UNIT-2

INTRODUCTION TO PROGRAMMING

Introduction:

Program: A set of structured instructions.

Programming: The process of writing specific instruction in a computer language.

Programmer/coder: Person who write such programs.

Programming languages: Language used in computer program.

What is java:

It is originally known as **'oak'**, a computer programming language (also known as high level language).

It has set of rules that determine how the instructions are to be written.

These rules are known as syntax.

History of Java:

James Gosling in 1991 developed the java language.

It is a **Write Once Run Anywhere** (WORA) type language providing no cost run time popular platforms.

Advantages/characteristics of java:

1.object oriented:

It is purely object-oriented language.

It helps the programmers to develop more reliable & error free code.

2. Platform independent:

Java is a platform independent which is portable and easily executed on all operating systems (windows, Linux, Solaris).

3. Simple:

Java is designed to be easy to learn.

4. Secure:

Java generates an intermediate code called Byte code which is very secure because nobody can understand if anyone tries to access the code.

5. Portable:

It ensures portability in 2 ways:

First java compiler generates byte code instructions that can be implemented on any machine.

Secondary the size of primitive data types are machine independent.

6. Robust:

Java eliminates error by focusing mainly on compile time error checking & run time checking.

7. High performance:

With the use of Just In Time (JIT) compilers, java enables high performance.

8. Distributed:

Java is designed for distributed environment of internet.

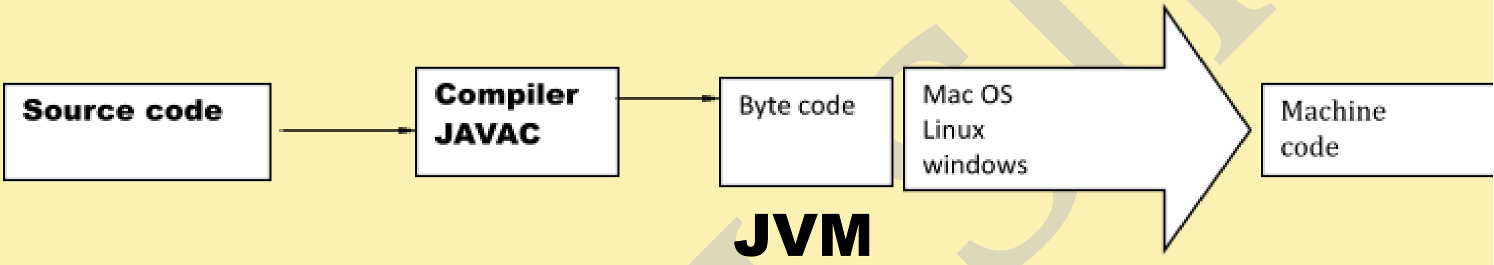
Java virtual machine (JVM)

JVM is a run time environment which acts as an interpreter & translates the byte code into object code (machine language).

It is a platform independent execution environment.

As C & C++ platform dependent so developers of java wanted it to be platform independent language.

Hence the concept of JVM was included in java.



<u>Source code:</u>

The core program written in any computer language (like C, C++, java).

It is a collection of computer instructions written by using any human readable computer language.

<u>Object code:</u>

The program in the form of machine instructions or binary instructions (computer readable form).

In case of java it is produced by JVM.

<u>Byte code:</u>

In java when source code is compiled, it doesn’t directly convert into object code rather it converts into byte code.

Byte code is machine instruction.

<u>Note:</u>

Javac–compiler that converts source code to byte code.

JVM – interpreter that converts byte code to machine language code.

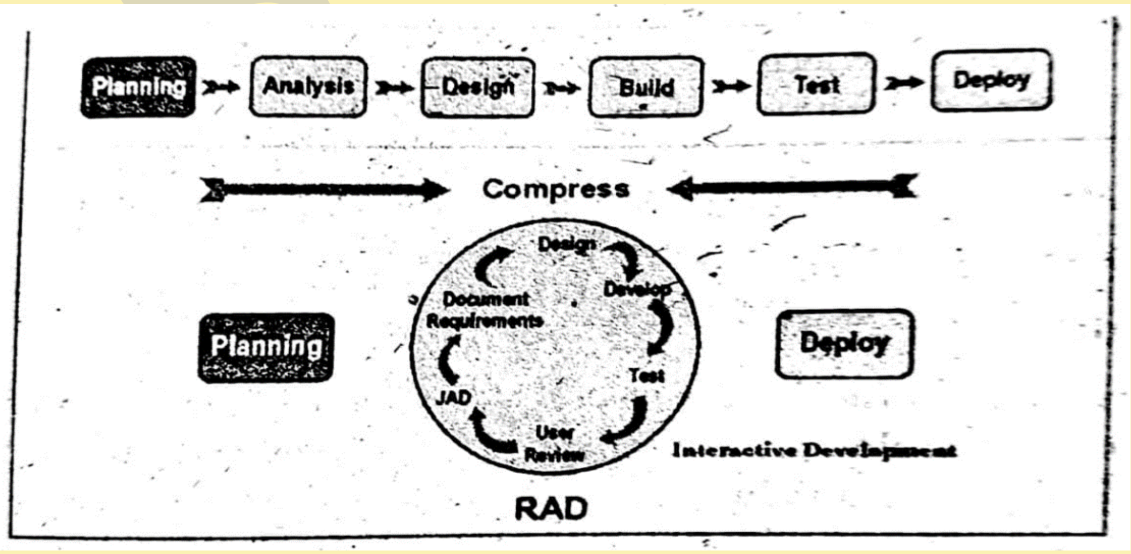
Source code – (program.java) →compile →byte code

