Lecture 5



Java PROGRAMMING (2)

BY: DR. MOHAMMED ABDELFATTAH

Lecture Objectives

- Java GUI
- GUI Components
- NetBeans Installation
- Swing Class Hierarchy
- JFrame Container
- JOptionPane
- Other Components
- Examples

Java GUI

Usually, the choice of a Java framework depends on the specific project needs and requirements. Some of the best frameworks include:

- Swing
- JavaFX
- Java SWT
- Java AWT
- Java Web Frameworks



More info.: https://medium.com/@christiegmarie89/what-are-the-best-java-gui-frameworks-in-2024-aca3ba21983b

Java Swing

- One of the well-known GUI frameworks, Java Swing, is a component of the Java Foundation Classes and provides a collection of tools for creating online applications.

- Swing is renowned for having a lightweight design that makes it ideal for slick and responsive user interfaces.

- This makes it perfect for a wide range of uses, from sophisticated commercial software systems to simple-to-use apps.

JavaFX

- Another popular Java GUI framework that was created as a substitute for Swing is JavaFX, which was constructed by Oracle.
- JavaFX is mostly utilized for contemporary UI development and multimedia integration.
- This framework's high degree of flexibility and extensive feature set make it an excellent option for creating aesthetically attractive and highly interactive applications.

Java AWT

- The acronym for the older Java GUI framework is AWT, or **Abstract Window Toolkit**.

- Java AWT's components are lightweight and native.

- This indicates that it renders using the platform's **native GUI** components. The AWT framework, which is frequently used in conjunction with JavaFX and Swing for complicated applications, is perfect for creating lightweight and straightforward user interfaces.

SWT

- The Eclipse Foundation created the well-known GUI framework known as **SWT**, or **Standard Widget Toolkit**.
- In contrast to Swing or JavaFX, SWT renders using native components. This gives it a native appearance and feel.
- Furthermore, a lot of Eclipse-centric apps are made with this GUI framework. It may be used for stand-alone applications with ease, though.

LibGDX

- **LibGDX** is a Java-oriented game development framework that is utilized for both **desktop and mobile game** development, whereas JavaFX, Swing, AWT, and SWT are commonly used for desktop applications.

- It offers a wide range of features and tools for making interactive apps and 2D and 3D games.

NetBeans

An integrated Java development environment is called NetBeans. Using a collection of modular software components known as modules, NetBeans enables the development of applications. NetBeans is compatible with Linux, Solaris, macOS, and Windows.

Link:

https://netbeans.apache.org/front/main/download/nb22/

NetBeans Installation



NetBeans Installation

Downloading Apache NetBeans 22

Apache NetBeans 22 was released on May 29, 2024.

Apache NetBeans 22 is available for download from your closest Apache mirror.

Binaries (Platform Independent):

- netbeans-22-bin.zip (SHA-512, PGP ASC)

