Розробник:

Огонькова Наталія Максимівна

Група КІТ-118б

Варіант №11

**Лабораторна робота № 16**

Розробка графічного інтерфейсу користувача

**Мета:**

* Придбання навичок використання засобів клієнтських технологій (Client Technologies) платформи Java SE**.**

**Вимоги:**

1. Розробити графічний інтерфейс користувача для програми рішення попередньої лабораторної роботи з використанням засобів JavaFX.
2. ОПИС ПРОГРАМИ
   1. Опис змінних

**private** String name;

**private** String unit;

**private** **int** count;

**private** **int** unit\_price;

**private** String data\_of\_receipt;

**private** String attribute;

**private** String value;

* 1. Ієрархія та структура класів

**class** RunProgram – точка входу в програму

**class** Main – містить об’єкт класу Container

**class** Shake – клас, з функціями анімації

**class** MyButtonSkin – клас, з функціями для кнопок

**class** AboutMenu – клас, що містить функції для вікна з меню

**class** Clear – клас, з функціями очищення контейнеру

**class** ControllerWelcome – клас, з функціями для стартового вікна

**class** Error – клас, з функціями для вікна з попередженням про помилку

**class** generateProducts – клас, с функціями для вікна введення даних

**class** HomeController – клас, з функціями для вікна з меню

**class** Number – клас, з функціями для вікна з кількістю елементів в контейнері

**class** Search – клас, з функціями пошуку необхідних елементів

**class** Show – клас демонстрації усіх даних в контейнері у вигляді таблиці

**class** Until – клас с описом параметризованих методів

**class** Container – клас, що містить функції

**class** Store – клас, що містить дані, введені користувачем

* 1. Текст програми

**class** RunProgram

package ua.khpi.oop.Ohonkova16.App;  
  
import javafx.application.Application;  
import javafx.stage.Stage;  
import ua.khpi.oop.Ohonkova16.Util.Util;  
  
  
public class RunProgram extends Application {  
  
 public static void main(String[] args) {  
 *launch*(args);  
 }  
  
 @Override  
 public void start(Stage primaryStage) {  
 Util.*createNewScene*("welcome.fxml", "Store");  
 }  
}

**class** Main

package ua.khpi.oop.Ohonkova16.App;  
  
import ua.khpi.oop.Ohonkova16.Container.Container;  
  
public class Main {  
 public static Container *container* = new Container();  
}

**class** Shake

package ua.khpi.oop.Ohonkova16.Animation;  
  
import javafx.animation.TranslateTransition;  
import javafx.scene.Node;  
import javafx.util.Duration;  
  
public class Shake {  
  
 private TranslateTransition transition;  
  
 public Shake(Node node) {  
 transition = new TranslateTransition(Duration.*millis*(100), node);  
 transition.setFromX(0f);  
 transition.setByX(10f);  
 transition.setCycleCount(3);  
 transition.setAutoReverse(true);  
 }  
  
 public void playAnim() {  
 transition.playFromStart();  
 }  
  
}

**class** MyButtonSkin

package ua.khpi.oop.Ohonkova16.Animation;  
  
import com.sun.javafx.scene.control.skin.ButtonSkin;  
import javafx.animation.FadeTransition;  
import javafx.scene.control.Button;  
import javafx.util.Duration;  
  
public class MyButtonSkin extends ButtonSkin {  
  
 public MyButtonSkin(Button control) {  
 super(control);  
  
 final FadeTransition fadeIn = new FadeTransition(Duration.*millis*(50));  
 fadeIn.setNode(control);  
 fadeIn.setToValue(0.5);  
 control.setOnMouseEntered(e -> fadeIn.playFromStart());  
  
 final FadeTransition fadeOut = new FadeTransition(Duration.*millis*(50));  
 fadeOut.setNode(control);  
 fadeOut.setToValue(1);  
 control.setOnMouseExited(e -> fadeOut.playFromStart());  
  
 control.setOpacity(1);  
 }  
  
}

**class** AboutMenu

package ua.khpi.oop.Ohonkova16.Controllers;  
  
  
import javafx.fxml.FXML;  
import javafx.scene.control.Button;  
import ua.khpi.oop.Ohonkova16.Animation.MyButtonSkin;  
  
public class AboutMenu {  
  
 @FXML  
 private Button okButton;  
  
 @FXML  
 void initialize() {  
 okButton.setSkin(new MyButtonSkin(okButton));  
  
 okButton.setOnAction(event -> okButton.getScene().getWindow().hide());  
  
 }  
}

**class** Clear

package ua.khpi.oop.Ohonkova16.Controllers;  
  
import java.io.IOException;  
  
import javafx.fxml.FXML;  
import javafx.scene.control.Button;  
import ua.khpi.oop.Ohonkova16.Animation.MyButtonSkin;  
import ua.khpi.oop.Ohonkova16.App.Main;  
import ua.khpi.oop.Ohonkova16.Util.Util;  
  
public class Clear {  
  
 @FXML  
 private Button okButton;  
  
 @FXML  
 private Button noButton;  
  
 @FXML  
 void initialize() {  
 okButton.setSkin(new MyButtonSkin(okButton));  
 noButton.setSkin(new MyButtonSkin(noButton));  
  
 okButton.setOnAction(event -> {  
 Main.*container*.clear();  
 try {  
 Util.*save*();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 okButton.getScene().getWindow().hide();  
 });  
  
 noButton.setOnAction(event -> {  
 noButton.getScene().getWindow().hide();  
 });  
 }  
}

**class** ControllerWelcome

package ua.khpi.oop.Ohonkova16.Controllers;  
  
import javafx.fxml.FXML;  
import javafx.scene.control.Button;  
import ua.khpi.oop.Ohonkova16.Animation.MyButtonSkin;  
import ua.khpi.oop.Ohonkova16.Util.Util;  
  
import java.io.IOException;  
  
public class ControllerWelcome {  
  
 @FXML  
 private Button doItButton;  
  
 @FXML  
 void initialize() {  
 doItButton.setSkin(new MyButtonSkin(doItButton));  
  
 doItButton.setOnAction(event -> {  
 Util.*createNewScene*("home.fxml", "Store");  
 try {  
 Util.*read*();  
 } catch (IOException | ClassNotFoundException e) {  
 e.printStackTrace();  
 }  
 doItButton.getScene().getWindow().hide();  
 });  
  