Byte code – (program. class) → JVM →machine code

Rapid Application Development (RAD):

- **Rapid Application Development is abbreviated as RAD.**
- **It is an object-oriented approach for software development.**
- **It includes method of development as well as software tools known as ICU (I - iterative development, C - Construction of prototype, U - use of CASE – Computer Aided Software Engineering tools)**
- **It includes analyzing, designing, building & using phases.**

Diagrammatic representation of RAD:



Two methods of RAD system:

1. NetBeans IDE:

It is a cross platform RAD tool for creating visual desktop & web application for Linux, windows, MacOS.

2. My eclipse:

It is a RAD environment focusing on web application development.

Advantages of RAD:

- Increase speed of software development.
- Reduced complexity of development.
- Simplicity & usability of GUI components.

Java Development Kit (JDK):

Java programs are mainly of two types:

Applets: These are the programs executed by web browser.

Applications: These are the programs executed directly on your machine.

JDK is a program development environment for writing java applets & application.

Java Development Kit includes serial tools:

1. JRE (Java Runtime Environment)

It used to run the predeveloped applets or applications in JDK.

2. Javac

The compiler is known as javac.

Converts source code → compiled byte code.

3. Java

Java interpreter / coder normally called as java.

Executes java applications.

4. Java doc

Generates source code → html files documentation.

5. Jar

It is an achiever to create jar files.

It contains class, image & sound files for java application or applets gathered into a single file &possibly compressed.

6.Applet Viewer

It provides environment for testing applets.

It loads html files & displays the application in a browser.

Integrated Development Environment (IDE):

- **It is abbreviated as IDE.**
- **It is a software tools which create & debug programs easily.**
- **IDE brings all programs easily.**

- IDE brings all programming tools into one environment place along with project management tools to increase programmer productivity.
- IDE include:

1.source code editor →programmers write source code.

2.compiler → source code→intermediate code

Interpreter → execute compiler generated byte code.

3.Build automation tools help to automate the processes like compiling, debugging & deployment.

4.GUI simplifies the creation of GUI.

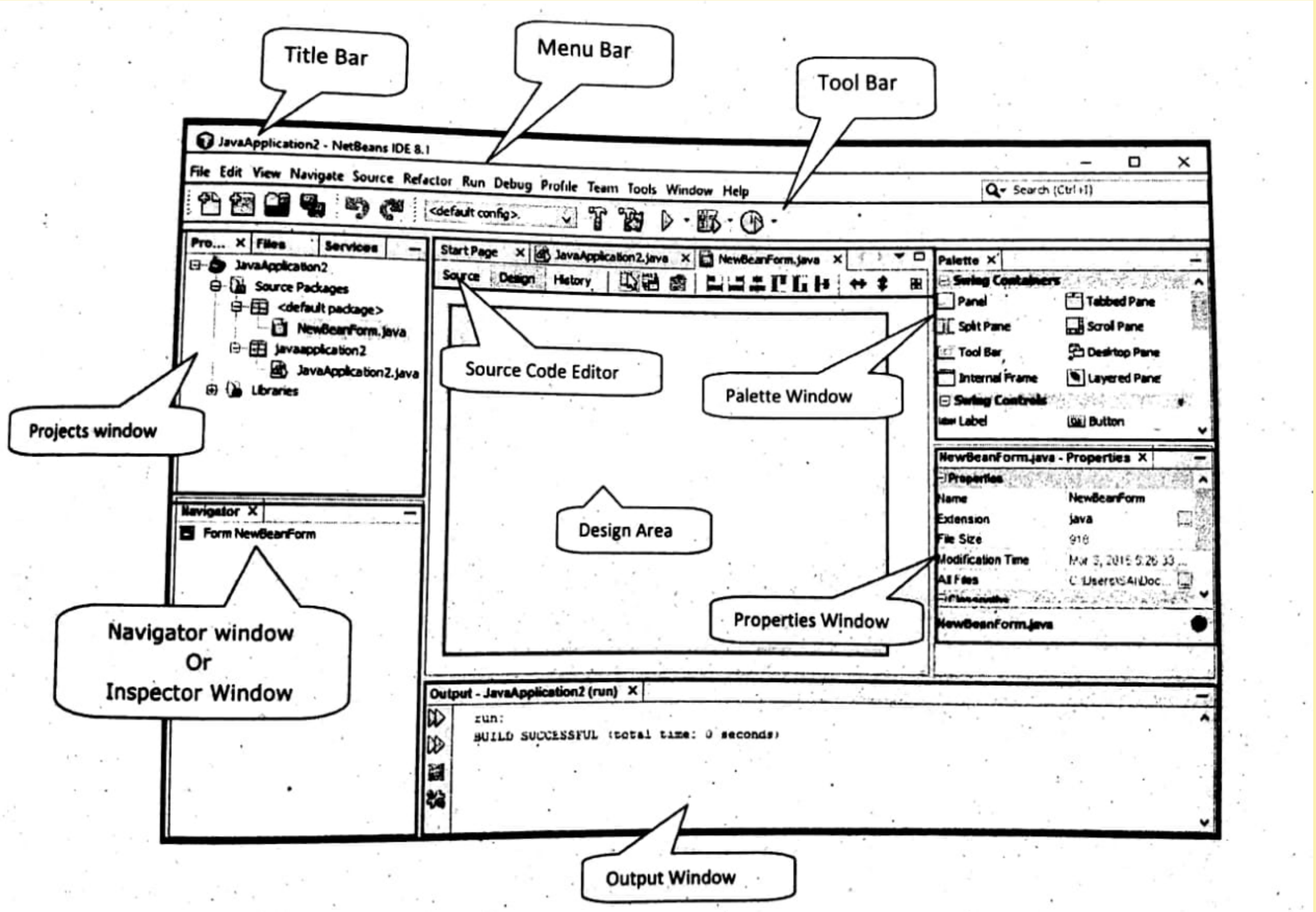
IDE tools:

Net Beans IDE:

- Net Beans IDE is an Integrated Development Environment used as a tool for programmers to edit, compile, debug & deploy programs.
- It is a free product developed under open source license with no restriction on use.
- The IDE runs on windows, Linux, MacOS and other UNIX based system.

- **NetBeans IDE incorporates many tools to facilitate creating, managing and running a program included features like a Graphical User Interface (GUI) for development, a fancy text editor, an o/p window, button, to compile and run the program, a debugger.**

Different parts of IDE windows:


Title Bar
Menu Bar
Tool Bar
JavaApplication2 - NetBeans IDE 8.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects window
Source Code Editor
Palette Window
Design Area
Navigator window Or Inspector Window
Properties Window
Output Window

```
run:
BUILD SUCCESSFUL (total time: 0 seconds)
```

The Net Beans IDE contains various controls and menus via which, a user can easily interact and work according to their requirements.

1. **Title Bar:** It displays the name of the application or a program.
2. **Menu Bar:** It provides the various commands used to work with Nets Beans IDE, such as file, edit, view, etc.
3. **Tool Bar:** It provides a way to quickly access the commonly used commands.
4. **Palette Window:** This window contains all the containers and other graphical components such as labels, panels, text fields etc.

5. **Projects Window:** This window shows a logical structure of different project content.
6. **Inspector Window:** This window displays a hierarchical tree structure of all the components used in the current form.
7. **Properties Window:** This window displays the editable properties of the currently selected graphical component.
8. **Design Area:** It is the primary area used for creating and editing of Java forms.
9. **Source Editor:** It is a fully featured text editor program which includes a compiler, debugger and other components of IDE.
10. **Output Window:** This window shows the output messages from the Net Beans IDE. Output window automatically show the messages after compilation of a program.

Integrated modules of IDE:

1.NetBeans profile:

It is a tool for monitoring of java application. GUI design file enables developer to prototype and position graphical component.

2.Java editors

Notepad: On windows machine we can use any simple text editor like notepad.

NetBeans: It is a java IDE that is open source and free.

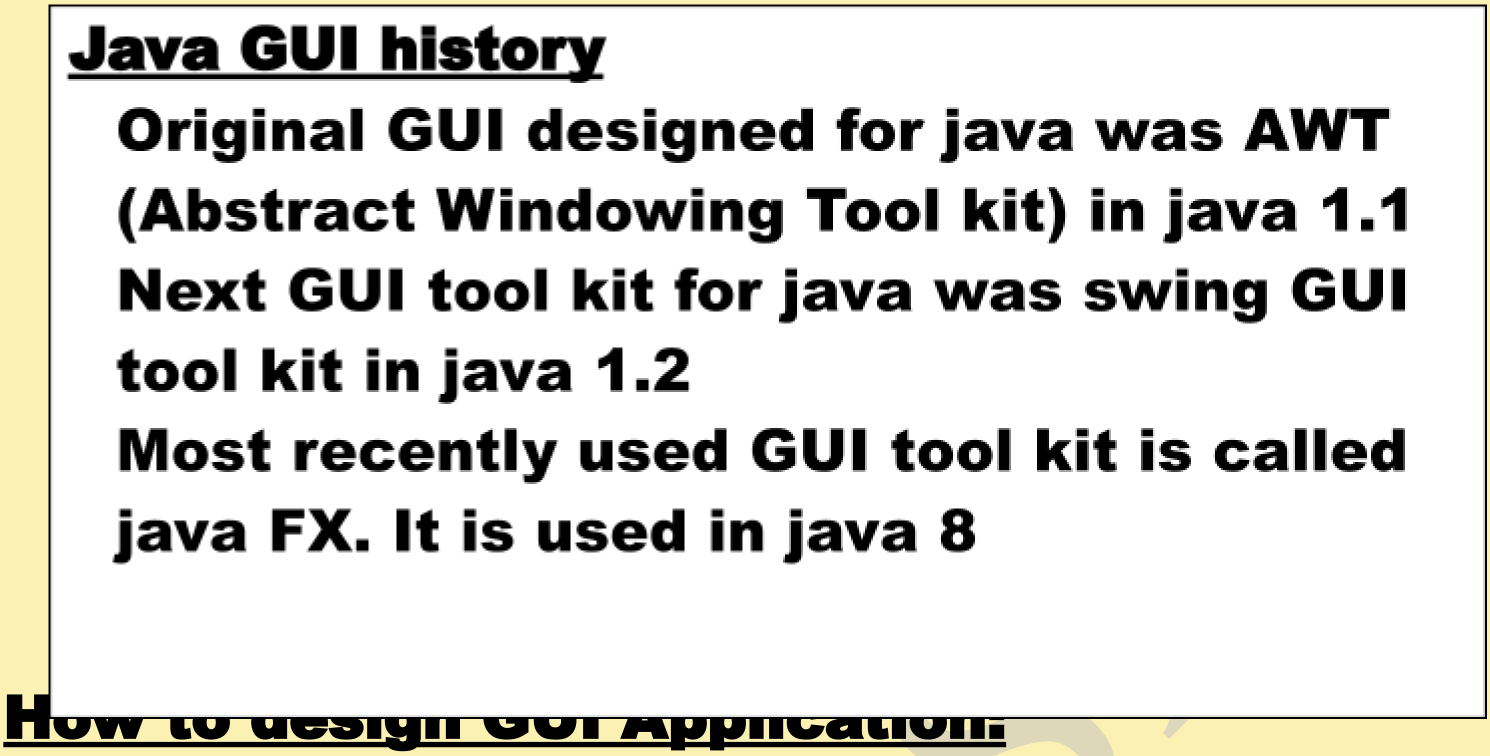
Eclipse: It is also a java IDE developed by eclipse open source community.