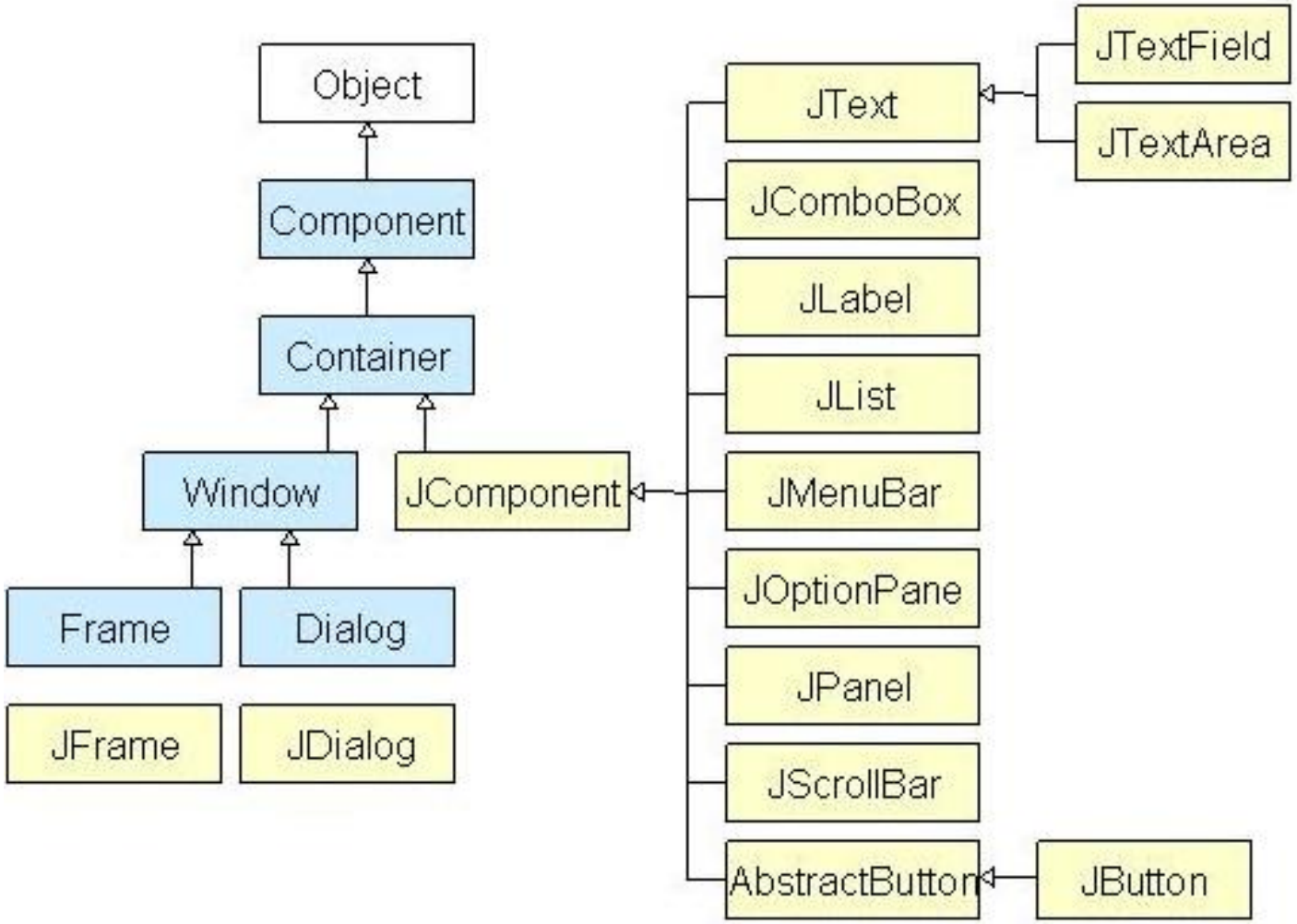
Installers and Packages:

- Apache-NetBeans-22-bin-windows-x64.exe (SHA-512, PGP ASC)
- Apache-NetBeans-22.pkg (SHA-512, PGP ASC)
- apache-netbeans_22-1_all.deb (SHA-512, PGP ASC)
- apache-netbeans-22-0.noarch.rpm (SHA-512, PGP ASC)
- Linux snap package

Source:

- netbeans-22-source.zip (SHA-512, PGP ASC)

