Curs2 PIG

https://docs.oracle.com/javase/tutorial/uiswing/index.html

https://docs.oracle.com/javase/tutorial/uiswing/examples/components/index.html

https://docs.oracle.com/javase/8/javafx/interoperability-tutorial/swing-fx-interoperability.htm

https://docs.oracle.com/javase/8/javase-clienttechnologies.htm

TEHNOLOGIA JFC (SWING)

Swing nu a existat în primele versiuni de Java. Tehnologia JFC (Java Foundation Classes) grupează un asamblu de facilităţi pentru construirea de interfeţe grafice şi interactivitate în cadrul aplicaţiilor Java. Prima versiune a bibliotecilor de clase a fost lansată în cadrul conferinţei JavaOne, în anul 1997.

Abstract Window Toolkit AWT defineşte un set de controale de baza, ferestre si dialog boxes ce suporta o interfaţa grafica utilizabila dar extrem de limitata. Un motiv pentru natura limitata a AWT este ca traduce diferitele componente vizuale în corespondentele specifice platformei utilizate sau legături. Acest lucru înseamnă ca look-and-feel – ul componentei este definit de platforma si nu de Java. Deoarece componentele AWT folosesc resurse de cod nativ sunt definite ca heavyweigh (de categorie grea).

Principalele pachete ce se folosesc în proiectarea şi dezvoltarea interfeţelor grafice sunt: **java.awt, java.awt.event, javax.swing, java.fx.** Acestea sunt incluse în pachetul JDK (Java Development Kit) care conţine clase fără de care nu se pot dezvolta aplicaţiile java.

De la lansarea aplicaţiei, Java AWT-ul (Abstract Window Toolkit) a făcut parte din JDK dar nu a fost suficient pentru a suporta interfeţe grafice complexe, totuşi putea suporta majoritatea operaţiilor valabile într-un formular HTML.

Java Foundation Classes (JFC) reprezintă un set de librării făcute pentru a ajuta programatorii în dezvoltarea aplicaţiilor pentru diferite corporaţii. Swing este doar una

dintre cele 5 librării care alcătuiesc JFC.

De asemenea JFC cuprinde şi Abstract Window Toolkit (AWT), Accessibility API şi 2D API precum şi suport pentru capabilităţile de tip drag and drop.

- AWT (Abstract Window Toolkit) - este interfaţă cu utilizatorul inclusă în toate versiunile de Java Development Kit (JDK).
- Accessibility

Pachetul Accessibility oferă ajutor utilizatorilor care au probleme cu interfeţele tradiţionale. Poate fi folosit împreună cu componente precum instrumente cu comandă vocală sau tastaturi în limbajul braille pentru a permite accesul la componentele Swing. Pachetul este împărţit în două părţi: API-ul Accessibility, care este inclus în Swing şi API-ul Accesibility Utilities care este distribuit separat.

- 2D API conţine clase folosite în implementarea formelor complexe, fonturi şi culori. Clasele 2D API nu fac parte din pachetul Swing.
- Drag and drop - funcţia drag and drop este una din cele mai cunoscute şi folosite în interfeţele grafice. Utilizatorul poate selecta un obiect ţinând click pe acesta şi îl poate muta într-o altă fereastră. Funcţia drag and drop nu face parte din clasa Swing.

Caracteristicile tehnologiei JFC

Java Foundation Classes (JFC) reprezintă un set de librării făcute pentru a ajuta programatorii în dezvoltarea aplicaţiilor pentru diferite corporaţii. Swing este doar una dintre cele 5 librării care alcătuiesc JFC.