 }  
}

**class** Error

package ua.khpi.oop.Ohonkova16.Controllers;  
  
import javafx.fxml.FXML;  
import javafx.scene.control.Button;  
import ua.khpi.oop.Ohonkova16.Animation.MyButtonSkin;  
  
public class Error {  
  
 @FXML  
 private Button okButton;  
  
 @FXML  
 void initialize() {  
 okButton.setSkin(new MyButtonSkin(okButton));  
  
 okButton.setOnAction(event -> {  
 okButton.getScene().getWindow().hide();  
 });  
  
 }  
}

**class** generateProducts

package ua.khpi.oop.Ohonkova16.Controllers;  
  
import java.io.IOException;  
import java.util.NoSuchElementException;  
  
import javafx.fxml.FXML;  
import javafx.scene.control.\*;  
import javafx.scene.image.ImageView;  
import ua.khpi.oop.Ohonkova16.Animation.Shake;  
import ua.khpi.oop.Ohonkova16.App.Main;  
import ua.khpi.oop.Ohonkova16.Head.Store;  
import ua.khpi.oop.Ohonkova16.Util.Util;  
  
public class generateProducts {  
  
 @FXML  
 private Menu fileMenu;  
  
 @FXML  
 private MenuItem backMenuItem;  
  
 @FXML  
 private MenuItem exitMenuItem;  
  
 @FXML  
 private MenuItem clearMenuItem;  
  
 @FXML  
 private ImageView backImage;  
  
 @FXML  
 private TextField nameField;  
  
 @FXML  
 private TextField unitField;  
  
 @FXML  
 private TextField countField;  
  
 @FXML  
 private TextField unit\_priceField;  
  
 @FXML  
 private Button saveButton;  
  
 @FXML  
 private TextField data\_of\_receiptField1;  
  
 @FXML  
 private TextField attributeField2;  
  
 @FXML  
 private TextField valueField21;  
  
  
  
 private void shakesField(){  
 Shake shake = new Shake(nameField);  
 Shake shake1 = new Shake(unitField);  
 Shake shake2 = new Shake(countField);  
 Shake shake3 = new Shake(unit\_priceField);  
 Shake shake4 = new Shake(data\_of\_receiptField1);  
 Shake shake5 = new Shake(attributeField2);  
 Shake shake6 = new Shake(valueField21);  
 shake.playAnim();  
 shake1.playAnim();  
 shake2.playAnim();  
 shake3.playAnim();  
 shake4.playAnim();  
 shake5.playAnim();  
 shake6.playAnim();  
 }  
  
 private void errorMessage() {  
 Util.*createNewScene*("error.fxml", "Error");  
 }  
  
  
 private Store generateProducts() {  
 Store recruitment = new Store();  
 try {  
 recruitment.setName(nameField.getText());  
 recruitment.setUnit(unitField.getText());  
 recruitment.setCount(Integer.*parseInt*(countField.getText()));  
 recruitment.setUnitPrice(Integer.*parseInt*(unit\_priceField.getText()));  
 recruitment.setDataOfReceipt(data\_of\_receiptField1.getText());  
 recruitment.setAttribute(attributeField2.getText());  
 recruitment.setValue(valueField21.getText());  
  
 } catch (Exception e) {  
 throw new NoSuchElementException();  
 }  
 return recruitment;  
 }  
  
 @FXML  
 void initialize() {  
  
 backImage.setOnMouseClicked(event -> {  
 Util.*createNewScene*("home.fxml", "Store");  
 backImage.getScene().getWindow().hide();  
 });  
  
 backMenuItem.setOnAction(event -> {  
 Util.*createNewScene*("home.fxml", "Store");  
 backImage.getScene().getWindow().hide();  
 });  
  
 exitMenuItem.setOnAction(event -> {  
 backImage.getScene().getWindow().hide();  
 });  
  
  
 saveButton.setOnAction(event -> {  
 try {  
 Main.*container*.add(generateProducts());  
 } catch (Exception e) {  
 System.*out*.println("Repeat entered:");  
 errorMessage();  
 shakesField();  
 Main.*container*.add(generateProducts());  
 }  
 try {  
 Util.*save*();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 Util.*createNewScene*("home.fxml", "Store");  
 saveButton.getScene().getWindow().hide();  
 });  
  
 clearMenuItem.setOnAction(event -> {  
 nameField.setText("");  
 unitField.setText("");  
 countField.setText("");  
 unit\_priceField.setText("");  
 data\_of\_receiptField1.setText("");  
 attributeField2.setText("");  
 valueField21.setText("");  
 });  
 }  
}

**class** HomeController

package ua.khpi.oop.Ohonkova16.Controllers;  
  
import javafx.fxml.FXML;  
import javafx.scene.control.Button;  
import javafx.scene.control.MenuItem;  
import javafx.scene.image.ImageView;  
import ua.khpi.oop.Ohonkova16.Animation.MyButtonSkin;  
import ua.khpi.oop.Ohonkova16.Util.Util;  
  
public class HomeController {  
 double x = 10;  
 double y = 10;  
 double w = 40;  
 double h = 180;  
 @FXML  
 private MenuItem newMenuItem;  
  
 @FXML  
 private MenuItem showAllMenuItem;  
  
 @FXML  
 private MenuItem exitMenuItem;  
  
 @FXML  
 private MenuItem clearMenuItem;  
  
 @FXML  
 private MenuItem numberMunuItem;  
  
 @FXML  
 private MenuItem AboutMenuItem;  
  
 @FXML  
 private Button newProductButton;  
  
 @FXML  
 private ImageView homeImageIco;  
  
 @FXML  
 private Button showAllProductsButton;  
  
 @FXML  
 private Button clearAllButton;  
  
 @FXML  
 private Button numberOfProductsButton;  
  
 private void addVacancy() {  
 Util.*createNewScene*("generateProducts.fxml", "Store");  
 }  
  
 @FXML  
 private ImageView closeImageView;  
  
