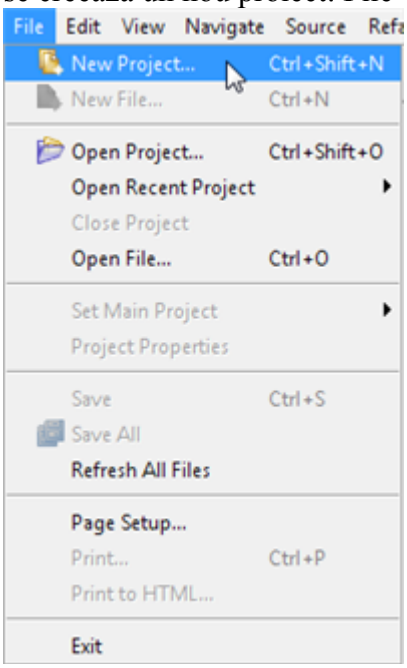
Laborator 3

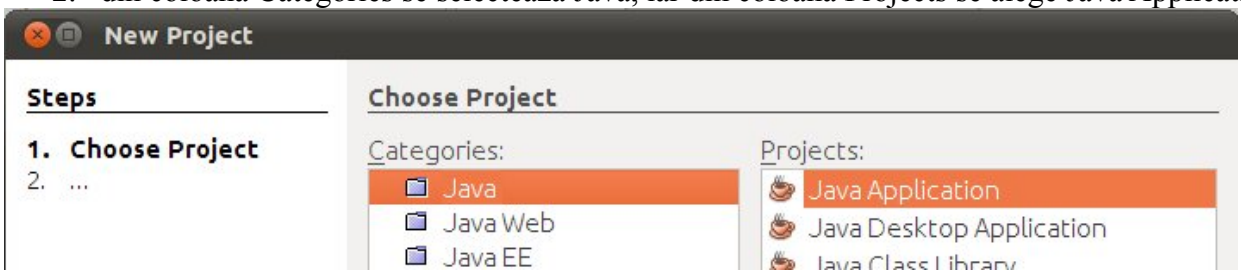
NetBeans IDE – dezvoltarea interfetelor grafice

NetBeans prezinta suport integrat pentru dezvoltarea proiectelor ce includ o interfata grafica cu utilizatorul. Pentru a crea un astfel de proiect se urmaresc urmatorii pasi:

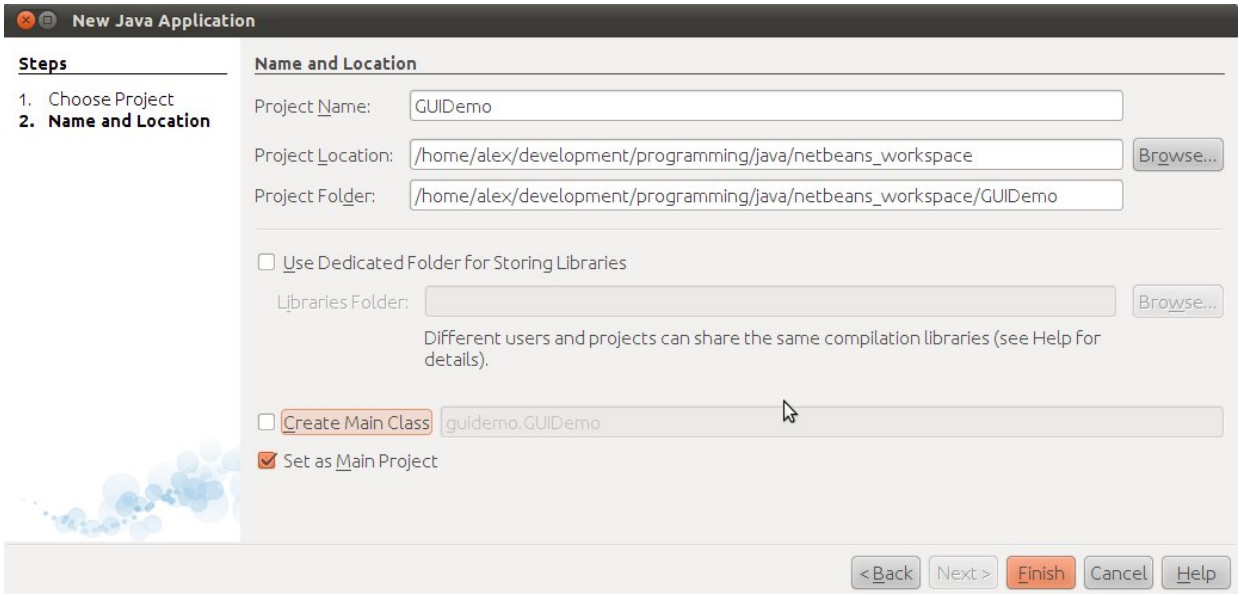
1. se creeaza un nou proiect: File → NewProject



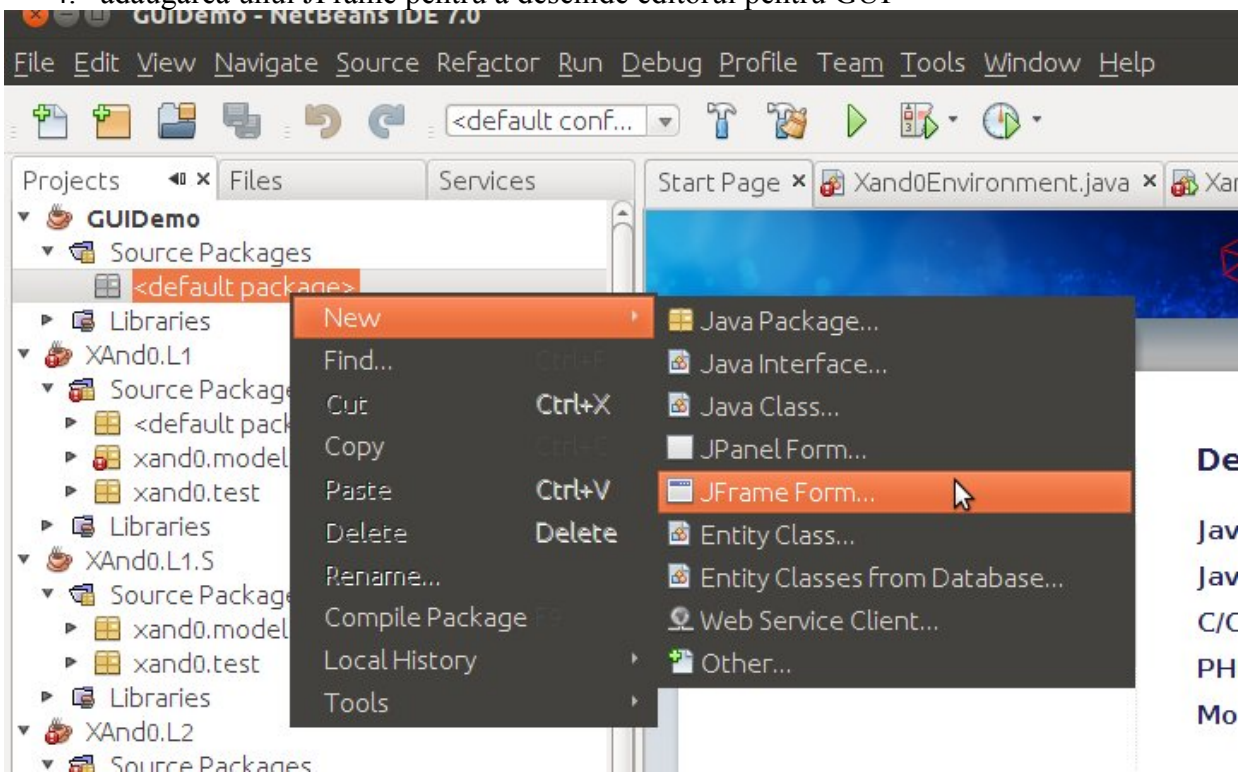
2. din coloana Categories se selecteaza Java, iar din coloana Projects se alege Java Application



