# Lucrarea 7

# Dezvoltarea aplicațiilor cu interfață grafică în Java – biblioteca Swing

Aplicațiile cu interfață grafică în Java pot să fie dezvoltate din cod (vezi exemplele din curs) sau cu ajutorul unui designer grafic. Dezvoltarea aplicațiilor cu interfață grafică din cod permite construirea de aplicații cu interfață dinamică şi oferă un control deplin asupra a ceea ce se întâmplă şi cum se întâmplă în cod, dar este lentă. Utilizarea unui designerului grafic crește viteza de dezvoltare a aplicației prin faptul că permite selectarea şi adăugarea componentelor în interfața grafică prin click-uri, mutarea componentelor în alte poziţii prin drag and drop, configurarea proprietăților componentelor într-o interfață grafică, într-un *Property editor*. Designer-ul grafic generează codul din spatele acțiunilor realizate în modul design. *Window Builder Pro* este un designer grafic care se instalează ca un *plugin* în Eclipse.

Instalarea Window Builder Pro se realizează urmând instrucțiunile de pe linkul:
http://www.eclipse.org/windowbuilder/download.php

După ce a fost instalat *WindowBuilder Pro* se creează un proiect nou în Eclipse şi apoi se creează un *JFrame* în acesta dând comanda *File* > *New* > *Other* > *Window Builder* > *Swing designer*> *JFrame (vezi figura 1).*

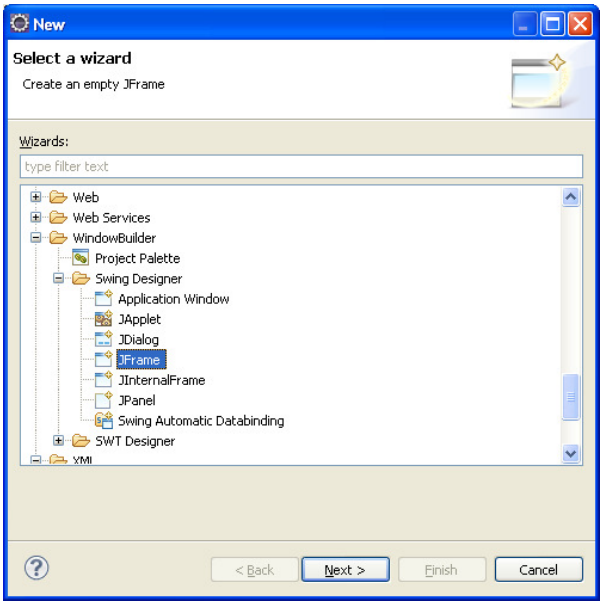


*Figura 1* – crearea unui *JFrame*

*JFrame*-ul creat poate fi vizualizat în modul *Design* sau în modul *Source*. Comutarea între cele două moduri se poate face folosind butoanele din partea stânga jos (vezi figura 2). Adăugarea componentelor în interfață în modul *design* se face selectând componenta dorită prin click în secţiunea *Pallete* (*JLabel, JTextField, JButton,* etc) şi apoi făcând click în poziția dorită pe *JFrame*. Containerele de componente cum ar fi containerul *JFrame*-ului sau *JPanel*-urile pot utiliza gestionari de componente (*LayoutManageri*) care decid unde amplasează controalele în cadrul containerelor, ce dimensiuni le dau în funcție de ordinea

adăugării componentelor şi pe baza unor reguli de funcționare. *LayoutManager*-ul implicit pentru containerul *JFrame*-ului este *BorderLayout* care împarte ecranul in 5 zone, nord, sud, est, vest centru şi permite adăugarea de componente în aceste zone. *LayoutManager*-ul implicit pentru *JPanel*-uri este *FlowLayout* care permite adăugarea componentelor una lângă alta. Când nu mai au loc sunt trecute pe rândul următor. Într-o primă etapă se recomandă amplasarea controalelor în poziții absolute. Pentru aceasta se selectează prin click *LayoutManager*-ul *AbsoluteLayout* şi se aplică prin click pe containerul *JFrame*-ului (sau se scrie în cod *contentPane.setLayout(null)*). În continuare pot să fie selectate componente în secțiunea *Pallete* şi aplicate în poziția dorită în interfață. Proprietățile componentelor se pot modifica din cod sau din modul design (daca se selectează prin click componenta se pot apoi modifica proprietățile acesteia în *Property Editor*.