 @FXML  
 void initialize() {  
 clearAllButton.setSkin(new MyButtonSkin(clearAllButton));  
 newProductButton.setSkin(new MyButtonSkin(newProductButton));  
 showAllProductsButton.setSkin(new MyButtonSkin(showAllProductsButton));  
 numberOfProductsButton.setSkin(new MyButtonSkin(numberOfProductsButton));  
  
 homeImageIco.setOnMouseClicked(event -> {  
 homeImageIco.getScene().getWindow().hide();  
 Util.*createNewScene*("welcome.fxml", "Store");  
 });  
  
 AboutMenuItem.setOnAction(event -> Util.*createNewScene*("aboutMenu.fxml", "About project"));  
  
 newMenuItem.setOnAction(event -> {  
 newProductButton.getScene().getWindow().hide();  
 addVacancy();  
 });  
  
 closeImageView.setOnMouseClicked(event -> closeImageView.getScene().getWindow().hide());  
  
 newProductButton.setOnAction(event -> {  
 newProductButton.getScene().getWindow().hide();  
 addVacancy();  
 });  
  
 exitMenuItem.setOnAction(event -> closeImageView.getScene().getWindow().hide());  
  
 showAllProductsButton.setOnAction(event -> {  
 Util.*createNewScene*("show.fxml", "Show");  
 showAllProductsButton.getScene().getWindow().hide();  
 });  
  
 clearAllButton.setOnAction(event -> Util.*createNewScene*("clear.fxml", "Clear"));  
  
 clearMenuItem.setOnAction(event -> Util.*createNewScene*("clear.fxml", "Clear"));  
  
 numberOfProductsButton.setOnAction(event -> Util.*createNewScene*("number.fxml", "Number of products"));  
  
 numberMunuItem.setOnAction(event -> Util.*createNewScene*("number.fxml", "Number of products"));  
  
 showAllMenuItem.setOnAction(event -> {  
 Util.*createNewScene*("show.fxml", "Show");  
 showAllProductsButton.getScene().getWindow().hide();  
 });  
  
 }  
}

**class** Number

package ua.khpi.oop.Ohonkova16.Controllers;  
  
import java.net.URL;  
import java.util.ResourceBundle;  
import javafx.fxml.FXML;  
import javafx.scene.control.Button;  
import javafx.scene.control.Label;  
import ua.khpi.oop.Ohonkova16.Animation.MyButtonSkin;  
import ua.khpi.oop.Ohonkova16.App.Main;  
  
public class Number {  
  
 @FXML  
 private ResourceBundle resources;  
  
 @FXML  
 private URL location;  
  
 @FXML  
 private Button okButton;  
  
 @FXML  
 private Label numberLabel;  
  
 @FXML  
 void initialize() {  
 okButton.setSkin(new MyButtonSkin(okButton));  
  
 okButton.setOnAction(event -> {  
 okButton.getScene().getWindow().hide();  
 });  
  
 numberLabel.setText(String.*valueOf*(Main.*container*.size()));  
  
 }  
}

**class** Search

package ua.khpi.oop.Ohonkova16.Controllers;  
  
import javafx.fxml.FXML;  
import javafx.scene.control.Button;  
import javafx.scene.control.Label;  
import ua.khpi.oop.Ohonkova16.Animation.MyButtonSkin;  
import ua.khpi.oop.Ohonkova16.App.Main;  
  
public class Search {  
  
  
 @FXML  
 private Label id;  
  
 @FXML  
 private Label name;  
  
 @FXML  
 private Label unit;  
  
 @FXML  
 private Label count;  
  
 @FXML  
 private Label unit\_price;  
  
 @FXML  
 private Label data\_of\_receipt;  
  
 @FXML  
 private Label attribute;  
  
 @FXML  
 private Label value;  
  
 @FXML  
 private Label error;  
  
 @FXML  
 private Button okButton;  
  
  
 private void show() {  
 if (Show.*getSearchId*().equals("")) {  
 error.setText("No such values!");  
 } else {  
 for (int i = 0; i < Main.*container*.size(); i++) {  
 if (String.*valueOf*(Main.*container*.getElementByIndex(i).getId()).equals(Show.*getSearchId*())) {  
 id.setText(String.*valueOf*(Main.*container*.getElementByIndex(i).getId()));  
 name.setText(String.*valueOf*(Main.*container*.getElementByIndex(i).getName()));  
 unit.setText(String.*valueOf*(Main.*container*.getElementByIndex(i).getUnit()));  
 count.setText(String.*valueOf*(Main.*container*.getElementByIndex(i).getCount()));  
 unit\_price.setText(String.*valueOf*(Main.*container*.getElementByIndex(i).getUnitPrice()));  
 data\_of\_receipt.setText(String.*valueOf*(Main.*container*.getElementByIndex(i).getDataOfReceipt()));  
 attribute.setText(String.*valueOf*(Main.*container*.getElementByIndex(i).getAttribute()));  
 value.setText(String.*valueOf*(Main.*container*.getElementByIndex(i).getValue()));  
 }  
 break;  
 }  
 }  
 }  
  
  
 @FXML  
 void initialize() {  
 show();  
  
 okButton.setSkin(new MyButtonSkin(okButton));  
  
 okButton.setOnAction(event -> {  
 okButton.getScene().getWindow().hide();  
 });  
 }  
}

**class** Show

package ua.khpi.oop.Ohonkova16.Controllers;  
  
import javafx.collections.FXCollections;  
import javafx.collections.ObservableList;  
import javafx.fxml.FXML;  
import javafx.scene.control.Button;  
import javafx.scene.control.TableColumn;  
import javafx.scene.control.TableView;  
import javafx.scene.control.TextField;  
import javafx.scene.control.cell.PropertyValueFactory;  
import javafx.scene.image.ImageView;  
import ua.khpi.oop.Ohonkova16.Animation.MyButtonSkin;  
import ua.khpi.oop.Ohonkova16.Animation.Shake;  
import ua.khpi.oop.Ohonkova16.App.Main;  
import ua.khpi.oop.Ohonkova16.Head.Store;  
import ua.khpi.oop.Ohonkova16.Util.Util;  
  
import java.io.IOException;  
  
public class Show {  
  
 private ObservableList<Store> list = FXCollections.*observableArrayList*();  
  
 @FXML  
 private TableView<Store> tableViewId;  
  
