Liam Baron

EDUV4813733@VOSSIE.NET

Details

The first Question I tackle creating a HashMap and a LinkedHashMap and the methods that can be used by them.  
Creating a Clinic app that takes data from the user and inserts it into the SQL Server table.

bASIC jAVA PROJECT

INDEX

[INFORMATION ABOUT PROJECT 2](#_Toc147388675)

[QUESTION 1 3](#_Toc147388676)

[QUESTION 2 5](#_Toc147388677)

[BIBLIOGRAPHY 9](#_Toc147388678)

# INFORMATION ABOUT PROJECT

* The version of Java that was used is Java version “20.0.2”
* The driver version that was used was version 12.4
* With the operating system being Windows 10

TO RUN QUESTION 1:

* “Javac Question1.java” then “ Java Question1 “

TO RUN QUESTION 2:

- “Javac -cp “file path of driver\ mssql-jdbc-12.4.1.jre11.jar” CreatingDatabase.java then

“Java -cp “file path of driver\ mssql-jdbc-12.4.1.jre11.jar” CreatingDatabase

APPLICATIONS THAT ARE NEEDED FOR QUESTION 2:

* Java.
* Microsoft SQL Server.
* An IDE for Java.
* SMSS to run SQL Server line of code.
* The drivers are also needed.

# QUESTION 1

import java.util.HashMap;

import java.util.LinkedHashMap;

import java.util.Map;

public class Quetion1 {

public static void main(String[] args) {

//Creating a HashMap.

HashMap<Integer, String> frenchToastIngredients = new HashMap<>();

//Inserting data into the HashMap.

frenchToastIngredients.put(1, "Bread");

frenchToastIngredients.put(2, "Eggs");

frenchToastIngredients.put(3, "Milk");

frenchToastIngredients.put(4, "Butter");

frenchToastIngredients.put(5, "Honey");

// Printing out the HashMap.

System.out.println(frenchToastIngredients);

// Finding a specific index as well as printing it out.

System.out.println(frenchToastIngredients.get(1));

// Removing an element from the HashMap. And printing out the changes.

frenchToastIngredients.remove(5);

System.out.println(frenchToastIngredients);

//Creating a Linked List.

LinkedHashMap<String, String> pokemon = new LinkedHashMap<>();

//Adding elements to the Linked List.

pokemon.put("Fire", "Charizard");

pokemon.put("Fighting", "Ratata");

pokemon.put("Electric", "Pikachu");

pokemon.put("Ground", "Diglet");

pokemon.put("Dark", "Ceruledge");

//Printing the Linked List.

System.out.println(pokemon);

}

}

1.2 The difference between a hashMap and a LinkedHashMap is that the LinkedHashMap extends from the HashMap class so it has the same features as the HashMap as well as its own features so this makes it a subclass. There insertion order in the LinkedHashMap. It only contains unique values. LinkedHashMap uses more memory than a HashMap. HashMap extends from the Abstract class.

# QUESTION 2

Steps to follow if you have dynamic ports enabled:

* Add the authorization file in to the system32 folder (You need to be admin).
* Change the port in the code. To find the correct port you can check the properties of tcp/ip or in the error log located in the SQL Server folder (You need to be admin).
* Before you can run the program make sure to create the database “CREATE DATABSE BasicJavaPro”. And create a table with this line of code in SQL Server: “CREATE TABLE patients (name VARCHAR(30), surname VARCHAR(30), idNum INT, cellNum INT, studNum VARCHAR(40))”
* Your line in cmd should be similar to this: java -cp "C:\Program Files (x86)\Java\jre-1.8\lib\ext\mssql-jdbc-12.4.1.jre11.jar" CreatingDatabase.java .

Else :

* Add the driver file into the same folder as the java file you downloaded.
* Before you can run the program make sure to create the database “CREATE DATABSE BasicJavaPro”. And create a table with this line of code in SQL Server: “CREATE TABLE patients (name VARCHAR(30), surname VARCHAR(30), idNum INT, cellNum INT, studNum VARCHAR(40))”
* Then run it:”java -cp mssql-jdbc-12.4.1.jre11.jar CreatingDatabase.java”.

