**Basics of Java Practice Exercise – Day 6**

Submitted By: Aniket Singh (Emp no. – 2605511)

**1.Create an interface Instrument which is having the abstract function play. Create three implementations from Instrument which is Piano, Flute, Guitar using Lambda Expressions. Override the play method inside all three classes printing a message as follows,**

**Note: You must not allow the user to declare an object of Instrument class.**

**• Create an array of 10 Instruments.**

**• Assign different type of instrument-to-Instrument reference.**

**• Check for the polymorphic behavior of play method.**

**• Use the instanceof operator to print that which object stored at which index of instrument array**

**Code –**

**package** practiceday6;

@FunctionalInterface

**interface** Instrument{

**void** play();

}

**public** **class** Instrument1 {

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

Instrument piano =()->System.***out***.println("Piano is playing tan tan tan tan");

Instrument flute=()->System.***out***.println("Flute is playing toot toot toot toot");

Instrument guitar=()->System.***out***.println("Guitar is playing tin tin tin");

Instrument[] ins=**new** Instrument[10];

**for**(**int** i=0;i<10;i++) {

**int** rand=(**int**)(Math.*random*()\*((3-1)+1));

**if**(rand==1) {

ins[i]=piano;

}

**else** **if**(rand==2) {

ins[i]=flute;

}

**else** {

ins[i]=guitar;

}

ins[i].play();

}

}

}

**HealthBox is an online Life Coaching application that helps its users to sign up and log in to** **seek the guidance of famous Life Coaches across the world from different expertise in one place. Users can book an appointment for Life Coach based on specialty. They can also see upcoming appointments and can reschedule or cancel the appointments.**

**Provide a menu for the user to choose from the below options and perform accordingly:**

**1. Schedule an appointment where customer can enter date, time, and zone to which they belong.**

**2. Print appointment details only if it is booked if not display appropriate message.**

**3. Reschedule an appointment.**

**4. Get Reminder where customer can check for the schedule one day prior to the appointment.**

**5. We can cancel the appointments before the scheduled date only if not display appropriate message.**

**6. Exit**

**Code –**

**import** java.util.Scanner;

**import** java.time.LocalDateTime;

**import** java.time.ZoneId;

**import** java.time.format.DateTimeFormatter;

**import** java.time.temporal.ChronoUnit;

**public** **class** HealthBox {

**private** **static** LocalDateTime *appointmentDateTime*;

**private** **static** ZoneId *appointmentZone*;

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

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

**int** choice;

**do** {

System.***out***.println("\nHealthBox - Life Coaching Application");

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

System.***out***.println("1. Schedule an appointment");

System.***out***.println("2. Print appointment details");

System.***out***.println("3. Reschedule an appointment");

System.***out***.println("4. Get Reminder");

System.***out***.println("5. Cancel appointment");

System.***out***.println("6. Exit");

System.***out***.print("Enter an Option: ");

choice = scanner.nextInt();

scanner.nextLine();

**switch** (choice) {

**case** 1:

*scheduleAppointment*(scanner);

**break**;

**case** 2:

*printAppointmentDetails*();

**break**;

**case** 3:

*rescheduleAppointment*(scanner);

**break**;

**case** 4:

*getReminder*();

**break**;

**case** 5:

*cancelAppointment*();

**break**;

**case** 6:

System.***out***.println("Exiting............!");

**break**;

**default**:

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

}

} **while** (choice != 6);

scanner.close();

}

**private** **static** **void** scheduleAppointment(Scanner scanner) {

System.***out***.print("\nEnter appointment Date and Time (YYYY-MM-DD HH:MM): ");

String dateTimeInput = scanner.nextLine();

System.***out***.println("Available Zones are");

System.***out***.println("A. America/Anchorage");

System.***out***.println("B. Europe/Paris");

System.***out***.println("C. Asia/Tokyo");

System.***out***.println("D. America/Phoenix");

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

System.***out***.println("Select the Zone");

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

String zoneInput = scanner.nextLine();

**try** {

*appointmentDateTime* = LocalDateTime.*parse*(dateTimeInput, DateTimeFormatter.*ofPattern*("YYYY-MM-DD HH:MM"));

*appointmentZone* = ZoneId.*of*(zoneInput);

System.***out***.println("Successfully Booked");

} **catch** (Exception e) {

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

}

}

**private** **static** **void** printAppointmentDetails() {

**if** (*appointmentDateTime* != **null** && *appointmentZone* != **null**) {

System.***out***.println("\nAppointment Details:");

System.***out***.println("Date and Time: " + *appointmentDateTime*.format(DateTimeFormatter.*ofPattern*("yyyy-MM-dd HH:mm")));