 @FXML  
 private TableColumn<Store, Integer> idProduct;  
  
 @FXML  
 private TableColumn<Store, String> nameId;  
  
 @FXML  
 private TableColumn<Store, String> unitId;  
  
 @FXML  
 private TableColumn<Store, Integer> countId;  
  
 @FXML  
 private TableColumn<Store, Integer> unit\_priceId;  
  
 @FXML  
 private TableColumn<Store, String> data\_of\_receiptId;  
  
 @FXML  
 private TableColumn<Store, String> attributeId;  
  
 @FXML  
 private TableColumn<Store, String> valueId;  
  
 @FXML  
 private ImageView refresh;  
  
 @FXML  
 private Button deleteId;  
  
 @FXML  
 private ImageView back;  
  
 @FXML  
 private TextField idField;  
  
 @FXML  
 private Button searchButton;  
  
 private static String *searchId*;  
  
 public static String getSearchId() {  
 return *searchId*;  
 }  
  
 private void search() {  
 *searchId* = idField.getText();  
 if (*searchId*.equals("")){  
 Shake shake = new Shake(idField);  
 shake.playAnim();  
 }  
 Util.*createNewScene*("search.fxml", "Search");  
 }  
  
 @FXML  
 void initialize() {  
 searchButton.setSkin(new MyButtonSkin(searchButton));  
 deleteId.setSkin(new MyButtonSkin(deleteId));  
  
 initData();  
 idProduct.setCellValueFactory(new PropertyValueFactory<Store, Integer>("id"));  
 nameId.setCellValueFactory(new PropertyValueFactory<Store, String>("name"));  
 unitId.setCellValueFactory(new PropertyValueFactory<Store, String>("unit"));  
 countId.setCellValueFactory(new PropertyValueFactory<Store, Integer>("count"));  
 unit\_priceId.setCellValueFactory(new PropertyValueFactory<Store, Integer>("UnitPrice"));  
 data\_of\_receiptId.setCellValueFactory(new PropertyValueFactory<Store, String>("DataOfReceipt"));  
 attributeId.setCellValueFactory(new PropertyValueFactory<Store, String>("attribute"));  
 valueId.setCellValueFactory(new PropertyValueFactory<Store, String>("value"));  
  
  
 tableViewId.setItems(list);  
  
 deleteId.setOnAction(event -> {  
 Main.*container*.remove(tableViewId.getSelectionModel().getSelectedItem());  
 try {  
 Util.*save*();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 });  
  
 refresh.setOnMouseClicked(event -> {  
 refresh.getScene().getWindow().hide();  
 Util.*createNewScene*("show.fxml", "Show");  
 });  
  
 back.setOnMouseClicked(event -> {  
 Util.*createNewScene*("home.fxml", "Store");  
 back.getScene().getWindow().hide();  
 });  
  
 searchButton.setOnAction(event -> search());  
  
 }  
  
 private void initData() {  
 for (int i = 0; i < Main.*container*.size(); i++) {  
 list.add(Main.*container*.getElementByIndex(i));  
 }  
 }  
}

**class** Until

package ua.khpi.oop.Ohonkova16.Util;  
  
import javafx.fxml.FXMLLoader;  
import javafx.scene.Parent;  
import javafx.scene.Scene;  
import javafx.scene.image.Image;  
import javafx.stage.Stage;  
import ua.khpi.oop.Ohonkova16.App.Main;  
import ua.khpi.oop.Ohonkova16.Container.Container;  
import ua.khpi.oop.Ohonkova16.Controllers.ControllerWelcome;  
  
import java.io.\*;  
  
public class Util {  
  
 public static void save() throws IOException {  
 FileOutputStream outputStream = new FileOutputStream("src/ua/khpi/oop/Ohonkova16/Save/data.bin");  
 ObjectOutputStream objectOutputStream = new ObjectOutputStream(outputStream);  
 objectOutputStream.writeObject(Main.*container*);  
 objectOutputStream.close();  
 }  
  
 public static void read() throws IOException, ClassNotFoundException {  
 FileInputStream fileInputStream = new FileInputStream("src/ua/khpi/oop/Ohonkova16/Save/data.bin");  
 ObjectInputStream objectInputStream = new ObjectInputStream(fileInputStream);  
 Main.*container* = (Container) objectInputStream.readObject();  
 }  
  
 public static void createNewScene(String fxml, String name){  
 FXMLLoader loader = new FXMLLoader();  
 loader.setLocation(ControllerWelcome.class.getResource("/ua/khpi/oop/Ohonkova16/View/" + fxml));  
 try {  
 loader.load();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 Parent root = loader.getRoot();  
 Stage stage = new Stage();  
 stage.getIcons().add(new Image("file:/ua/khpi/oop/Ohonkova16/Assist/customer\_person\_people\_man\_you\_1625.ico"));  
 stage.setTitle(name);  
 stage.setScene(new Scene(root));  
 stage.show();  
 }  
}

**class** Container

package ua.khpi.oop.Ohonkova16.Container;  
  
import ua.khpi.oop.Ohonkova16.Head.Store;  
  
import java.io.\*;  
import java.util.LinkedList;  
  
public class Container implements Serializable {  
  
 LinkedList<Store> linkedList = new LinkedList<>();  
  
 public boolean notEmpty() {  
 return linkedList.size() > 0;  
 }  
  
  
 public void clear() {  
 linkedList.clear();  
 }  
  
  
 public void add(final Store store) {  
 addLast(store);  
 }  
  
  
 public void saveAll() {  
 try {  
 File file = new File("save.txt");  
 if (!file.exists()) {  
 file.createNewFile();  
 }  
 PrintWriter pw = new PrintWriter(file);  
 System.*out*.println();  
 pw.println(linkedList.size());  
 for (int i = 0; i < linkedList.size(); i++) {  
 pw.println(getElementByIndex(i));  
 }  
 pw.close();  
 } catch (IOException e) {  
 System.*out*.println("Error" + e);  
 }  
 }  
  
