Curs 10 PPOO

Conf. univ. dr. Cristian CIUREA

Departamentul de Informatică și Cibernetică Economică

cristian.ciurea@ie.ase.ro

Agenda

- Spring Framework
- Swing versus AWT
- JavaFX

Spring Framework

- **Spring** Framework este o platformă open source pentru simplificarea scrierii aplicațiilor în limbajul Java.
- Deși este folosit în principal pentru platforma Java EE, **Spring** poate fi utilizat pe orice aplicație Java.
- **Spring** este considerat o alternativă la modelul Enterpise JavaBeans (EJB).
- **Spring** a fost lansat în anul 2002 de către Rod Johnson împreună cu cartea sa, *Expert One-on-One: J2EE Design and Development*. În iunie 2003 Spring a intrat sub licența Apache 2.0.

Spring Framework

Pași pentru instalare și utilizare **Spring** Framework:

- Instalare Java Development Kit (JDK)
- Instalare Apache Common Logging API
- Configurare Eclipse
- Configurare biblioteci Spring Framework

Spring Framework

Beneficii utilizare **Spring** Framework:

- Spring permite dezvoltatorilor să dezvolte aplicații enterprise;
- Spring este organizat într-o manieră modulară;
- Spring utilizează unele dintre tehnologiile existente, cum ar fi ORM, JEE, Quartz și JDK, precum și alte tehnologii de vizualizare;
- Testarea unei aplicatii scrise cu Spring este simplă, deoarece codul dependent de mediu este transmis și integrat în cadrul framework-ului;
- Spring este un framework web MVC bine conceput, care oferă o alternativă foarte bună pentru alte framework-uri, cum ar fi Struts sau altele mai puțin populare.

Swing versus AWT

- **Swing** este un set de instrumente GUI pentru Java. Este parte din Oracle Java Foundation Classes (JFC) - un API pentru a furniza o interfață grafică cu utilizatorul (GUI) pentru aplicațiile Java.
- **Swing** a fost dezvoltat pentru a oferi un set mai sofisticat de componente GUI decât cele anterioare din **Abstract Window Toolkit (AWT)**.
- În plus față de componentele clasice, cum ar fi butoanele, checkbox-urile și etichetele, **Swing** oferă mai multe componente avansate, cum ar fi panoul de tab-uri (tabbed panel), scroll panes, tabele și liste.
- Spre deosebire de componentele **AWT**, componentele **Swing** nu sunt implementate prin cod specific platformei. Acestea sunt scrise în întregime în Java și sunt independente de platformă.

Swing versus AWT

- Începând cu versiunea Java 1.1 au apărut Java Foundation Classes (JFC) care includ un set numeros de componente denumite Swing.
- În vederea scrierii aplicatiilor care folosesc aceste componente trebuie importat package-ul **javax.swing.***
- În general, ferestrele care stau la baza aplicatiilor Swing sunt derivate din clasa **JFrame** care contine margini, titlu, butoane pentru închiderea, respectiv minimizarea fereastrei.

Swing versus AWT

- Clasele din pachetul **javax.swing.*** sunt bazate pe arhitectura MVC. Modelul este reprezentarea logică, view-ul este reprezentarea vizuală și controller-ul specifică comportamentul.
- Putem avea mai multe view-uri asociate aceluiași model. De exemplu, putem vizualiza aceleași date sub formă de tabel sau sub formă de grafic.
- În **AWT**, fiecare componentă grafică are o clasă nativă (clasa peer) care comunică cu componenta nativă a sistemului de operare. Componentele din pachetul Swing nu mai au aceste clase peer și pot arăta în mod diferit pe aceeași platformă din cauză că respectă arhitectura MVC.

Swing versus AWT

Pachetul Swing conține două tipuri de componente:

- containere (**JFrame, JApplet, JWindow, JDialog**);
- componente "lightweight" (**JButton, JPanel, Jlist**, etc.).

Containerele reprezintă cadrul în care aceste componente pot exista.