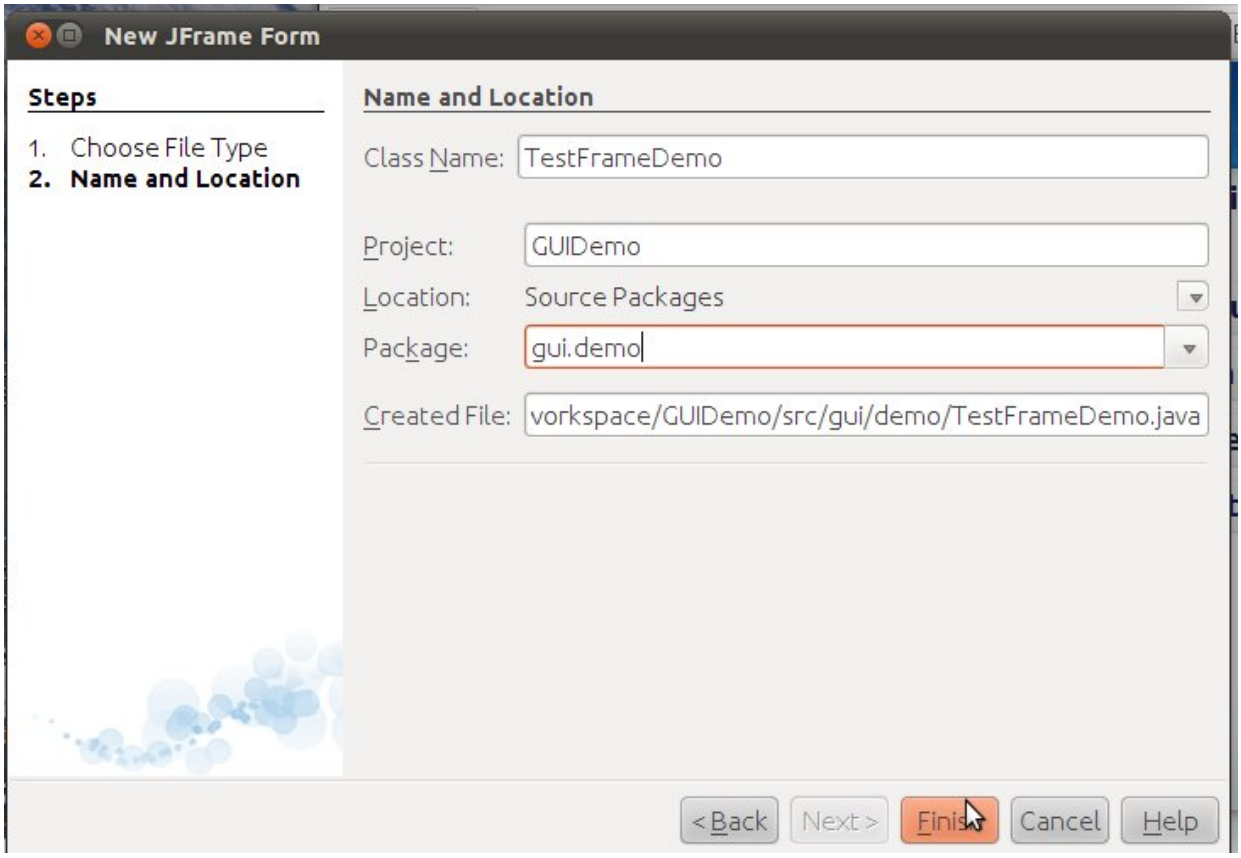
3. stabiliti numele proiectului (e.g. GUIDemo), deselectati Create Main Class



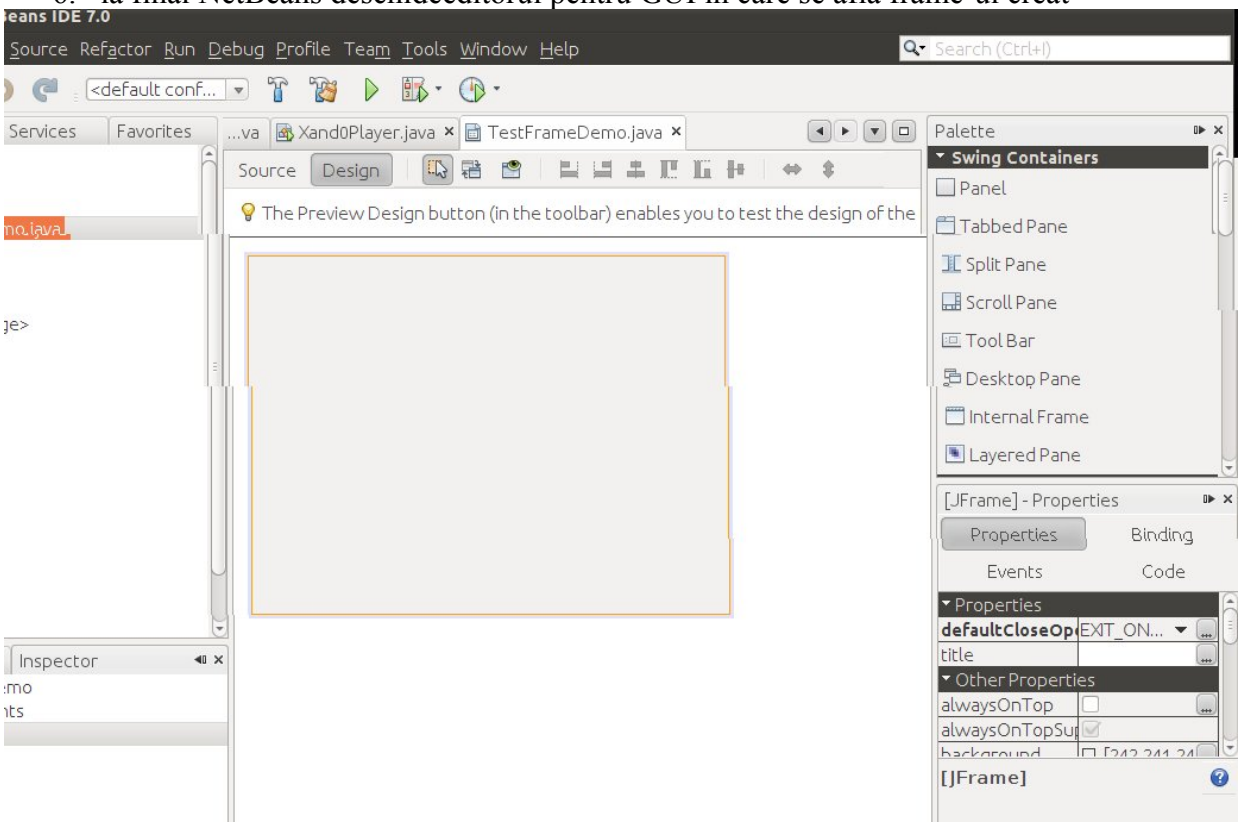
4. adaugarea unui JFrame pentru a deschide editorul pentru GUI