CODE:

import java.util.Scanner;

import java.sql.\*;

public class CreatingDatabase {

public static void main(String[] args) {

Scanner scan = new Scanner(System.in);

// initializing variables.

int port = 1433;

String connectionURL = "jdbc:sqlserver://localhost:" + port + ";database=BasicJavaPro;integratedSecurity=true;encrypt=false";

String patientName;

String patientSurname;

String idNumTemp;

String cellNumTemp;

int idNum = 0;

int cellNum = 0;

String studNum = "1";

/\*java -cp mssql-jdbc-12.4.1.jre11.jar CreatingDatabase.java \*/

while (true) {

// This will ask the user to input data.

System.out.println("Welcome to Eduvos Clinic App");

System.out.print("Enter Patient name: ");

patientName = scan.nextLine();

System.out.print("Enter Patient surname: ");

patientSurname = scan.nextLine();

System.out.print("Enter ID Number: ");

idNumTemp = scan.nextLine();

System.out.print("Enter Cell Number: ");

cellNumTemp = scan.nextLine();

System.out.print("Enter Student Number: ");

studNum = scan.nextLine();

/\* These are to catch whether or not ID number is and int and the same

applies to cell number. \*/

try {

idNum = Integer.parseInt(idNumTemp);

cellNum = Integer.parseInt(cellNumTemp);

} catch (Exception x) {

System.out.println("Enter Numbers for ID or Cell Number");

System.out.println();

continue;

}

if (studNum.equals("0")) {

break;

}

try {

// Creating connection between Java and SQL Server using JDBC

Connection con = DriverManager.getConnection(connectionURL);

System.out.println("Connected to the database Eduvos Clinic");

// Insertion of data into the table named "patients".

String insertSql = "INSERT INTO patients VALUES (" + "'" + patientName + "'" + "," + "'" + patientSurname + "'" + "," + idNum + "," + cellNum + "," + "'" + studNum + "'" + ")";

PreparedStatement prepsInsertData = con.prepareStatement(insertSql);

prepsInsertData.execute();

System.out.println("Patient information has been added to the database");

System.out.println();

} catch (SQLException y) {

y.printStackTrace();

}

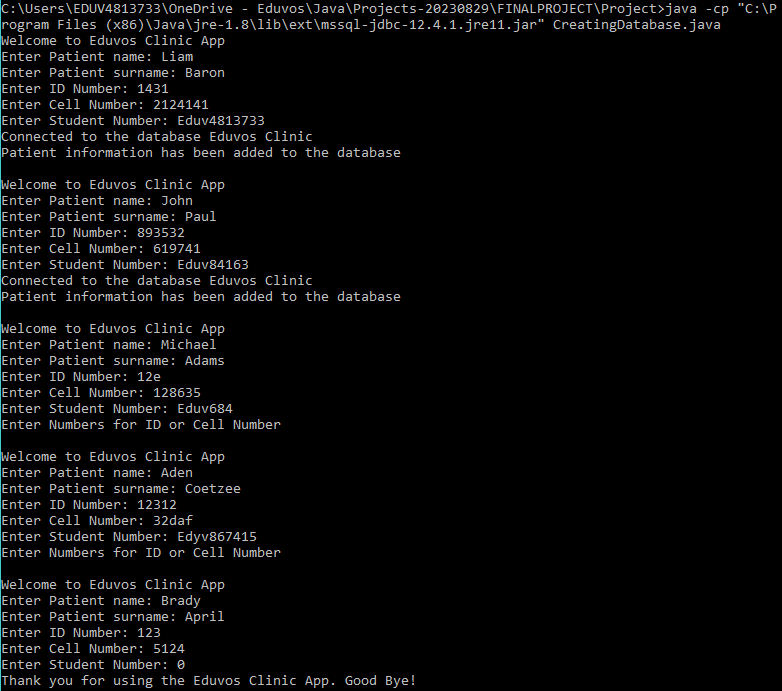
}

System.out.println("Thank you for using the Eduvos Clinic App. Good Bye!");

}

}

Examples of the code:



And the result being displayed in the table:

A screenshot of a computer

Description automatically generated

# BIBLIOGRAPHY

HashMap and LinkedHashMap :

* <https://www.javatpoint.com/linkedhashmap-vs-hashmap-in-java#:~:text=The%20Major%20Difference%20between%20the,the%20ordering%20of%20the%20elements>.

Downloading Java:

* <https://www.oracle.com/za/java/technologies/downloads/>

This helped me with understanding ports for connecting Java to SQL Server:

* <https://blog.devart.com/sql-server-ports.html>