Swing versus AWT

Diferente:

- **Swing**:
  - suportă modelul MVC
  - componentele necesită pachetul **java.swing.***
  - componentele sunt independente de platformă
  - rulează mai rapid decât AWT
  - pluggable look and feel
- **AWT**:
  - nu suportă modelul MVC
  - componentele necesită pachetul **java.awt.***
  - componentele sunt dependente de platformă
  - rulează mai lent
  - native look and feel

JavaFX

- Există mai multe biblioteci de dezvoltare a interfețelor grafice în Java, cea mai cunoscută fiind **Java Swing** și cea mai nouă fiind **JavaFX**.
- **JavaFX** este cea mai nouă implementare a platformei de dezvoltare pentru interfețe GUI de client. Cu toate că este încă într-un stadiu incipient, are avantaje față de vechiul **Swing**.

JavaFX

Avantaje **JavaFX**:

- integrare completă cu Java SE și JDK, începând cu versiunea 7, ceea ce implică faptul că aplicațiile JavaFX vor putea fi dezvoltate și rulate de către orice client cu această versiune de Java;
- inițial JavaFX a fost un limbaj de scripting, dar acum Oracle pune la dispoziție un API pentru dezvoltarea aplicațiilor direct în Java, pentru un mai bun management și reutilizare a codului;
- un nou motor (engine) grafic, denumit Prism, care face uz de accelerarea hardware oferită de GPU-urile moderne, precum și un nou manager de ferestre (Window Toolkit) numit Glass;
- un nou limbaj bazat pe XML, denumit FXML, folosit pentru descrierea interfețelor grafice, astfel încât să nu fie nevoie de recompilarea codului la fiecare modificare;

JavaFX

Avantaje **JavaFX:**

- o componentă care poate afișa conținut web și care poate fi integrată în orice interfață grafică JavaFX;
- o serie de componente noi de interfață, precum grafice, tabele, meniuri și panouri;
- un sistem de a împacheta aplicațiile astfel încât acestea să fie livrate cu toate bibliotecile necesare execuției;
- portabilitate pe Linux, Windows și Mac OS X;
- spre deosebire de Swing, aceeași aplicație poate fi rulată de sine stătător, ca applet, sau ca aplicație de tip Web Start.

JavaFX

- Ca și Java Swing, **JavaFX** se bazează pe o ierarhie de clase care implementează diferite componente și containere ce reprezintă elementele grafice. Analog clasei **JFrame**, în JavaFX, clasa care descrie fereastra principală a unei aplicații este **javafx.stage.Stage**.
- Un obiect de tip **Stage** conține, la un moment dat, o singură scenă (**javafx.scene.Scene**). Această scenă este inițializată dându-i-se dimensiunile scenei și container-ul care conține toate celălalte elemente din fereastră. Acest container este, de regulă, un panou, care pe lângă rolul de container, specifică și modul în care sunt afișate componentele, analog **LayoutManager**-ului din Swing.

JavaFX

- **JavaFX** este o bibliotecă Java pentru proiectarea, crearea, testarea și implementarea aplicațiilor GUI multi-platforma. Este destinat să înlocuiască Swing ca bibliotecă GUI standard pentru Java.
- Biblioteca **JavaFX** este scrisă în Java și este disponibilă pentru limbajele care pot fi executate pe platforma JVM, respectiv **Java**, **Groovy** și **JRuby**. Aplicațiile JavaFX sunt, de asemenea, independente de platformă.

JavaFX

- Există mai multe biblioteci de dezvoltare a interfețelor grafice în Java, cea mai cunoscută fiind **Java Swing** și cea mai nouă fiind **JavaFX**.
- **JavaFX** este cea mai nouă implementare a platformei de dezvoltare pentru interfețe GUI de client. Cu toate că este încă într-un stadiu incipient, are avantaje față de vechiul **Swing**.