System.***out***.println("Time Zone: " + *appointmentZone*);

} **else** {

System.***out***.println("\nNo appointment is scheduled.");

}

}

**private** **static** **void** rescheduleAppointment(Scanner scanner) {

**if** (*appointmentDateTime* != **null** && *appointmentZone* != **null**) {

System.***out***.print("Enter new appointment date and time (YYYY-MM-DD HH:MM): ");

String newDateTimeInput = scanner.nextLine();

**try** {

*appointmentDateTime* = LocalDateTime.*parse*(newDateTimeInput, DateTimeFormatter.*ofPattern*("yyyy-MM-dd HH:mm"));

System.***out***.println("Appointment rescheduled successfully.");

} **catch** (Exception e) {

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

}

} **else** {

System.***out***.println("No appointment is scheduled to reschedule.");

}

}

**private** **static** **void** getReminder() {

**if** (*appointmentDateTime* != **null** && *appointmentZone* != **null**) {

LocalDateTime now = LocalDateTime.*now*(*appointmentZone*);

**long** hoursUntilAppointment = ChronoUnit.***HOURS***.between(now, *appointmentDateTime*);

**if** (hoursUntilAppointment >= 24 && hoursUntilAppointment < 48) {

System.***out***.println("Reminder: Your appointment is scheduled for tomorrow.");

} **else** {

System.***out***.println("No upcoming appointment for tomorrow.");

}

} **else** {

System.***out***.println("No appointment is scheduled.");

}

}

**private** **static** **void** cancelAppointment() {

**if** (*appointmentDateTime* != **null** && *appointmentZone* != **null**) {

LocalDateTime now = LocalDateTime.*now*(*appointmentZone*);

**if** (now.isBefore(*appointmentDateTime*)) {

*appointmentDateTime* = **null**;

*appointmentZone* = **null**;

System.***out***.println("Appointment canceled successfully.");

} **else** {

System.***out***.println("Cannot cancel an appointment that has already passed or is today.");

}

} **else** {

System.***out***.println("No appointment is scheduled to cancel.");

}

}

}

**3.1 Astound has recently opened its internet services in India. The company wants users to register for their internet services. Presently, there are two kinds of users, i.e., employees and students. However, the company has established the following criteria for user authentication:**

**• Firstly, for both the type of users, Name and two Phone No (one alternate phone number) is compulsory.**

**• Also, the user should have passport. If user is not having a passport, he/she can provide**

**anyone of the following combinations.**

**I. License number and pan card.**

**II. Voter Id and License number. Adding on criteria mandatory for an Employee:**

**• Provide employee Id.**

**Design a generic class of name Register having methods to generate registration id and display based on the type of user.**

**Code –**

Employee.java

**package** practiceday6;

**import** java.util.Arrays;

**public** **class** Employee {

String name;

**long**[] phoneNo;

String passportNo;

**int** licenseNo;

String panCardNo;

**int** votedId,employeeId;

**public** Employee(String name, **long**[] phoneNo, String passportNo, **int** employeeId) {

**super**();

**this**.name = name;

**this**.phoneNo = phoneNo;

**this**.passportNo = passportNo;

**this**.employeeId = employeeId;

}

**public** Employee(String name, **long**[] phoneNo, **int** licenseNo, String panCardNo, **int** employeeId) {

**super**();

**this**.name = name;

**this**.phoneNo = phoneNo;

**this**.licenseNo = licenseNo;

**this**.panCardNo = panCardNo;

**this**.employeeId = employeeId;

}

**public** Employee(String name, **long**[] phoneNo, **int** licenseNo, **int** votedId, **int** employeeId) {

**super**();

**this**.name = name;

**this**.phoneNo = phoneNo;

**this**.licenseNo = licenseNo;

**this**.votedId = votedId;

**this**.employeeId = employeeId;

}

@Override

**public** String toString() {

**if**(**this**.passportNo!=**null**) {

**return** "Name = " + name + "\nPhoneNo = " + Arrays.*toString*(phoneNo) + "\nPassportNo = " + passportNo

+ "\nEmployeeId = "

+ employeeId;

}

**else** **if**(**this**.panCardNo!=**null**) {

**return** "Name = " + name + "\nPhoneNo = " + Arrays.*toString*(phoneNo)

+ "\nLicenseNo = " + licenseNo + "\nPanCardNo = " + panCardNo + "\nEmployeeId = "

+ employeeId;

}

**else** {

**return** "Name = " + name + "\nPhoneNo = " + Arrays.*toString*(phoneNo)

+ "\nLicenseNo = " + licenseNo + "\nVotedId = " + votedId + "\nEmployeeId = "

+ employeeId;