 public void saveRec() {  
 try {  
 File file = new File("save.txt");  
 if (!file.exists()) {  
 file.createNewFile();  
 }  
 PrintWriter pw = new PrintWriter(file);  
 Store temp;  
 System.*out*.println();  
 pw.println(linkedList.size());  
 for (int i = 0; i < linkedList.size(); i++) {  
 temp = getElementByIndex(i);  
 pw.println(temp.getName());  
 pw.println(temp.getUnit());  
 pw.println(temp.getCount());  
 pw.println(temp.getUnitPrice());  
 pw.println(temp.getDataOfReceipt());  
 pw.println(temp.getAttribute());  
 pw.println(temp.getValue());  
 }  
 pw.close();  
 } catch (IOException e) {  
 System.*out*.println("Error" + e);  
 }  
 }  
  
  
 public void readAll() {  
 try (BufferedReader br = new BufferedReader(new FileReader("save.txt"))) {  
 Object temp;  
 String line;  
 line = br.readLine();  
 int count = Integer.*parseInt*(line);  
 for (int i = 0; i < count; i++) {  
 line = br.readLine();  
 temp = line;  
 add((Store) temp);  
 }  
 } catch (IOException ex) {  
 ex.printStackTrace();  
 }  
 }  
  
  
 public void readRec() throws IOException {  
 BufferedReader br = null;  
 Store temp = new Store();  
 try {  
 br = new BufferedReader(new FileReader("save.txt"));  
 String line;  
 line = br.readLine();  
 int count = Integer.*parseInt*(line);  
 for (int i = 0; i < count; i++) {  
 line = br.readLine();  
 temp.setName(line);  
 line = br.readLine();  
 temp.setUnit(line);  
 line = br.readLine();  
 temp.setCount(Integer.*parseInt*(line));  
 line = br.readLine();  
 temp.setUnitPrice(Integer.*parseInt*(line));  
 line = br.readLine();  
 temp.setDataOfReceipt(line);  
 line = br.readLine();  
 temp.setAttribute(line);  
 line = br.readLine();  
 temp.setValue(line);  
 add(new Store(temp));  
 }  
 } catch (IOException ex) {  
 ex.printStackTrace();  
 } finally {  
 br.close();  
 }  
 }  
  
 public Object[] toArray() {  
 return linkedList.toArray();  
 }  
  
  
 public void addLast(final Store obj) {  
 linkedList.addLast(obj);  
 }  
  
  
 public void addFirst(final Store obj) {  
 linkedList.addFirst(obj);  
 }  
  
  
 public int size() {  
 return linkedList.size();  
 }  
  
 public Store getElementByIndex(final int index) {  
 return linkedList.get(index);  
 }  
  
 public void sortProducts() {  
 linkedList.sort(Store.*compareByProducts*);  
 }  
  
 public void sortPrice() {  
 linkedList.sort(Store.*compareByPrice*);  
 }  
  
 public void sortDate() {  
 linkedList.sort(Store.*compareByDate*);  
 }  
  
  
 public int search(Store obj) {  
 int index = 0;  
 for (Store store : linkedList) {  
 if (store == obj) {  
 return index;  
 }  
 }  
 return -1;  
 }  
  
 public void saveSerializable() {  
 try {  
 ObjectOutputStream oo = new ObjectOutputStream(new FileOutputStream("save.ser"));  
 oo.writeObject(linkedList);  
 oo.close();  
 } catch (IOException ioe) {  
 System.*out*.println("Error");  
 }  
 }  
  
 public void readSerializable() {  
 try {  
 FileInputStream fileInput = new FileInputStream("save.ser");  
 ObjectInputStream objectInput = new ObjectInputStream(fileInput);  
 LinkedList<Store> temp ;  
 temp = (LinkedList<Store>) objectInput.readObject();  
 System.*out*.println(temp);  
 objectInput.close();  
 } catch (Exception exc) {  
 System.*out*.println("Error");  
 }  
 }  
  
 public void show() {  
 int count = 0;  
 for (Store s : linkedList) {  
 System.*out*.println("#" + (++count));  
 System.*out*.println(s.toString());  
 }  
 }  
  
 public void remove(Store recruitment){  
 linkedList.remove(recruitment);  
 }  
  
  
}

**class** Store

package ua.khpi.oop.Ohonkova16.Head;  
  
import java.io.Serializable;  
import java.util.Comparator;  
import java.util.InputMismatchException;  
import java.util.Scanner;  
import java.util.regex.Matcher;  
import java.util.regex.Pattern;  
  
public class Store implements Serializable {  
 private String name;  
 private String unit;  
 private int count;  
 private int unit\_price;  
  
 private String data\_of\_receipt;  
 private String attribute;  
 private String value;  
 private int id;  
  
 public int getId() {  
 return id;  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
  
 public Store() {  
 name = null;  
 unit = null;  
 count = 0;  
 unit\_price = 0;  
 data\_of\_receipt = null;  
 attribute = null;  
 value = null;  
 id = (int) (Math.*random*() \* 9999);  
 }  
   
 public Store(final Store obj) {  
 name = obj.name;  
 unit = obj.unit;  
 count = obj.count;  
 unit\_price = obj.unit\_price;  
 data\_of\_receipt = obj.data\_of\_receipt;  
 attribute = obj.attribute;  
 value = obj.value;  
 }  
  
 public void setAttribute(String attribute) {  
 if (checkAttribute(attribute)) {  
 this.attribute = attribute;  
 } else {  
 throw new InputMismatchException();  
 }  
 }  
  
  
 private boolean checkAttribute(final String attribute) {  
 Pattern pattern = Pattern.*compile*("[\\s\\w%$+#@^()=!\_\\\\-]\*", Pattern.*CASE\_INSENSITIVE*);  
 Matcher matcher = pattern.matcher(attribute);  
 return matcher.matches();  
 }  
  
  
 public void setName(String name) {  
 if (checkName(name)) {  
 this.name = name;  
 } else {  
 throw new InputMismatchException();  
 }  
 }  
  
 private boolean checkName(final String name) {  
 Pattern pattern = Pattern.*compile*("[a-z]\*-?\\s?", Pattern.*CASE\_INSENSITIVE*);  
 Matcher matcher = pattern.matcher(name);  
 return matcher.matches();  
 }  
  
 public void setUnit(String unit) {  
 if (checkUnit(unit)) {  
 this.unit = unit;  
 } else {  
 throw new InputMismatchException();  
 }  
 }  
  
