МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ”ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра інформаційних

систем та мереж

Звіт з лабораторної роботи №3

з дисципліни "Технології програмування в розподілених інформаційних системах"

на тему:

**"Розроблення програми для розподіленого середовища"**

Варіант №24

Виконав:

Студент групи КН-38

Стасишин О.З.

Прийняв:

доцент каф. ІСМ, к.т.н.

Щербак С. С.

ЛЬВІВ – 2018

**Мета роботи:** придбання і закріплення навиків в роботі з Firebase, в інтеграції даних, в модульному програмуванні.

**Завдання на роботу:**

Для заданої прикладної області розробити опис об'єктів цієї області. Розробити додаток на основі БД Firebase, що реалізують базові операції над цими об'єктами, зокрема:

текстове введення-виведення;

присвоювання;

видалення;

редагування

задання константних значень;

порівняння (не менше 2-х типів);

пошук за шаблоном.

БД повинна зберігати не менше 10 значень конкретних об’єктів. Додаток повинен забезпечувати введення з клавіатури та зчитування з файлу початкових даних в БД та відображення даних з підтримкою сортування по алфавітному і по числовому параметру.

**Індивідуальне завдання:**



**AddDataBase.java**

package addDataBase;  
  
import javafx.fxml.FXMLLoader;  
import javafx.scene.Scene;  
import javafx.scene.layout.Pane;  
import javafx.stage.Stage;  
  
import java.io.IOException;  
  
public class AddDataBase {  
 public AddDataBase() throws IOException {  
 FXMLLoader loader = new FXMLLoader(getClass().getResource("/AddDataBase.fxml"));  
 Pane load = (Pane) loader.load();  
 Stage stage = new Stage();  
 stage.setTitle("Додавання до бази даних");  
 Scene scene = new Scene(load);  
 stage.setScene(scene);  
 stage.show();  
 }  
  
}

**ControllerAddDataBase.java**

package addDataBase;  
  
import com.google.api.core.ApiFuture;  
import com.google.auth.oauth2.GoogleCredentials;  
import com.google.cloud.firestore.DocumentReference;  
import com.google.cloud.firestore.Firestore;  
  
import com.google.cloud.firestore.WriteResult;  
import com.google.firebase.FirebaseApp;  
import com.google.firebase.FirebaseOptions;  
import com.google.firebase.cloud.FirestoreClient;  
  
import javafx.event.ActionEvent;  
import javafx.fxml.FXML;  
import javafx.scene.control.Button;  
import javafx.scene.control.TextField;  
import javafx.stage.Stage;  
  
import java.io.FileInputStream;  
import java.io.IOException;  
import java.io.InputStream;  
import java.util.HashMap;  
import java.util.LinkedHashMap;  
import java.util.Map;  
import java.util.Scanner;  
  
public class ControllerAddDataBase {  
 @FXML  
 private TextField display1;  
 @FXML  
 private TextField display2;  
 @FXML  
 private TextField display3;  
 @FXML  
 private TextField display4;  
 @FXML  
 private TextField display5;  
  
 @FXML  
 private Button addData;  
  
 public void addData(ActionEvent actionEvent) throws Exception {  
 InputStream serviceAccount = new FileInputStream("C:\\Users\\User\\Downloads\\TPRIS-fff677a0ea61.json");  
 GoogleCredentials credentials = GoogleCredentials.*fromStream*(serviceAccount);  
 FirebaseOptions options = new FirebaseOptions.Builder()  
 .setCredentials(credentials)  
 .build();  
 FirebaseApp.*initializeApp*(options);  
  
 Firestore db = FirestoreClient.*getFirestore*();  
  
 DocumentReference docRef = db.collection("EOM").document(display1.getText());  
 Map<String , String> data = new LinkedHashMap<String, String>();  
 data.put("Type" , display2.getText());  
 data.put("Frequency" , display3.getText());  
 data.put("Capacity of RAM" , display4.getText());  
 data.put("Hard disk capacity" , display5.getText());  
 ApiFuture<WriteResult> result = docRef.set(data);  
  
 Stage stage = (Stage) addData.getScene().getWindow();  
 stage.close();  
  