Concepts of GUI (Graphical user interface):

- A GUI is a graphical user interface to the computer.
- Applications use elements of GUI that come with operating system.
- Examples of GUI includes windows, pull down menus, buttons, scroll bars, iconic images, the mouse.
- GUI operating system monitors each type of windows for its functions through activities, these activities are called events.
- New forms of input are called events and style of programming used to process is called event-driven programming.
- Event is an occurrence of an activity.

Rapid Application Development with IDE:

- It is a programming system that enables programmers to quickly build working programs.
- RAD system provides number of tools to help build **Graphical User Interface.**
- Some of the most popular RAD systems for windows are visual basic and NetBeans is highly suitable on java programs.



Java GUI history

Original GUI designed for java was AWT (Abstract Windowing Tool kit) in java 1.1

Next GUI tool kit for java was swing GUI tool kit in java 1.2

Most recently used GUI tool kit is called java FX. It is used in java 8

How to design GUI Application:

Basic step to create and display GUI application:

1.Create forms: forms can be created within existing projects.

Form contains containers, sub-containers and components.

2.Edit forms: a component within a form can be modified directly or through the project editor.

3. Preview forms: with this designing capability the form can be tested without compiling or running it.

4.Deploy the GUI application:

Applications are distributed as JAR files.

Jar should contain all necessary libraries.

GUI programs in java is created by using following three types of software:

1.Graphical components:

It includes text box, button or menu. Each component has properties such as size, color, title, alignment etc.

2.<u>Event:</u>

It gets information about the events and respond to them.

3. <u>Event handle</u>:

It does the work for the user in case of event occurs.

<u>Role of API in java:</u>

- **API (Application Programming Interface) is a collection of pre-written packages, classes and interface which helps the interaction between humans and computers.**
- **Java programming books are performed by API's classes which helps to minimize no of lines written within pieces of code.**

<u>JDK components:</u>

- **Java compiler**
- **Java Virtual Machine (JVM)**
- **Java Application Programming Interface (API)**
- **The API is a library of available java classes, packages, and interfaces.**

The three type are as follows

(i) Official java core API, which is bundled with JDK download.
(ii) Optional official java API's, which may be downloaded when needed.
(iii) Un-official API's, which are third party API's that may be downloaded from source websites.

Three Frames of java:

- **The First frame shows an API component (classes and packages).**
- **The Second frame shows an interfaces, classes and exceptions of a particular selected packages.**
- **The Third primary frame provides an overview of all API packages to show the index, class hierarchy and help sections.**

Java GUI:

- **Java AWT (Abstract Windowing Toolkit) is an API to develop GUI.**
- **They are platform depended; the components display according to the view of operating system.**
- **Java AWT package provides classes such as text field, label, text area, radio button, check box, choice, list etc.**

Java GUI tool kit:

GUI is created with the following objects, Components (defers screen element such as text field, scroll bar, menu)

Events

Listeners

Graphical components available in java:

They are of 2 types:

- **Container control**
- **Child control**
- **Container controls hold other control within it. Ex- panel or frame**
- **Controls inside the container are called child controls. Ex- button, list**

NOTE: GUI application contains top lens, contain that hold another context in it.