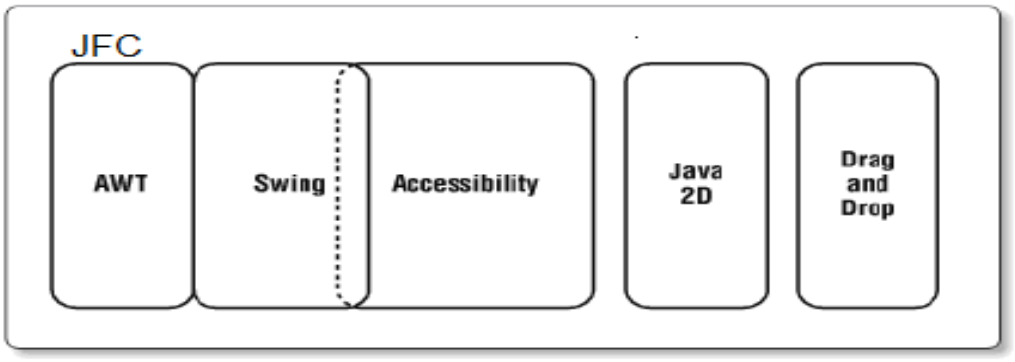


Figura Legăturile dintre clasele ce formează JFC

Caracteristică	Descriere
Componente GUI Swing	Componente GUI: Ferestre, dialoguri, meniuri, butoane, liste etc.
Suport pentru aspect	Componentele grafice pot avea acelaşi aspect, independent de platformă, sau aspecte corespunzătoare anumitor platfome
Suport pentru accesibilitate	Inputul de la utilizator poate fi obţinut prin cititoare de ecran şi monitoare Braille.
Grafică 2D	Permite încorporarea elementelor de grafică 2D, text, imagine în aplicaţii şi applet-uri. De asemenea, include suport performant pentru imprimare.
Suport Drag-and-Drop	Extinde această facilitate, realizând comunicarea între aplicaţiile Java şi cele native.
Internaţionalizare	Permite dezvoltarea aplicaţiilor ce prelucrează text ce conţine caractere corespunzătoare diferitelor limbi.

"Swing" reprezintă numele de cod al proiectului care a dezvoltat noile componente. *Neoficial,* este frecvent folosit pentru a desemna JFC.

Biblioteca de clase care oferă servicii grafice se numeşte **java.awt**, **AWT** fiind prescurtarea de la **Abstract Window Toolkit** şi este pachetul care care a suferit cele mai multe modificări în trecerea de la o versiune JDK la alta.

Structura de clase din Swing este asemănătoare cu cea din AWT, în sensul că toate componentele interfeţei grafice sunt derivate dintr-un singur părinte numit JComponent (care este derivat din clasa AWT Container).

Pachetul de clase Swing reprezintă soluţia furnizată de Sun pentru crearea unor interfeţe utilizator grafice complet portabile pe orice platformă.

În Swing, toate numele claselor încep cu litera J, şi atunci când este posibil, numele este acelaşi cu cel al clasei AWT pe care o înlocuieşte.

În principiu, crearea unei aplicaţii grafice presupune următoarele lucruri:

- crearea unei suprafete de afişare (jFrame cum ar fi o fereastră) pe care vor fi aşezate obiectele grafice care servesc la comunicarea cu utilizatorul (butoane, controale de editare, texte, etc);
- crearea şi aşezarea obiectelor grafice pe suprafata de afişare (la coordonatele x,y .....);
- definirea unor acţiuni (action...) care trebuie să se execute în momentul când utilizatorul interacţionează cu obiectele grafice ale aplicatiei;
- "ascultarea" --- interfata listner evenimentelor generate de obiecte, în momentul interacţiunii cu utilizatorul şi executarea acţiunilor corespunzătoare aşa cum au fost ele definite.

Majoritatea obiectelor grafice sunt subclase ale clasei Component, clasa care defineşte generic o componentă grafică şi care poate interacţiona cu utilizatorul. Singura excepţie o constituie meniurile care descind din clasa JMenuComponent.

Print-o componentă grafică se înţelege orice obiect care are o reprezentare grafică ce poate fi afişată pe ecran şi care poate interacţiona cu utilizatorul. Exemple de componente grafice sunt **ferestrele, butoanele, bare de defilare etc.** In general, toate componentele sunt definte de clase proprii ce se găsesc în pachetul java.aw, clasa Component fiind superclasa abstractă a tuturor acestor clase.

Crearea obiectelor grafice **nu** realizează automat şi afişarea lor pe ecran. Mai întâi ele trebuie aşezate pe o suprafaţă de afişare, care poate fi o fereastră sau suprafaţa unui applet, şi vor deveni vizibile în momentul în care suprafaţa pe care sunt afişate va fi vizibilă. O astfel de suprafaţă pe care se asează obiectele grafice reprezintă o instanţă a unei clase obţinută prin extensia clasei Container; din acest motiv suprafeţele de afişare vor mai fi numite şi containere. Clasa Container este o subclasă aparte a clasei Component, fiind la rândul ei superclasa tuturor suprafeţelor de afişare Java (ferestre, applet-uri, etc).