5. se denumeste clasa GUI

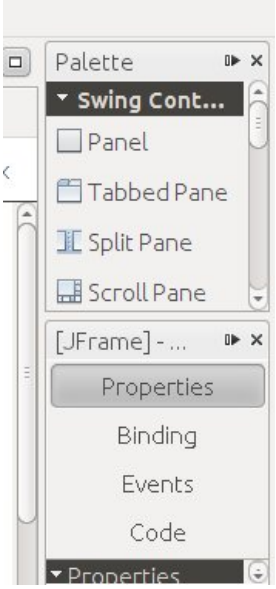


6. la final NetBeans deschideeditorul pentru GUI in care se afla frame-ul creat

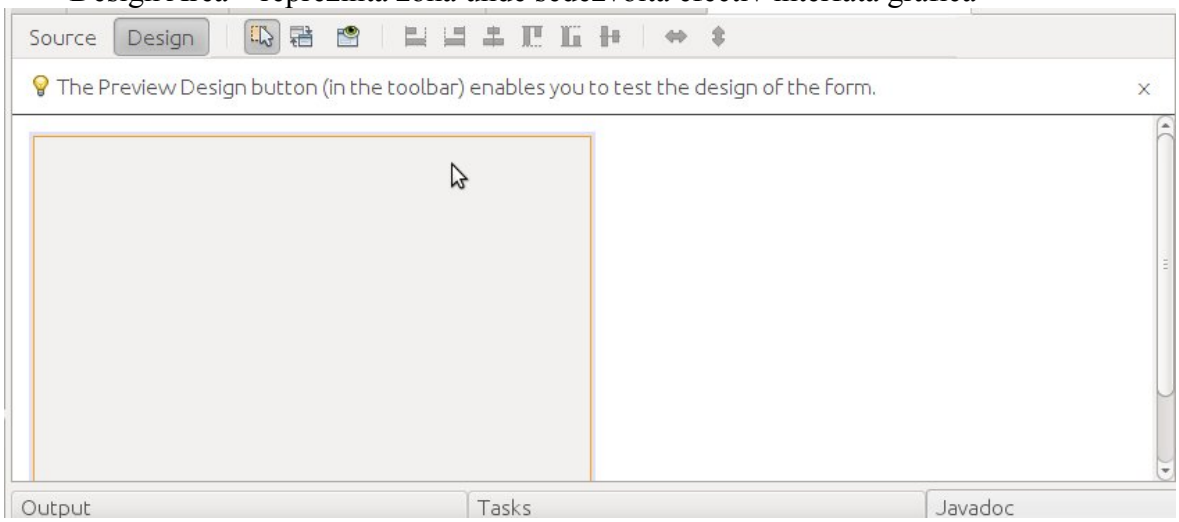


Editorul deschis de NETBeans pentru realizarea interfetelor grafice detine cateva zone de interes in dezvoltarea unui GUI:

- Palette – detine componenteleSwingcarealcatuiesc un GUI (vezi sectiunea urmatoare)

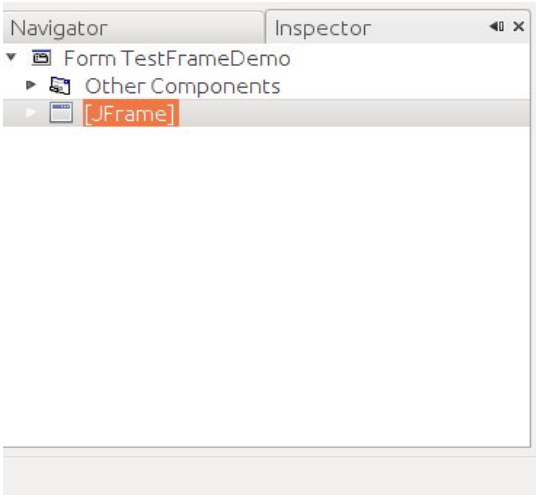


- Design Area – reprezinta zona unde sedezvolta efectiv interfata grafica



- Property Editor – permite parametrizarea si actualizarea caracteristicilor (dimensiune,pozitie, culori, text.mesaje, etc) pentru componenta graficaselectata in Design Area

- Inspector – ofera o reprezentare grafica a componentelor aplicatiei dezvoltate



Exercitiu: Adaugati un buton prin Drag-and-Drop din Palette in Design Area si observati cum acestaeste adaugat si reprezentat in Inspector.

Swing – API Java pentru dezvoltarea aplicatiilor GUI

Swing si JFC

Swing este un API ce face parte dintr-un set complex de componente, API-uri si trasaturi arhitecturale specifice dezvoltarii aplicatiilor de tip GUI denumit JFC – Java Foundation Classes. JFCeste utilizat in mare pentru imbogatirea experientei grafice a utilizatorului pri9n adaugarea diverselor functionalitati siinteractivitati grafice necesare la nivelul unui GUI modern.

Toolkit-ul Swing include un set amplu de componente pentru dezvoltarea interfetelor grafice, mai exactinclude toate componentele necesare in dezvoltarea aplicatiilor curente: tabele, controale, list de controale, arbori de controale, butoane, etichete, etc.

