**Day-6 Java Assignment**

**Problem Statement 1: Java 8- Lambda Expressions**

**package** day6.com;

**public** **class** InstrumentDemo {

**interface** Instrument {

**void** play();

  }

**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 strum strum strum strum");

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

  instruments[0] = piano;

  instruments[1] = flute;

  instruments[2] = guitar;

  instruments[3] = piano;

  instruments[4] = flute;

  instruments[5] = guitar;

  instruments[6] = piano;

  instruments[7] = flute;

  instruments[8] = guitar;

  instruments[9] = piano;

**for** (Instrument instrument : instruments) {

  instrument.play();

  }

**for** (**int** i = 0; i < instruments.length; i++) {

**if** (instruments[i] **instanceof** Instrument) {

  System.***out***.print("Element at index " + i + " is of type: ");

**if** (instruments[i] == piano) {

  System.***out***.println("Piano");

  } **else** **if** (instruments[i] == flute) {

  System.***out***.println("Flute");

  } **else** **if** (instruments[i] == guitar) {

  System.***out***.println("Guitar");

  }

  }

  }

  }

  }

**2: New Date-Time API in Java 8**

**package** day6.com;

**import** java.time.LocalDate;

**import** java.time.LocalTime;

**import** java.time.ZoneId;

**import** java.time.ZonedDateTime;

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

**import** java.util.Scanner;

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

**private** **static** ZonedDateTime *appointment*;

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

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

DateTimeFormatter dateFormatter = DateTimeFormatter.*ofPattern*("dd/MM/yyyy");

DateTimeFormatter timeFormatter = DateTimeFormatter.*ofPattern*("HH:mm");

DateTimeFormatter dateTimeFormatter = DateTimeFormatter.*ofPattern*("EEE, MMM dd yyyy hh:mm a");

**while** (**true**) {

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 the Appointments.");

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

System.***out***.println("Enter an Option");

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

scanner.nextLine(); // Consume newline

**switch** (option) {

**case** 1:

System.***out***.println("Enter Date (dd/mm/yyyy):");

String dateInput = scanner.nextLine();

LocalDate date = LocalDate.*parse*(dateInput, dateFormatter);

System.***out***.println("Enter Time (hh:mm):");

String timeInput = scanner.nextLine();

LocalTime time = LocalTime.*parse*(timeInput, timeFormatter);

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("Select the Zone (A/B/C/D):");

String zoneOption = scanner.nextLine();

ZoneId zone = *getZoneId*(zoneOption);

*appointment* = ZonedDateTime.*of*(date, time, zone);

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

**break**;

**case** 2:

**if** (*appointment* != **null**) {

System.***out***.println(*appointment*.format(dateTimeFormatter));

}

**else**

{

System.***out***.println("No appointment booked.");

}

**break**;

**case** 3:

**if** (*appointment* != **null**)

{

System.***out***.println("Current Appointment Date is " + *appointment*.format(dateTimeFormatter));

System.***out***.println("Kindly Enter Number of Days to be postponed:");

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

scanner.nextLine(); // Consume newline

System.***out***.println("Enter the new time in HH:mm:");

String newTimeInput = scanner.nextLine();

LocalTime newTime = LocalTime.*parse*(newTimeInput, timeFormatter);

*appointment* = *appointment*.plusDays(days).with(newTime);

System.***out***.println("Your Appointment has been rescheduled to: " + *appointment*.format(dateTimeFormatter));

}

**else**

{

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

}

**break**;

**case** 4:

**if** (*appointment* != **null**)

{

ZonedDateTime reminderTime = *appointment*.minusDays(1);

System.***out***.println("Reminder: Your appointment is on " + *appointment*.format(dateTimeFormatter) + ", you will be reminded on " + reminderTime.format(dateTimeFormatter));

}

**else**

{

System.***out***.println("No appointment booked.");

}

**break**;

**case** 5:

**if** (*appointment* != **null** && *appointment*.isAfter(ZonedDateTime.*now*())) {

*appointment* = **null**;

System.***out***.println("Appointment has been cancelled!!");

}

**else**

{

System.***out***.println("No appointment to cancel or it's already past the appointment date.");

}

**break**;

**case** 6:

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

scanner.close();