 private boolean checkUnit(final String unit) {  
 Pattern pattern = Pattern.*compile*("[\\s\\w%$+#@^()=!\_\\\\-]\*", Pattern.*CASE\_INSENSITIVE*);  
 Matcher matcher = pattern.matcher(unit);  
 return matcher.matches();  
 }  
  
 public void setValue(String value) {  
 if (checkValue(value)) {  
 this.value = value;  
 } else {  
 throw new InputMismatchException();  
 }  
 }  
  
 private boolean checkValue(final String value) {  
 Pattern pattern = Pattern.*compile*("[a-z]\*-?\\s?", Pattern.*CASE\_INSENSITIVE*);  
 Matcher matcher = pattern.matcher(value);  
 return matcher.matches();  
 }  
  
 public void setDataOfReceipt(String data\_of\_receipt) {  
 if (checkDataOfReceipt(data\_of\_receipt)) {  
 this.data\_of\_receipt = data\_of\_receipt;  
 } else {  
 throw new InputMismatchException();  
 }  
 }  
  
 private boolean checkDataOfReceipt(final String data\_of\_receipt) {  
 Pattern pattern = Pattern.*compile*("[0-9]{2}.[0-9]{2}.[0-9]{4}");  
 Matcher matcher = pattern.matcher(data\_of\_receipt);  
 return matcher.matches();  
 }  
  
  
  
 public void setUnitPrice(int unit\_price) {  
 this.unit\_price = unit\_price;  
 }  
 public void setCount(int count) {  
 this.count = count;  
 }  
  
  
 public String getName() {  
 return name;  
 }  
 public String getUnit() {  
 return unit;  
 }  
 public int getCount() {  
 return count;  
 }  
 public int getUnitPrice() {  
 return unit\_price;  
 }  
 public String getDataOfReceipt() {  
 return data\_of\_receipt;  
 }  
 public String getAttribute() {  
 return attribute;  
 }  
 public String getValue() {  
 return value;  
 }  
  
  
 public void generateProducts() {  
 Scanner scan = new Scanner(System.*in*);  
 Scanner scan2 = new Scanner(System.*in*);  
 int choose = 0;  
 System.*out*.print("Введите название товара: ");  
 name = scan2.nextLine();  
 System.*out*.print("Введите единицу измирения: ");  
 unit = scan2.nextLine();  
 System.*out*.print("Введите количество товара: ");  
 count = scan.nextInt();  
 System.*out*.print("Введите цену за единицу товара: ");  
 unit\_price = scan.nextInt();  
 System.*out*.print("Введите дату получения: ");  
 data\_of\_receipt = scan2.nextLine();  
 System.*out*.print("Свойства: ");  
 attribute = scan2.nextLine();  
 System.*out*.print("Значение: ");  
 value = scan2.nextLine();  
  
 }  
  
  
 @Override  
 public String toString() {  
 return "Name = " + name + "\n" +"Unit = " + unit +"\n" + "Count = "  
 + count + "\n" +"Unit\_price = " + unit\_price + "\n" +"Data\_of\_receipt = " + data\_of\_receipt + "\n" + "Attribute = "  
 + attribute + "\n" +"Value = " + value;  
 }  
  
 public static final Comparator<Store> *compareByProducts* = Comparator.*comparing*(Store::getName);  
 public static final Comparator<Store> *compareByPrice* = Comparator.*comparingInt*(Store::getUnitPrice);  
 public static final Comparator<Store> *compareByDate* = Comparator.*comparing*(Store::getDataOfReceipt);  
   
   
}

1. ВАРІАНТИ ВИКОРИСТАННЯ

Програму можна використовувати для створення контейнеру з даними, що вводить користувач, стосовно теми «магазин».

