**ปฏิบัติการที่ 12**

**Management Application for MongoDB**

**วัตถุประสงค์**

* **เพื่อให้นิสิตเข้าใจการใช้งานเบื้องต้นของโปรแกรมฐานข้อมูล MongoDB กับโปรแกรมภาษา Java เพื่อใช้ในการสร้าง Application**

1. **การทดลองการประยุกต์ใช้ฐานข้อมูล MongoDB กับภาษา Java เพื่อพัฒนาพัฒนาโปรแกรม โดยมีขั้นตอนการติดตั้งดังนี้**
2. ดาวโหลด MongoDB java driver คือ mongo-java-driver-3.11.2.jar ที่ลิงค์นี้

https://drive.google.com/file/d/1JSwWdpakjVMAkMMqHYsJZ2hOMeb5t24U/view?usp=sharing

1. สร้างโฟลเด้อเป็นรหัสนิสิตที่ตำแหน่ง F:\65xxxxx โดยในโฟลเด้อมีโฟลเด้อย่อยชื่อ lib และนำไฟล์

mongo-java-driver-3.11.2.jar ไปเก็บไว้ในโฟลเด้อ lib

1. การเขียนโปรแกรมด้วยภาษา Java เพื่อติอต่อฐานข้อมูล MongoDB โดยพิมพ์คำสั่งต่อไปนี้ โดยออกแบบให้อยู่ในรูปแบบของ GUI

* ข้อมูลวิธีการใช้งานเบื้องต้นการสร้าง GUI JAVA ด้วย NetBeans ตามลิ้ง:

http://www.thaicreate.com/java/java-gui-netbeans.html

1. **ตัวอย่างคำสั่งการจัดการฐานข้อมูลด้วย java code**

การเขียนโปรแกรมด้วยภาษา Java เพื่อแสดงชื่อฐานข้อมูลในระบบ

import com.mongodb.MongoClient;

import java.util.List;

public class MongoDisplayDB{

public static void main(String[] args) {

try {

MongoClient mongo = new MongoClient("localhost", 27017);

List<String> dbs = mongo.getDatabaseNames();

for(String db : dbs){

System.out.println(db);

}

} catch (Exception e) {

}

}

}

การเขียนโปรแกรมด้วยภาษา Java เพื่อแสดงชื่อ Collections ในฐานข้อมูลที่กำหนด

import com.mongodb.DB;

import com.mongodb.MongoClient;

import java.util.Set;

public class MongoDisplayAllCollections{

public static void main(String[] args) {

try {

MongoClient mongo = new MongoClient("localhost", 27017);

DB db = mongo.getDB("test");

Set<String> tables = db.getCollectionNames();

for(String coll : tables){

System.out.println(coll);

} } catch (Exception e) {

}

}

}

1. **การเขียนโปรแกรมแสดงชื่อฐานข้อมูลใน DB Server ด้วยภาษา Java โดยมีเงื่อนไขดังต่อไปนี้**
   1. เมื่อทำการเปิดโปรแกรมให้แสดงจอดังต่อไปนี้

|  |  |  |
| --- | --- | --- |
| No. | Action | Detail |
| 1 | Interface design Show DB Name |  |
| 2 | ตั้งชื่อไฟล์ | **ShowDB.java** |
| 3 | รายละเอียด | * Title Bar แสดงข้อความ “**Show DB Name**” * แสดงข้อความ “**ติดต่อฐานข้อมูล และ แสดงฐานข้อมูลในระบบ**” * ลิสต์แสดงรายชื่อฐานข้อมูล * ปุ่มกดแสดงฐานข้อมูลทั้งหมด * ปุ่มกด “Exit” ปิดโปรแกรม |
| 4 | หน้าจอเมื่อใช้งานจริง |  |
| 5 | รายละเอียด | ดึงข้อมูลรายการรายชื่อฐานข้อมูลทั้งหมดใน Server มาแสดงใน List |

1.2 โค้ดโปรแกรมแสดงชื่อฐานข้อมูลทั้งหมดใน DB Server

* จัดเรียงให้สวยงาม
* มี Comment อธิบายโค้ดเข้าใจง่าย
* ตั้งชื่อตัวแปรให้สอดคล้องกับข้อมูลที่เก็บ ห้าม a1 a2 a3
* นำเฉพาะ method ที่เราเขียนมาใหม่ลงในตาราง