JavaFX

Avantaje **JavaFX**:

- integrare completă cu Java SE și JDK, începând cu versiunea 7, ceea ce implică faptul că aplicațiile JavaFX vor putea fi dezvoltate și rulate de către orice client cu această versiune de Java;
- inițial JavaFX a fost un limbaj de scripting, dar acum Oracle pune la dispoziție un API pentru dezvoltarea aplicațiilor direct în Java, pentru un mai bun management și reutilizare a codului;
- un nou motor (engine) grafic, denumit Prism, care face uz de accelerarea hardware oferită de GPU-urile moderne, precum și un nou manager de ferestre (Window Toolkit) numit Glass;
- un nou limbaj bazat pe XML, denumit FXML, folosit pentru descrierea interfețelor grafice, astfel încât să nu fie nevoie de recompilarea codului la fiecare modificare;

JavaFX

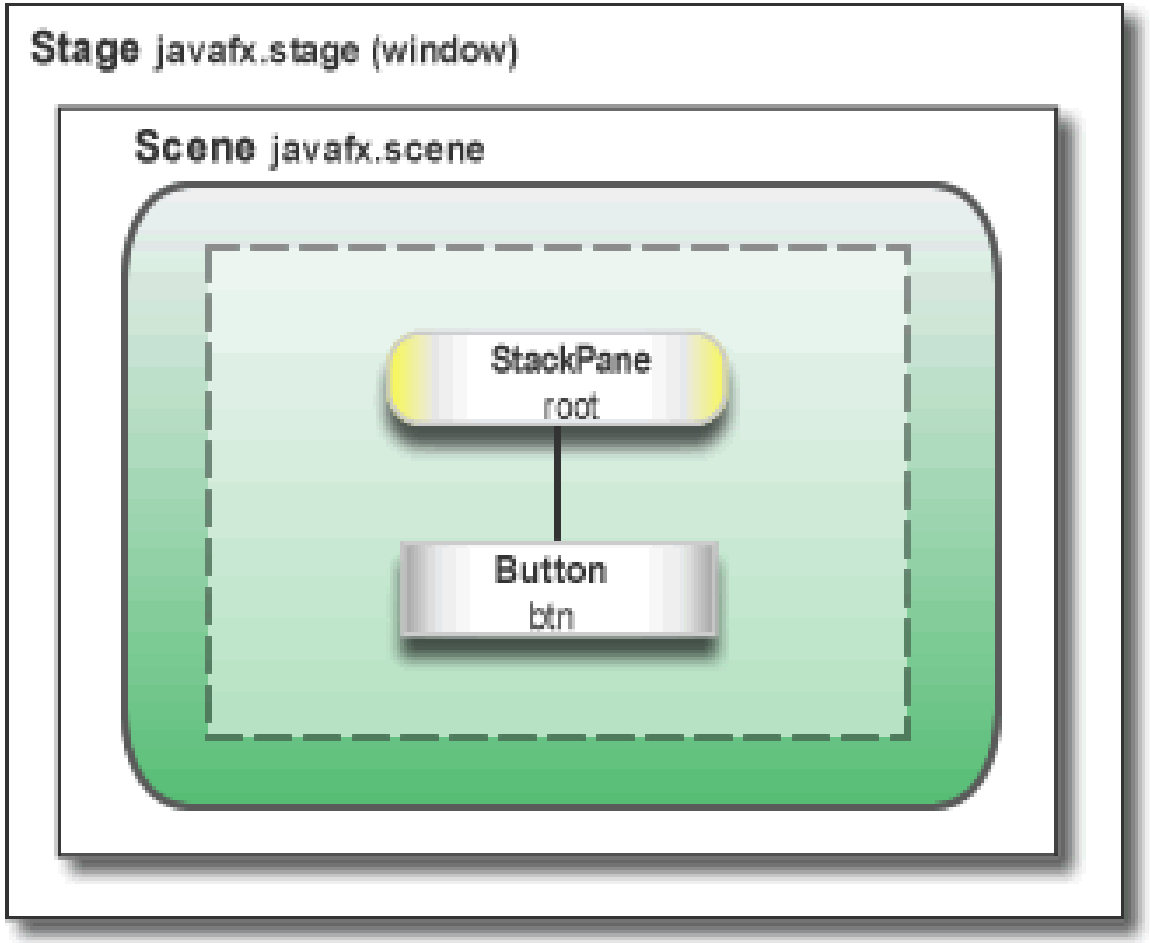
Avantaje **JavaFX:**

- o componentă care poate afișa conținut web și care poate fi integrată în orice interfață grafică JavaFX;
- o serie de componente noi de interfață, precum grafice, tabele, meniuri și panouri;
- un sistem de a împacheta aplicațiile astfel încât acestea să fie livrate cu toate bibliotecile necesare execuției;
- portabilitate pe Linux, Windows și Mac OS X;
- spre deosebire de Swing, aceeași aplicație poate fi rulată de sine stătător, ca applet, sau ca aplicație de tip Web Start.