Interfaţa grafică serveşte interacţiunii cu utilizatorul. De cele mai multe ori programul trebuie să facă o anumită prelucrare în momentul în care utilizatorul a efectuat o acţiune şi, prin urmare, obiectele grafice trebuie să genereze evenimente în funcţie de acţiunea pe care au suferit-o (actiune transmisă de la tastatură, mouse, etc.). Un eveniment este produs de o acţiune a utilizatorului asupra unui obiect grafic, deci evenimentele nu trebuiesc generate de programator, dar intr-un program trebuie specificat codul care se execută la aparitia unui eveniment.

Interceptarea evenimentelor se realizează prin intermediul unor clase de tip listener (ascultator, consumator de evenimente), clase care sunt definite în pachetul java.awt.event. In Java, orice componentă poate "consuma" evenimentele generate de o altă componenta grafică.

Componentele Swing

Un GUI Swing este compus din doua elemente principale: componente si containere. Totuşi, aceasta distincţie este mai mult conceptual deoarece toate containerele sunt de asemenea si componente. Diferenţa între cele doua se găsesc în scopul destinat. Termenul folosit în general este ca o componenta este un control vizual independent cum ar fi un slider sau un push button.

Un container cuprinde un grup de componente, prin urmare un containere este un tip special de componenta care este conceput pentru a tine alte componente. Totuşi

pentru ca o componenta sa fie afişată trebuie sa fie ţinuta într-un container. Prin urmare toate GUI – urile Swing vor avea cel puţin un container. Deoarece containerele sunt componente, o componenta poate tine si alte componente. Aceasta permite Swing sa defineasc ce este o ierarhie container la baza căreia trebuie sa fie container un top-level.

Componente

In general componentele Swing sunt derivate din clasa JComponent. Singura excepţie la aceasta sunt cele 4 containere top-level descrise în următoarea secţiune. **JComponent** furnizează funcţionalitate comuna pentru toate componentele. Ca de exemplu, JComponent suporta aspectul bazat pe plug in. JComponent moşteneşte din clasa AWT Container si Component. Rezulta ca o componenta swing este construita si compatibila cu componenta AWT.

Toate componentele Swing sunt reprezentate de clase si definite în pachetul javax.swing. Următorul tabel arata numele claselor pentru componentele Swing (inclusiv cele folosite ca si container).

JApplet	JButton	JCheckBox	JCheckBoxMenuItem
JColorChooser	JComboBox	JComponent	JDesktopPane
JDialog	JEditorPane	JFileChooser	JFormattedTextField
JFrame	JInternalFrame	JLabel	JLayer
JLayeredPane	JList	JMenu	JMenuBar
JMenuItem	JOPtionPane	JPanel	JPasswordField
JPopupMenu	JProgressBar	JRadioButton	JRadioButtonMenuItem
JRootPane	JScroolBar	JScroolPane	JSeparator
JSlider	JSpinner	JSplitPane	JTabbedPane
JTable	JTextArea	JTextField	JTextPane
JTogglebutton	JToolBar	JToolTip	JTree

JViewport	JWindow		

De notat ca toate clasele component încep cu litera J. De exemplu, clasa pentru un label este JLabel; clasa pentru un buton tip push este JButton; si clasa pentru bara scrool este JScrollBar.

Containere

Swing defineşte doua tipuri de containere. Primele sunt containere top-level: JFrame, JApplet, JWindow, and JDialog. Aceste containere nu moştenesc JComponent. Dar în schimb moştenesc din AWT clasele Component si Container. Spre deosebire de celelalte componente Swing care sunt lightweight containerele top-level sunt containere heavyweight. Aceasta face containerele top level un caz special pentru librăria de componente Swing.

După cum si numele precizează un container top-level trebuie sa fie la baza ierarhiei de componente. O componenta top-level nu este conţinută în nici un alt container. De asemenea orice ierarhie de componente trebuie sa înceapă cu un container top-level. Cel mai folosit în aplicaţii este JFrame iar cel mai folosit pentru Applet este JApplet.