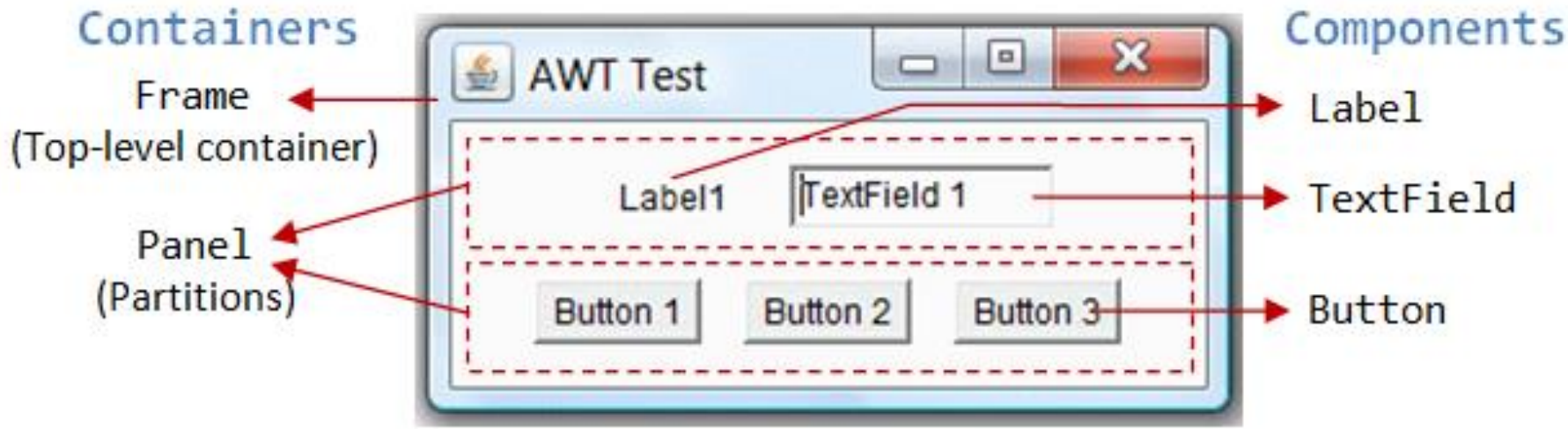
Java Swing Class Hierarchy Diagram



GUI Components in Java

You can use Java's plenty of **GUI** components to construct graphical user interfaces. The six most popular Java Swing GUI elements are as follows:

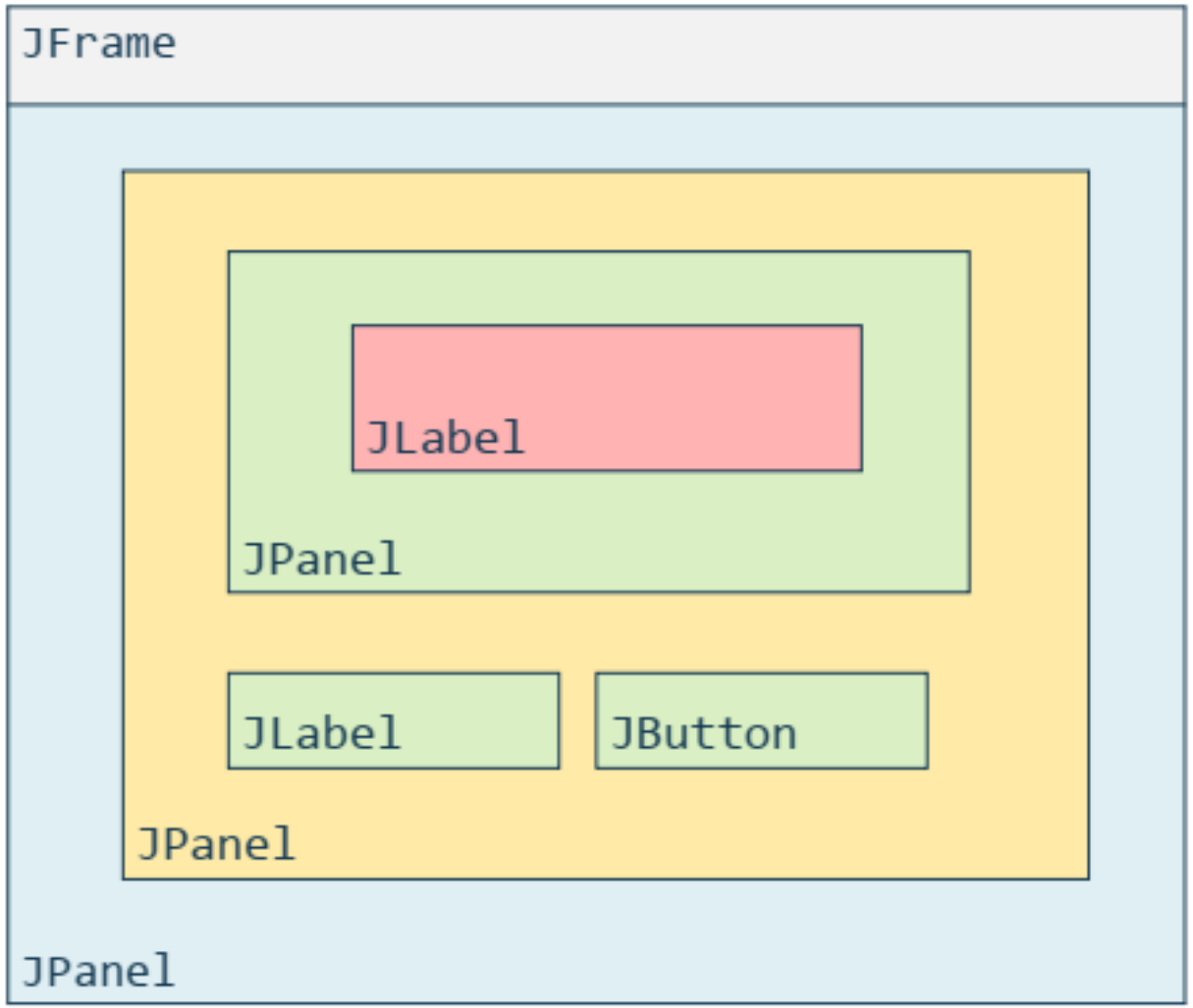
- JFrame
- JPanel
- JButton
- JLabel
- JTextField
- JTextField