SWING:

- **Swing is a GUI widget tool kit for java.**
- **It is a part of parcel Java Foundation Classes (JFC) which is an API for doing a Graphical User Interface (GUI) java program.**
- **Swings are written entirely in java and they are platform independent.**
  So they are termed as light weight.
- **It includes all the components from a modern tool kit like table controls, list controls, tree controls, buttons and labels.**

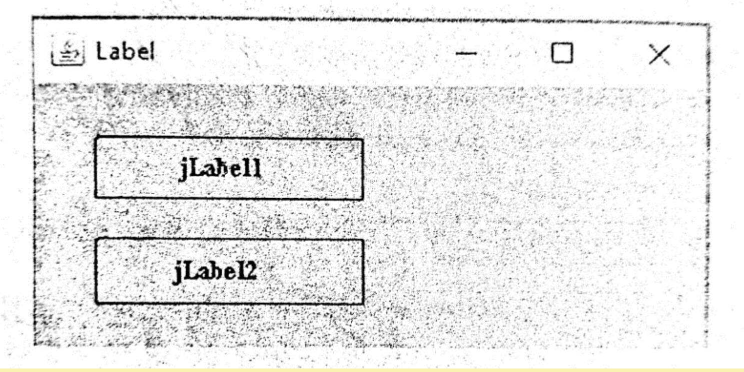
Difference between Java AWT and Java swing:

Java AWT	Java swing
1. AWT components are platform Depended.	1.Javaswingcomponents are platform Independent.
2.Components are heavy weight.	2.Components are light weight.
3.Provides less components than swing	3.Provide more powerful components Such as tables, lists, scroll panel.

Components of SWING:

1. Label:

- It is used to display text that a user cannot edit directly.
- It is created through jLabel component.
- A label displays a single line of read only text.



Properties

- **Background:** Set the background color of the label.
  Color of label is displayed when opaque property is true.
- **Enabled:** It is used to set weather the label is active or not or at run.
- **Font:** This property is used for set font style for text.
- **Foreground:** It sets the foreground color or it display in the label.
- **Text:** Used to specify the text to be displayed in the label.

Methods

- **void setText(string)**
  Ex. <label_name> setText ("Name");
- **String, getText()**
  Ex. String str = <Label_name> . getText();

2. Text field

- Text field is used to display information entered by the user at the run time.
- Numeric data cannot be displayed in a text format.
- It is derived from jTextField component.

Properties

1. Background: It sets the background color of the text field.

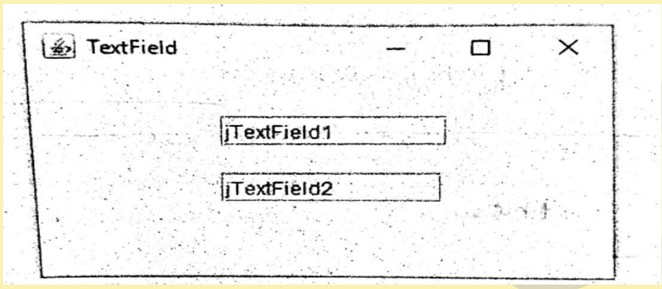
2. Edittable: It is used to edit context of the text field. It should be set to true that the user can edit it on run time.

3. Font: It sets a font style for the text to be displayed in the font field.

4. Enabled: It sets the font field as activated or deactivated.

5. Foreground: It sets the foreground color of text to be displayed in the text field.

6.Text: Specifies the text should be displayed in the text field.

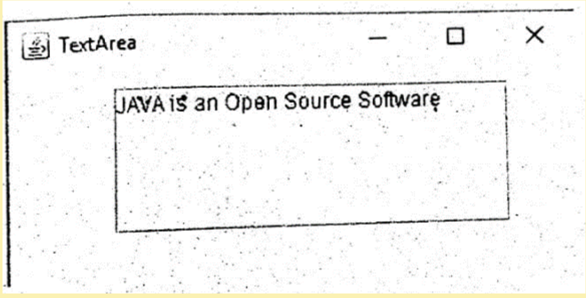


Methods

- **void setText (string)**→used to set the text displayed by the text field.
  Ex:<text field _name>. set .text("Name");
- **String getText ()**→used to fetch the text displayed in a text field.
  String str= <text field _name>. getText();
- **IsEditable ()**→ returns the current state of text field.
  If value true than user can edit the text available.
- **IsEnabled ()** → enables whether the text is in activated state or deactivated.

3. Text Area

- It provides large space for text entries.
- It is created through jTextArea component class.



JAVA is an Open Source Software

Properties

1 . Background

2 . Foreground

3 . Enabled

4 . Editable

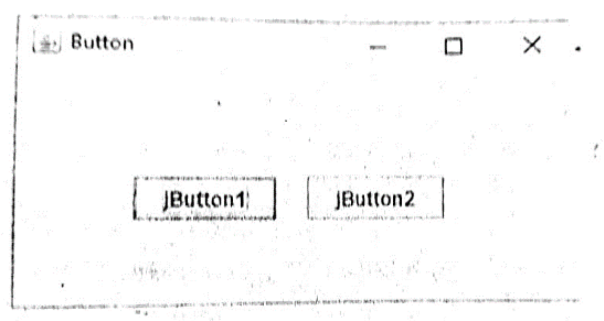
Methods

String getText()

Void setText(String)

4. Button

- A button is created through jButton class.
- Button is generally known as command button.
- Used generally to begin, interrupt or end a particular process.
- A button is used either by clicking on or by texting to it and pressing enter.



Properties

1 .Background :sets the background color of the button.

2 .Enabled

3 .Font

4 .Foreground

5 .Text

6.Label : It is used to set the label of the button.

Methods

Void setText(string):is used to set text which will be displayed on the button.

