

## Лабораторна робота №6

**Тема:** Графічні інтерфейси користувача в Java.

**Мета:** Вивчити особливості роботи з інтерфейсами у Java..

**Завдання:** Здобути навички створення та відлагодження проектів обробки даних з використанням графічних інтерфейсів користувача мовою Java.

(max: 5 балів)

### Теоретичні відомості:

Однією з найпопулярніших технологій створення GUI-додатків є JavaFX. JavaFX представляє інструментарій для створення кросплатформових графічних додатків мовою Java. За допомогою JavaFX можна створювати програми для різних операційних систем: Windows, MacOS, Linux і для найрізноманітніших пристроїв: десктопи, смартфони, планшети, вбудовані пристрої, ТВ.

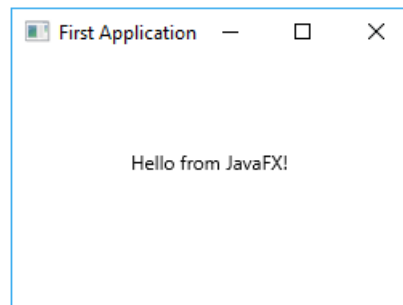
Серед особливостей JavaFX слід зазначити хардверні прискорення, створення GUI за допомогою CSS і XML (FXML), можливість використовувати контролі JavaFX в Swing, а також набір нових красивих контролів, в тому числі для малювання діаграм і 3D.

Найпростіша програма для JavaFX:

```
import javafx.application.Application;
import javafx.stage.Stage;
import javafx.scene.*;

public class Main extends Application{
    public static void main(String[] args) { Application.launch(args); }
    @Override //stage представляє графічний інтерфейс
    public void start(Stage stage) {
        Text text = new Text("Hello from JavaFX!"); //визначення напису
        text.setLayoutY(80); // задання положення напису по осі Y
        text.setLayoutX(100); // задання положення напису по осі X
        Group group = new Group(text); // створення групи елементів
        Scene scene = new Scene(group); //створення сцени
        stage.setScene(scene); //визначення сцени для об'єкта stage
        stage.setTitle("First Application"); // задання заголовку
        stage.setWidth(300); // задання ширини вікна
        stage.setHeight(250); // задання висоти вікна
        stage.show();
    } // scene – контейнер верхнього рівня
}
```

Результат:



Головний клас додатку JavaFX повинен успадковуватись від класу `javafx.application.Application`. Об'єкт цього класу проходить ряд етапів життєвого циклу. Всі ці етапи представлені методами класу ***Application***, які викликаються автоматично середовищем JavaFX:

- `init()` ініціалізує додаток до його запуску; метод не повинен використовуватися для створення графічного інтерфейсу або окремих його частин;

- *start(Stage stage)* призначений для визначення в ньому графічного інтерфейсу;
- *stop()* викликається після закриття програми, наприклад, після того, як користувач натиснув на хрестик в правому верхньому куті.

#### Процес роботи додатку:

1. Запуск виконуючого середовища JavaFX.
2. Виклик конструктора класу, який розширює клас *Application*.
3. Виклик методу *init()*.
4. Виклик методу *start(javafx.stage.Stage)*, в який середовище JavaFX передає створений об'єкт *Stage*. Таким чином, додаток починає працювати.
5. Очікування закриття додатку.
6. При завершенні роботи додатку середовище викликає методу *stop()*.

Всі ці методи можна перевизначити в головному класі програми. Але, як правило, визначається один метод *start()*, в якому визначаються налаштування графічного інтерфейсу.

Основою для створення графічного інтерфейсу в JavaFX є клас *javafx.stage.Stage*. Фактично, він є контейнером, в який поміщаються всі інші компоненти інтерфейсу. Його конкретна реалізація залежить від платформи, на якій запускається додаток. Так, на десктопах це буде окреме графічне вікно, а на мобільних пристроях інтерфейс може представляти весь екран пристрою. Використовуючи свої методи, *Stage* управляє позиціонуванням, розмірами і деякими іншими налаштуваннями вікна програми.

Клас *javafx.scene.Scene* представляє контейнер для всіх графічних елементів всередині об'єкта *Stage* у вигляді графа, який називається *Scene Graph*. Всі вузли цього графа, тобто всі вкладені елементи повинні представляти клас *javafx.scene.Node*. Але **кореневої вузол** цього графа повинен представляти об'єкт класу, який успадковується від *javafx.scene.Parent*. *Parent* – це, фактично, контейнер, що може містити інші елементи.