Cateva dintre capabilitatile oferite de Swing si JFC:

- **Componente Swing pentru GUI**
  - Toolkit-ul Swingdetine o varietate mare de componente – butoane, check box-uri, tabele, containere pentru text – fiecare oferind functionalitati sofisticate pentru a acoperio gama larga din cerintele aplicatiilor.
- **API pentru grafica 2D**
  - Pentru aplicatii care necesita componente grafice, la nivelul GUI, ce nu se regasesc in setul componentelor oferite de API-ul Swing JFC asigura Java 2D API si pentru ca Swing estedezvoltat peste acest API interactiunea siutilizarea acestuia la nivelul componentelor Swing este triviala.
- **Suport pentru formatul general de reprezentare al interfetei - Look-and-Feel**
  - Swing ofera suport pentru selecctia si dezvoltarea formatului de reprezentare vizuala a compoentelor graficele ce formeaza interfata grafica astfel incat acestea sa fie omogene in contextul reprezentarii vizuale caracteristice diverselor sisteme de operare.
- **Transfer de date**
  - Transferul datelor prin operatii de tip cut, copy, paste, drag-and-drop este asigurat in mod natural lanivelul toolkit-ului Swing asigurandu-se mecanismul necesar pentru a se realiza acesteoperatii in mod trivial intre orice componente Swing, intre aplicatii Java si intre aplicatii Java si aplicatii native.
- **Internationalizare**
  - Aceasta caracteristica ofera suportul necesar aplicatiilor de a fi dezvoltate astfel incat sa

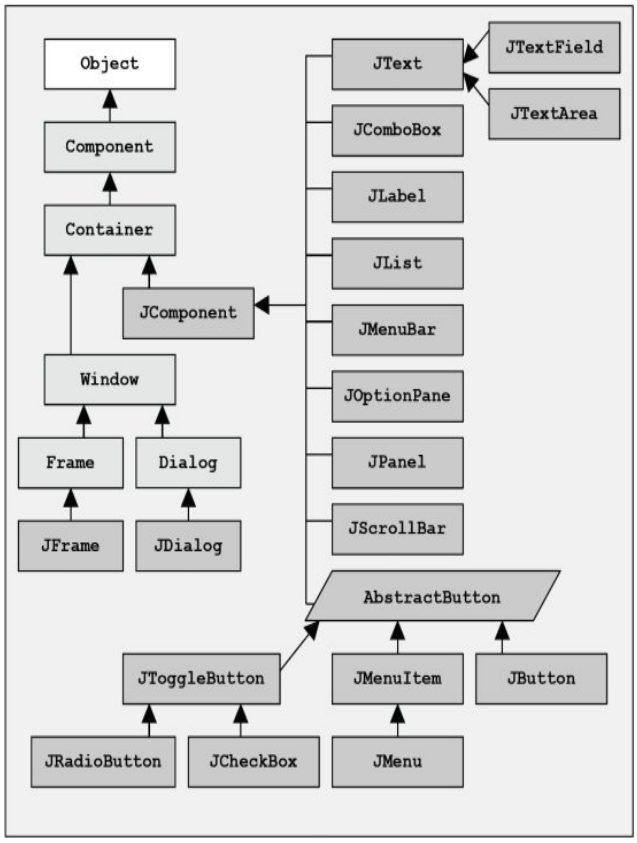
se poata asigura o interactiune cu utilizatorii folsoind limba nativa a acestora precum si conventiile culturale specifice acestora. (Datele de intrare se pot achizitiona atat folosind atat caractere latine cat si caractere chinezesti, japoneze, coreene)

- **API pentru accesibilitate**
  - Se asigura suport pentru tehnologii de asistare utilizate la nivelul aplicatiilor ce utilizeaza un suport software si hardware special (e.g. cumulare de informatii despre aplicatii care ruleaza pentru a o reprezenta intr-un format diferit cum ar fi Braille).
- **Undo Framework API**
  - Suport pentru implementarea facila a operatiilor de tip undo-redo.
- **Suport pentru instalarea si configurarea flexibila**
  - Se asigura un suport suplimentar astfel incat aplicatiile sa poata fi utilizate intr-un cadru cat mai larg de containere de aplicatii (e.g. rularea unui applet in cadrul unei ferestre deschide intr-un browser web).

API-ul Swing este ”puternic, flexibil si imens”[Sun], detine 18 pachete pubnlice fiecare detinand un numar impresionant de clase.

javax.accessibility	javax.swing.plaf	javax.swing.text
javax.swing	javax.swing.plaf.basic	javax.swing.text.html
javax.swing.border	javax.swing.plaf.metal	javax.swing.text.html.parser
javax.swing.event	javax.swing.plaf.multi	javax.swing.text.rtf
javax.swinf.filechooser	javax.swing.plaf.synth	javax.swing.tree
javax.swing.colorchooser	javax.swing.table	javax.swing.undo

O parte astructurii ierarhice aAPI-ului Swing este descrisa in figura de mai joscare prezinta componentele Swin (gri inchis) si cum extind componentele AWT(gri deschis) [Java COurse – Ch.06].



Componente Swing

Orice componenta grafica din API-ul toolkit-ului Swing, al careinume incepe cu “J” extinde clasa JComponent al carei API este descris in detaliu aici. Singurele componente Swing care nu respecta aceasta conventie sunt containerele de tip top-level cum ar fi JFrame sau JDialog.

JComponent extinde clasa Container care este o specializare a clasei Component. Clasa Component asigura suportul necesar pentru indiciile de pozitionare a componentelor, pentru desenarea acestora sau pentru declansarea evenimentelor de interfata. Clasa Container adauga suportul necesar pentru inserarea componentelor la nivelul unui container si pozitionarea acestora in functie de necesitatile aplicatiei dezvoltate.

API-ul clasei JComponent ofera informatiile necesare despre mecanismele oferite pentru parametrizarea componentelor grafice Swing. Astfel se ofera suport pentru:

- stabilirea particularitatilor de aspect grafic
  - void setBorder(Border)
Border getBorder()
  - void setForeground(Color)
void setBackground(Color)
  - Color getForeground()
Color getBackground()
  - void setOpaque(boolean)
boolean isOpaque()
  - void setFont(Font)
Font getFont()
  - void setCursor(Cursor)
Cursor getCursor()
- scrierea si citirea starii unei componente
  - void setComponentPopupMenu(JPopupMenu)
  - void setTransferHandler(TransferHandler)
TransferHandler getTransferHandler()
  - void setToolTipText(String)
  - void setName(String)
String getName()
  - boolean isShowing()
  - void setEnabled(boolean)
boolean isEnabled()
  - void setVisible(boolean)
boolean isVisible()
- tratarea evenimentelor
  - void addHierarchyListener(hierarchyListener l)
void removeHierarchyListener(hierarchyListener l)
  - void addMouseListener(MouseListener)
void removeMouseListener(MouseListener)
  - void addMouseMotionListener(MouseMotionListener)
void removeMouseMotionListener(MouseMotionListener)
  - void addKeyListener(KeyListener)
void removeKeyListener(KeyListener)
  - void addComponentListener(ComponentListener)

void removeComponentListener(ComponentListener)
    - boolean contains(int, int)
      boolean contains(Point)
    - Component getComponentAt(int, int)
      Component getComponentAt(Point)
    - Component setComponentZOrder(component comp, int index)
    - Component getComponentZOrder(component comp)
- desenarea componentelor
    - void repaint()
      void repaint(int, int, int, int)
    - void repaint(Rectangle)
    - void revalidate()
    - void paintComponent(Graphics)
- tratarea ierarhiei de apartenenta la un container
    - Component add(Component)
      Component add(Component, int)
      void add(Component, Object)
    - void remove(int)
      void remove(Component)
      void removeAll()
    - JRootPane getRootPane()
    - Container getTopLevelAncestor()
    - Container getParent()
    - int getComponentCount()
    - Component getComponent(int)
      Component[] getComponents()
    - Component getComponentZOrder(int)
    - Component[] getComponentZOrder()
- pozitionarea componentelor
    - void setPreferredSize(Dimension)
      void setMaximumSize(Dimension)
      void setMinimumSize(Dimension)
    - Dimension getPreferredSize()
      Dimension getMaximumSize()
      Dimension getMinimumSize()
    - void setAlignmentX(float)
      void setAlignmentY(float)
    - float getAlignmentX()
      float getAlignmentY()
    - void setLayout(LayoutManager)
      LayoutManager getLayout()
    - void applyComponentOrientation(ComponentOrientation)
      void setComponentOrientation(ComponentOrientation)
- obtinerea informatiilor de dimensiune si pozitie
    - int getWidth()
      int getHeight()

