МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра систем штучного інтелекту

Лабораторна робота №15

з дисципліни
«Об'єктно-орієнтоване програмування»

Виконав:
студент групи КН-110
Натяглий А.М.
Викладач:
Гасько Р.Т.

Львів – 2018 р.

Тема: Колекції в JavaРозробка графічного інтерфейсу користувача
Мета:
Отримання навичок використання засобів клієнтських технологій (Client Technologies) платформи Java SE .

Код програми:

```
package
ua.lpnuai.oop.natyahlyi15;

import javafx.beans.property.SimpleStringProperty;

import java.text.ParseException;
import java.text.SimpleDateFormat;

public class Product {

    private SimpleStringProperty title;
    private SimpleStringProperty price;
    private SimpleStringProperty quantity;
    private SimpleStringProperty unit;
    private SimpleStringProperty time;

    public Product () {
    }

    public Product (String s1, String s2, String s3, String
s4, String s5) {

        title = new SimpleStringProperty(s1);
        price = new SimpleStringProperty(s2);
        quantity = new SimpleStringProperty(s3);
        unit = new SimpleStringProperty(s4);
        time = new SimpleStringProperty(s5);
    }

    public String getTitle() {
        return title.get();
    }
    public void setTitle(String s) {
        if ((s.matches("^(?!\\s*$).+"))){
            title.set(s);
```

```
        } else {
            title.set("Введіть коректну назву!");
        }


    }


    public String getQuantity() {


        return quantity.get();
    }
    public void setQuantity(String s) {
        if (s.matches("^\\d+")) {
            quantity.set(s);
        } else {
            quantity.set("Введіть коректну кількість!");
        }
    }


    public String getPrice() {


        return price.get();
    }
    public void setPrice(String s) {
        if (s.matches("^\\d+(\\.\\d+)?")) {
            price.set(s + " UAH");
        } else {
            price.set("Введіть коректну ціну!");
        }


    }


    public String getUnit() {


        return unit.get();
    }
    public void setUnit(String s) {
        unit.set(s);
    }


    public String getTime() {
```

```
            return time.get();
        }
        public void setTime(String s) {
            SimpleDateFormat format = new
SimpleDateFormat("dd.MM.yyyy");


            try {
                format.parse(s);
                time.set(s);
            }
            catch(ParseException e){
                time.set("Введіть коректну дату!");
            }




        }


        @Override
        public String toString() {
            if (getQuantity().equals("Введіть коректну
кількість!")){
                return "Назва: " + getTitle() + "\n" +
                        "Ціна: " + getPrice() + "\n" +
                        "Кількість: " + getQuantity() + "\n" +
                        "Дата: " + getTime() + "\n";
            } else {
                return "Назва: " + getTitle() + "\n" +
                        "Ціна: " + getPrice() + "\n" +
                        "Кількість: " + getQuantity() + " " +
getUnit() + "\n" +
                        "Дата: " + getTime() + "\n";
            }
        }
}
```

```
package
ua.lpnuai.oop.natyahlyi15
;
```

```
import javafx.application.Application;
import javafx.scene.input.KeyEvent;
import javafx.stage.Stage;
import javafx.scene.Scene;
import javafx.scene.layout.VBox;
```

```
import javafx.scene.layout.HBox;
import javafx.scene.text.Font;
import javafx.scene.text.FontWeight;
import javafx.scene.text.Text;
import javafx.scene.paint.Color;
import javafx.scene.control.Label;
import javafx.scene.control.TableView;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.control.TableColumn;
import javafx.scene.control.cell.TextFieldTableCell;
import javafx.scene.control.TableColumn.CellEditEvent;
import javafx.scene.control.Button;
import javafx.geometry.Pos;
import javafx.geometry.Insets;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;


import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.List;
import java.util.ArrayList;
import javafx.beans.value.ChangeListener;
import javafx.beans.value.ObservableValue;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;



public class Shop
        extends Application {


    private TableView<Product> table;
    private ObservableList<Product> data;
    private Text actionStatus;


    public static void main(String [] args) {