JavaFX

- Ca și Java Swing, **JavaFX** se bazează pe o ierarhie de clase care implementează diferite componente și containere ce reprezintă elementele grafice. Analog clasei **JFrame**, în JavaFX, clasa care descrie fereastra principală a unei aplicații este **javafx.stage.Stage**.
- Un obiect de tip **Stage** conține, la un moment dat, o singură scenă (**javafx.scene.Scene**). Această scenă este inițializată dându-i-se dimensiunile scenei și container-ul care conține toate celălalte elemente din fereastră. Acest container este, de regulă, un panou, care pe lângă rolul de container, specifică și modul în care sunt afișate componentele, analog **LayoutManager**-ului din Swing.

JavaFX


Stage javafx.stage (window)
Scene javafx.scene
StackPane
root
Button
btn

JavaFX

- Pentru a ajuta dezvoltatorii să proiecteze aplicațiile, JavaFX oferă un instrument de design denumit **JavaFX Scene Builder**. Se pot draga componente UI într-un panou de conținut **JavaFX Content pane**, iar instrumentul generează codul **FXML** care poate fi utilizat într-un IDE precum NetBeans sau Eclipse.
- FXML este un limbaj de marcatori bazat pe XML care permite dezvoltatorilor să creeze o interfață cu utilizatorul (UI) într-o aplicație JavaFX, separat de implementarea logicii aplicației.