Для визначення **кореневого вузла** в *Scene* використовується один з **конструкторів об'єкта Scene**. Основні з них:

- *Scene(Parent root)* створює *Scene* з кореневим вузлом *root*;
- *Scene(Parent root, double width, double height)* створює *Scene* з кореневим вузлом *root*, з шириною *width* і висотою *height*;
- *Scene(Parent root, Paint fill)* створює *Scene* з кореневим вузлом *root* і задає фоновий колір;
- *Scene (Parent root, double width, double height, Paint fill)* створює *Scene* з кореневим вузлом *root*, з шириною *width* і висотою *height* і задає фоновий колір.

При цьому всі конструктори приймають в якості першого параметра кореневий вузол, тому при створенні об'єкта *Scene* доведеться визначити і кореневий вузол. Розглянемо код програми

```
import javafx.application.Application;
import javafx.stage.Stage;
import javafx.scene.Scene;
import javafx.scene.Group;
import javafx.scene.layout.FlowPane;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
public class Main extends Application {
    public static void main (String [] args) {
        Application.launch (args);
    }
    @Override
    public void start (Stage stage) {
        Label label = new Label( "Hello"); // текстова мітка
        Button button = new Button( "Button"); // кнопка
```

```

        Group group = new Group(button); // вкладений вузол Group
        FlowPane root = new FlowPane(label, group); // кореневий вузол
        Scene scene = new Scene(root, 300, 150); // створення Scene
        stage.setScene(scene); // задання Scene для Stage
        stage.setTitle("Hello JavaFX");
        stage.show();
    }
}

```

Тут в якості кореневого вузла *root* виступає клас *FlowPane*, який розташовує вкладені елементи підряд і який, як і *Group*, успадкований від класу *Parent.FlowPane* містить об'єкт *Label* (текстова мітка) і об'єкт *Group*. А об'єкт *Group*, в свою чергу, містить об'єкт *Button* (кнопка).

Для взаємодії з користувачем в JavaFX використовується **модель подій**. У цій моделі є джерело події – деякий елемент управління (кнопка, ...), який генерує подію, і є один або кілька слухачів або обробників події, які виконуються при настанні події. Коли елемент управління генерує подію, то обробник обробляє цю подію.

Сцена програми та її граф показані на рисунку 6.1.:

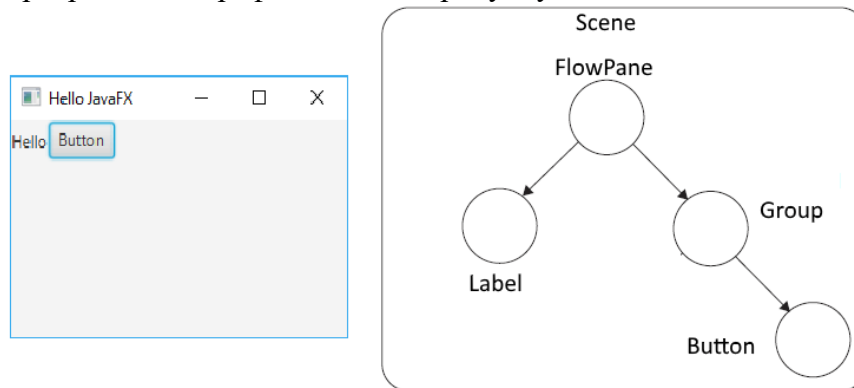


Рис. 6.1. Сцена та її граф

Розглянемо базовий приклад обробки подій:

```

import javafx.application.Application;
import javafx.stage.Stage;
import javafx.scene.Scene;
import javafx.scene.Group;
import javafx.scene.control.Button;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
public class Main extends Application {
    public static void main (String [] args) { Application.launch(args); }
    @Override
    public void start (Stage stage){
        Button btn = new Button();
        btn.setText("Потрібно натиснути!");
        btn.setOnAction(new EventHandler <ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                btn.setText("Ви натиснули на кнопку!");
            }
        });
        Group root = new Group(btn);
        Scene scene = new Scene(root);
        stage.setScene(scene);
    }
}

```

```

        stage.setTitle("Hello JavaFX");
        stage.setWidth(250);
        stage.setHeight(200);
        stage.show();
    }
}

```

Тут створено кнопку, яка представлена об'єктом класу *Button*. При натисканні на кнопку виникає подія натискання, і ця подія обробляється.

Базовим класом для всіх подій є **клас *javafx.event.Event***, який успадкований від класу *java.util.EventObject*. При генерації події, наприклад, при натисканні на кнопку, створюється **об'єкт *Event***, через який передається інформація про подію. В даному випадку при натисканні на кнопку буде генеруватися подія *javafx.event.ActionEvent*.