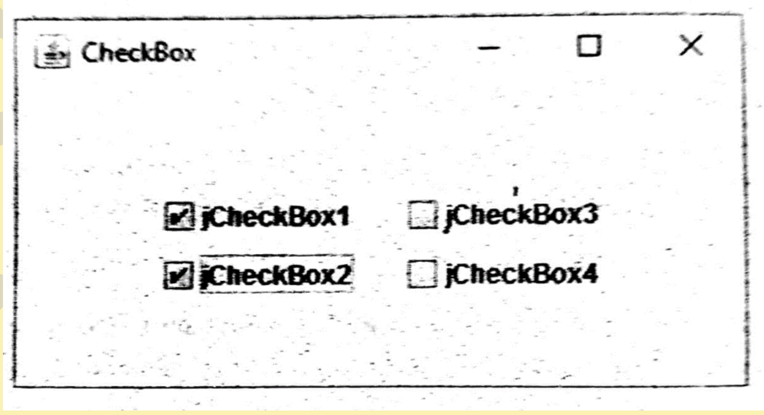
<button _ name> .setText ("Push Button")

String getText() :method sued to fetch the text displayed on the button.

String STR = <button _ name> .getText()

5. Checkbox

- **It is a graphical component that can be either "on" (true) or "off" (false) state.**
- **It is created through jCheckBox class.**
- **Each time the user clicks it, its state changes.**



Properties

1.Button Group : It sets group of buttons in which current control belong.

2.Font : Specifies the font color of the checkbox.

3.Foreground : Sets the font color of the checkbox.

4.Label : Used to set the checkbox label.

5.Selected : It sets the current value of the checkbox. if true it is displayed as checked otherwise unchecked.

6.Text : Sets the text should be displayed with the checkbox.

Methods

void setText(String STR) : It sets the text which will be displayed with the checkbox.

For Ex :<checkbox _name> .setText("New Checkbox");

String getText() : It fetch the text displayed with the checkbox.

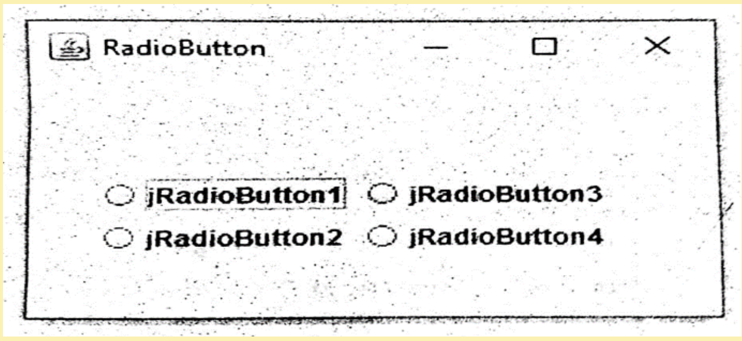
Boolean isSelected() :relates the current state of the checkbox.

void setSelected(Boolean B) :used to set the current state of the checkbox.

6. **Radio button**

It is a group of mutually exclusive buttons in which only one button can be selected at a time.

It is supported by jRadioButton class.



Properties

1 .Background

2 .Button Group :It sets the name of group of Radio button.

3 .Enabled

4 .Font :It sets the font style of the text to be displayed on the radio button.

5 .Foreground :Sets the font color of he text to be displayed with radio button.

6 .Label :Sets the text displayed with any other control.

7 .Selected :Sets the current state of the radio button.

8 .Text

Methods

void setText(Boolean b) :Sets the text when will be displayed with the radio button.

Ex :<radio button _name> .setText("New");

String getText() :It is used to fetch the text displayed with the radio button.

String STR =<radiobutton _name> .getText();

Boolean isselected() :it returns the current state of the radio button.

viod setSelected(Boolean) :it sets the current state of the radio button for the selection process.

7. **Frame**

- It is a top-level window with a title and a border.
- Frame is created using jFrame components.

- **Frame is a super class which provides basic attributes and behavior of a window . A title bar and buttons to minimize, maximize and close the window.**
- **It provides a home for the other components of the interface.**

Properties

1 .Background

2 .Title

3 .Default close operation : This property sets hide in close.

Methods

Integer DefaultCloseOperation() :when user initiates close on the frame it returns the operation.

Void setdefaultCloseOperation() :sets the operation happen by default when user initiates close on the frame.

String getTitle: gets the title of the frame.

void setTitle(String): sets the title for this frame to be specified title.

8. Dialog Box

It is a small separator sub-window which appears on the screen for providing / requesting to/from the user.

It is an independent sub-window to carry temporary notice apart from the main window.

The type of controls:

JDialog:

It displays a dialog box which has minimize, maximize and close icons in the title bar of the dialog box.

It presents error messages or warning a user.

JOptionPane:

It creates pop-up window with void contents.

It displays message box or an i/p box.

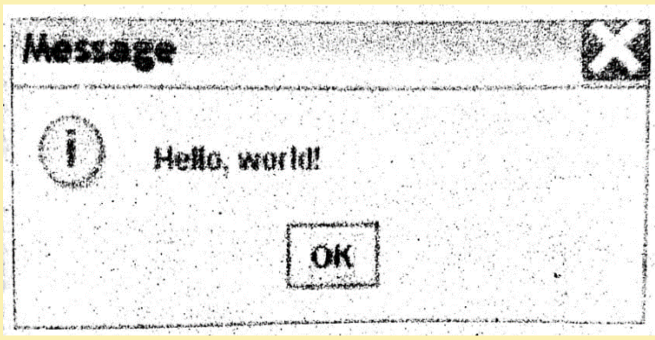
Structure of JOptionPane

(i) **Icon:** The icon determines the type of dialog box or message box; There are four default options available. These are error, information, warning and question.

