**Day-6 Basics of Java**

Problem Statement 1: Java 8- Lambda Expressions

//Interface

**package** com.mycom.lambda;

**public** **interface** Instruments {

**void** play();

}

// Main Class

**import** java.util.ArrayList;

**import** java.util.List;

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

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

List<Instruments> instruments = **new** ArrayList<>(10);

instruments.add(() -> System.***out***.println("Piano is playing tan tan tan tan"));

instruments.add(() -> System.***out***.println("Flute is playing toot toot toot toot"));

instruments.add(() -> System.***out***.println("Guitar is playing strum strum strum"));

**for** (**int** i = 3; i < 10; i++) {

**int** instrumentIndex = i % 3;

**if** (instrumentIndex == 0) {

instruments.add(() -> System.***out***.println("Piano is playing tan tan tan tan"));

} **else** **if** (instrumentIndex == 1) {

instruments.add(() -> System.***out***.println("Flute is playing toot toot toot toot"));

} **else** {

instruments.add(() -> System.***out***.println("Guitar is playing strum strum strum"));

}

}

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

instrument.play();

}

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

Instruments instrument = instruments.get(i);

System.***out***.print("Instrument at index " + i + " is a ");

**if** (instrument.toString().contains("tan")) {

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

} **else** **if** (instrument.toString().contains("toot")) {

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

} **else** {

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

}

}

}

}

Problem Statement 2: New Date-Time API in Java 8

**package** com.mycom.lambda;

**import** java.time.\*;

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

**import** java.util.Scanner;

**public** **class** AppointmentScheduler {

**private** **static** ZonedDateTime *appointment* = **null**;

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

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

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

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

**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("=========================");

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

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

scanner.nextLine();

**switch** (option) {

**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.*exit*(0);

**break**;

**default**:

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

}

}

}

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

System.***out***.print("Enter Date (dd/MM/yyyy): ");

String dateInput = scanner.nextLine();

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

System.***out***.print("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***.print("Select the Zone: ");

String zoneInput = scanner.nextLine().toUpperCase();

ZoneId zone = **null**;

**switch** (zoneInput) {

**case** "A":

zone = ZoneId.*of*("America/Anchorage");

**break**;

**case** "B":

zone = ZoneId.*of*("Europe/Paris");

**break**;

**case** "C":

zone = ZoneId.*of*("Asia/Tokyo");

**break**;

**case** "D":

zone = ZoneId.*of*("America/Phoenix");

**break**;

**default**:

System.***out***.println("Invalid zone! Appointment not scheduled.");

**return**;

}

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

System.***out***.println("Successfully Booked: " + *appointment*);

}

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

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

System.***out***.println("Appointment Details: " + *appointment*);

}

**else** {

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

}

}

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

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

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

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

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

scanner.nextLine();

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

String newTimeInput = scanner.nextLine();

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

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

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

} **else** {

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

}

}

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

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

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

System.***out***.println("Reminder: " + reminder);

} **else** {

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

}

}

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

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

*appointment* = **null**;

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

} **else** {

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

}

}

}

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

// Employee Class

**package** com.mycom.lambda.employee;

**import** java.util.Arrays;

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

**private** String name;

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

**private** String passportNo;

**private** Integer licenseNo;

**private** String panCardNo;

**private** Integer 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, Integer licenseNo, String panCardNo) {

**this**.name = name;

**this**.phoneNo = phoneNo;

**this**.employeeId = employeeId;

**this**.licenseNo = licenseNo;

**this**.panCardNo = panCardNo;

}

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

**this**.name = name;

**this**.phoneNo = phoneNo;

**this**.employeeId = employeeId;

**this**.voterId = voterId;

**this**.licenseNo = licenseNo;

}

@Override

**public** String toString() {

**return** String.*format*("Name: %s\n Phone No's: %s\n Employee Id: %d\n Passport No: %s\n License No: %d\n Pan No: %s\n Voter Id: %d",

name, Arrays.*toString*(phoneNo), employeeId, passportNo, licenseNo, panCardNo, voterId);

}

}

//Student Class

**import** java.util.Arrays;

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

**private** String name;

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

**private** String passportNo;

**private** Integer licenseNo;

**private** String panCardNo;

**private** Integer voterId;

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

**this**.name = name;

**this**.phoneNo = phoneNo;

**this**.passportNo = passportNo;

}

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

**this**.name = name;

**this**.phoneNo = phoneNo;

**this**.licenseNo = licenseNo;

**this**.panCardNo = panCardNo;

}

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

**this**.name = name;

**this**.phoneNo = phoneNo;

**this**.voterId = voterId;

**this**.licenseNo = licenseNo;

}

@Override

**public** String toString() {

**return** String.*format*("Name: %s\nPhone No's: %s\nLicense No: %d\nPan No: %s\nVoter Id: %d",

name, Arrays.*toString*(phoneNo), licenseNo, panCardNo, voterId);

}

}

// Register class

**import** java.util.Random;

**public** **class** Register<T> {

**private** String registerId;

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

String characters = "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789";

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

StringBuilder registerId = **new** StringBuilder();

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

registerId.append(characters.charAt(random.nextInt(characters.length())));

}

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

**return** **this**.registerId;

}

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

System.***out***.println("Hurray!! you availed a discount of 10%");

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

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

}

}

// Tester Main Class

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

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

**long**[] phoneNos1 = {9997389981L, 9094930952L};

Employee emp1 = **new** Employee("Anu", phoneNos1, 1101, "LA788333DH");

**long**[] phoneNos2 = {9090493455L, 9389849551L};

Employee emp2 = **new** Employee("Andrew", phoneNos2, 1114, 2199, "SJLDD1781J");

**long**[] phoneNos3 = {7676384945L, 8727387400L};

Employee emp3 = **new** Employee("Rose", phoneNos3, 1010, 837427, 2777);

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

empRegister.generateRegisterId(8);

empRegister.display(emp1);

empRegister.generateRegisterId(8);

empRegister.display(emp2);

empRegister.generateRegisterId(8);

empRegister.display(emp3);

**long**[] phoneNos4 = {9038474875L, 8359493029L};

Student stu1 = **new** Student("John", phoneNos4, 2210, "DUPPS2781K");

**long**[] phoneNos5 = {8793479495L, 9239402595L};

Student stu2 = **new** Student("Albert", phoneNos5, 293049, 7168);

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

stuRegister.generateRegisterId(8);

stuRegister.display(stu1);

stuRegister.generateRegisterId(8);

stuRegister.display(stu2);

}

}