**return**;

**default**:

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

}

}

}

**private** **static** ZoneId getZoneId(String zoneOption) {

**switch** (zoneOption.toUpperCase()) {

**case** "A":

**return** ZoneId.*of*("America/Anchorage");

**case** "B":

**return** ZoneId.*of*("Europe/Paris");

**case** "C":

**return** ZoneId.*of*("Asia/Tokyo");

**case** "D":

**return** ZoneId.*of*("America/Phoenix");

**default**:

**throw** **new** IllegalArgumentException("Invalid zone option");

}

}

}

**3: Design the highly general and reusable code with Generic classes**

**package** day6.com;

**import** java.util.Random;

**class** Employee {

**private** String name;

**private** **long**[] phoneNo;

**private** String passportNo;

**private** String licenseNo;

**private** String panCardNo;

**private** String voterId;

**private** Integer employeeId;

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

**this**.name = name;

**this**.phoneNo = phoneNo;

**this**.employeeId = employeeId;

**this**.passportNo = passportNo;

}

**public** Employee(String name, **long**[] phoneNo, Integer employeeId, String licenseNo, String voterId) {

**this**.name = name;

**this**.phoneNo = phoneNo;

**this**.employeeId = employeeId;

**this**.licenseNo = licenseNo;

**this**.voterId = voterId;

}

**public** String toString() {

StringBuilder sb = **new** StringBuilder();

sb.append("Name: ").append(name).append("\n");

sb.append("Phone No's: ");

**for** (**long** phone : phoneNo) {

sb.append(phone).append(", ");

}

sb.delete(sb.length() - 2, sb.length());

sb.append("\nEmployee Id: ").append(employeeId).append("\n");

**if** (passportNo != **null**) sb.append("Passport No: ").append(passportNo).append("\n");

**if** (licenseNo != **null**) sb.append("License No: ").append(licenseNo).append("\n");

**if** (panCardNo != **null**) sb.append("Pan No: ").append(panCardNo).append("\n");

**if** (voterId != **null**) sb.append("Voter Id: ").append(voterId).append("\n");

**return** sb.toString();

}

}

**class** Student {

**private** String name;

**private** **long**[] phoneNo;

**private** String passportNo;

**private** String licenseNo;

**private** String panCardNo;

**private** String voterId;

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

**this**.name = name;

**this**.phoneNo = phoneNo;

**this**.passportNo = passportNo;

}

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

**this**.name = name;

**this**.phoneNo = phoneNo;

**this**.voterId = voterId;

**this**.licenseNo = licenseNo;

}

**public** String toString() {

StringBuilder sb = **new** StringBuilder();

sb.append("Name: ").append(name).append("\n");

sb.append("Phone No's: ");

**for** (**long** phone : phoneNo) {

sb.append(phone).append(", ");

}

sb.delete(sb.length() - 2, sb.length());

**if** (passportNo != **null**) sb.append("\nPassport No: ").append(passportNo).append("\n");

**if** (licenseNo != **null**) sb.append("\nLicense No: ").append(licenseNo).append("\n");

**if** (panCardNo != **null**) sb.append("Pan No: ").append(panCardNo).append("\n");

**if** (voterId != **null**) sb.append("Voter Id: ").append(voterId).append("\n");

**return** sb.toString();

}

}

**class** Register<T> {

**private** String registerId;

**public** **void** generateRegisterId(**int** n) {

StringBuilder sb = **new** StringBuilder(n);

Random random = **new** Random();

String characters = "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789";

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

**int** index = random.nextInt(characters.length());

sb.append(characters.charAt(index));

}

**this**.registerId = sb.toString();

}

**public** **void** display(T user) {

System.***out***.println("Hurray!! you availed a discount of " + (user **instanceof** Student ? "22%" : "10%"));

System.***out***.println("Registered Id: " + **this**.registerId);

System.***out***.println(user.toString());

}

}

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

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

**long**[] phoneNumbers1 = {99973899811L, 9094930952L};

  Employee employee1 = **new** Employee("Arun", phoneNumbers1, 1101, "LA788333DG");

**long**[] phoneNumbers2 = {90904934551L, 9389849551L};

  Employee employee2 = **new** Employee("Aruna", phoneNumbers2, 1114, "2199", "SJLDD1761J");

**long**[] phoneNumbers3 = {7676384945L, 8727387400L};

  Employee employee3 = **new** Employee("Rose", phoneNumbers3, 1010, "2777", "837427");

**long**[] phoneNumbers4 = {9038474875L, 8359493029L};

  Student student1 = **new** Student("Shiva", phoneNumbers4, "2210", "DUPPS2781J");

**long**[] phoneNumbers5 = {87934794951L, 9239402595L};

  Student student2 = **new** Student("Naveen", phoneNumbers5, "293049", "7168");

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

  employeeRegister.generateRegisterId(7);

  System.***out***.println("Details of the Employee:");

  employeeRegister.display(employee1);

  employeeRegister.generateRegisterId(7);

  System.***out***.println("Details of the Employee:");

  employeeRegister.display(employee2);

  employeeRegister.generateRegisterId(7);

  System.***out***.println("Details of the Employee:");

  employeeRegister.display(employee3);

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

  studentRegister.generateRegisterId(7);

  System.***out***.println("Details of the Student:");

  studentRegister.display(student1);

  studentRegister.generateRegisterId(7);

  System.***out***.println("Details of the Student:");

  studentRegister.display(student2);

  }

  }

has context menu

has context menu