**DAY-6 PRACTICE EXERCISE**

**BASICS OF JAVA**

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

Instrument Interface

**package** org.software.com;

**interface** Instrument {

**void** play();

}

InstrumentTest Class

**public** **class** InstrumentTest {

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

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

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

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

Instrument[] instruments = { piano, flute, guitar, piano, flute, guitar, piano, flute, guitar, piano };

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

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

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

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

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

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

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

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

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

}

instruments[i].play();

}

}

}

}

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

HealthBoxApp Class

**package** org.software.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 HealthBoxApp {

**private** LocalDate date;

**private** LocalTime time;

**private** String zone;

**private** void getReminder() {

if (date == null || time == null || zone == null) {

**System**.out.println("No Appointments are Available");

return;

}

ZoneId zoneId = ZoneId.of(zone);

ZonedDateTime zonedDateTime = ZonedDateTime.of(date, time, zoneId);

DateTimeFormatter myFormatObj = DateTimeFormatter.ofPattern("E, MMM dd yyyy hh:mm:ss a");

String formattedDate = zonedDateTime.format(myFormatObj) + " " + getZone();

**System**.out.println(formattedDate);

}

**public** **void** printAppointment() {

if (date == null || time == null || zone == null) {

**System**.out.println("No Appointments are Available");

return;

}

ZoneId zoneId = ZoneId.of(zone);

ZonedDateTime zonedDateTime = ZonedDateTime.of(date, time, zoneId);

DateTimeFormatter myFormatObj = DateTimeFormatter.ofPattern("dd-MM-yyyy hh:mm:ss a");

String formattedDate = zonedDateTime.format(myFormatObj) + " " + getZone();

**System**.out.println(formattedDate);

}

public String getZone() {

**switch** (zone) {

case "America/Anchorage":

return "AKST";

case "Europe/Paris":

return "CEST";

case "Asia/Tokyo":

return "JST";

case "America/Phoenix":

return "MST";

default:

return "";

}

}

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

HealthBoxApp app = new HealthBoxApp();

Scanner sc = 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("Enter an Option:");

int choice = sc.nextInt();

sc.nextLine();

switch (choice) {

case 1:

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

String appDate = sc.nextLine();

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

app.date = LocalDate.parse(appDate, dateFormatter);

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

String appTime = sc.nextLine();

app.time = LocalTime.parse(appTime);

**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:");

String zoneChoice = sc.nextLine();

switch (zoneChoice) {

case "A":

app.zone = "America/Anchorage";

break;

case "B":

app.zone = "Europe/Paris";

break;

case "C":

app.zone = "Asia/Tokyo";

break;

case "D":

app.zone = "America/Phoenix";

break;

default:

System.out.println("Invalid zone choice.");

continue;

}

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

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

break;

case 2:

app.printAppointment();

break;

case 3:

if (app.date == null) {

**System**.out.println("No Appointments to Reschedule");

break;

}

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

app.getReminder();

**System**.out.println("Enter the number of days to be postponed:");

int daysToPostpone = sc.nextInt();

sc.nextLine();

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

String newTimeStr = sc.nextLine();

LocalTime newTime = LocalTime.parse(newTimeStr);

app.date = app.date.plusDays(daysToPostpone);

app.time = newTime;

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

app.getReminder();

break;

case 4:

app.getReminder();

break;

case 5:

app.date = null;

app.time = null;

app.zone = null;

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

break;

case 6:

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

sc.close();

return;

default:

**System**.out.println("Sorry, Invalid Option");

}

}

}

}

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

**Generic classes**

Employee Class

**package** org.software.com;

**import** java.util.Arrays;

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

String name;

**long**[] phoneNo;

String passportNo;

**int** licenseNo;

String pancardNo;

**int** voterId;

**int** employeeId;

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

**this**.name = name;

**this**.phoneNo = phoneNo;

**this**.passportNo = passportNo;

**this**.employeeId = employeeId;

}

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

**this**.name = name;

**this**.phoneNo = phoneNo;

**this**.licenseNo = licenseNo;

**this**.pancardNo = pancardNo;

**this**.employeeId = employeeId;

}

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

**this**.name = name;

**this**.phoneNo = phoneNo;

**this**.licenseNo = licenseNo;

**this**.voterId = voterId;

**this**.employeeId = employeeId;

}

@Override

**public** String toString() {

System.***out***.println("Name : " + name);

