

# Laborator 9

În acest laborator vom construi o aplicație completă cu accent pe construcția interfeței grafice, pentru a vedea cum putem lucra cu diverse elemente.

## Etapa 1

Să înțelegem ce se cere. Vrem să facem o aplicație în care să introducem studenți și notele lor. Un student are un nume, prenume și o listă de note.

O notă, are o valoare (de la 1 la 10, întreaga) și o disciplină (string).

Pe prima pagină a aplicației vedem un tabel cu studenți, coloanele nume, prenume, note, unde în coloana note vedem câte note are studentul (un număr).

Aplicația are un meniu "Principal" și un meniu "Adăugare".

În meniul principal avem "Salvare date" care deschide un dialog în care alegem fișierul unde să fie salvate datele și apoi salvează datele.

Mai avem și "Încărcare date" care deschide un dialog în care alegem fișierul din care să încărcam datele.

Mai avem și "Ștergere date" care curăță "baza de date" curentă și "Închidere" care ne scoate din aplicație.

În meniul adăugare, avem "Adăuga student" care deschide o fereastră în care adăugăm un student nou (nume și prenume, nu și note) și mai avem "Adăuga notă" care deschide o fereastră în care adăugăm un dropdown cu numele și prenumele studenților din bază, un dropdown cu disciplină (Avem doar 5 discipline, Analiză matematică, Algebră, Geometrie, Baze de date, Programare distribuită) și un câmp unde punem o valoare întreagă, notă efectivă. Evident în ambele ferestre avem câte un buton cu "Ok" respectiv "Cancel" care execută acțiunile efectiv (ok) sau doar închide fereastra respectivă (cancel).

## Etapa 2

Construim clasele de care avem nevoie, getter/setter.

Nu uitați ce trebuie să facem ca să putem să salvăm pe disc datele!

## Etapa 3

Construim pagina principală.

Mai întâi facem o fereastră goală (ce componentă folosim? ce mai trebuie să facem ca să o vedem pe ecran?)

Apoi trebuie să adăugăm un meniu.

Informații complete legate de meniuri găsiți aici:

<https://docs.oracle.com/javase/tutorial/uiswing/components/menu.html>

Ideea este ca și la meniuri trebuie să avem o ierarhie de componente, dar nu pot să participe decât anumite componente.

Componenta de top level este JMenuBar, si o instanta de-a ei trebuie setata in JFrame cu setJMenuBar.

Un JMenuBar contine mai multe JMenu. Fiecare JMenu reprezinta un meniu (de exemplu, File, Edit, Options, Help) iar constructorul lui primeste un sir de caractere reprezentand textul de pe meniu.

Fiecare JMenu contine mai multe JMenuItem. JMenuItem reprezinta o optiune din meniu.

Cum adaugam un JMenu la un JMenuBar? Dar un JMenuItem la un JMenu? Hint: toate sunt componente.

Probabil ati observat ca exista shortcuturi pentru meniuri. De exemplu, putem sa apasam Alt, se activeaza meniu, si apoi F si se deschide meniul File. Faptul ca atunci cand apasam F se deschide File este realizat prin apelul lui

```
menu.setMenmonic(KeyEvent.VK_F);
```

sau pentru JMenuItem, folosind al doilea parametru al constructorului sau.

Construiti meniul! Testati, ca sa verificati ca arata cum doriti.

## Etapa 4

Adaugarea tabelului cu studenti.

In Java, un tabel se realizeaza foarte usor, folosind componenta JTable.

Informatii complete gasiti aici:

<https://docs.oracle.com/javase/tutorial/uiswing/components/table.html>

Pentru a configura un tabel ne vom folosi de o implementare particulara a clasei

```
javax.swing.table.AbstractTableModel
```

Aceasta clasa contine tot ce ii trebuie tabelului pentru a fi afisat cum trebuie. Clasa noastra, care extinde AbstractTableModel, trebuie sa implementeze cateva metode:

1. `int getColumnCount` -> returneaza numarul de coloane.
2. `int getRowCount` -> returneaza numarul de randuri
3. `String getColumnName(int col)` -> returneaza titlul coloanei cu numarul col (incepem de la 0)
4. `Object getValueAt(int row, int col)` -> returneaza continutul casutei [row,col].