    - Dimension getSize()
      Dimension getSize(Dimension)
    - int getX()
      int getY()
    - Rectangle getBounds()
      Rectangle getBounds(Rectangle)
    - Point getLocation()
      Point getLocation(Point)
      Point getLocationOnScreen()
    - Insets getInsets()
- specificarea dimensiunii si pozitiei absolute
    - void setLocation(int, int)
      void setLocation(Point)
    - void setSize(int, int)
      void setSize(Dimension)
    - void setBounds(int, int, int, int)
      void setBounds(Rectangle)

Printre functionalitatile oferite de clasa JComponent descendentilor sai se regasesc:

- indicii de utilizare – Tool-Tips – prin utilizarea metodei `setToolTipText`
- desenarea componentei si atasarea marginilor – prin utilizarea metodelor `setBorder` si suprascrierea `paintComponent`
- utilizarea unui look-and-feel specific – un obiect instanta a clasei JComponent detine un obiect instanta a clasei ComponentUI - care realizeaza desenarea componentei in contextul aplicatiei, trateaza evenimentele, determina dimensiunile componentei, etc – ce specilizare a clasei ComponentUI este utilizata la instantierea acestui obiect depinde de aspectul curentutilizat -look and fell. Aspectul de reprezentare se poate manipula prin utilizarea `UIManager.setLookAndFeel`.
- suport pentru scheme de pozitionare – layout - `setMinimumSize`, `setMaximumSize`, `setAlignmentX`, and `setAlignmentY`.
- Suport pentru oepratii de tip drag-and-drop- DnD
- bufferizare dubla pentru o desenare fluenta a componentei pe ecran
- corelarea actiunilor unei componente relativ la combinatii de taste/chei – key binding.

Containere de nivel intalt – Top-Level Containers

Toolkit-ul Swing ofera trei containere de nivel inalt descrise in API prin clasele:

- JFrame
- JDialog
- JApplet

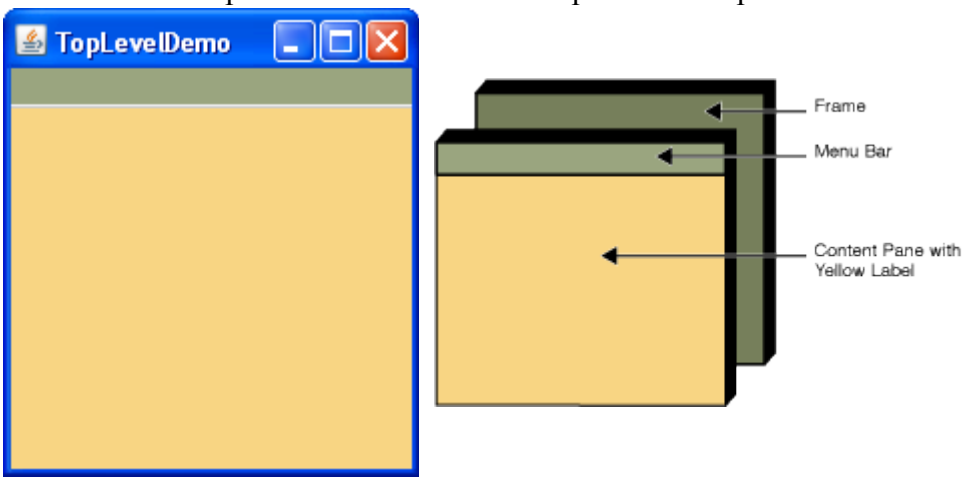
La utilizarea containerelor top-level este indicat:

- pentru a fi desentae pe ecran fiecare componenta GUI trebuei sa apartina unei ierarhii de apartenenta la containere – *containment hierarchy.* O astfel deierarhie descrie un arbore care are ca radacina un container de nivel intalt iar componentele formeaza frunzele arborerlui.
- orice componenta GUI poate fi continuta doar de un container. Adaugarea unei componente detinuta de un container intr-un alt container are drept efect eliminarea componentei din containerul initial si mutarea in containerul pentru care se doreste

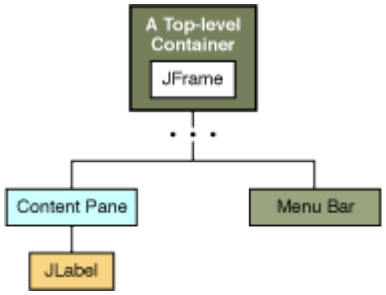
adaugarea.

- orice container de nivel inalt detine un panou de continut – content pane – care contine componentele grafice vizibile.
- orice container top-level prezinta optiunea de a detine o bara de meniu, pozitionata, prin conventie, in cadrul containerului dar in afara panoului de continut.

In figura de mai jos s-a creat un containerde tip JFrame care contine o bara de meniu – colorata in verde - si content pane-ul care contine o componenta de tip JLabel colorata in galben.



Ierarhia de apartenenta este descrisa in figura de mai jos.



Aplicatiile dezvoltate folosind API-ul Swing pentru realizarea interfetei grafice necesita cel putin un container de nivel inalt care sa contina componentele grafice asociate aplicatiei. Acest container reprezinta radacina arborelui ierarhic de apartenenta a componentelor grafice fata de container. In cazul in care sedezvolta aplicatii de tip standalone/desktop atunci va exista cel putino ierarhie a carei radacina va fi un container de tip JFrame. Ierarhiile de apartenenta sunt descoperite prin evaluarea cerintelor aplicatiei relativ la GUI – e.g. daca o aplicatie necesita treiferestre principale si un dialog atunci vor exista 4 ierarhii de apartenenta si astfelvorfiimplicate 4 containere de nivel inalt: 3 de tip JFrame si un JDialog.