System.***out***.println("Phone No : " + Arrays.*toString*(phoneNo));

System.***out***.println("Emp Id : " + employeeId);

**if** (passportNo != **null**)

System.***out***.println("Passport No : " + passportNo);

**if** (licenseNo != 0)

System.***out***.println("License No : " + licenseNo);

**if** (pancardNo != **null**)

System.***out***.println("Pancard No : " + pancardNo);

**if** (voterId != 0)

System.***out***.println("Voter Id : " + voterId);

**return** "";

}

}

Student Class

**package** org.software.com;

**import** java.util.Arrays;

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

String name;

**long**[] phoneNo;

String passportNo;

**int** licenseNo;

String pancardNo;

**int** voterId;

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

**this**.name = name;

**this**.phoneNo = phoneNo;

**this**.passportNo = passportNo;

}

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

**this**.name = name;

**this**.phoneNo = phoneNo;

**this**.licenseNo = licenseNo;

**this**.pancardNo = pancardNo;

}

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

**this**.name = name;

**this**.phoneNo = phoneNo;

**this**.licenseNo = licenseNo;

**this**.voterId = voterId;

}

@Override

**public** String toString() {

System.***out***.println("Name : " + name);

System.***out***.println("Phone No : " + Arrays.*toString*(phoneNo));

**if** (passportNo != **null**)

System.***out***.println("Passport No : " + passportNo);

**if** (licenseNo != 0)

System.***out***.println("License No : " + licenseNo);

**if** (pancardNo != **null**)

System.***out***.println("Pancard No : " + pancardNo);

**if** (voterId != 0)

System.***out***.println("Voter Id : " + voterId);

**return** "";

}

}

Registration<T> Class

**package** org.software.com;

**import** java.util.Random;

**public** **class** Registration<T> {

String registerId;

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

**if** (obj **instanceof** Employee) {

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

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

} **else** {

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

System.***out***.println("Hurray!! you availed a discount of 22%");

}

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

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

}

String generateRegisterId(**int** n) {

String characters = "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789";

Random rm = **new** Random();

StringBuilder id = **new** StringBuilder();

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

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

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

}

registerId = id.toString();

**return** registerId;

}

}

Tester Class

**package** org.software.com;

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

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

Registration<Employee> reEmp = **new** Registration<Employee>();

Registration<Student> reStd = **new** Registration<Student>();

**long**[] phoneNo = { 9712345678L, 9876543210L };

// EMPLOYEE TEST CASE 1 Passport

Employee emp1=**new** Employee("Chaitanya", phoneNo, 1234, "SRVRJKS2");

reEmp.generateRegisterId(7);

reEmp.display(emp1);

// EMPLOYEE TEST CASE 2 licenseNo and Pancard No

Employee emp2 = **new** Employee("Karthik", phoneNo, 1456, 7473681, "GSHGD5364H");

reEmp.generateRegisterId(7);

reEmp.display(emp2);

// EMPLOYEE TEST CASE 3 licenseNo and Voter Id

Employee emp3 = **new** Employee("Rajashekhar Reddy", phoneNo, 9876, 6467236, 737637483);

reEmp.generateRegisterId(7);

reEmp.display(emp3);

// STUDENT TEST CASE 1 Passport

Student st1 = **new** Student("Sreekanth Reddy", phoneNo, "SRVRJKS2");

reStd.generateRegisterId(7);

reStd.display(st1);

// STUDENT TEST CASE 2 licenseNo and Pancard No

Student st2 = **new** Student("Akhila", phoneNo, 7473681, "GSHGD5364H");

reStd.generateRegisterId(7);

reStd.display(st2);

// STUDENT TEST CASE 3 licenseNo and Voter Id

Student st3 = **new** Student("Arun", phoneNo, 6467236, 737637483);

reStd.generateRegisterId(7);

reStd.display(st3);

}

}