(ii) **Message:** The message option is used to set the text information that we want to convey through the dialog box.

(iii) **Input Area:** The input area allows the user to provide a response in the form of an input. For user response, we can use jTextField, jComboBox or jList.

(iv) **Button:** The button area displays a set of buttons such as OK/ Cancel, Yes/ No etc. The buttons can be customized according to the user's requirements.



Dialog type-

1.Input dialog—

- It is used to enter data through two buttons ok and cancel.
- Input dialog () method is used to create dialog box.

2. Conform dialog—

- It asks user about the confirmation of information.
- Confirm dialogue() method is used.

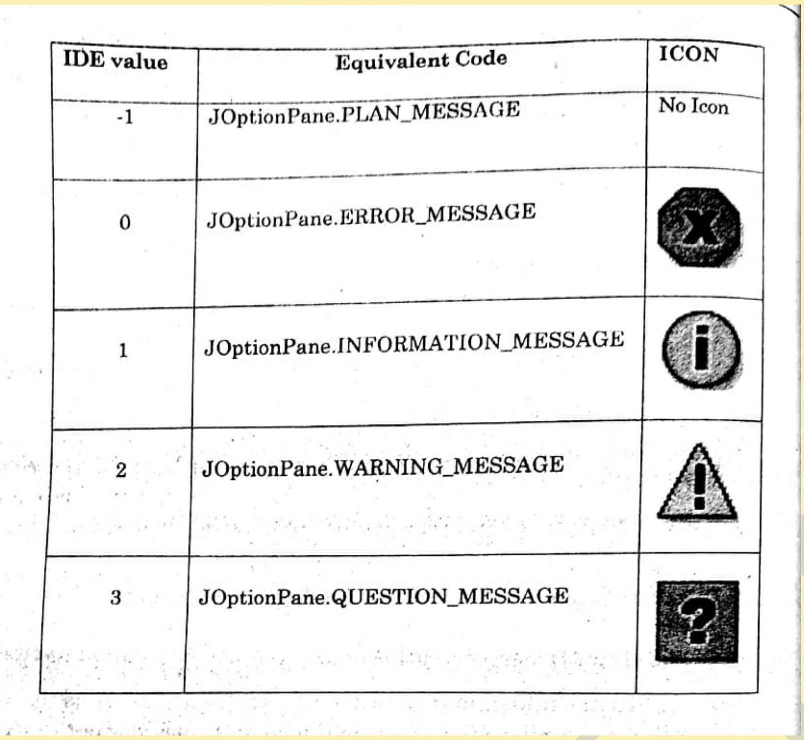
3. Message dialog—

- It displays message some information to user
- Show message dialog() method is used

4. Option dialog—

- It is used to create dialog box according to users need.
- Show option dialog() method is used.

Message type—



IDE value	Equivalent Code	ICON
-1	JOptionPane.PLAN_MESSAGE	No Icon
0	JOptionPane.ERROR_MESSAGE	X
1	JOptionPane.INFORMATION_MESSAGE	i
2	JOptionPane.WARNING_MESSAGE	!
3	JOptionPane.QUESTION_MESSAGE	?

9.<u>Panel—</u>

- It is a light weight container used for holding components which include jButton, jLabel etc.
- It is a create through jpanel component.

Methods—

- **<u>String paramstring()— </u>** Returns a string representation of this panel.
- **<u>AccessibleContentgetAccessibleContext ()—</u>** it adds the specified component to the panel.
- **<u>void add(component)—</u>** Adds the specified component to the panel.
- **<u>Integer componentCount()— </u>** Gets the number of component in this panel.
- **<u>void remove() – </u>** Removes the specified component.

10<u>. Scroll Pane</u>

- A scroll pane is created through jScrollpane component.
- It provides a scrollable view of a component.
- It provides a way of scrolling horizontally and vertically if the widget becomes bigger than the view port size.
- It is primary used in jTextArea, jList and jTable.

11.Password Field:

- It is used to accept the password for an application during the execution of a program.
- It is created through jPassword field component.
- The text displayed is in the form of an echo char (eco character). The default character is.
- It is used for security purpose and it demote show the character the user types.

Properties:

- **Back ground:** It is used to set the back colour for the password field.
- **Font:** This control sets the font for the text of a password field.
- **Fore ground:** It is used to set the colour of the text to be display on password field.
- **Text:** It is used to set the text for the password field.

- **Echo char:** It is used to set an eco-character, that is character to be display in the password field.

Methods—

- **void setEchoChar(char)—**sets the echo character for the password field.
- **Char getEchoChar()—**this method returns echo character of the password field.
- **char [] getPassword—**it returns the text displayed on the password field.
- **void selectAll()—**selects all character of text field.

12. **Combo Box:**

- This let the user choose of several choices.
- It has 2 forms—
  1. **Uneditable combobox** — It features a button and a drop-down unit of values.
  2. **Editable combobox**— It features a text box with a button
     - User can type the value in the text field and click the button to displaying a drop-down list.
     - It is create through jComboBox component.

