallet balance

walletBalance -= cameraPrice;

// Update the status of the rented camera

rentedCamera.setStatus("Rented");

System.***out***.printf("Your transaction for camera %s %s with rent %.2f has been successfully added.%n",

rentedCamera.getBrand(),rentedCamera.getModel(), rentedCamera.getPrice());

} **else** {

// Insufficient balance, display error message

System.***out***.println("Transaction failed due to insufficient amount.");

System.***out***.println("Please deposit amount to your wallet.");

}

} **else** {

// Invalid camera ID or camera already rented, display error message

System.***out***.println("Invalid camera ID or camera already rented. Transaction failed.");

}

}

**private** **static** **void** handleMyCamera(List<Camera> cameras, Scanner scanner) {

**boolean** backToMenu = **false**;

**while** (!backToMenu) {

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

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

System.***out***.println("3. View My Cameras");

System.***out***.println("4. Go to Previous Menu");

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

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

scanner.nextLine(); // Consume newline character

**switch** (choice) {

**case** 1:

// Add camera logic can be implemented here

// Consume newline character

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

String brand = scanner.nextLine();

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

String model = scanner.nextLine();

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

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

scanner.nextLine(); // Consume newline character

// Create a new Camera object and add it to the list

Camera newCamera = **new** Camera( choice, brand, model, price, brand);

cameras.add(newCamera);

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

**break**;

**case** 2:

System.***out***.println("Camera ID\tBrand\t\tModel\tPrice\tStatus");

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

System.***out***.printf("%d\t\t%s\t%s\t%.2f\t%s\n",

camera.getId(), camera.getBrand(), camera.getModel(),

camera.getPrice(), camera.getStatus());

}

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

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

scanner.nextLine(); // Consume newline character

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

Iterator<Camera> iterator = cameras.iterator();

**while** (iterator.hasNext()) {

Camera camera = iterator.next();

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

iterator.remove();

found = **true**;

**break**;

}

}

**if** (found) {

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

} **else** {

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

}

**break**;

**case** 3:

System.***out***.println("My Cameras:");

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

System.***out***.printf("%-3s | %-12s | %-10s | %-8s | %-8s\n",

"ID", "Brand", "Model", "Price", "Status");

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

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

System.***out***.printf("%-3d | %-12s | %-10s | %-8.2f | %-8s\n",

camera.getId(), camera.getBrand(), camera.getModel(),

camera.getPrice(), camera.getStatus());

}

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

**break**;

**case** 4:

backToMenu = **true**;

**break**;

**default**:

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

**break**;

}

}

}

**public** **static** Camera binarySearchCameraByID(List<Camera> cameras, **int** id) {

**int** low = 0;

**int** high = cameras.size() - 1;

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

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

Camera camera = cameras.get(mid);

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

**return** camera;

} **else** **if** (camera.getId() < id) {

low = mid + 1;

} **else** {

high = mid - 1;

}

}

**return** **null**; // Camera not found

}

**public** **static** **void** bubbleSort(List<Camera> cameras) {

**int** n = cameras.size();

**for** (**int** i = 0; i < n - 1; i++) {

**for** (**int** j = 0; j < n - i - 1; j++) {

**if** (cameras.get(j).getId() > cameras.get(j + 1).getId()) {

Collections.*swap*(cameras, j, j + 1);

}

}

}

}

**public** **static** **void** mergeSort(List<Camera> cameras) {

**if** (cameras.size() <= 1) {

**return**;

}

**int** mid = cameras.size() / 2;

List<Camera> left = **new** ArrayList<>(cameras.subList(0, mid));

List<Camera> right = **new** ArrayList<>(cameras.subList(mid, cameras.size()));

*mergeSort*(left);

*mergeSort*(right);

*merge*(cameras, left, right);

}

**private** **static** **void** merge(List<Camera> cameras, List<Camera> left, List<Camera> right) {

**int** i = 0, j = 0, k = 0;

**while** (i < left.size() && j < right.size()) {

**if** (left.get(i).getId() <= right.get(j).getId()) {

cameras.set(k++, left.get(i++));

} **else** {

cameras.set(k++, right.get(j++));

}

}

**while** (i < left.size()) {

cameras.set(k++, left.get(i++));

}

**while** (j < right.size()) {

cameras.set(k++, right.get(j++));

}

}

**private** **static** **void** handleViewAllCameras(List<Camera> cameras) {

System.***out***.println("All Cameras:");

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

System.***out***.println("ID | Brand | Model | Price | Status ");

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

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

System.***out***.printf("%-4d | %-12s | %-10s | %.2f | %s\n",

camera.getId(), camera.getBrand(), camera.getModel(),

camera.getPrice(), camera.getStatus());

}

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

}

}