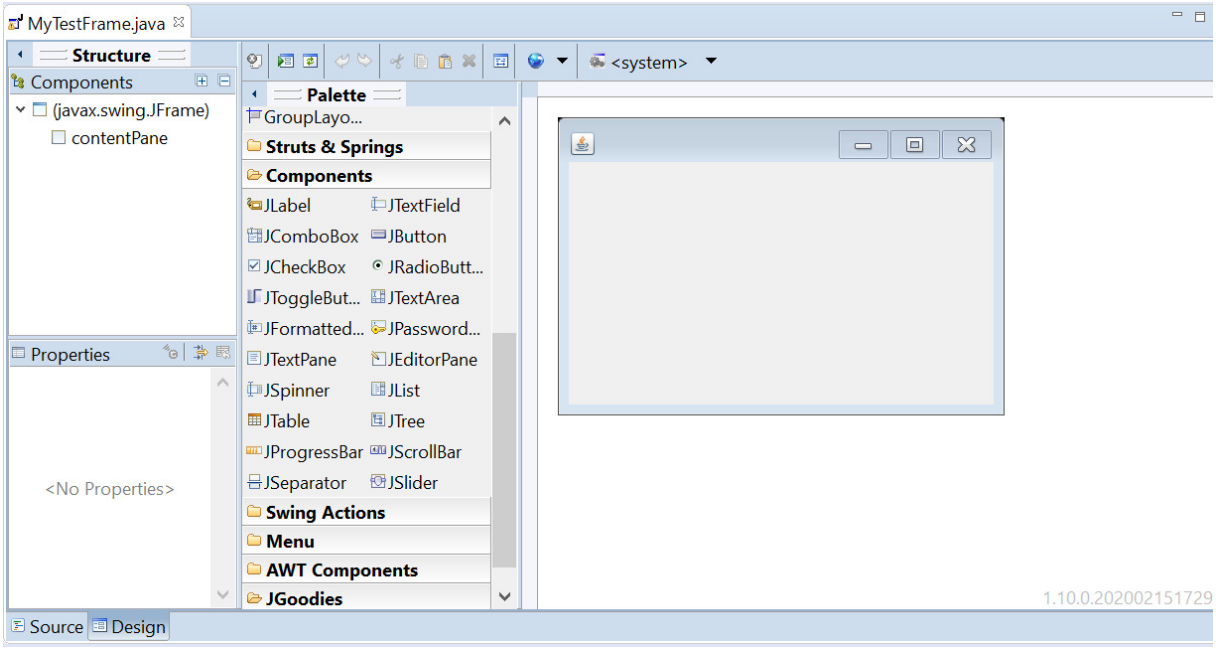


*Figura 2*- designer-ul grafic *Window Builder Pro*

Codul generat pentru *JFrame*-ul creat, fără a face nici o modificare în designer-ul grafic este prezentat mai jos:

```
package exemplu;

import java.awt.BorderLayout;
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;

public class MyTestFrame extends JFrame {
	private JPanel contentPane;
	/**
	 * Launch the application.
	 */
```

```
	public static void main(String[] args) {
		EventQueue.invokeLater(new Runnable() {
			public void run() {
				try {
					MyTestFrame frame = new MyTestFrame();
					frame.setVisible(true);
				} catch (Exception e) {
					e.printStackTrace();
				}
			}
		});
	}

	/**
	 * Create the frame.
	 */
	public MyTestFrame() {
		setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		setBounds(100, 100, 450, 300);
		contentPane = new JPanel();
		contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
		contentPane.setLayout(new BorderLayout(0, 0));
		setContentPane(contentPane);
	}
}
```

S-a creat clasa *MyTestFrame* care extinde *JFrame*, prin urmare are acces la toate proprietăţile pe care aceasta le expune. În constructorul clasei se stabileşte acţiunea care se va întâmpla la rulare când se va face click pe X-ul din colţul din dreapta sus şi anume închiderea aplicaţiei. Metoda *setBounds()* stabileşte locaţia ferestrei (primii doi parametri) şi dimensiunea acesteia (ultimii doi parametri). Se instanţiază *JPanel*-ul numit *contentPane*. Se seteză o bordură goală de 5 pixeli faţă de marginile de sus, stânga, jos şi dreapta. Se setează pentru *JPanel* gestionarul de componente *BorderLayout* cu golurile specificate ca şi parametru între componente pe orizontală şi pe verticală. Se setează *JPanel-ul* creat ca şi container al ferestrei.