5. `Class getColumnClass(int col) -> returneaza tipul savat in coloana respectiva (din fericire noi salvam doar Stringuri, deci putem returna String.class)`
6. `boolean isCellEditable(int row, int col) -> returneaza true daca putem sa modificam cazuta [row,col]. Noi nu vrem sa modificam nicio casuta.`
7. `void setValueAt(Object value, int row, int col) -> care seteaza in obiectul din spate (in model) datele preluate de la user din tabel, din casuta [row,col];`

Dupa ce am definit o clasa model corecta, tot ce trebuie sa facem este ca construim un `JTable` si sa ii pasam constructorului acestuia o instanta a modelului nostru.

Nu uitati, clasa noastra poate contine si alte lucruri in afara de implementarile obligatorii!

Definiti clasa model (folositi ce nume vreti).

Construiti un tabel.

Adaugati tabelul in fereastra noastra (ce trebuie sa facem inainte pentru a adauga tabelul?)

## Etapa 5

Sa adaugam construim fereastra copil in care adaugam studenti.

Pentru a deschide o fereastra copil, folosim componenta `JDialog`. Puteti sa va ganditi la ea ca la un frame. Constructorul lui `JDialog` insa primeste ca parametru:

1. `JFrame`ul parinte
2. Titlul dialogului
3. `true/false` daca dialogul este modal (adica daca ne forteaza sa il inchidem inainte de a face altceva). Noi vrem sa fie modal.

Ar trebui sa stiti cum sa construiti tot ce se afla in `JDialog`, dar unde punem codul care il deschide?

`JMenuItem` are o metoda numita `addActionListener` care primeste ca parametru o implementare a lui `ActionListener` (ca la butoane), iar codul din unica sa metoda este cel care se executa.

Construiti `JDialog` si adaugati un action listener corespunzator pentru `JMenuItem` (si toate elementele din dialog!)

## Etapa 6

Facem o procedura similara pentru Adaugare Nota, doar ca de data aceasta vom folosi o noua componenta, `JComboBox`. Informatii complete aici:

<https://docs.oracle.com/javase/tutorial/uiswing/components/combobox.html>

dar pe scurt, constructorul primeste un `Array` cu optiunile posibile, avem o metoda `getSelectedIndex` care ne returneaza numarul itemului selectat si `getItemAt(index)` care ne returneaza obiectul de la indexul respectiv.

Faceti si acest dialog.

## Etapa 7

Implementati ascultatorul pentru meniul "Iesire"

## Etapa 8

Pentru a implementa salvarea, vom folosi o alta componenta, JFileChooser.

Aceasta componenta este destul de independenta si deschide un dialog modal in care userul poate selecta un fisier de pe disc.

Informatii complete aici:

<https://docs.oracle.com/javase/tutorial/uiswing/components/filechooser.html>

Dupa ce am construit un obiect nou de tip JFileChooser (cu constructorul fara argumente), chemam metoda showSaveDialog(parentFrame) care returneaza un intreg. In functie de valoarea returnata, stim ce a facut userul, iar daca a fost returnat

JFileChooser.APPROVE\_OPTION inseamna ca userul a apasat pe Save. In acest caz, putem sa folosim getSelectedFile() pentru a obtine numele fisierului selectat. Odata ce avem fisierul, este simplu sa facem salvarea.

Implementati acum aceasta functionalitate. Atentie la tratarea exceptiilor!

## Etapa 9

Haideti sa implementam incarcarea. Vom proceda aproape la fel ca mai sus, doar ca vom folosi showOpenDialog si vom citi din fisier.

Implementati acum aceasta functionalitate. Atentie la tratarea exceptiilor!

## Etapa 10

Implementati golirea bazei de date, din meniul principal.

## Etapa 11

Observam ca dupa ce modificam datele din spatele tabelului (din model) tabelul nu se schimba.

De ce? Pentru ca noi nu anuntam tabelul cand ni se schimba datele din model.

Pentru a face acest lucru, trebuie sa chemam metoda fireTableDataChanged de pe modelul tabelului dupa fiecare modificare.

## Etapa 12

Daca tabelul este destul de mare, atunci nu mai intra in fereastra. Putem sa adaugam un scrollbar, iar asta o facem foarte simplu adaugand tabelul ca un copil al unei componente numita JScrollPane. Informatii complete aici:

<https://docs.oracle.com/javase/tutorial/uiswing/components/scrollpane.html>

insa puteti sa folositi implementarea default in acest caz, fara customizari.