  
 }  
}

**ControllerDeleteDataBase.java**

package deleteDataBase;  
  
import com.google.api.core.ApiFuture;  
import com.google.auth.oauth2.GoogleCredentials;  
import com.google.cloud.firestore.DocumentReference;  
import com.google.cloud.firestore.Firestore;  
  
import com.google.cloud.firestore.WriteResult;  
import com.google.firebase.FirebaseApp;  
import com.google.firebase.FirebaseOptions;  
import com.google.firebase.cloud.FirestoreClient;  
  
import javafx.event.ActionEvent;  
import javafx.fxml.FXML;  
import javafx.scene.control.Button;  
import javafx.scene.control.TextField;  
import javafx.stage.Stage;  
  
import java.io.FileInputStream;  
import java.io.IOException;  
import java.io.InputStream;  
import java.util.HashMap;  
import java.util.Map;  
import java.util.Scanner;  
  
public class ControllerDeleteDataBase {  
  
 @FXML  
 private TextField deleteData;  
  
 @FXML  
 private Button deleteDataButton;  
  
  
 public void deleteDataButton(ActionEvent actionEvent) throws Exception {  
 InputStream serviceAccount = new FileInputStream("C:\\Users\\User\\Downloads\\TPRIS-fff677a0ea61.json");  
 GoogleCredentials credentials = GoogleCredentials.*fromStream*(serviceAccount);  
 FirebaseOptions options = new FirebaseOptions.Builder()  
 .setCredentials(credentials)  
 .build();  
 FirebaseApp.*initializeApp*(options);  
  
 Firestore db = FirestoreClient.*getFirestore*();  
  
 ApiFuture<com.google.cloud.firestore.WriteResult> result = db.collection("EOM").document(deleteData.getText()).delete();  
  
 Stage stage = (Stage) deleteDataButton.getScene().getWindow();  
 stage.close();  
 }  
}

**ControllerEditingDataBase.java**

import java.io.InputStream;  
import java.util.HashMap;  
import java.util.Map;  
  
public class ControllerEditingDataBase {  
 @FXML  
 private TextField display1;  
 @FXML  
 private TextField display2;  
 @FXML  
 private TextField display3;  
 @FXML  
 private TextField display4;  
 @FXML  
 private TextField display5;  
  
  
  
 @FXML  
 private Button addData;  
 public void updateData(ActionEvent actionEvent) throws Exception{  
  
 InputStream serviceAccount = new FileInputStream("C:\\Users\\User\\Downloads\\TPRIS-fff677a0ea61.json");  
 GoogleCredentials credentials = GoogleCredentials.*fromStream*(serviceAccount);  
 FirebaseOptions options = new FirebaseOptions.Builder()  
 .setCredentials(credentials)  
 .build();  
 FirebaseApp.*initializeApp*(options);  
  
 Firestore db = FirestoreClient.*getFirestore*();  
  
 DocumentReference docRef = db.collection("EOM").document(display1.getText());  
 Map<String , Object> data = new HashMap<String, Object>();  
 data.put("Type" , display2.getText());  
 data.put("Frequency" , display3.getText());  
 data.put("Capacity of RAM" , display4.getText());  
 data.put("Hard disk capacity" , display5.getText());  
 ApiFuture<WriteResult> result = docRef.update(data);  
  
 Stage stage = (Stage) addData.getScene().getWindow();  
 stage.close();  
  
  
 }  
  
}

**Controller.java**

package sample;  
  
import EditingDataBase.EditingDataBase;  
import addDataBase.AddDataBase;  
import deleteDataBase.DeleteDataBase;  
import javafx.event.ActionEvent;  
import searchDataBase.SearchDataBase;  
import showDataBase.ShowDataBase;  
  
public class Controller {  
  
  
 public void addDataBase(ActionEvent actionEvent) {  
 try{  
 new AddDataBase();  
  
 }catch(Exception e){  
 e.printStackTrace();  
 }  
 }  
  
 public void deleteDataBase(ActionEvent actionEvent) {  
  
 try{  
 new DeleteDataBase();  
  
 }catch (Exception e){  
 e.printStackTrace();  
 }  
 }  
  
 public void editingDataBase(ActionEvent actionEvent) {  
 try{  
 new EditingDataBase();  
 }catch (Exception e){  
 e.printStackTrace();  
 }  
  
 }  
  
 public void searchDataBase(ActionEvent actionEvent) {  
 try{  
 new SearchDataBase();  
 }catch (Exception e){  
 e.printStackTrace();  
 }  
 }  
  
 public void comparsionDataBase(ActionEvent actionEvent) {  
 }  
  
 public void showDataBase(ActionEvent actionEvent) {  
 try{  
  
 new ShowDataBase();  
  