|  |  |
| --- | --- |
| Line No. | File: ShowDB.java |
| 1  2  3  4  5  6  7 | /\*\* import code here\*\*/  import com.mongodb.MongoClient;  import com.mongodb.MongoClientOptions;  import com.mongodb.ServerAddress;  import javax.swing.DefaultListModel;  import javax.swing.JOptionPane;  // modelListDocument : DefaultListModel      // using for set to show list components so it's private      private final *DefaultListModel* modelListDocument = **new** DefaultListModel();  // Using for alert everything.      public *void* AlertBox(*String* *title*, *String* *description*, *int* *messageType*)      {          JOptionPane.showMessageDialog(null, description, title, messageType);      }  //  Using for exit button      private *void* BTNExitActionPerformed(*java*.*awt*.*event*.*ActionEvent* *evt*) {          System.exit(0); // exit program with exit code 0.  }  private *void* BTNViewDatabaseActionPerformed(*java*.*awt*.*event*.*ActionEvent* *evt*) {//GEN-FIRST:event\_BTNViewDatabaseActionPerformed          try          {              // options when connect to database  *MongoClientOptions* connOptions = MongoClientOptions.builder()                  .connectTimeout(5000) // connection timeout set to 5 ms                  .serverSelectionTimeout(5000) // selection timeout set to 5 ms                  .build();                // address of connection with port              // we using local database  *ServerAddress* connectAddress = **new** ServerAddress("localhost", 27017);                // connect to database  *MongoClient* client = **new** MongoClient(connectAddress, connOptions);                  modelListDocument.clear(); // clear model list before add docs name.              for(*String* docName : client.getDatabaseNames())              {                  modelListDocument.addElement(docName); // loop and add each doc name.              }                // set model to LDocumentList              // then it's will re-render show list.              LDocumentList.setModel(modelListDocument);          }          catch(*Exception* *e*)          {              // alert box error              AlertBox("Something's went wrong!", e.getMessage(), JOptionPane.ERROR\_MESSAGE);          }      } |

1. **การเขียนโปรแกรมแสดงชื่อ Collection ในฐานข้อมูล “BigC” ด้วยภาษา Java โดยมีเงื่อนไขดังต่อไปนี้**
   1. เมื่อทำการเปิดโปรแกรมให้แสดงจอดังต่อไปนี้

|  |  |  |
| --- | --- | --- |
| No. | Action | Detail |
| 1 | Interface Design Show Collection Name |  |
| 2 | ตั้งชื่อไฟล์ | **ShowCollection.java** |
| 3 | รายละเอียด | * Title Bar สดงข้อความ “Show Collection Name” * ปุ่มกดแสดงฐานข้อมูลทั้งหมด * เลือกชื่อฐานข้อมูลจากลิสต์ * กดปุ่มแสดง Collection * ชื่อ Collection ในฐานข้อมูลที่เลือกจะแสดงใน List |
| 4 | หน้าจอเมื่อใช้งานจริง |  |
| 5 | รายละเอียด | ชื่อ Collection ในฐานข้อมูลที่เลือกจะแสดงใน List |

2.2 โค้ดโปรแกรม ShowCollection

* จัดเรียงให้สวยงาม
* มี Comment อธิบายโค้ดเข้าใจง่าย
* ตั้งชื่อตัวแปรให้สอดคล้องกับข้อมูลที่เก็บ ห้าม a1 a2 a3
* นำเฉพาะ method ที่เราเขียนมาใหม่ลงในตาราง