În programul principal se creează un fir de execuţie folosind expresia unei clase anonime şi implementând interfaţa *Runnable*. Codul din metoda *run()* reprezintă codul firului de execuţie. În aceasta se creează un obiect de *MyTestFrame* şi se afişează pe ecran. Metoda statică *invokeLater()* din clasa *EventQueue* determină firul să fie rulat după ce toate evenimentele care aşteaptă sunt procesate.

# Tema

1. Realizați o aplicație cu interfață grafică de tip calculator, similară cu cea din figura 3. Situațiile de excepție care pot să apară vor fi tratate prin afișarea de mesaje corespunzătoare (de exemplu: Împărțire la 0, valoare lipsă, valoare necorespunzătoare - dacă în loc de cifre se introduc litere). După finalizarea aplicației se va crea fișierul *jar* rulabil, cu ajutorul comenzii *File > Export > Java > Runnable jar file*. Se alege pentru opțiunea *Launch configuration* clasa *main*-ului şi se specifica prin *Export destination* locaţia şi denumirea fișierului *jar* care va fi creat. Fișierul *jar* creat se rulează prin dublu click.

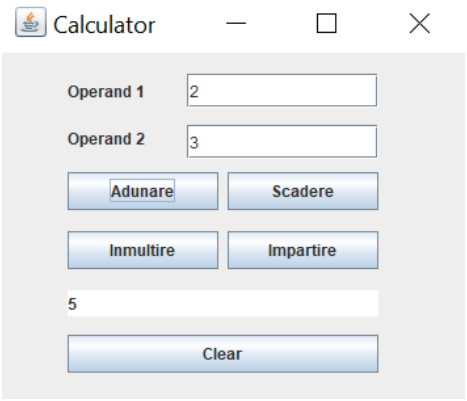


*Figura 3* - Calculator

## *Indicații de rezolvare:*

Creați un proiect nou în Eclipse şi în interiorul acestui proiect un *JFrame.* Deschideți *JFrame*-ul în modul *Design*. Setaţi pentru containerul JFame-ului *LayoutManager AbsoluteLayout* (vezi indicaţiile de mai sus). În continuare puteți amplasa în containerul *JFrame*-ului componente în ce poziţii doriţi si puteți să le daţi ce dimensiune doriți. Calculatorul are în containerul JFrame-ului 3 tipuri de componente: *JLabel*, *JTextField* şi *JButton*. Pentru toate componentele designer-ul grafic creează variabile membre private în clasa *JFrame*-ului. Fiecare componentă trebuie selectată si modificate din secțiunea *Properties* proprietățile acesteia. Pentru fiecare componentă se recomandă modificarea proprietății *Variable name* prin introducerea unui nume sugestiv. Primele 3 litere ar trebui sa indice tipul de componenta (lbl – label, btn – buton, txt- caseta de text) şi apoi ar trebui introdusă o denumire care să indice rolul componentei. Câteva exemple de denumiri recomandate sunt următoarele: *lblOperand, txtOperand1, btnAdunare*. Cu excepția casetelor de text, pentru celelalte componente trebuie editată proprietatea *text*, care reprezintă textul afișat de componentă.

Rezultatul operației aritmetice va fi afișat într-un *JLabel*, de culoare alba.

După construirea interfeței grafice urmează scrierea codului care se va executa când se va face click pe butoane. Pentru aceasta se face dublu click pe buton în modul *Design*. Se va comuta pe modul *Source*, cursorul fiind poziționat pe metoda *actionPerformed()*, metodă care se va executa la rulare atunci când se face click pe buton. Designer-ul grafic generează câte o clasă anonimă pentru click-ul pe fiecare buton. Extragerea unei valori dintr-o caseta de text se face cu ajutorul metodei *getText()*.