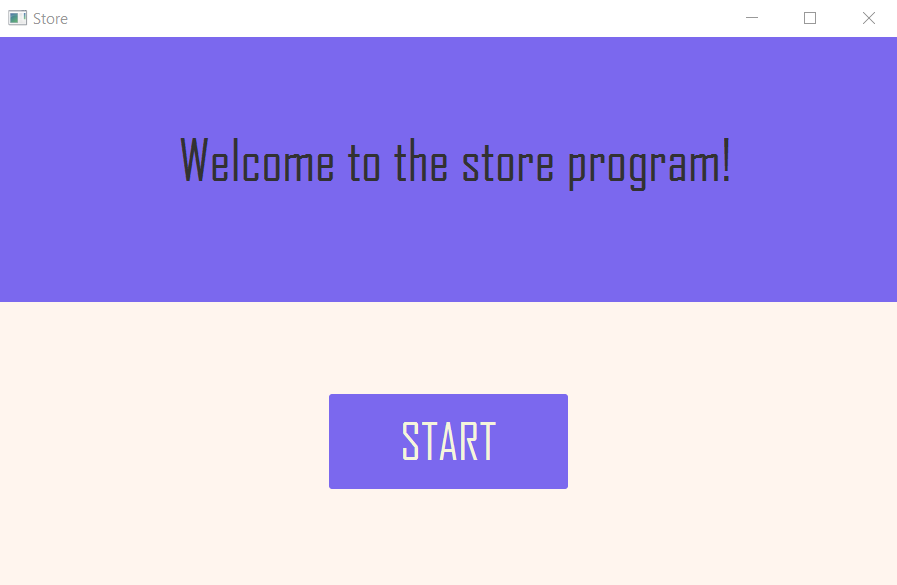


Рисунок 1 – Стартове вікно програми

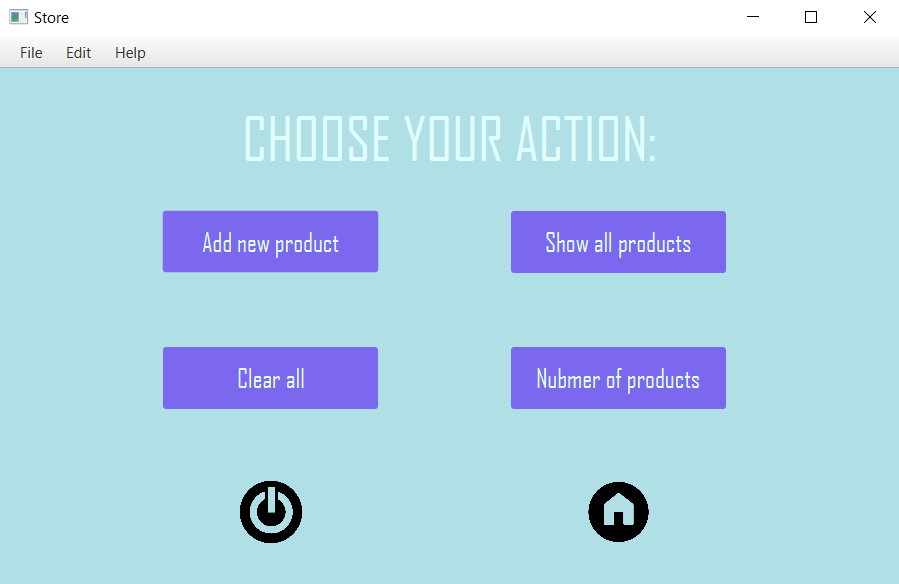


Рисунок 2 – Меню

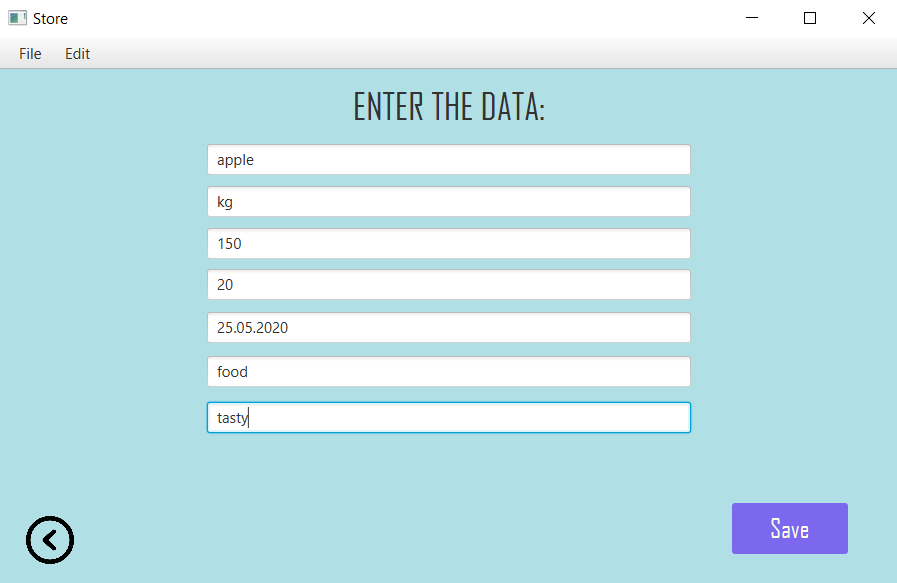


Рисунок 3 – Введення даних

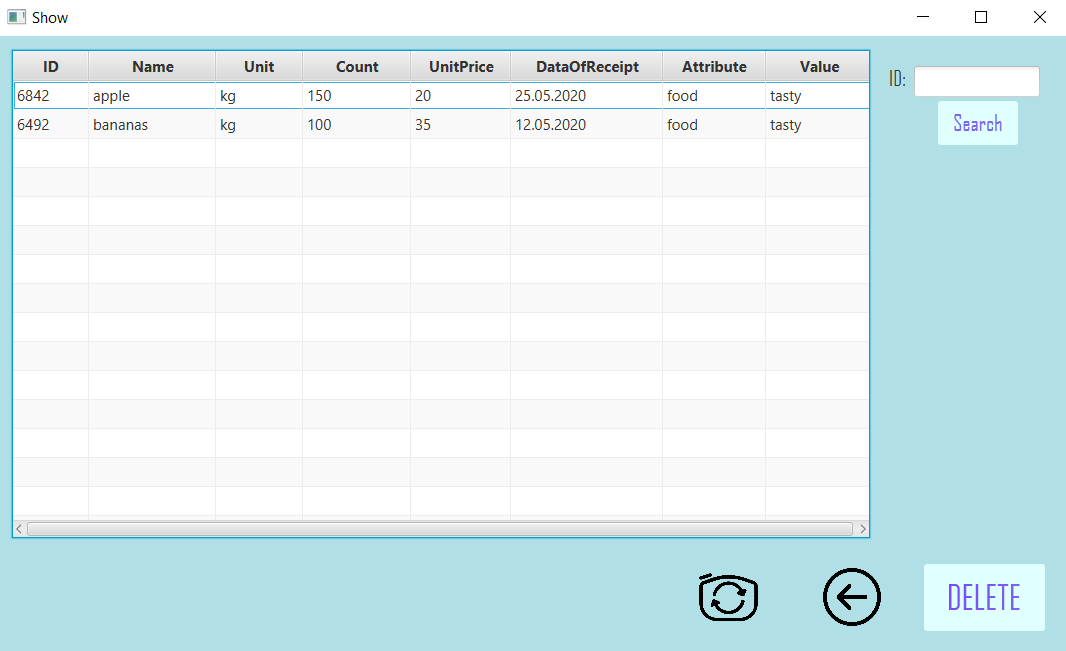


Рисунок 4 – Таблиця даних

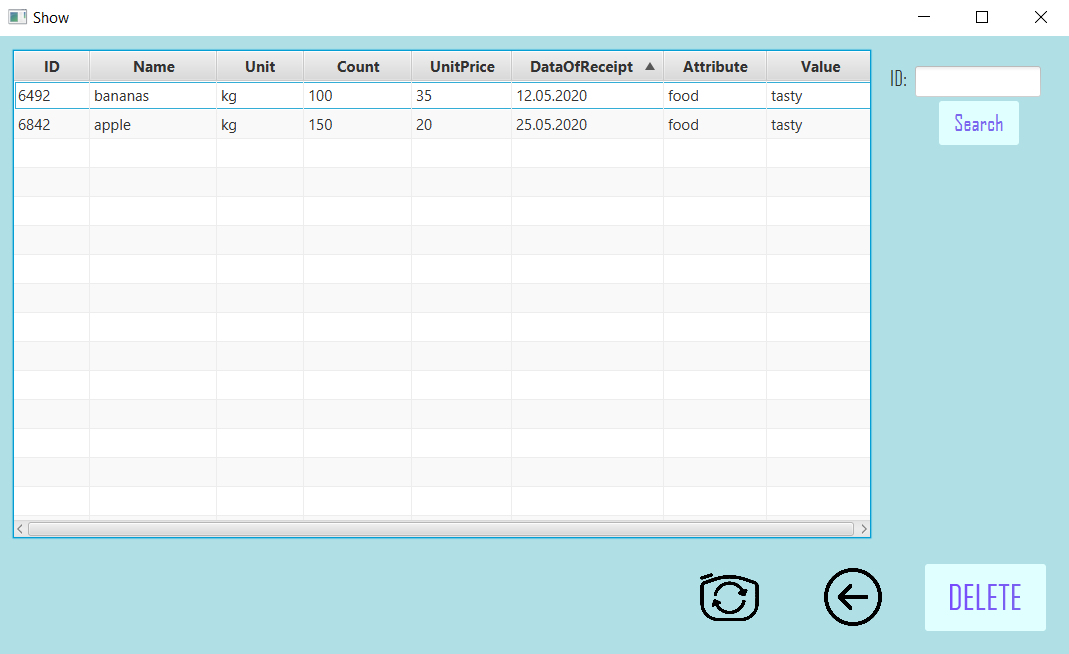


Рисунок 5 – Сортування даних за датою

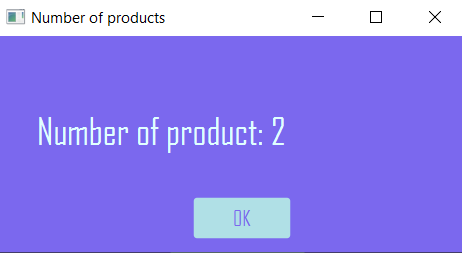


Рисунок 6 – Кількість елементів в контейнері

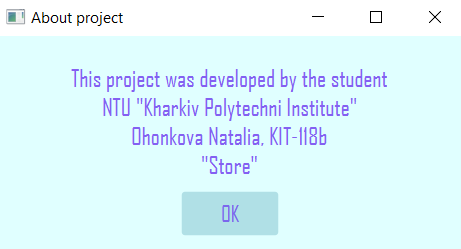


Рисунок 7 – Дані програми



Рисунок 8 – Вікно помилки при введенні даних

