TP10

Java DataBase Connection (JDBC)

Remarks:

- ✓ All classes should have validations
- ✓ All data from 10.3 to 10.5 are stored in Database using Java DataBase Connection (JDBC)

The try-with-resources Statement

The try-with-resources statement is a try statement that declares one or more resources. A resource is an object that must be closed after the program is finished with it. The try-with-resources statement ensures that each resource is closed at the end of the statement.

Any object that implements *java.lang.AutoCloseable*, which includes all objects which implement *java.io.Closeable*, can be used as a resource.

Example:

```
public class TryRes {
    public static void main(String[] args) {
        try(Scanner sc = new Scanner(System.in)){
            System.out.println("Input sentence: ");
            String str = sc.nextLine();
        } // will call sc.close() automatically

    }
}
```

Example 2:

```
import java.io.*;
public class TryReadFile {
    static String readFirstLineFromFile(String path) throws IOException {
        try (BufferedReader br = new BufferedReader(new FileReader(path))) {
            return br.readLine();
        } //will call br.close() automatically

    }
    public static void main(String[] args) {
        try {
            var text = readFirstLineFromFile("data.txt");
            System.out.println(text);
        } catch (IOException e) {
            System.out.println("OOPs Sth went wrong!");
        }
    }
}
```

Java Database connection (JDBC)

We can connect to Database like MySQL, SQLite, many mores using JDBC drivers. Common steps in connection are:

1. Establish a connection
2. Create JDBC Statements
3. Execute SQL Statements
4. GET ResultSet
5. Close connections

Driver for MySQL server: https://dev.mysql.com/downloads/connector/j/

Driver for SQLite DB:
https://www.sqlite.org/java/raw/doc/overview.html?name=0a704f4b7294a3d63e6ea2b612daa3b997c4b5f1

And many mores.

import java.sql.*;

Load the vendor specific driver

```
Class.forName("com.mysql.cj.jdbc.Driver");
```

Make the connection

```
Connection con = DriverManager.getConnection(
  "jdbc:mysql://localhost:3306/i4db?user=root&password=secret");
```

Create statement for query:

```
Statement stmt = con.createStatement() ;
String createStudentTable = "Create table students " +
     "(ID Integer not null, Name VARCHAR(32), " + "Marks Integer)";
stmt.executeUpdate(createStudentTable);
String insertStudent = "Insert into students values‟ +
  "(e20226789,abc,100)";
stmt.executeUpdate(insertStudent);
```

Data manipulation

+ Selecting data

```
String queryStudent = "select * from students";
ResultSet rs = Stmt.executeQuery(queryStudent);
while (rs.next()) {
  int ssn = rs.getInt("ID");
  String name = rs.getString("NAME");
  int marks = rs.getInt("MARKS");
}
```

+ Inserting data

```
String queryStudent = "insert into students(Name,Marks)values('Sophy',9)";
int affectedRowCount = Stmt.executeUpdate(queryStudent);
```

+ Deleting data

```
String queryStudent = "delete * from students where ID=1";
int affectedRowCount = Stmt.executeUpdate(queryStudent);
```

TP10.1. Date class

Create the DateUtil class, which will contain the date information (using java.util.Calendar class).
Define methods for:

1. Subtraction operation of two dates (the result should be presented as a number of days between dates), and save the history of substraction in database (ID, date_start, date_end, n_days, operation_type, changed_date)
   ex: 1, '2022-01-01', '2022-01-28', 28, 'Substraction', '2022-01-28 08:08:08'

2. Operation of increasing the date by a certain number of days. Also store history in database.
   ex: 2, '2022-01-28', '2022-01-31', 3, 'Increment', '2022-01-28 08:09:08'
   3, '2020-02-01', '2022-01-20', -11, 'Increment', '2022-01-28 08:19:08'

TP10.2. ChangesLog class

Using class created above as reference to create the ChangesLog class with the following features:

1. Override toString() method that returns a string containing last 5 changes in date history
2. Listing changes by date
3. Listing changes by week
4. Listing changes by date range
5. List all changes

TP10.3. Customer class

Create class Customer that are waiting in queue for ordering food at PopularStore. It contains the following features:

1. Fields: number, date entering into queue, order status (waiting to order, waiting for food, ordering, …)
2. Constructors
3. Data input, output, accessors and mutators

TP10.4. Queue class

Using previous tasks as references and reuse classes of it. Create new class name CustomerQueue represents a list of customers that are waiting for ordering and those are waiting for ordered foods. This class should be able to:

- Get customers count
- Add new customer to queue (last one will serve last too like FIFO)
- Remove a customer from queue (FIFO)
- Serve a customer follow FIFO

The maximum customers can be queued is 100. If it reached the maximum, customer cannot be added to queue. There should be not display on screen in this class.

TP10.5. Popular Store class

Create new class name PopularStore represents a popular store sales information. This class should be able to:

- Serve a customer (duration of serving is random between 1 to 10 minutes, if pass this maximum, the customer will be moved to tail of the queue). In serving a customer, cashier should:
  - Record things customer ordered
  - Record money the customer paid
  - Record duration of serving
  - Print receipt

Example:

```
Start serving time: 08:30:00
What do you like?
A Hamburger and a Coke

Amount to pay ($): 3.5

Pay by (0: cash, 1: credit card): 1
In case credit card, please input credit card number: 1234 5874 4856 1254
and pass code: 235

Receipt:
---------------------------------------
Customer No.: 00005
A Hamburger and a Coke = 3.5$
(In credit card **** **** **** 1254)

------------ Thanks you!!! ------------

Issue date: 2017/10/03 08:36:02
---------------------------------------

End serving time: 08:36:02
Serving duration: 00:05:58
```

- Add customer to waiting list (queue) for serving
- Display list of customers in queue
- List all served customers (orders history)