Al doilea tip de containere suportat de Swing sunt containerele lightweight. Containerele lightweiht moştenesc JComponent. Un exemplu de containere de categorie uşoară este JPanel, care este un container general. Containerele de categorie uşoară sunt de obicei folosite în organizarea si gestionarea grupurilor asemănătoare de componente pentru ca un container uşor poate fi conţinut de alt containere. Prin urmare se foloseşte Containerele lightweight ca JPanel pentru a crea subgrupuri de controale asemănătoarea ce sunt conţinute într-un alt container.

Panouri de conţinut de nivel înalt

Fiecare container de nivel înalt defineşte un set de *panouri.* La cel mai înalt nivel în ierarhie sta o instanţa a **JRootPane.**

JRootPane este un container minimal ce are ca scop gestionarea celorlalte panouri. De asemenea, ajuta la gestionarea barei de meniu care este opţională. Panourile care umple panoul rădăcina sunt denumite *glass pane, content pane,* si *layered pane.*

Glass pane este un panou top-level. Stă deasupra si acoperă complet toate celelalte panouri. Implicit este o instanţa transparenta a JPanel. Panoul Glass are activate interacţionări cu mouse-ul ce afectează întreg containerul (fată un control individual) sau de exemplu pentru a desena peste oricare alta componenta. În majoritatea cazurilor nu este neapărata folosirea acestuia dar în caz de necesitate el există.

Layered pane este o instanţa a **JLayeredPane.** Layered pane permite componentelor sa li se acorde o valoare de adâncime. Aceasta valoare determina care componenta suprapune pe cealaltă. (Astfel, layered pane permite specificare unei ordini pe axa Z pentru o componenta, deşi acest lucru nu este ceva ce este necesar întotdeauna.) Layered pane ţine content pane -ul si opţional bara de meniu.

Deşi glass pane - ul si layered pane –urile sunt parte integrata a operaţiilor unui container de nivel înalt si are atribuţii importante, o mare parte din ceea ce oferă se întâmplă în spatele aplicaţiei. Panoul cu care aplicaţia va interacţiona cel mai mult este content pane –u, deoarece acesta este panoul în care se adaugă componentele vizuale. Cu alte cuvinte, atunci când adăugaţi o componentă, cum ar fi un buton, într-un container de nivel înalt, va fi adăugat în content pane. în mod implicit, panoul de conţinut este o instanţă opaca a **JPanel**.

Pachetele Swing

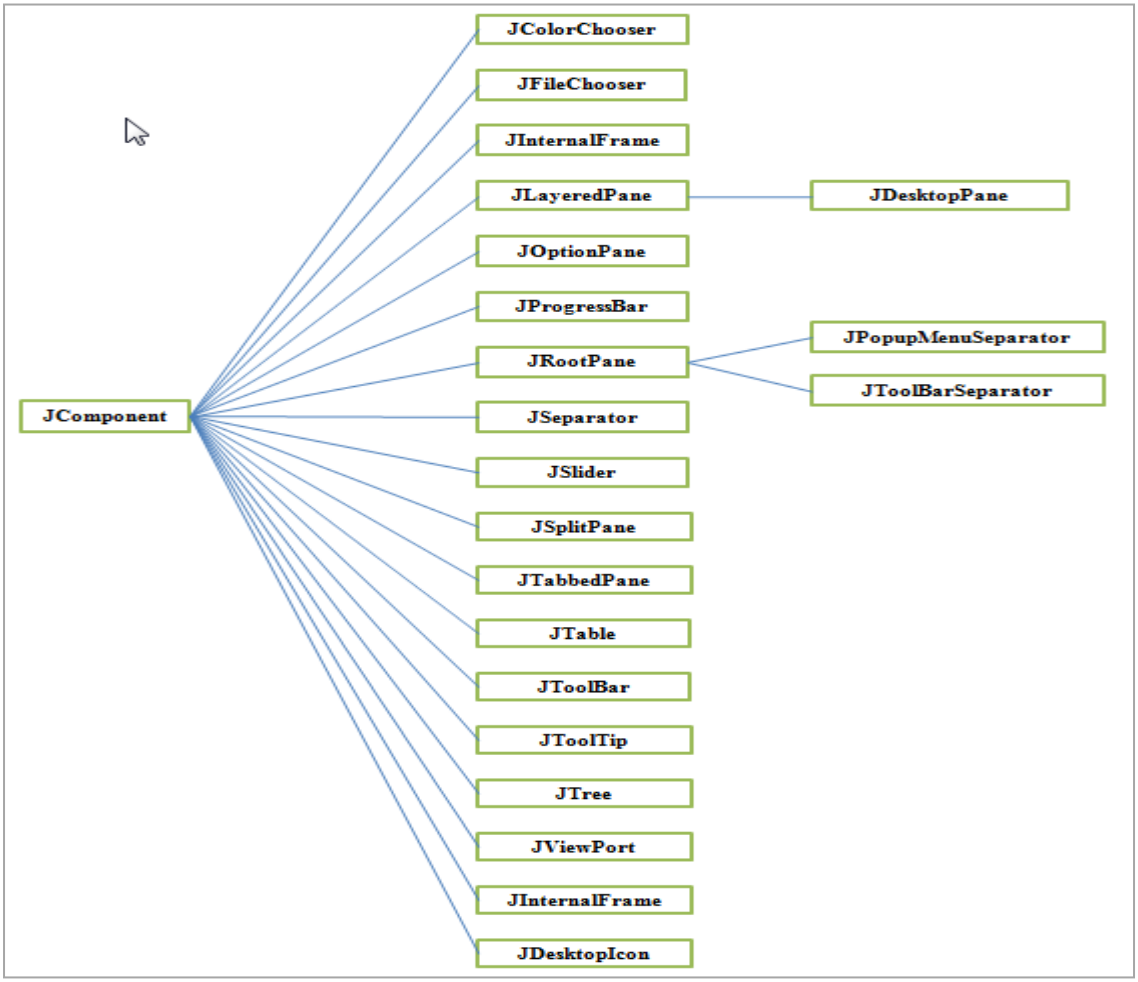
Swing este un subsistem foarte mare şi face uz de mai multe pachete care sunt in tabelul alturat.