Adaugarea componentelor in Content Pane

Pentru a adauga componente in Content Pane detinut de un container de nivel inalt se excuta urmatorii pasi:

1. se obtine referinta la content pane sau se obtine referinta cate un obiect capabil sa detina componente grafice (e.g. JPanel):
2. folosind metodele de tip **add** ale obiectului referit de referinta obtinuta precedent se adauga

componetele grafice:

3. in cazul in care s-a utilizat un nou obiect pentru formarea content pane-ului, se actualizeaza referinta catre acesta la nivelul containerului de nivel inalt:

```
public class TestFrameDemo {

	public static void main(String[] args) {

		SwingUtilities.invokeLater(new Runnable() {

			@Override
			public void run() {
				// TODO Auto-generated method stub
				JFrame frame = new JFrame("Some Title");

				// step 1
				Container contentPane = frame.getContentPane();
				JPanel otherContentPane = new JPanel();

				// step 2
				contentPane.add(new JLabel("Some Label"));
				otherContentPane.add(new JLabel("Some other label"));

				// step 3: use this step if using a dedicated JPanel
object
				frame.setContentPane(otherContentPane);


				// YOU SHOULD ALWAYS PACK AND SET THE FRAME TO
VISIBLE AFTER YOU've ADDED THE SWING WIDGETS
				frame.pack();
				frame.setVisible(true);

			}
		});
	}

}
```

Adaugarea barei de meniu

Desi orice container de nivel inalt poate fi dotat cu o bara de meniu, in practica utilizarea acestora este specifica in special in cazul containerelor de tip frame sau applet.

Pentru formarea unei bare de meniu se urmaresc pasii:

1. se creeaza obiect instanta a clasei JMenuBar
2. se creeaza obiecte instante ale clasei JMenu
3. se populeaza meniurile cu elemente de tip JMenuItem
4. se populeaza bara de meniu cu obiectele de tip meniu
5. se adauga bara de meniu containerului denivel inalt

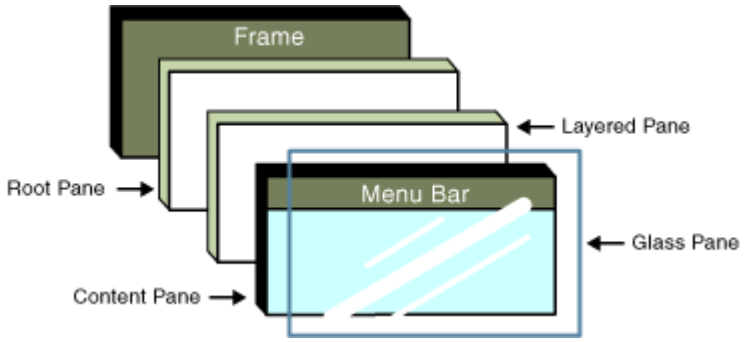
```
package gui.demo.frame;

import java.awt.Color;
import java.awt.Container;

import javax.swing.JFrame;
```

```java
import javax.swing.JLabel;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;

public class TestFrameDemo {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		SwingUtilities.invokeLater(new Runnable() {

			@Override
			public void run() {
				// TODO Auto-generated method stub
				JFrame frame = new JFrame("Some Title");

				Container contentPane = frame.getContentPane();
				JPanel otherContentPane = new JPanel();

				contentPane.add(new JLabel("Some Label"));
				otherContentPane.add(new JLabel("Some other label"));

				frame.setContentPane(otherContentPane);

				// step 1
				JMenuBar menuBar = new JMenuBar();

				// step 2
				JMenu fileMenu = new JMenu("File");
				JMenu editMenu = new JMenu("Edit");

				// step 3
				fileMenu.add(new JMenuItem("Open"));
				fileMenu.add(new JMenuItem("Save"));
				fileMenu.add(new JMenuItem("Close"));

				editMenu.add(new JMenuItem("Copy"));
				editMenu.add(new JMenuItem("Paste"));
				editMenu.add(new JMenuItem("Undo"));
				editMenu.add(new JMenuItem("Redo"));

				// step 4
				menuBar.add(fileMenu);
				menuBar.add(editMenu);

				// step 5
				frame.setJMenuBar(menuBar);

				frame.pack();
				frame.setVisible(true);
			}
		});
	}

}
```

Panoul Radacina – Root Pane

Fiecare container denivel inalt detine un container intermediar denumit *root pane*. Acesta administreaza content pane si bara de meniu impreuna cu alte acteva containere. Componentele oferite de root pane unui container de nivel inalt sunt:

- root pane
- layered pane – contine bara de meniu si content pane si gestioneaza ordonarea in adancime a componentelor (axa Z)
- content pane
- glass pane - utilizat la interceptarea evenimentelor care au loc "peste" containerul denivel inalt



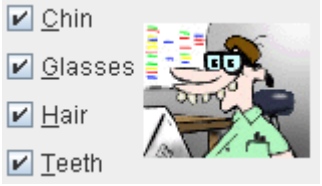
Exercitiu: Sa se implementeze in cadrul unor proiecte demonstrative, realizate folosind NetBean IDE, exemplele prezentate mai sus.

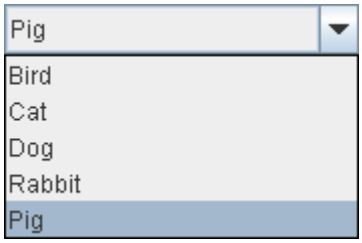
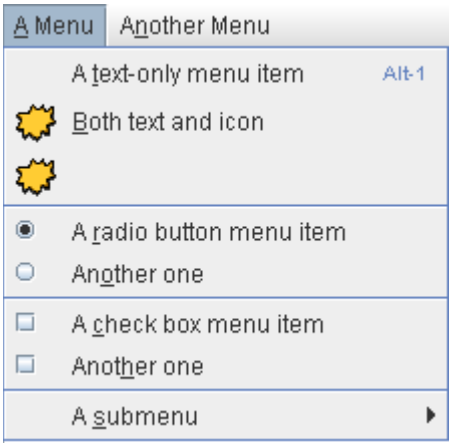
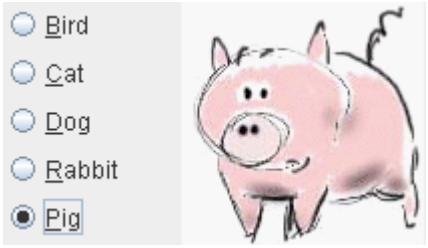
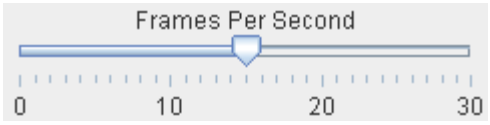
Swing widgets

