

Java FX

Asist. univ. dr. Sergiu Limboi



Ce este JavaFX?

- JavaFX este framework-ul modern pentru GUI (interfețe grafice) în Java.
- Este succesorul lui Swing și folosește concepte moderne:
 - arhitectură bazată pe scene (Scene Graph)
 - binding-uri de proprietăți
 - FXML pentru declararea UI
 - CSS pentru stil
 - controale moderne și layout-uri dinamice
 - suport pentru grafică 2D, animații, multimedia și grafică 3D
 - JavaFX este folosit pentru aplicații desktop, tool-uri interne, dashboard-uri și chiar aplicații educaționale.

Arhitectura JavaFX

- Toate elementele UI într-o aplicație JavaFX formează un **arbore** — „Scene Graph”.

Stage (fereastra aplicației)

└─ Scene (conținutul ferestrei)

└─ Root Node (layout principal: VBox, HBox etc.)

└─ Controls (Button, TableView, TextField...)

└─ Layouts (VBox, HBox, GridPane...)

└─ Nested layouts & nodes

- Stage = fereastra aplicației
- Scene = conținutul grafic
- Scene Graph = structura UI ca arbore
- Nodes = noduri (totul e un nod: butoane, tabele, forme, texte)

CICLUL DE EXECUȚIE AL UNEI APLICAȚII JAVA FX

Programul începe în `main()`

Se execută `launch(args)` → pornește platforma JavaFX

JavaFX creează thread-ul „JavaFX Application Thread”

Se apelează metoda `start(Stage primaryStage)`

Tu creezi scena, încarci FXML, pui UI-ul pe scenă

Se afișează fereastra (`stage.show()`)

FXML

FXML este un limbaj XML care descrie UI-ul:

- *separă complet UI-ul de logică*
- similar cu HTML + controller → ceea ce este excelent pentru MVC
- e încărcat cu FXMLLoader

Exemplu de încărcare:

```
FXMLLoader loader = new FXMLLoader(Main.class.getResource("app.fxml"));
Parent root = loader.load();
Scene scene = new Scene(root);
stage.setScene(scene);
stage.show();
```

Ce face FXML Loader?

- Citește fișierul XML (FXML)
- Creează instanțe pentru tot ceea ce găsește în XML
- Creează un controller (clasa declarată cu `fx:controller`)
- Leagă folosind reflecție controalele:
 - `fx:id="table" ↔ @FXML private TableView table;`
- Apelează **automat** `initialize()`
- Returnează UI-ul construit



Controller

- Controllerul este clasa care „conduce” UI-ul.
- Responsabilități:
 - gestionează evenimente (click, schimbare selecție, taste)
 - actualizează UI-ul
 - comunică cu logica aplicației (service, repository)
 - ține referințe la controale (prin @FXML)
- Ciclul de viață:
 - FXMLLoader creează controllerul
 - Injectează controalele marcate cu @FXML
 - Apelează initialize()
 - Apelează metodele @FXML la evenimente

Layout-uri

- JavaFX are layout-uri automate:
 - VBox — vertical
 - HBox — orizontal
 - BorderPane — sus/centru/jos/stânga/dreapta
 - GridPane — tabelar
 - StackPane — elemente suprapuse

Table View

- TableView funcționează pe baza:
 - ObservableList (lista de rânduri)
 - TableColumn (coloane)
 - CellValueFactory (cum se extrage valoarea unui câmp)
 - CellFactory (cum se afișează celula – opțional)
 - Exemplu: column name → metoda getName() din model:
- `nameCol.setCellValueFactory(new PropertyValueFactory<>("name"));`

Recap

Concept

Scene Graph

FXML

Controller

Binding

ObservableList

PropertyValueFactory

Layout

Stage

Scene

Charts

Explicație scurtă

Structura arborescentă a UI-ului JavaFX

Limbaj XML pentru declararea UI

Logica UI, gestionarea evenimentelor

Legarea automată între date și UI

Listă care notifică automat UI-ul

Extrage valori din model pentru TableView

Organizare vizuală (VBox, HBox, GridPane etc.)

Fereastra principală

Conținutul ferestrei

Grafice moderne bazate pe XYChart

Ce este module-info.java ?

- un „manifest XML” al aplicației, dar scris în Java, care spune:
 - ce module folosești
 - ce pachete vrei să exporti
 - ce pachete sunt deschise pentru reflecție (necesar pentru FXML, TableView etc.)

Exemplu module-info

```
module crud {  
    requires javafx.controls;  
    requires javafx.fxml;  
  
    exports crud;  
    opens crud to javafx.fxml, javafx.base;  
}
```

- **module crud { ... }** Definește numele modulului tău.
- **requires...** Spune compilatorului și JVM: „modulele astea *trebuie* să fie prezente, altfel nu pot porni.”
- **exports crud...** → Spune: „pachetul crud este vizibil altor module.” Fără exports crud,, clasa ta Main NU poate fi folosită ca punct de intrare. JVM literalmente nu poate vedea clasa ta pentru main().
- **opens crud** to javafx.fxml, javafx.base;
 - Asta este **critică** mai ales pentru JavaFX:
 - **Ce face?** Permite accesul prin **reflecție** la pachet.
- Reflection = JavaFX poate:
 - crea controllerul din FXML
 - seta câmpuri private @FXML
 - accesa proprietăți pentru TableView

Erori/probleme la JavaFX

- Java 17 este versiune stabila pentru JavaFX
- Gradle 7.x **NU suportă Java 21.**
- `java.lang.IllegalAccessException: module javafx.fxml cannot access class crud.Controller` → Controller-ul **nu este deschis pentru FXML**. (In `module-info.java` adaugam **`opens crud to javafx.fxml;`**)