

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНОМУ  
УНІВЕРСИТЕТУ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

**Кафедра систем штучного інтелекту**

**Лабораторна робота №15  
Розробка графічного інтерфейсу користувача**

**Мета з дисципліни.  
«Об’єктно-орієнтоване програмування»**

**Виконала:**  
студентка групи КН-108  
Гетьман Соломії  
**Викладач:**  
Гасько Р.Т.

Львів – 2018 р.

# Мета

Отримання навичок використання засобів клієнтських технологій (Client Technologies) платформи Java SE

## Задача

Розробити графічний інтерфейс користувача для програми рішення попередньої лабораторної роботи з використанням засобів JavaFX

## Код Програми

```
package
ua.lpnuai.oop.hetman15;

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("Навчальні курси");
    // Products label
    Label label = new Label("Студенти");
    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("Ім'я");
    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("Вступний бал");
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("Номер телефону");
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("Пік вступу");
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("Стасюк Андрій", "108", "190",
"0965494392", "17.05.2018"));
        list.add(new Product("Гетьман Соломія", "108", "190",
"0965494392", "17.05.2018"));
        list.add(new Product("Петро Телішевський", "108", "180",
"0965494392", "17.05.2018"));
        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.setFocusModel().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.setFocusModel().focus(ix);
    }
}
}

```