|  |  |
| --- | --- |
| Line No. | File: ShowCollection.java |
| 1  2  3  4  5  6  7 | /\*\* import code here\*\*/  import com.mongodb.MongoClient;  import com.mongodb.MongoClientOptions;  import com.mongodb.ServerAddress;  import com.mongodb.client.MongoDatabase;  import javax.swing.DefaultListModel;  import javax.swing.JOptionPane;  import javax.swing.event.ListSelectionEvent;  // modelListDocument : DefaultListModel      private final *DefaultListModel* modelListDocument = **new** DefaultListModel();      // modelListCollection : DefaultListModel      private final *DefaultListModel* modelListCollection = **new** DefaultListModel();      // selectedDocementName : String      // using when user select document of database      private *String* selectedDocementName;        // using for manage database when connect      private *MongoClient* connection;  /\*\*       \* Creates new form ShowCollectionName       \*/      public ShowCollectionName() {          initComponents();            // after we init all components          // we'll add listener when user select doc in list.          LDocumentList.addListSelectionListener(                  (*ListSelectionEvent* arg0) *->* {                      OnSelectDocument(arg0);                  }          );      }  // Using for alert everything.      public *void* AlertBox(*String* *title*, *String* *description*, *int* *messageType*) {          JOptionPane.showMessageDialog(null, description, title, messageType);      }      public *void* OnSelectDocument(*ListSelectionEvent* *arg0*) {          // Implement later should check it's valid or not null or not empty string!!!          if (!arg0.getValueIsAdjusting()) { // Ensure that this event is not fired twice due to adjustments              selectedDocementName = LDocumentList.getSelectedValue();              System.out.println("selectedDocementName=" + selectedDocementName);          }      }  private *void* BTNViewDatabaseActionPerformed(*java*.*awt*.*event*.*ActionEvent* *evt*) {//GEN-FIRST:event\_BTNViewDatabaseActionPerformed          try {              // options when connect to database  *MongoClientOptions* connOptions = MongoClientOptions.builder()                      .connectTimeout(5000) // connection timeout set to 5 ms                      .serverSelectionTimeout(5000) // selection timeout set to 5 ms                      .build();              // address of connection with port              // we using local database  *ServerAddress* connectAddress = **new** ServerAddress("localhost", 27017);              // connect to database              connection = **new** MongoClient(connectAddress, connOptions);              modelListDocument.clear(); // clear model list before add docs name.              for (*String* docName : connection.getDatabaseNames()) {                  modelListDocument.addElement(docName); // loop and add each doc name.              }              // set model to LDocumentList              // then it's will re-render show list.              LDocumentList.setModel(modelListDocument);          } catch (*Exception* *e*) {              // alert box error              AlertBox("Something's went wrong!", e.getMessage(), JOptionPane.ERROR\_MESSAGE);          }      }//GEN-LAST:event\_BTNViewDatabaseActionPerformed      private *void* BTNExitActionPerformed(*java*.*awt*.*event*.*ActionEvent* *evt*) {//GEN-FIRST:event\_BTNExitActionPerformed          System.exit(0); // exit program with exit code 0.      }//GEN-LAST:event\_BTNExitActionPerformed      private *void* BTNViewCollectionsActionPerformed(*java*.*awt*.*event*.*ActionEvent* *evt*) {//GEN-FIRST:event\_BTNViewCollectionsActionPerformed          try          {              // get database of selection docs  *MongoDatabase* database = connection.getDatabase(selectedDocementName);              // loop each collection name for add to model.              modelListCollection.clear();              for(*String* collectionName : database.listCollectionNames())              {                 modelListCollection.addElement(collectionName);              }                // set model to display.              LCollectionList.setModel(modelListCollection);          }          catch (*Exception* *e*)          {              // alert box error              AlertBox("Something's went wrong!", e.getMessage(), JOptionPane.ERROR\_MESSAGE);          }      }//GEN-LAST:event\_BTNViewCollectionsActionPerformed |

1. การเขียนโปรแกรม Main Menu เพื่อเชื่อมโยงหน้าจอทุกหน้าจอ

|  |  |  |
| --- | --- | --- |
| 1 | **File name**: MainMenu.java |  |
|  | รายละเอียด เมนูหลัก | - Title bar “User Management System Development for BigC Company”  - ฐานข้อมูล  - จัดการข้อมูลผู้ใช้ |
| 2 | หน้าจอเมื่อใช้งานจริง |  |
|  | รายละเอียด  เมนูย่อย | 1. “แสดงชื่อฐานข้อมูล” แสดงหน้าจอแสดงชื่อฐานข้อมูล  2. “แสดงชื่อ Collection” แสดงหน้าจอแสดงชื่อ Collection  3. “Exit” ปิดโปรแกรม |

3.1 โค้ดโปรแกรมหน้าจอ Main Menu

* จัดเรียงให้สวยงาม
* มี Comment อธิบายโค้ดเข้าใจง่าย
* ตั้งชื่อตัวแปรให้สอดคล้องกับข้อมูลที่เก็บ
* นำเฉพาะ method ที่เราเขียนมาใหม่ลงในตาราง

|  |  |
| --- | --- |
| Line No. | File: MainMenu.java |
| 1  2  3  4  5  6  7 | /\*\* import code here\*\*/      public Management() {          initComponents();          MDatabase.setText("ฐานข้อมูล");          MDatabase.getItem(0).setText("แสดงชื่อฐานข้อมูล");          MDatabase.getItem(1).setText("แสดงชื่อ Collection");          MDatabase.getItem(2).setText("Exit");          PanelShowDatabases.setVisible(false);          PanelShowCollections.setVisible(false);            // Show All Databases listener          MDatabase.getItem(0).addActionListener((e) *->* {              // TODO: Implement later.              System.out.println("Show All Databases listener executed.");              PanelShowCollections.setVisible(false);// hide collections panel              PanelShowDatabases.setVisible(true); // show databases panel          });          // Show All Collections listener          MDatabase.getItem(1).addActionListener((e) *->* {              // TODO: Implement later.              System.out.println("Show All Collections listener executed.");              PanelShowDatabases.setVisible(false); // hide databases panel              PanelShowCollections.setVisible(true); // show collections panel          });          // Exit listener          MDatabase.getItem(2).addActionListener((e) *->* {              // TODO: Implement later.              System.out.println("Exit listener executed.");              System.exit(0); // exit program.          });          MUserManagement.setText("จัดการข้อมูลผู้ใช้");      } |