 }catch(Exception e){  
 e.printStackTrace();  
 }  
 }  
}

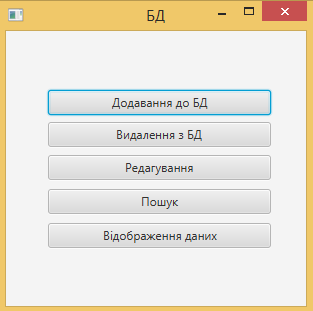
**SearchDataBase.java**

package searchDataBase;  
  
import com.google.api.core.ApiFuture;  
import com.google.auth.oauth2.GoogleCredentials;  
import com.google.cloud.firestore.Firestore;  
import com.google.cloud.firestore.QueryDocumentSnapshot;  
import com.google.cloud.firestore.QuerySnapshot;  
import com.google.firebase.FirebaseApp;  
import com.google.firebase.FirebaseOptions;  
import com.google.firebase.cloud.FirestoreClient;  
import javafx.event.ActionEvent;  
import javafx.fxml.FXML;  
import javafx.fxml.FXMLLoader;  
import javafx.scene.Node;  
import javafx.scene.Parent;  
import javafx.scene.Scene;  
import javafx.scene.control.Label;  
import javafx.scene.control.TextField;  
import javafx.stage.Modality;  
import javafx.stage.Stage;  
  
  
import java.io.FileInputStream;  
import java.io.InputStream;  
import java.util.List;  
  
  
public class ControllerSearchDataBase {  
  
 @FXML  
 private TextField searchData;  
  
 @FXML  
 private Label label1;  
 @FXML  
 private Label label2;  
 @FXML  
 private Label label3;  
 @FXML  
 private Label label4;  
 @FXML  
 private Label label5;  
  
 public void searchDataButton(ActionEvent actionEvent) throws Exception {  
  
  
 InputStream serviceAccount = new FileInputStream("C:\\Users\\User\\Downloads\\TPRIS-fff677a0ea61.json");  
 GoogleCredentials credentials = GoogleCredentials.*fromStream*(serviceAccount);  
 FirebaseOptions options = new FirebaseOptions.Builder()  
 .setCredentials(credentials)  
 .build();  
 FirebaseApp.*initializeApp*(options);  
  
 Firestore db = FirestoreClient.*getFirestore*();  
  
 ApiFuture<QuerySnapshot> future = db.collection("EOM").get();  
  
 List<QueryDocumentSnapshot> documents = future.get().getDocuments();  
 for(QueryDocumentSnapshot document : documents) {  
 /\*eomData.add(new showDataBase.EOM(document.getId(), document.getString("Type"),  
 document.getString("Frequency"), document.getString("Capacity of RAM"), document.getString("Hard disk capacity")));\*/  
 if(searchData.getText().equals(document.getId())){  
 label1.setText(document.getId());  
 label2.setText(document.getString("Type"));  
 label3.setText(document.getString("Frequency"));  
 label4.setText(document.getString("Capacity of RAM"));  
 label5.setText(document.getString("Hard disk capacity"));  
  
  
 }  
  
 }  
  
  
 }

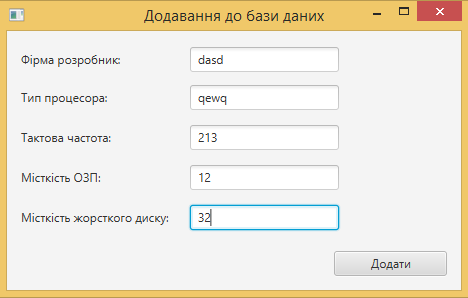
**ControllerShowDataBase.java**

import java.util.ResourceBundle;  
  
  
public class ControllerShowDataBase implements Initializable {  
 private ObservableList<EOM> eomData = FXCollections.*observableArrayList*();  
  
 @FXML  
 private TableView<EOM> tableEOM;  
  
 @FXML  
 private TableColumn<EOM, String> company;  
 @FXML  
 private TableColumn<EOM, String> Type;  
 @FXML  
 private TableColumn<EOM, String> frequency;  
 @FXML  
 private TableColumn<EOM, String> RAM;  
 @FXML  
 private TableColumn<EOM, String> hardDrive;  
  
  
 @FXML  
 public void initialize(URL location, ResourceBundle resources) {  
 try {  
 initData();  
  
 company.setCellValueFactory(new PropertyValueFactory<EOM, String>("company"));  
 Type.setCellValueFactory(new PropertyValueFactory<EOM, String>("typeProcessor"));  
 frequency.setCellValueFactory(new PropertyValueFactory<EOM, String>("frequency"));  
 RAM.setCellValueFactory(new PropertyValueFactory<EOM, String>("RAM"));  
 hardDrive.setCellValueFactory(new PropertyValueFactory<EOM, String>("hardDrive"));  
  
 tableEOM.setItems(eomData);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 private void initData() throws Exception {  
 InputStream serviceAccount = new FileInputStream("C:\\Users\\User\\Downloads\\tpris-a7cca-a1a6eeff232e.json");  
 GoogleCredentials credentials = GoogleCredentials.*fromStream*(serviceAccount);  
 FirebaseOptions options = new FirebaseOptions.Builder()  
 .setCredentials(credentials)  
 .build();  
 FirebaseApp.*initializeApp*(options);  
  