javax.swing.*	javax.swing.plaf.basic	javax.swing.text
javax.swing.border	javax.swing.plaf.metal	javax.swing.text.html

javax.swing.colorchooser	javax.swing.plaf.multi	javax.swing.text.html.parser
javax.swing.event	javax.swing.plaf.nimbus	javax.swing.text.rtf
javax.swing.filechooser	javax.swing.plaf.synth	javax.swing.tree
javax.swing.plaf	javax.swing.table	javax.swing.undo

Pachetul principal este **javax.swing.** Acest pachet trebuie să fie importat în orice program care utilizează Swing. Acesta conţine clasele care implementează componentele Swing de bază, cum ar fi butoane, check boxes si etichetele.

Principalele componentele Swing definite in pachetul javax.swing.*;



Componentele prezentate, folosite pentru crearea interfeţelor grafice Swing pot fi grupate astfel:

- **Componente atomice**

- JLabel, JButton, JCheckBox, JRadioButton, JToggleButton, JSlider, JProgressBar, JSeparator

- **Componente complexe**
  - **JTable, JTree, JComboBox, JSpinner, JList, JFileChooser, ColorChooser, JOptionPane**

- **Componente pentru editare de text**
  - JTextField, JFormattedTextField, JPasswordField, JTextArea, JEditorPane, JTextPane

- **Meniuri**
  - JMenuBar, JMenu, JPopupMenu,
  - JMenuItem
  - JCheckboxMenuItem, JRadioButtonMenuItem

Containere - reprezintă suprafeţe de afişare pe care pot fi plasate alte componente, eventual chiar alte containere. Superclasa componentelor de acest tip este Container, din modelul AWT.

Containere intermediare - reprezintă suprafeţe de afişare cu ajutorul cărora pot fi organizate mai eficient componentele aplicaţiei, putând fi imbricate. Acestea sunt:

JPanel, JScrollPane, JTabbedPane, JSplitPane, JLayeredPane, JDesktopPane, JRootPane

JPanel este cel mai simplu container, avand aceeaşi funcţionalitate ca şi clasa Panel din AWT, fiind folosit pentru gruparea mai multor componente Swing şi plasarea lor împreună pe o altă suprafaţă de afişare. Gestionarul de poziţionare implicit este FlowLayout, acesta putând fi schimbat însă, chiar în momentul construirii obiectului JPanel, sau ulterior cu metoda setLayout. Adăugarea de componente se realizează ca pentru orice container, folosind metoda **add( ).**

JScrollPane este o clasă foarte importantă în arhitectura modelului Swing, deoarece oferă **suport pentru derularea pe orizontală** şi verticală **a componentelor a căror reprezentare completă nu încape în suprafaţa asociată**, nici o componentă Swing neoferind suport intrinsec pentru această operaţie.