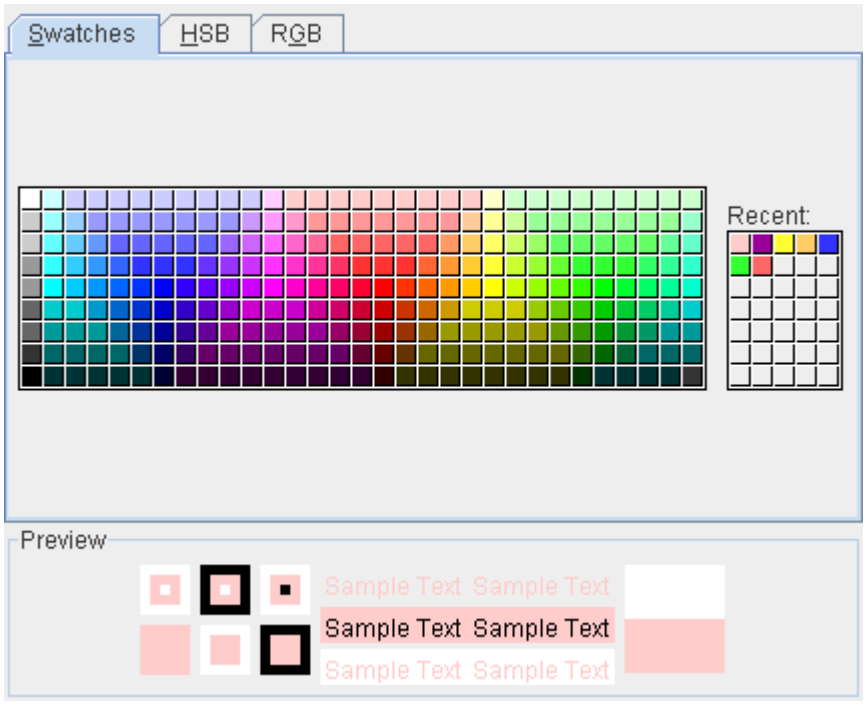
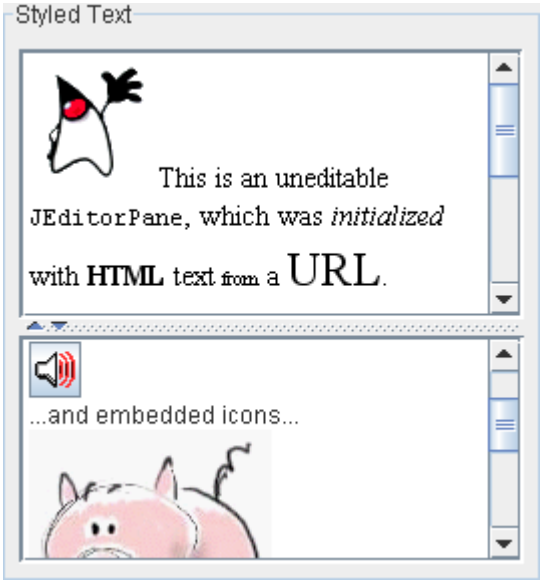
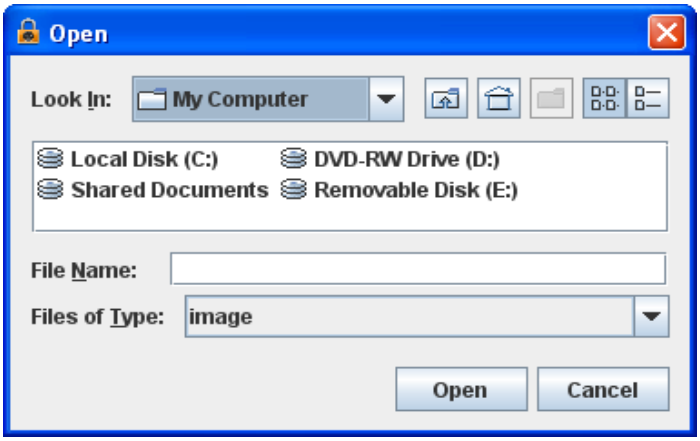
Un widget reprezinta o componenta grafica sau un ansamblu de componente grafice particularizate de programator in vederea indeplinirii cerintelor elaborate pentru parteaGUI a unei aplicatii. Swing ofera o serie de componente grafice ce extind JComponent, fiecare adaugand anumite caracteristici specifice elementului grafic pecare il integreaza. (e.g un buton prezinta o serie de imagini asociate si un comportament care tranziteaza intre aceste imagini, ceformeaza fundalul butonului, la apasarea butonului pentru a se realiza o indicare vizuala a faptului ca butonul a fost actionat).

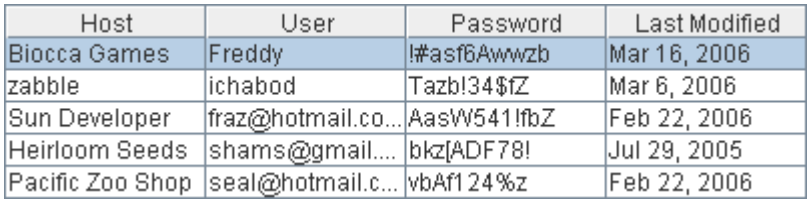
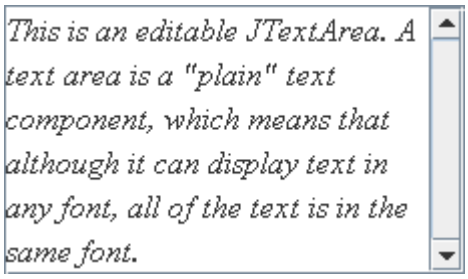
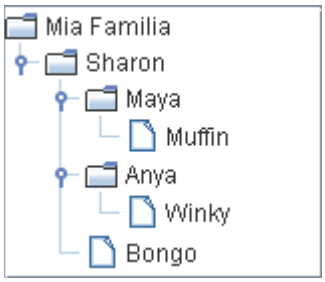
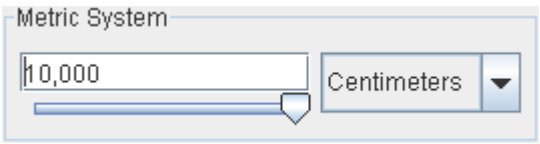
Nota: Pentru o utilizare eficienta a componentelor grafice oferite de toolkit-ul Swing este indicata acomodarea cu documentatia oferita prin API-ul Swing precum si parcurgerea link-urilor individuale din tabelul de mai jos – punctual in functie de componenta ce se doreste folosita.