        Application.launch(args);
    }


    @Override
    public void start(Stage primaryStage) {
```

```
        primaryStage.setTitle("Магазин");


        // Products label
        Label label = new Label("Товари");
        label.setTextFill(Color.DARKBLUE);
        label.setFont(Font.font("Calibri", FontWeight.BOLD,
32));
        HBox labelHb = new HBox();
        labelHb.setAlignment(Pos.CENTER);
        labelHb.getChildren().add(label);


        // Table view, data, columns and properties


        table = new TableView<>();
        data = getInitialTableData();
        table.setItems(data);
        table.setEditable(true);


        TableColumn titleCol = new TableColumn("Назва");
        titleCol.setCellValueFactory(new
PropertyValueFactory<Product, String>("title"));

titleCol.setCellFactory(TextFieldTableCell.forTableColumn());
        titleCol.setOnEditCommit(new
EventHandler<CellEditEvent<Product, String>>() {
            @Override
            public void handle(CellEditEvent<Product, String>
t) {


                ((Product) t.getTableView().getItems().get(
                        t.getTablePosition().getRow())
                ).setTitle(t.getNewValue());
                actionStatus.setText(((Product)
t.getTableView().getItems().get(
                        t.getTablePosition().getRow())
                ).toString());
                table.refresh();
            }
        });


        TableColumn priceCol = new TableColumn("Ціна");
```

```
        priceCol.setCellValueFactory(new
PropertyValueFactory<Product, String>("price"));

priceCol.setCellFactory(TextFieldTableCell.forTableColumn());
        priceCol.setOnEditCommit(new
EventHandler<CellEditEvent<Product, String>>() {
            @Override
            public void handle(CellEditEvent<Product, String>
t) {


                ((Product) t.getTableView().getItems().get(
                        t.getTablePosition().getRow())
                ).setPrice(t.getNewValue());
                actionStatus.setText(((Product)
t.getTableView().getItems().get(
                        t.getTablePosition().getRow())
                ).toString());
                table.refresh();
            }
        });


        TableColumn qCol = new TableColumn("Кількість");
        qCol.setCellValueFactory(new
PropertyValueFactory<Product, String>("quantity"));

qCol.setCellFactory(TextFieldTableCell.forTableColumn());
        qCol.setOnEditCommit(new
EventHandler<CellEditEvent<Product, String>>() {
            @Override
            public void handle(CellEditEvent<Product, String>
t) {


                ((Product) t.getTableView().getItems().get(
                        t.getTablePosition().getRow())
                ).setQuantity(t.getNewValue());
                actionStatus.setText(((Product)
t.getTableView().getItems().get(
                        t.getTablePosition().getRow())
                ).toString());
                table.refresh();
            }
        });


        TableColumn unitCol = new TableColumn("О.В.");
```

```
        unitCol.setCellValueFactory(new
PropertyValueFactory<Product, String>("unit"));

unitCol.setCellFactory(TextFieldTableCell.forTableColumn());
        unitCol.setOnEditCommit(new
EventHandler<CellEditEvent<Product, String>>() {
            @Override
            public void handle(CellEditEvent<Product, String>
t) {


                ((Product) t.getTableView().getItems().get(
                        t.getTablePosition().getRow())
                ).setUnit(t.getNewValue());
                actionStatus.setText(((Product)
t.getTableView().getItems().get(
                        t.getTablePosition().getRow())
                ).toString());
                table.refresh();
            }
        });



        TableColumn timeCol = new TableColumn("Створено");
        timeCol.setCellValueFactory(new
PropertyValueFactory<Product, String>("time"));

timeCol.setCellFactory(TextFieldTableCell.forTableColumn());
        timeCol.setOnEditCommit(new
EventHandler<CellEditEvent<Product, String>>() {
            @Override
            public void handle(CellEditEvent<Product, String>
t) {


                ((Product) t.getTableView().getItems().get(
                        t.getTablePosition().getRow())
                ).setTime(t.getNewValue());
                actionStatus.setText(((Product)
t.getTableView().getItems().get(
