

Source Code

Camera rental Application -

Developer : Shivam Jadhav

```
import java.util.ArrayList;

import java.util.Scanner;

public class CameraRentalApplication {

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

    private static double walletAmount = 0;

    public static void main(String[] args) {

        cameras.add(new Camera(1, "Canon ", "EOS R6", 100.0, "Available"));

        cameras.add(new Camera(2, "Nikon ", "Z6 II", 120.0, "Available"));

        cameras.add(new Camera(3, "Sony ", "Alpha", 150.0, "Available"));

        cameras.add(new Camera(4, "Samsung", "X-T4", 80.0, "Available"));

        cameras.add(new Camera(5, "Panasonic", "Lumix", 200.0, "Available"));


        try {

            System.out.println("\n");

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

            System.out.println("|      WELCOME TO CAMERA RENTAL APP      |");

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

            System.out.println("\nPlease login to continue");

            System.out.print("\nLogin ID: ");

            Scanner sc = new Scanner(System.in);

            String loginId= sc.next();

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

            String password= sc.next();
```

```
        if(password.equals("admin")) {  
            System.out.println("\nLogin SUCCESSFUL !");  
        }  
        else{  
            throw new Exception("Incorrect password!");  
        }  
    }
```

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

```
boolean isRunning = true;
```

```
while (isRunning) {
```

```
    System.out.println("\nEnter an option number to proceed:");
```

```
    System.out.println("1. My cameras");
```

```
    System.out.println("2. Rent a camera");
```

```
    System.out.println("3. Add or view wallet amount");
```

```
    System.out.println("4. List all cameras");
```

```
    System.out.println("5. Close the application");
```

```
    int option = scanner.nextInt();
```

```
    switch (option) {
```

```
        case 1:
```

```
            myCamera(scanner);
```

```
            break;
```

```
        case 2:
```

```
            rentCamera(scanner);
```

```
            break;
```

```
        case 3:
```

```
            addOrViewWalletAmount(scanner);
```

```
            break;
```

```
        case 4:
```

```

        listCameras();

        break;

    case 5:

        isRunning = false;

        break;

    default:

        System.out.println("Invalid option selected.");

    }

}

}catch (Exception e) {

    System.out.println(e.getMessage());

}

}

private static void listCameras() {

    if (cameras.isEmpty()) {

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

    } else {

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

        System.out.println("ID    "+"Brand    "+"Model    "+"Rate(Per day)    "+"Status");

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

        for (Camera camera : cameras) {

            System.out.printf("%s    %s    %s    $%.2f    %s\n", camera.getid(), camera.getBrand(),
camera.getModel(), camera.getPerDayRentalAmount(), camera.getStatus());

        }

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

    }

}

```

```
}
```

```
private static void rentCamera(Scanner scanner) {
```

```
    if (cameras.isEmpty()) {
```

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

```
    } else {
```

```
        listCameras();
```

```
        System.out.print("\nEnter the number of the camera you want to rent: ");
```

```
        int cameraNumber = scanner.nextInt();
```

```
        if (cameraNumber < 1 || cameraNumber > cameras.size()) {
```

```
            System.out.println("Invalid camera number selected.");
```

```
        } else {
```

```
            Camera selectedCamera = cameras.get(cameraNumber - 1);
```

```
            double rentalAmount = selectedCamera.getPerDayRentalAmount();
```

```
            if (walletAmount < rentalAmount) {
```

```
                System.out.printf("\nYou don't have enough balance in your wallet to rent this camera.  
Current balance: $%.2f\n", walletAmount);
```

```
            } else {
```

```
                walletAmount -= rentalAmount;
```

```
                System.out.printf("You have rented %s %s for $%.2f per day. Your current wallet balance is  
$%.2f.\n", selectedCamera.getBrand(), selectedCamera.getModel(), rentalAmount, walletAmount);
```

```
            }
```

```
        }
```

```
    }
```

```
}
```

```
private static void myCamera(Scanner scanner) {
```

```
    Scanner s = new Scanner(System.in);
```

```

boolean isRunning = true;

while (isRunning) {
    System.out.println("Enter an option number to proceed:");
    System.out.println("1. Add cameras");
    System.out.println("2. Remove camera");
    System.out.println("3. view cameras");
    System.out.println("4. Back to menu");

    int option = scanner.nextInt();
    switch (option) {
        case 1:
            addCamera(s);

            break;
        case 2:
            removeCamera(s);
            break;
        case 3:
            viewCamera(s);
            break;
        case 4:
            return;

        default:
            System.out.println("Invalid option selected.");
    }
}

private static void addCamera(Scanner scanner) {

```

```

System.out.println("Enter camera ID:");

int id = scanner.nextInt();

scanner.nextLine();


System.out.println("Enter camera brand:");

String brand = scanner.nextLine();


System.out.println("Enter camera model:");

String model = scanner.nextLine();


System.out.println("Enter per day rental amount:");

double perDayRentalAmount = scanner.nextDouble();

scanner.nextLine();


System.out.println("Enter camera status (Available/Not Available:");

String status = scanner.nextLine();


cameras.add(new Camera(id, brand, model, perDayRentalAmount, status));


System.out.println("Camera added successfully!");
}


private static void removeCamera(Scanner scanner) {
    listCameras();

    System.out.println("Enter the ID of the camera you want to remove:");

    int id = scanner.nextInt();

    for (int i = 0; i < cameras.size(); i++) {
        if (cameras.get(i).id == id) {
            cameras.remove(i);

            System.out.println("Camera with ID " + id + " has been removed.");

            return;
        }
    }
}

```

```

    }

    }

    System.out.println("Camera with ID " + id + " not found.");
}

private static void viewCamera(Scanner scanner) {
    System.out.println("List of all cameras:");

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

    for (Camera camera : cameras) {
        System.out.println(camera.id + "      " + camera.brand + "      " + camera.model + "      " +
camera.perDayRentalAmount + "      " + camera.status);
    }

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

}

private static void addOrViewWalletAmount(Scanner scanner) {
    System.out.printf("Your current wallet balance is $%.2f.\n", walletAmount);

    System.out.print("Do you want to add more money to your wallet? (y/n): ");

    String choice = scanner.next();

    if (choice.equalsIgnoreCase("y")) {
        System.out.print("\nEnter the amount you want to add: ");

        double amountToAdd = scanner.nextDouble();

        walletAmount += amountToAdd;

        System.out.printf("%.2f has been added to your wallet. Your current balance is $%.2f.\n",
amountToAdd, walletAmount);
    }

}
}

```

```
private static class Camera {  
    private int id;  
        private String brand;  
    private String model;  
    private double perDayRentalAmount;  
    private String status;  
  
    public Camera(int id, String brand, String model, double perDayRentalAmount, String status) {  
        this.id = id;  
        this.brand = brand;  
        this.model = model;  
        this.perDayRentalAmount = perDayRentalAmount;  
        this.status = status;  
    }  
  
    public int getId() {  
        return id;  
    }  
  
    public String getBrand() {  
        return brand;  
    }  
  
    public String getModel() {  
        return model;  
    }  
  
    public double getPerDayRentalAmount() {  
        return perDayRentalAmount;  
    }  
}
```



```
public String getStatus() {  
    return status;  
}  
}  
}
```