Clasa **JComponent** este superclasa tuturor componentelor Swing, mai puţin a celor care descriu containere de nivel înalt JFrame, JDialog, JApplet. Deoarece **JComponent extinde clasa Container**, deci şi Component, ea moşteneşte funcţionalitatea generală a **containerelor şi componentelor AWT, furnizând bineînţeles şi o serie întreagă de noi facilităţi.**

Containere de nivel înalt

Un container top-level este un container care nu este inclus în nici un alt container.

Din acestea fac parte:

- **JFrame, JDialog, JWindow, JInternalFrame, JApplet**

Pentru a fi afişate pe ecran componentele grafice ale unei aplicaţii trebuie plasate pe o suprafaţă de afişare (container). Fiecare componentă poate fi conţinută doar într-un singur container, adăugarea ei pe o suprafaţă nouă de afişare determinând eliminarea ei de pe vechiul container pe care fusese plasată. Deoarece containerele pot fi încapsulate în alte containere, o componentă va face parte la un moment dat dintr-o ierarhie. Rădăcina acestei ierarhii trebuie să fie un aşa numit container de nivel înalt, care este reprezentat de una din clasele ***JFrame, JDialog*** sau ***JApplet.***

În general orice aplicaţie Java independentă bazată pe Swing conţine cel puţin un container de nivel înalt reprezentat de fereastra principală a programului, instanţă a clasei Jframe si una sau mai multe ferestre aditionale.

Clasele care gestioneaza ferestrele in Swing sunt:

Clasa JFrame care permite crearea ferestrei aplicatiei, care contine bara de titlu, marginea, butoanele system: minimizare, maximizare si inchidere.

Clasa JWindow care permite crearea unei ferestre simple, fara bara de titlu, meniu sau butoane sistem.

Clasa JDialog care permite crearea ferestrelor de dialog. Ferestrele de dialog pot fi modale sau nu. Ferestrele modale

blocheaza focus-ul altor ferestre pana la inchiderea dialogului curent.

Pentru a crea ferestre de afisare a unor mesaje se poate utiliza direct o functie statica, fara a mai crea explicit un obiect tip dialog:

```
JOptionPane.showMessageDialog(frame, "Student la Informatica.");
```

Pentru ferestre de dialog standard exista clase specializate: ***JFileChooser*** si ***JColorChooser***. Acestea pot fi utilizate pentru a selecta un fisier sau a alege o culoare.

Caracteristicile tehnologiei JFC

Java Foundation Classes (JFC) reprezintă un set de librării făcute pentru a ajuta programatorii în dezvoltarea aplicaţiilor pentru diferite corporaţii. Swing este doar una dintre cele 5 librării care alcătuiesc JFC.

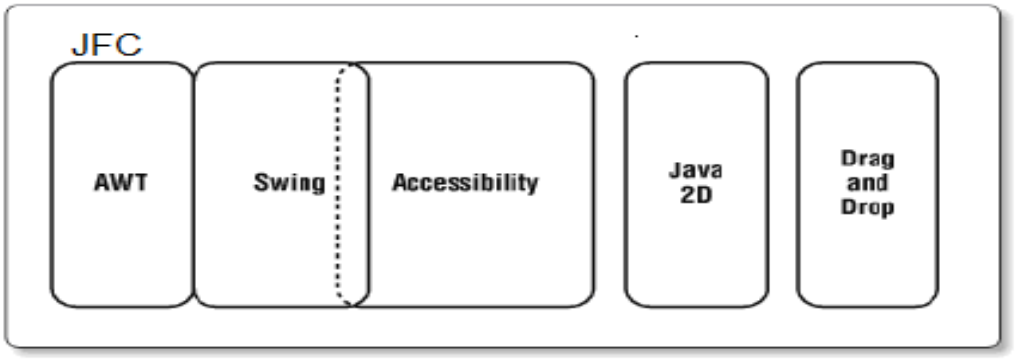


Figura Legăturile dintre clasele ce formează JFC

Exemplu de cod pentru crearea unei ferestre principale:

```
public static void main(String args[]) {
  JFrame win = new JFrame("Student la Informatica");
  win.setSize(200, 200);
  win.show();
 }
```