JavaFX

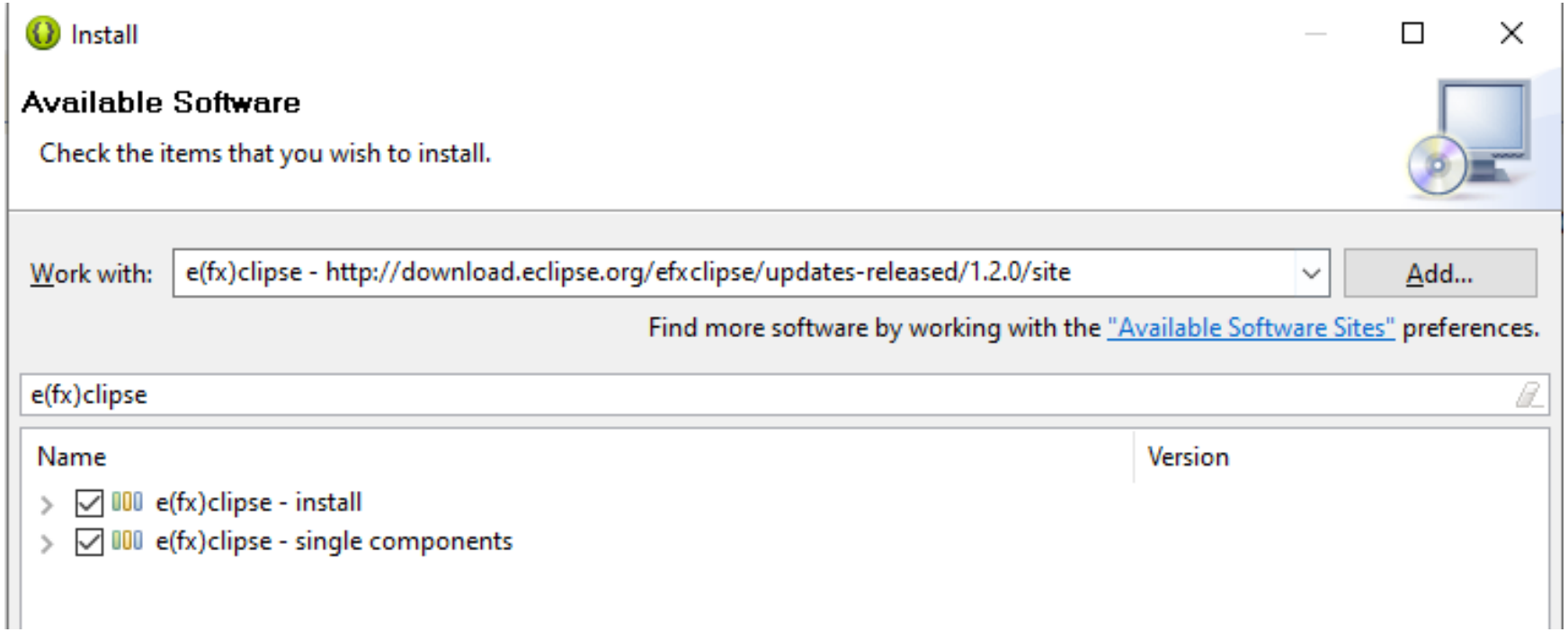
- Într-o aplicație JavaFX, se poate încorpora conținut Swing folosind clasa Swing Node. În mod similar, se pot actualiza aplicațiile Swing existente cu caracteristici JavaFX, cum ar fi conținut web încorporat și suport grafic bogat (**Swing Interoperability**).
- JavaFX oferă suport canvas in pachetul **javafx.scene.canvas**, care deține un set de clase pe care le putem folosi pentru a desena direct într-o zonă a scenei JavaFX. JavaFX oferă, de asemenea, clase pentru suport tipărire în pachetul **javafx.print**.

JavaFX

Ciclul de viață al unei aplicații JavaFX:

- punctul de intrare pentru aplicațiile JavaFX este clasa **Application**; de fiecare dată când este lansată o aplicație JavaFX se derulează următoarele etape, în ordinea aceasta:
- 1. se creează o instanță a clasei Application specificate.
- 2. se apelează metoda **init()** a clasei Application.
- 3. se apeleaza metoda **start()**.
- 4. in acest moment, aplicația este vizibilă pe ecran; iesirea din aplicatie se face atunci când aplicația apelează **Platform.exit()** sau
- când ultima fereastră a aplicației este închisă.
- 5. inainte de a ieși, se apeleaza metoda **stop()** din clasa Application; metoda stop() se poate supradefini pentru a distruge orice resurse utilizate de aplicație.

JavaFX


Install
Available Software
Check the items that you wish to install.
Work with:
e(fx)clipse - http://download.eclipse.org/efxclipse/updates-released/1.2.0/site
Add...
Find more software by working with the "Available Software Sites" preferences.
e(fx)clipse
Name
Version
e(fx)clipse - install
e(fx)clipse - single components

Bibliografie

- [1] Jonathan Knudsen, Patrick Niemeyer – *Learning Java, 3rd Edition*, O'Reilly.
- [2] http://www.itcsolutions.eu
- [3] http://www.acs.ase.ro
- [4] https://en.wikipedia.org/wiki/Swing_(Java)
- [5] http://wiki.dcae.pub.ro/index.php/Graphical_User_Interface_(GUI)_-_Java_Swing_%C8%99i_JavaFX
- [6] https://www.javaguides.net/2019/07/javafx-hello-world-example-tutorial.html?m=1
- [7] https://ro.wikipedia.org/wiki/Spring_Framework