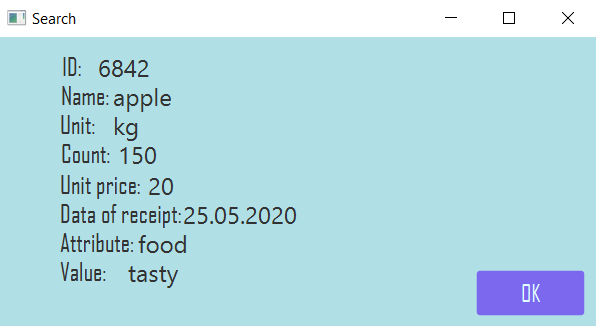


Рисунок 9 – Вікно пошуку по індексу

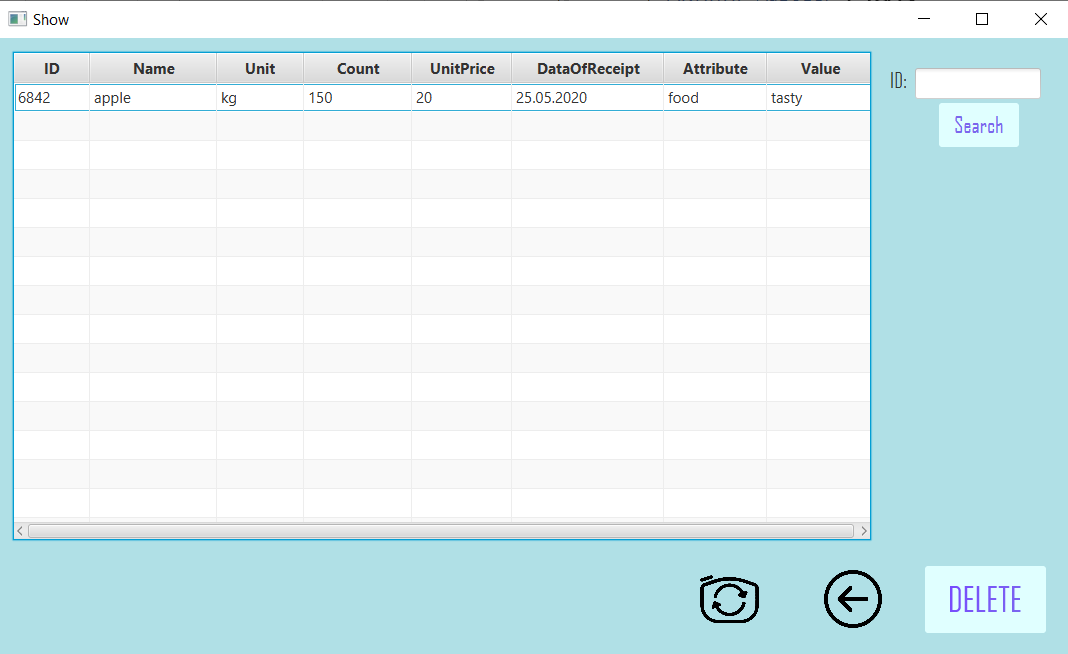


Рисунок 10 – Видалення елементів



Рисунок 11 – Очищення контейнера

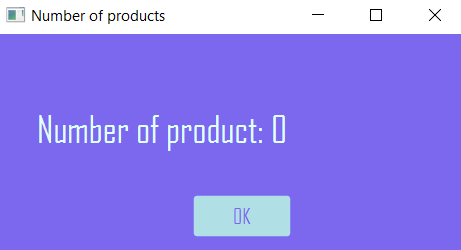


Рисунок 12 – Результат

ВИСНОВОК

При виконанні лабораторної роботи набуто практичних навичок щодо використання засобів клієнтських технологій (Client Technologies) платформи Java SE**.** Розробленографічний інтерфейс користувача у SceneBuilder для програми рішення попередньої лабораторної роботи з використанням засобів JavaFX. Програма виконується без помилок.