 Firestore db = FirestoreClient.*getFirestore*();  
  
 ApiFuture<QuerySnapshot> future = db.collection("EOM").get();  
  
 List<QueryDocumentSnapshot> documents = future.get().getDocuments();  
 for (QueryDocumentSnapshot document : documents) {  
 eomData.add(new EOM(document.getId(), document.getString("Type"),  
 document.getString("Frequency"), document.getString("Capacity of RAM"), document.getString("Hard disk capacity")));  
  
 }  
  
 }  
  
 public class EOM {  
  
 private String company;  
 private String typeProcessor;  
 private String frequency;  
 private String RAM;  
 private String hardDrive;  
  
 public EOM(String company, String typeProcessor, String frequency, String RAM, String hardDrive) {  
 this.company = company;  
 this.typeProcessor = typeProcessor;  
 this.frequency = frequency;  
 this.RAM = RAM;  
 this.hardDrive = hardDrive;  
 }  
  
 public EOM() {  
 }  
  
 public String getCompany() {  
 return company;  
 }  
  
 public void setCompany(String company) {  
 this.company = company;  
 }  
  
 public String getTypeProcessor() {  
 return typeProcessor;  
 }  
  
 public void setTypeProcessor(String typeProcessor) {  
 this.typeProcessor = typeProcessor;  
 }  
  
 public String getFrequency() {  
 return frequency;  
 }  
  
 public void setFrequency(String frequency) {  
 this.frequency = frequency;  
 }  
  
 public String getRAM() {  
 return RAM;  
 }  
  
 public void setRAM(String RAM) {  
 this.RAM = RAM;  
 }  
  
 public String getHardDrive() {  
 return hardDrive;  
 }  
  
 public void setHardDrive(String hardDrive) {  
 this.hardDrive = hardDrive;  
 }  
  
 }  
}

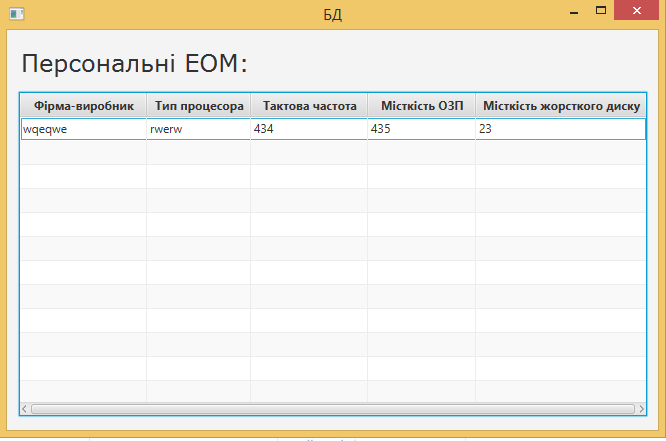
**Результати виконання програми:**



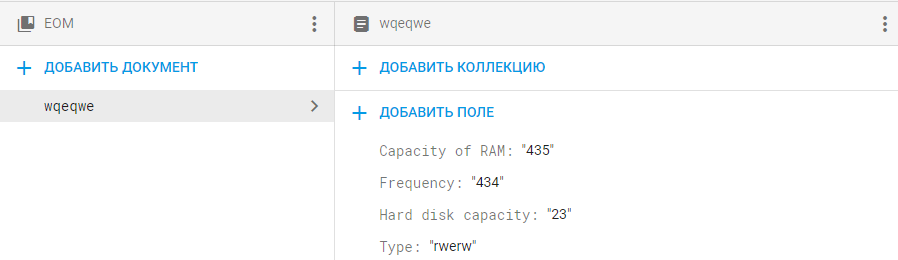
*Рис. 1. Графічний інтерфейс користувача*



*Рис. 2. Додавання в базу даних*



*Рис. 4. Виведення даних*



*Рис. 5. База даних*

**Висновок:** програма призначена для роботи з Firebase. Програма реалізує усі необхідні завдання. На цій лабораторній роботі було придбано і закріплено навики в роботі з Firebase, в інтеграції даних, в модульному програмуванні. Програма написана на мові Java у середовищі Intellij IDEA.