Для визначення обробника події використовується функціональний **інтерфейс *EventHandler***.

```

public interface EventHandler <T extends Event>
{ void handle (T event); }

```

Інтерфейс *EventHandler* типізується типом, який успадкований від класу *Event* і який, фактично, є подією. У нашому випадку це клас *ActionEvent*.

Для прикріплення обробника події *EventHandler* до події елемента управління застосовується метод *btn.setOnAction()*, в який передається реалізація інтерфейсу *EventHandler*. У методі *handle* визначаються дії, які будуть виконуватися при натисканні на кнопку:

```

btn.setOnAction (new EventHandler <ActionEvent> () {
    @Override
    public void handle(ActionEvent event) {
        btn.setText("Ви натиснули на кнопку!");
    }
});

```

В обробнику вище змінюється надпис на кнопці. Часто замість визначення явної реалізації інтерфейсу *EventHandler* застосовуються лямбда-вирази. Наприклад, можна переписати метод *btn.setOnAction()* наступним чином:

```

btn.setOnAction(e->{btn.setText("Ви натиснули на кнопку!"); });

```

Для спрощення організації та роботи з інтерфейсом в JavaFX використовується мова розмітки FXML, яка створена на основі XML. FXML дозволяє визначити інтерфейс програми декларативним чином подібно до того, як веб-сторінки визначаються за допомогою HTML.

Інтерфейс програми також можна визначити за допомогою **програми *Scene Builder***. З допомогою останньої, користувацький інтерфейс (UI) створюється шляхом перетягування елементів управління з палітри. Ця інформація зберігається у файлі FXML.

До елементів управління в JavaFX належать: кнопки та мітки, *CheckBox*, *ToggleButton*, *RadioButton*, текстові елементи управління, *ScrollPane*, *Slider*, *ListView*, *ComboBox*, *ChoiceBox*, *TreeView*, *ToolTip*, *TableView*.

### **Індивідуальні завдання:**

(Для виконання індивідуальних завдань № варіанта є для 532 групи порядковим номером прізвища студента в списку групи, для 531 – вказаний у додатку (окремий документ). Усі проекти та, за наявності, тести до них завантажити у власні репозиторії на [Git Hub](#). Посилання на репозиторії проектів вказати у звіті та обов'язково долучати скріншоти успішного виконання)

**Написати програму для створення GUI-додатку, що містить дві вкладки: на першій розв'язується задача до завдання 1 згідно вашого варіанту, на другій – завдання 2. Для введення даних використати елементи діалогових вікон. Для виведення результату використати:**

**А) стандартні вікна повідомлень (інформаційне вікно)**

**Б) текстові або написові (label) поля віконної форми.**

**Для створення GUI-додатку можна використати засоби бібліотеки Swing або JavaFX на ваш вибір.**

**Завдання 1: Створити програму для виконання обчислень згідно свого варіанта. Дані контрольного прикладу надано для перевірки коректності обчислень.**

**(2 бала)**

1. Дано два числа  $a$  та  $b$ . Знайти їх середнє арифметичне та середнє геометричне.

Вхідні дані:  $a = 3, b = 27$ .

Вихідні дані:  $sa = 15, sg = 9$ .

2. З початку доби пройшло  $N$  секунд ( $N$  - ціле). Знайти кількість повних хвилин, що пройшли з початку доби.

Вхідні дані:  $N = 550$ .

Вихідні дані:  $m = 9$ .

3. Дано  $a$  та  $b$  – катети прямокутного трикутника. Знайти гіпотенузу  $c$  та його периметр.

Вхідні дані:  $a = 3, b = 4$ .

Вихідні дані:  $c = 5, p = 12$ .

4. Дано довжини ребер прямокутного паралелепіпеда  $a, b$  та  $c$ . Знайти його об'єм та площу поверхні.

Вхідні дані:  $a = 1, b = 2, c = 3$ .

Вихідні дані:  $V = 6, S = 22$ .

5. Дано координати трьох вершин трикутника  $(x_1, y_1), (x_2, y_2)$  та  $(x_3, y_3)$ . Використовуючи формулу обчислення відстані між двома точками на площині знайти його периметр та площу.

Вхідні дані:  $x_1 = 0, y_1 = 0, x_2 = 0, y_2 = 3, x_3 = 4, y_3 = 0$ .

Вихідні дані:  $P = 12, S = 6$ .

6. Дано циліндр  $R$ . Знайти площу його бічної поверхні та об'єм.

Вхідні дані:  $r = 3, h = 10$ .

Вихідні дані:  $S = 188,4, V = 244,92$ .

7. Дано трьохзначне число. Вивести першу та останню цифру даного числа.