Properties:

1. **Background:** it is used to set background colour of the combo box.

2. **ButtonGroup:** It sets the group of buttons in which the correct control belongs.
3. **Editable:** it is used to edit the contents of the textfield of the combobox.
4. **Enabled:** It is used to set the combobox as activated and deactivated on the runtime.
5. **Font:** It is used to sent a font style for the text to be displayed.
6. **Foreground:** It is used to foreground colour of the text to be displayed.
7. **Mood**—specifics the data model of the combo box.
8. **Selected index**—this property is used to get or set the index specifying the correctly selected item.
9. **Selected them**—this property is used to get or set correctly selected item in the combo box.
10. **Text** –It is used to specify the text to be displayed on the textfield.

Methods:

1. **Object getSelectedItem()**—it returns currently selected list of jest from the data model.
2. **int getSelectedIndex()**—it returns an integer specifying the currently selected list data.
   0—specifices first them in the list

1—if no even is selected or if the currently selected item Is not in the list

3. **voidsetModel(Combo Box Model)—it sets the data model that the jComboBox user to obtain the list of items.**

13. List:

- **It displays list of objects and allows user to selected one or more items.**
- **It is created through jList component.**
- **Due to many items in list they are put in scroll pare.**

The basic difference between list and combo box is—

  - **List allows selecting more than one at a time.**
  - **Combo box allows selecting only one them at a time.**

Properties:

- **Background: It is used to set the background colour of the text field.**
- **Model: sets an associated object that stores**
- **Selected index: used to select an item initially default value is -1 (no item selected)**
- **Editable: This is used to edit the content of text field. If it is set to true then the**

text available in the text field can be edited by the user at run time.

- **Enabled:** It is used to set the text field as activated and deactivated.
- **Font:** This is used to set a fond type for the text to be displayed in the text field.
- **Foreground:** This is used to set the foreground colour of the text to be display in the text field.
- **Text:** This is used to specify the text to be display in the text field.
- **Select item:** returns the selected item from the list.

Methods —

- **Object getSelected value():** It returns the selected call index when only one item is selected.

  If no items selected then it will return null.
- **Object getSelected values():** It returns all selected raw in the form of an array from the list.

Basic component handling methods

Several methods by which the various properties of components can be handled and modified are—

Some of the methods are as follows in

1. **setText():**

- **This method is used to set the text of component (button, label, text field, text area, check box) at run time.**

Syntax: <textfield_name>.setText ("Name");

Example –jButton1. setText ("enter")

jButton 1—component__name

"Enter"—text displayed on button

setText—method name

<u>2. getText()—</u>

- **This method is used to pitch out or retire the text contacted by the specified component (button label, text field, text area, check box etc.) at run time.**

Syntax – component_name.getText ();

Ex – string str = jTextField 1.getText ();

STR – string type variable which holds the value of text.

<u>3. isSelected()</u>

- **It gets the current state of component.**
- **It is in the form of true or face.**

Syntax – component_name.isSelected

Ex – Boolean= jCheckBox1. isSelected()

<u>4. setSelected()</u>

- **used to set the current state of the component as selected or unselected.**

Syntax – component_name.setSelected();

Ex –jCheckBox1. setSelected(true);

Component of event handling—–

It has 3 main components

1. Events—it is change of state of an object.
2. Event source—it is an object that generates an event.
3. Listener—it listens to the event it gets roughed when an event occurred.

Events are supported by no of java package is like java.util, java.io,java.applet and java.awt.event.

DEVELOPING GENERAL APPLICATION:

Creating A New Project:

To create a new application project called "JavaApplication1"

1. Choose File → New Project. Alternately, click the New Project icon in the toolbar.
2. From the Categories pane select Java and in the Projects pane, choose Java Application.→ Click Next.
3. Enter a name (in this case JavaApplication1) in the Project Name field and specify the project location by clicking on the Browse button. By default, thee project is saved in the NetBeans Projects folder in My Documents and so this is the default Project location displayed in this field.
4. Ensure that the Set as Main Project checkbox is selected and clear the Create Main Class field.
5. Click Finish.

Creating a new Form:

To creating a jFrame Form container:

1. **In the Projects window, right-click the Book node and choose New → jFrame Form as shown in Figure.**
2. **Enter Form Example 1 as the Class Name. This will be the name of your form.**
3. **Enter Book as the package. This should be the name given while creating the Project.**
4. **Click Finish.**

Adding a Button Component to a Form:

We want to add a button so follow the given steps to add a jButton to the form:

1. **In the Palette window, select the jButton component from the Swing Controls category.**
2. **Move the cursor over the Form. When the guidelines appear (as displayed in Figure) indicating that the JButton is positioned in the desired location, click to place the button.**

Executing a File:

Step 1: Create a new Project

Step 2: Add a jFrame form

Step 3: Add the desired component form the Palette window using drag and drop feature.

Step 4: Associate code with the component by double clicking the component.

Step 5: Add the source code.

Step 6: Test the form by pressing Shift + F6

Note : **In Non-GUI methods, to run the program press F6 key from the key board and for GUI method, to run the program press Shift + F6 key board.**