2. Sa se realizeze o aplicație cu interfață grafică similară cu cea din figura 4:

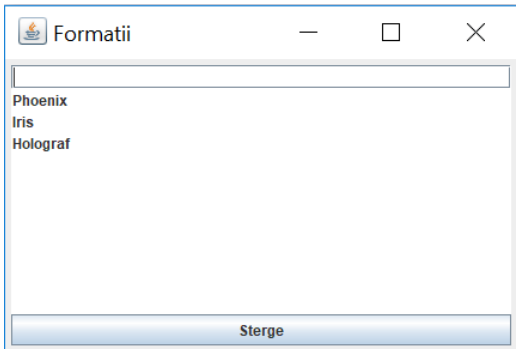


*Figura 4* - formaţii

Numele unei formaţii se introduce în caseta de text şi apoi se apasa tasta *enter*, acţiune care determină adăugarea denumirii formaţiei în caseta *JList* de dedesupt. Acţionarea butonului de ştergere determină ştergerea formaţiilor selectate în caseta JList (ştergere multiplă).

## *Indicaţii de rezolvare:*

Interfaţa grafică se va realiza utilizând gestionarul de componente *BorderLayout*. În nord se va adauga caseta de text, în centru controlul *JList* iar în sud butonul de stergere.

Adăugarea unui eveniment pe caseta de text se poate face în modul *Design* prin click dreapta pe casetă, apoi *Add event handler* > *action* > *actionPerformed*.

Adăugarea în *JList* se poate realiza cu ajutorul unui obiect *DefaultListModel* asociat *JList*-ului (vezi exemplul din curs).

3. Să se realizeze o aplicaţie care permite introducerea unor filme lansate în ultimii 5 ani şi afişarea acestora în format tabelar. Aplicaţia va avea o interfaţă grafică similara cu cea din figura 5:

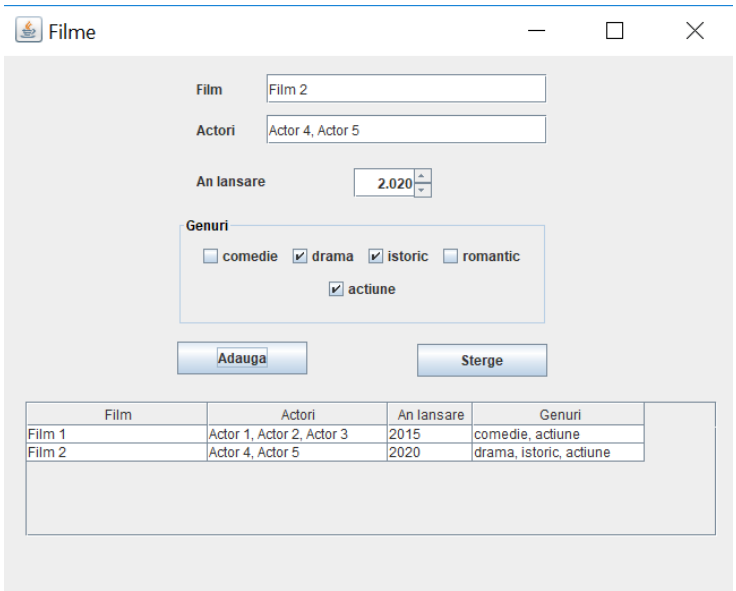


*Figura 5* - filme

# *Indicaţii de rezolvare:*

Contanerul *JFrame*-ului va utiliza gestionarul de componente *AbsolutLayout*.

Pentru anul de lansare se va utiliza o componentă *JSpinner* configurată prin proprietatea model astfel încat valoarea minimă să fie 2015, valoarea maximă 2020 şi pasul de incrementare 1. Preluarea valorii selectate se realizeaza cu ajutorul metodei *getValue()*.

Apăsarea butonului *Adauga* va determina introducerea filmului într-un control *JTable*. Acţionarea butonului de ştergere va determina stergerea filmelor selectate în *JTable*.

Adăugarea, respectiv ştergerea filmelor în *JTable* se va realiza cu ajutorul unui obiect *DefaultTableModel* asociat *JTable*-ului.