Вхідні дані:  $d = 357$ .

Вихідні дані:  $d1 = 3, d2 = 7$ .

8. Дано чотиризначне число. Визначити суму та добуток його цифр.

Вхідні дані:  $d = 1352$ .

Вихідні дані:  $S = 11, D = 30$ .

9. Дано трьохзначне число  $x$ . В ньому закреслили його останню цифру. Після цього цифри, які залишилися поміняли місцями і зліва до них дописали закреслену цифру числа  $x$ . Отримали число 654. Знайти число  $x$ .

Вхідні дані:  $d = 654$ .

Вихідні дані:  $x = 456$ .

10. З початку доби пройшло  $n$  секунд. Визначити скільки годин, хвилин та секунд пройшло з початку доби. Вивести інформацію в наступному вигляді: "З початку доби пройшло ... годин ... хвилин ... секунд".

Вхідні дані:  $n = 3680$ .

*Вихідні дані:* "З початку доби пройшло 1 годин 1 хвилин 20 секунд".

11. Дано значення змінних  $A, B, C$ . Змінити їх значення, перемістивши значення  $A$  в  $B$ ,  $B$  – в  $C$ ,  $C$  – в  $A$ .

*Вхідні дані:*  $A = 4, B = 5, C = 6$ .

*Вихідні дані:*  $A = 6, B = 4, C = 5$ .

12. Знайти значення функції  $y = x^3 - 4x^2 - 5x + 9 + \cos x$  при заданому значенні  $x$ .

*Вхідні дані:*  $x = 0$ .

*Вихідні дані:*  $y = 10$ .

13. Дано  $a$  довжину ребра куба. Знайти об'єм куба та площу його поверхні.

*Вхідні дані:*  $a = 4$ .

*Вихідні дані:*  $V = 64, S = 96$ .

14. Дано два числа  $a$  та  $b$ . Знайти їх середнє арифметичне та середнє геометричне.

*Вхідні дані:*  $a = 3, b = 27$ .

*Вихідні дані:*  $sa = 15, sg = 9$ .

15. З початку доби пройшло  $N$  секунд ( $N$  – ціле). Знайти кількість повних хвилин, що пройшли з початку доби.

*Вхідні дані:*  $N = 550$ .

*Вихідні дані:*  $m = 9$ .

16. Дано  $a$  та  $b$  – катети прямокутного трикутника. Знайти гіпотенузу  $c$  та його периметр.

*Вхідні дані:*  $a = 3, b = 4$ .

*Вихідні дані:*  $c = 5, p = 12$ .

17. Дано довжини ребер прямокутного паралелепіпеда  $a, b$  та  $c$ . Знайти його об'єм та площу поверхні.

*Вхідні дані:*  $a = 1, b = 2, c = 3$ .

*Вихідні дані:*  $V = 6, S = 22$ .

18. Дано координати трьох вершин трикутника  $(x_1, y_1), (x_2, y_2)$  та  $(x_3, y_3)$ . Використовуючи формулу обчислення відстані між двома точками на площині знайти його периметр та площу.

*Вхідні дані:*  $x_1 = 0, y_1 = 0, x_2 = 0, y_2 = 3, x_3 = 4, y_3 = 0$ .

*Вихідні дані:*  $P = 12, S = 6$ .

19. Дано циліндр  $R$ . Знайти площу його бічної поверхні та об'єм.

*Вхідні дані:*  $r = 3, h = 10$ .

*Вихідні дані:*  $S = 188,4, V = 244,92$ .

20. Дано трьохзначне число. Вивести першу та останню цифру даного числа.

*Вхідні дані:*  $d = 357$ .

*Вихідні дані:*  $d1 = 3, d2 = 7$ .

21. Дано чотиризначне число. Визначити суму та добуток його цифр.

*Вхідні дані:*  $d = 1352$ .

*Вихідні дані:*  $S = 11, D = 30$ .

22. Дано трьохзначне число  $x$ . В ньому закреслили його останню цифру. Після цього цифри, які залишилися поміняли місцями і зліва до них дописали закреслену цифру числа  $x$ . Отримали число 654. Знайти число  $x$ .

*Вхідні дані:*  $d = 654$ .

*Вихідні дані:*  $x = 456$ .

23. З початку доби пройшло  $n$  секунд. Визначити скільки годин, хвилин та секунд пройшло з початку доби. Вивести інформацію в наступному вигляді: "З початку доби пройшло ... годин ... хвилин ... секунд".

*Вхідні дані:*  $n = 3680$ .

*Вихідні дані:* "З початку доби пройшло 1 годин 1 хвилин 20 секунд".