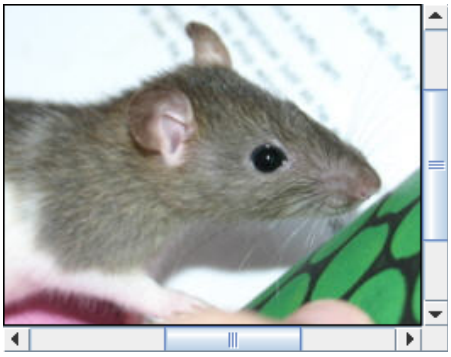
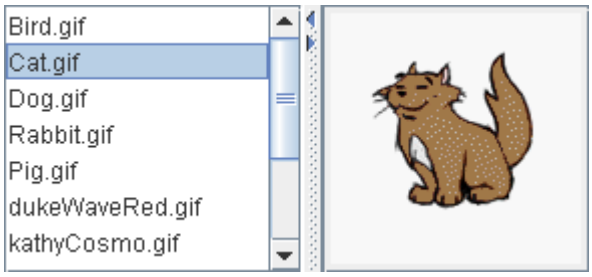
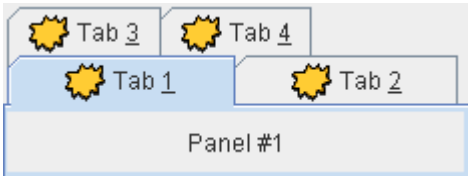
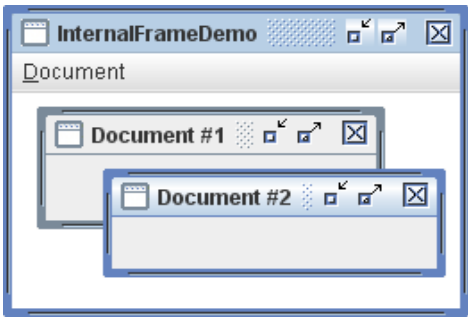
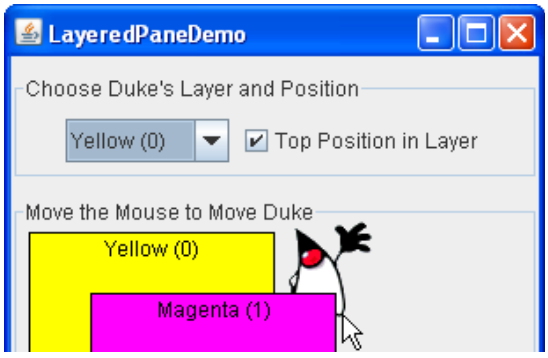
e.g. pentru componenta JFileChooser in documentatia API se ofera modalitatea deinitializare si utilizare a acestei componente.

Controale de baza	
	JButton
	JCheckBox

	JComboBox
	JList
	JMenu
	JRadioButton
	JSlider
	JSpinner
	JTextField
	JPasswordField
Afisari interactive pentru date organizate	

	JColorChooser
	JEditorPane si JTextPane
	JFileChooser

	JTable
	JTextArea
	JTree
Afisarea de date needitabile	
	JLabel
	JProgressBar
	JSeparator
	JToolTip
Containere de uz general	
	JPanel

	JScrollPane
	JSplitPane
	JTabbedPane
	JToolBar
Containere dedicate	
	JInternalFrame
	JLayeredPane

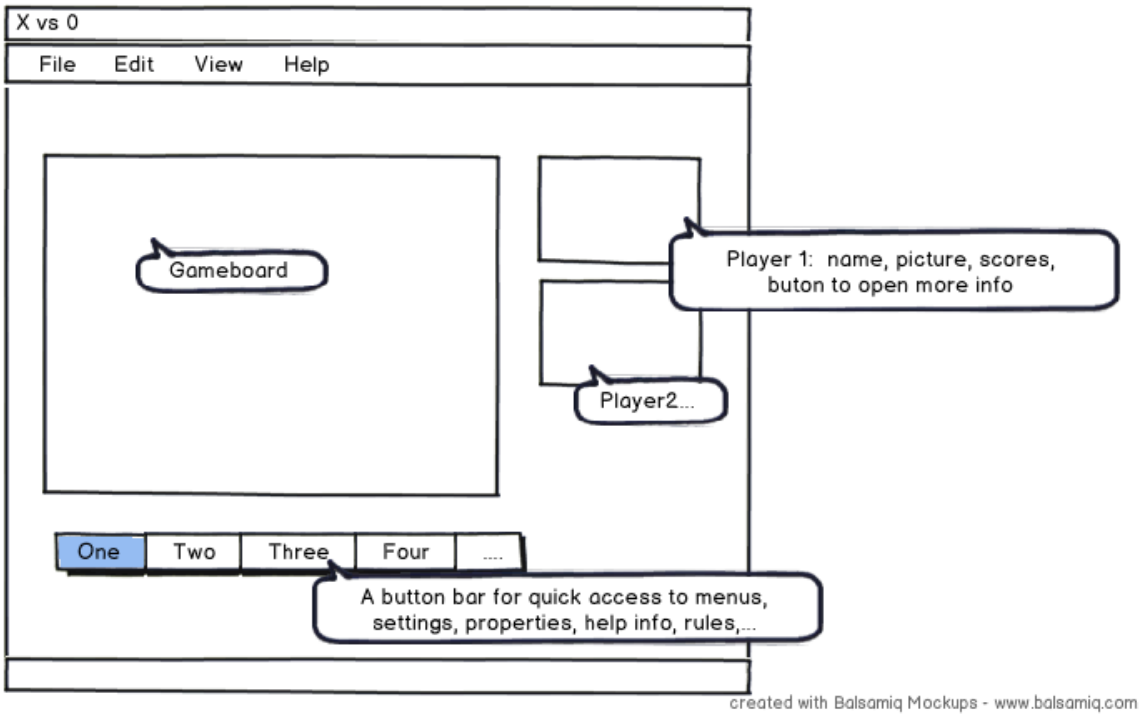
Exercitiu: Realizati un mic proiect demonstrativ care la inceperea aplicatiei deschide o fereastra care reprezinta un JFileChooser. Se va selecta un fisier text iar continutul lui va fi afisat intr-o noua fereastra intr-o zona detip JTextArea.

“X and 0” GUI

Exercitiu: Pornind de la macheta interfetei grafice, reprezentata in figura de mai jos:

1. realizati o analiza asupra cerintelor necesare aplicatiei (atat din punct de vedere functional cat si din punct de vedere al interactiunii la nivel grafic cuutilizatorul); identificati componentele grafice Swing necesare acestui GUI si completati macheta cu orice alte widget-uri/containere considerati necesare; in cazul in care analist-ul decide utilizarea unui widget complex (e.g. zona de informatii jucator) acesta se poate detaliaseparat (fie ptrintr-o macheta de tipul celei din figurade mai jos, fie prin detalii in format text)
2. completati detaliile (e.g. butoanele One, Two, Three) astfel incat sa se obtina un GUI functional din punctul de vedere al interactiunii cu utilizatorii:
    - modificati macheta existenta
    - parametrizati componentele alese (e.g. completarea barei de meniucu meniuri elocvente)
3. realizati o implementare minimala a acesteia folosind NetBeans IDE

NOTA: Analiza si design-ul unui GUI urmaresc linii generale pentru dezvoltare insa reprezinta o munca individuala. In etapa analitica trebuie motivate: selectarea unui anumit model de organizare grafica a componentelor, modelul culorilor folosite (e.g. o alegere nu tocmai inspirata de schema de culori este un background galben si un font alb), componentele utilizate - in special containerele (e.g. zona dedicata informatiilor utilizatorilor reprezintaun widget complet in care exista un container JPanel care poate contine un JLabel cu numele jucatorului, un JLabel cu avatarul sau imaginea reprezentativa a jucatorului, posibil un JTextArea needitabil pentru afisarea ultimelor scoruri,un buton cu o grafica speciala pentru accesarea unor informatii suplimentare desprejucator, un buton special dedicat fiecarui jucator pentru editarea si modificarea profilului acestuia), mecanisme speciale de actualizare si redesenare. In etapa de design sevor respecta si actualiza specificatiile elaborate in etapa analitica.



NOTA: O descriere detaliata a tuturor conceptelor precedente se poate consulta la aceasta adresa.