}

}

}

Student.java

**package** practiceday6;

**import** java.util.Arrays;

**public** **class** Student {

String name;

**long**[] phoneNo;

String passportNo;

**int** licenseNo;

String panCardNo;

**int** votedId;

**public** Student(String name, **long**[] phoneNo, String passportNo) {

**super**();

**this**.name = name;

**this**.phoneNo = phoneNo;

**this**.passportNo = passportNo;

}

**public** Student(String name, **long**[] phoneNo, **int** licenseNo, String panCardNo) {

**super**();

**this**.name = name;

**this**.phoneNo = phoneNo;

**this**.licenseNo = licenseNo;

**this**.panCardNo = panCardNo;

}

**public** Student(String name, **long**[] phoneNo, **int** licenseNo, **int** votedId) {

**super**();

**this**.name = name;

**this**.phoneNo = phoneNo;

**this**.licenseNo = licenseNo;

**this**.votedId = votedId;

}

@Override

**public** String toString() {

**if**(**this**.passportNo!=**null**) {

**return** "Name = " + name + "\nPhoneNo = " + Arrays.*toString*(phoneNo) + "\nPassportNo = " + passportNo;

}

**else** **if**(**this**.panCardNo!=**null**) {

**return** "Name = " + name + "\nPhoneNo = " + Arrays.*toString*(phoneNo)

+ "\nLicenseNo = " + licenseNo + "\nPanCardNo = " + panCardNo;

}

**else** {

**return** "Name = " + name + "\nPhoneNo = " + Arrays.*toString*(phoneNo)

+ "\nLicenseNo = " + licenseNo + "\nVotedId = " + votedId;

}

}

}

Register.java

package practiceday6;

import java.util.Random;

import java.lang.\*;

public class Register<T> {

private String registerId;

public void display(T obj) {

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

}

public String generateRegisterId(int n) {

String chars="ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789";

StringBuilder registerId=new StringBuilder();

Random rand=new Random();

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

registerId.append(chars.charAt(rand.nextInt(chars.length())));

}

return registerId.toString();

}

}

Tester.java

**package** practiceday6;

**import** java.util.Arrays;

**public** **class** Tester {

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

**long**[] empPhone= {9889756321L,7896541258L};

Employee emp=**new** Employee("Arun",empPhone,"LA788333DG",1101);

Register<Employee> empRegister=**new** Register<>();

String empRegId=empRegister.generateRegisterId(9);

System.***out***.println("==========Employee Details==========\n");

System.***out***.println("Registration Id: "+empRegId);

empRegister.display(emp);

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

**long**[] empPhone2= {9090493455L,9898941258L};

Employee emp2=**new** Employee("Andrew",empPhone2,2199,"SJLDD176J",1114);

Register<Employee> empRegister2=**new** Register<>();

String empRegId2=empRegister2.generateRegisterId(5);

System.***out***.println("==========Employee Details==========\n");

System.***out***.println("Registration Id: "+empRegId2);

empRegister2.display(emp2);

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

**long**[] empPhone3= {8090493455L,8998941258L};

Employee emp3=**new** Employee("Rose",empPhone3,2777,837427,1010);

Register<Employee> empRegister3=**new** Register<>();

String empRegId3=empRegister3.generateRegisterId(7);

System.***out***.println("==========Employee Details==========\n");

System.***out***.println("Registration Id: "+empRegId3);

empRegister3.display(emp3);

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

**long**[] stuPhone1= {8090493455L,8998941258L};

Student stu1=**new** Student("Aniket",stuPhone1,"ASD1344");

Register<Student> stuRegister1=**new** Register<>();

String stuRegId1=stuRegister1.generateRegisterId(4);

System.***out***.println("==========Student Details==========\n");

System.***out***.println("Registration Id: "+stuRegId1);

stuRegister1.display(stu1);

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

**long**[] stuPhone2= {9030493455L,7098941258L};

Student stu2=**new** Student("Joseph",stuPhone1,2210,"DUPPS2781J");

Register<Student> stuRegister2=**new** Register<>();

String stuRegId2=stuRegister2.generateRegisterId(10);

System.***out***.println("==========Student Details==========\n");

System.***out***.println("Registration Id: "+stuRegId2);

stuRegister2.display(stu2);

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

**long**[] stuPhone3= {8090493455L,8998941258L};

Student stu3=**new** Student("Albert",stuPhone3,7168,293049);

Register<Student> stuRegister3=**new** Register<>();

String stuRegId3=stuRegister3.generateRegisterId(8);

System.***out***.println("==========Student Details==========\n");

System.***out***.println("Registration Id: "+stuRegId3);

stuRegister3.display(stu3);

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

}

}