24. Знайти значення функції  $y = x^3 - 4x^2 - 5x + 9 + \cos x$  при заданому значенні  $x$ .

Вхідні дані:  $x = 0$ .

Вихідні дані:  $y = 10$ .

25. Дано  $a$  довжину ребра куба. Знайти об'єм куба та площу його поверхні.

Вхідні дані:  $a = 4$ .

Вихідні дані:  $V = 64, S = 96$ .

**Завдання 2: Створити програму згідно свого варіанта, використати конструкцію `switch...case` (це було одне з завдань до лабораторної роботи №1)**

**(3 бала)**

1. Нехай в 9-поверховому будинку по 5 квартир на поверсі. Написати програму, яка за введеним номером квартири повідомляє номер поверху, на якій вона знаходиться.

2. За введеним номером у списку групи (від 1 до 10) вивести прізвище та ім'я одnogрупника.

3. Ввести ціле число з діапазону 10..20. Вивести його назву словом. Наприклад 11 – «одинадцять»

4. Для 9 маршрутів трамваю за веденим їх номером вивести початкову та кінцеву зупинки.

5. Написати програму, яка для 8 одnogрупників виводить їх ім'я за введеним прізвищем.

6. Написати програму, яка за заданим поштовим індексом виводить назву міста, за яким цей індекс закріплений. Наприклад 58000 – «Чернівці». Не менше 10 міст обрати для виконання.

7. Написати програму, яка виводить назву знака зодіака за його порядковим номером.

8. Задано ціле число  $n$ ,  $1 \leq n \leq 24$ , яке вказує порядковий номер гонини в добі. За введеним значенням  $n$  надрукувати назву відповідної години. Наприклад 1 – «перша година».

9. Дано порядковий номер місяця. В залежності від введеного значення вивести пору року.

10. Написати програму, яка в залежності від порядкового номера дня тижня (1,2,...7) виводить на екран його назву (понеділок,...)

11. Написати програму, яка в залежності від порядкового номера місяця (1,2,...12) виводить на екран його назву (січень,...грудень).

12. Написати програму, яка в залежності від порядкового номера кольору у спектрі (1,2,...7) виводить його назву (червоний, помаранчевий, жовтий, зелений, блакитний, синій, фіолетовий)

13. Дано ціле число  $n$  в діапазоні 20 – 69, що визначає вік (в роках). Вивести рядок-опис вказаного віку, забезпечивши правильне співставлення числа зі словом "рік", наприклад: 20 – "20 років", 32 – "32 роки", 41 – "41 рік".

14. Нехай в 9-поверховому будинку по 5 квартир на поверсі. Написати програму, яка за введеним номером квартири повідомляє номер поверху, на якій вона знаходиться.

15. За введеним номером у списку групи (від 1 до 10) вивести прізвище та ім'я одnogрупника.

16. Ввести ціле число з діапазону 10..20. Вивести його назву словом. Наприклад 11 – «одинадцять»

17. Для 9 маршрутів трамваю за веденим їх номером вивести початкову та кінцеву зупинки.

18. Дано натуральне число  $n$  ( $n \leq 100$ ), яке вказує вік людини. Додати до цього числа відповідно слова: „рік”, „роки”, „років”. Наприклад: 1 рік, 12 років, 94 роки.

19. Створити діалогову програму, яка запитує вік користувача і визначає, до якої вікової категорії він належить:

1) від 1 до 10 років – дитина;

2) від 11 до 15 років – підліток;

- 3) від 16 до 20 років – юнак;
  - 4) від 21 до 30 років – молода людина;
  - 5) після 31 року – доросла людина.
20. Задано ціле число  $n$ ,  $1 \leq n \leq 12$ , яке вказує порядковий номер місяця в році. За введеним значенням  $n$  надрукувати назву відповідного місяця.
21. Дано порядковий номер місяця. В залежності від введеного значення вивести пору року.
22. Написати програму, яка в залежності від порядкового номера дня тижня (1,2,...7) виводить на екран його назву (понеділок,...)
23. Написати програму, яка в залежності від порядкового номера місяця (1,2,...12) виводить на екран його назву (січень,...грудень).
24. Написати програму, яка в залежності від порядкового номера кольору у спектрі (1,2,...7) виводить його назву (червоний, помаранчевий, жовтий, зелений, блакитний, синій, фіолетовий)
25. Дано ціле число  $n$  в діапазоні 0 – 50, що визначає вік (в роках). Вивести рядок-опис вказаного віку, забезпечивши правильне співставлення числа зі словом "рік", наприклад: 20 – "20 років", 32 – "32 роки", 41 – "41 рік".