                        t.getTablePosition().getRow())
                ).toString());
                table.refresh();
            }
        });
```

```
        table.getColumns().setAll(titleCol, priceCol, qCol,
unitCol, timeCol);
        table.setPrefWidth(700);
        table.setPrefHeight(700);

table.setColumnResizePolicy(TableView.CONSTRAINED_RESIZE_POLICY
);



table.getSelectionModel().selectedIndexProperty().addListener(
                new RowSelectChangeListener());


        // Add and delete buttons
        Button addbtn = new Button("Добавити");
        addbtn.setOnAction(new AddButtonListener());
        Button delbtn = new Button("Видалити");
        delbtn.setOnAction(new DeleteButtonListener());
        HBox buttonHb = new HBox(10);
        buttonHb.setAlignment(Pos.CENTER);
        buttonHb.getChildren().addAll(addbtn, delbtn);


        // Status message text
        actionStatus = new Text();
        actionStatus.setFill(Color.FIREBRICK);


        // Vbox
        VBox vbox = new VBox(20);
        vbox.setPadding(new Insets(25, 25, 25, 25));;
        vbox.getChildren().addAll(labelHb, table, buttonHb,
actionStatus);


        // Scene
        Scene scene = new Scene(vbox, 600, 550); // w x h
        primaryStage.setScene(scene);
        primaryStage.show();
        scene.setOnKeyPressed(new EventHandler<KeyEvent>() {
            @Override
            public void handle(KeyEvent event) {
                switch (event.getCode()) {
                    case ENTER:    table.refresh();
```

```
                }
            }
        });


        // Select the first row
        table.getSelectionModel().select(0);
        Product product =
table.getSelectionModel().getSelectedItem();
        actionStatus.setText(product.toString());


    } // start()


    private class RowSelectChangeListener implements
ChangeListener<Number> {


        @Override
        public void changed(ObservableValue<? extends Number>
ov,
                            Number oldVal, Number newVal) {


            int ix = newVal.intValue();


            if ((ix < 0) || (ix >= data.size())) {


                return; // invalid data
            }


            Product product = data.get(ix);
            actionStatus.setText(product.toString());
            table.refresh();
        }
    }


    private ObservableList<Product> getInitialTableData() {


        List<Product> list = new ArrayList<>();
```

```
        list.add(new Product("Штани", "40 UAH", "10", "шт.",
"12.12.2012"));
        list.add(new Product("Штани", "40 UAH", "10", "шт.",
"12.12.2012"));
        list.add(new Product("Штани", "40 UAH", "10", "шт.",
"12.12.2012"));
        list.add(new Product("Штани", "40 UAH", "10", "шт.",
"12.12.2012"));
        list.add(new Product("Штани", "40 UAH", "10", "шт.",
"12.12.2012"));

        ObservableList<Product> data =
FXCollections.observableList(list);

        return data;
    }

    private class AddButtonListener implements
EventHandler<ActionEvent> {

        @Override
        public void handle(ActionEvent e) {
            String  sdf = new
SimpleDateFormat("dd.MM.yyyy").format(new Date());
            Product product = new Product("...", "...", "...",
"...", sdf);
            data.add(product);
            int row = data.size() - 1;

            // Select the new row
            table.requestFocus();
            table.getSelectionModel().select(row);
            table.getFocusModel().focus(row);

            actionStatus.setText("Заповніть поля новго
продукту!");
        }
    }

    private class DeleteButtonListener implements
EventHandler<ActionEvent> {
```

```
        @Override
        public void handle(ActionEvent e) {

            // Get selected row and delete
            int ix =
table.getSelectionModel().getSelectedIndex();
            Product product = (Product)
table.getSelectionModel().getSelectedItem();
            data.remove(ix);
            actionStatus.setText("Deleted: " +
product.toString());

            // Select a row

            if (table.getItems().size() == 0) {

                actionStatus.setText("No data in table !");
                return;
            }

            if (ix != 0) {

                ix = ix -1;
            }

            table.requestFocus();
            table.getSelectionModel().select(ix);
            table.getFocusModel().focus(ix);
        }
    }
}
```