Pentru a accesa continutul unei ferestre se va folosi functia ***getContentPane***():

```
JFrame win = new JFrame("Student la Informatica");

Container c = win.getContentPane();
```

Pentru a obtine inchiderea automata a aplicatiei atunci cand se apasa butonul de Close, se va utiliza metoda:

```
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

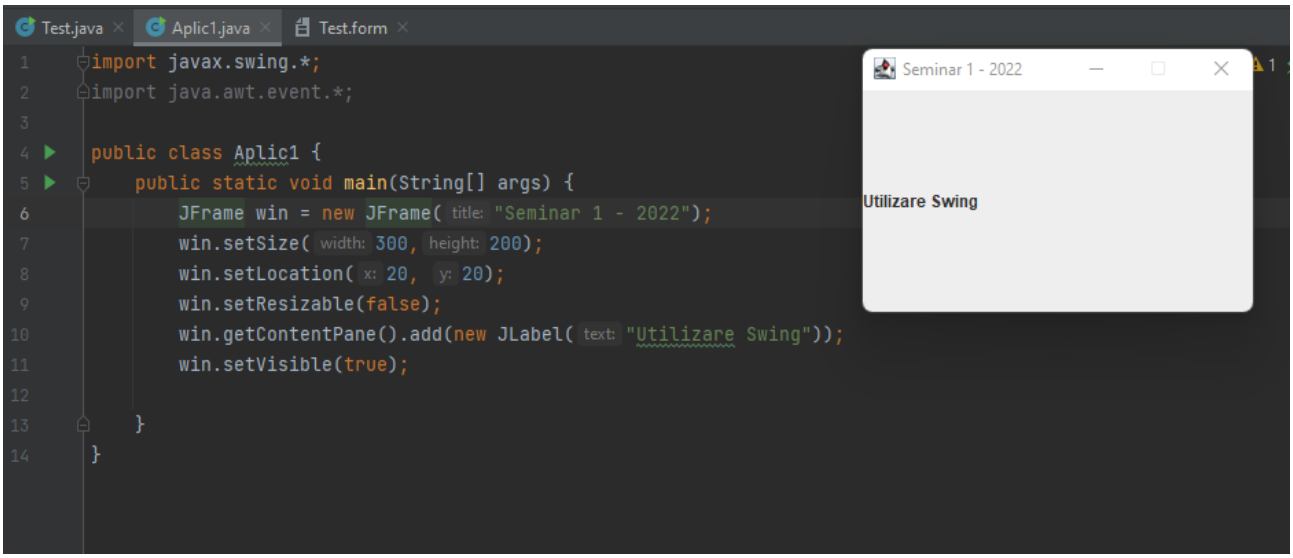
Stabilirea principalelor trăsături pentru ferestrele tip *frame* se realizează pe baza următoarelor metode:

- *setSize(*lăţime, lungime*)* pentru stabilirea dimesiunilor;
- *setResizeable(*true|flase*)* pentru a activa sau dezactiva posibilitatea redimesionării ferestrei de către utilizator;
- *setTitle(*"text"*)* pentru desemna un titlu afişat în bara superioară a ferestrei;
- *setLocation(x,y)* pentru a desemna originea colţului stânga-sus al ferestrei, de remarcat faptul că originea (0, 0) de la care sunt consideraţi factorii x şi y este colţul stânga sus al cadrului superior şi nu cel din stânga-jos;
- *setVisible(true|flase)* pentru a afişa sau ascunde fereastra, echivalent cu metodele *show()* şi *hide(),* recomandate a fi înlocuite cu această metodă.
- *pack();*

Pentru a accesa continutul unei ferestre se va folosi functia ***getContentPane***():

- ```
  JFrame win = new JFrame("Student la Informatica");
  ```
- ```
  Container c = win.getContentPane();
  ```
-
- Pentru a obtine inchiderea automata a aplicatiei atunci cand se apasa butonul de Close, se va utiliza metoda:
- ```
  setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
  ```
  ```

APLICAȚIA 1 curs



APLICAŢIA 2 curs