

Візуальний UI JavaFX.

JavaFX

Історія JavaFX почалася в першій половині 2000-х років, коли Кріс Олівер (Chris Oliver) з компанії SeeBeyond, розробив для створення графічних інтерфейсів нову мову F3 (Froms Follows Functions).

У 2005 році SeeBeyond була придбана компанією Sun Microsystems (на той момент розвивала мову Java). F3 перейменували в JavaFX, а Кріс Олівер продовжив роботу над новою платформою вже в рамках компанії Sun. 4 грудня 2008 року вийшов JavaFX 1.0 SDK. Після придбання Sun Microsystems компанією Oracle в 2010 році була анонсована, а в 2011 році вийшла в реліз версія JavaFX 2.0. Скриптова мова, що була основою перших версій була вилучена, а платформа була повністю переписана фактично з нуля і почала працювати на Java.

Історія виникнення і розвитку

Версія JavaFX 2.0 дозволила створювати додатки за допомогою будь-якої мови, який підтримувала JVM. Були додані нові API, інтеграція зі Swing і багато інших речей.

Наступними віхами в розвитку платформи стали версії JavaFX 8 і особливо JavaFX 9, яка вийшла у вересні 2017 року разом з Java 9 і привнесла в платформу модульність. І якщо раніше JavaFX поставлялася разом з Java SE, то зараз JavaFX відокремлена від основної функціональності Java SE і використовується як окремий модуль.

Остання версія фреймворку - JavaFX 15 - вийшла у вересні 2020 року. На даний момент JavaFX являє кращий спосіб для створення графічних додатків за допомогою мови Java, який прийшов на зміну AWT і Swing.

Програма Java FX

```
public static void main(String[] args) {  
    Application.launch(args);  
}
```

launch викликає метод

```
public void start(Stage primaryStage){}
```

primaryStage - вікно- контейнер(залежить від ОС), що передається в метод зворотнього (Callback) виклику для заповнення і запуску

Контроли(Ноди - базовий клас для контейнерів і контролів) в контейнер, Контейнер на сцену, Сцену на Стейдж ()

```
primaryStage.setScene(scene);  
primaryStage.show();
```

Java FX приклад

```
public class HelloApplication extends Application {  
    @Override  
    public void start(Stage primaryStage) {  
        // Елементи інтерфейсу  
        Label label = new Label("Привіт, JavaFX!");  
        Button button = new Button("Натисни мене");  
        // ....  
        // Код, що описує UI вікна....  
    }  
  
    public static void main(String[] args) {  
        launch();  
    }  
}
```

Java FX приклад

```
public class HelloApplication extends Application {  
    @Override  
    public void start(Stage primaryStage) {  
        // Елементи інтерфейсу  
        Label label = new Label("Привіт, JavaFX!");  
        Button button = new Button("Натисни мене");  
        // Обробка події кнопки  
        button.setOnAction(event ->  
            label.setText("Ти натиснув кнопку!"));  
        // Контейнер  
        VBox root = new VBox(10); // 10 — відступ між елементами  
        root.setStyle("-fx-padding: 20; -fx-alignment: center;");  
        root.getChildren().addAll(label, button);  
        //...  
    }  
    //...  
}
```

Java FX приклад

```
public class HelloApplication extends Application {  
    @Override  
    public void start(Stage primaryStage) {  
        // Елементи інтерфейсу  
        // .....  
        // Сцена  
        Scene scene = new Scene(root, 300, 200);  
        // Налаштування сцени та показ  
        primaryStage.setTitle("JavaFX без FXML");  
        primaryStage.setScene(scene);  
        primaryStage.show();  
    }  
    // .....  
}
```

Обробка події кнопки в FX

Лямбда реалізовує такий інтерфейс обробника

```
new EventHandler<ActionEvent>(){@Override  
public void handle(ActionEvent event)}):  
btn.setOnAction((ActionEvent event) -> {  
    System.out.println("Hello World!");  
});
```

Деякі елементи керування:

Label, Button, TextField, TextArea,
PasswordField, List

Java FXML

FXML — підвид мови XML, яка дозволяє **описувати графічний інтерфейс JavaFX** окремо від логіки програми. Забезпечує розділ завдань між UI-дизайнером і програмістом.

FXML-файл описує структуру вікна: розмітку, компоненти, властивості, події.

Переваги:

- Зручна організація коду за рахунок розділення UI та логіки.
- Швидка розробка, зручна для дизайну інтерфейсу.
- Використання візуального редактора FXML (**Scene Builder**).

Java FXML

Загрузка контейнера з XML

Parent root =

```
FXMLLoader.load(getClass().getResource(  
                                "FXMLDocument.fxml"));
```

Обробкою керує контроллер прописується в
FXMLDocument.fxml

елементи і обробники помічаються анотацією
@FXML

імена елементів по *fx:id*

IntelliJ Idea має непоганий Сценбилдер,

Але не вміє писати контролерів

Матеріали з використання Java FX

1. Офіційний сайт проекту: <https://openjfx.io/>,
документація <https://openjfx.io/openjfx-docs/>
2. Короткий огляд <https://www.baeldung.com/javafx>
3. Детальний огляд
<https://www.geeksforgeeks.org/javafx-tutorial/>
4. Приклади використання компонентів
<https://jenkov.com/tutorials/javafx/index.html>
5. Навчальні матеріали
<https://www.javatpoint.com/javafx-tutorial>