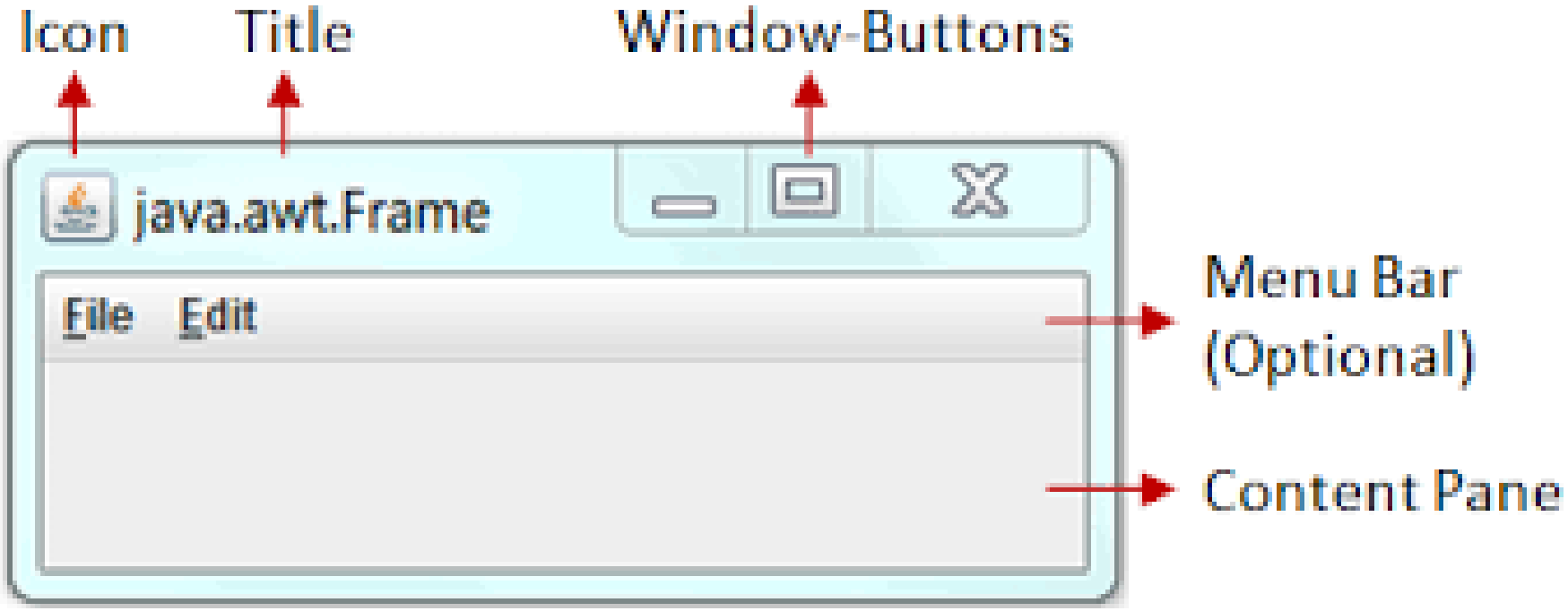
Containment Hierarchy

✓ To make a component visible, its containment hierarchy must be included into a **JFrame** o another window object.

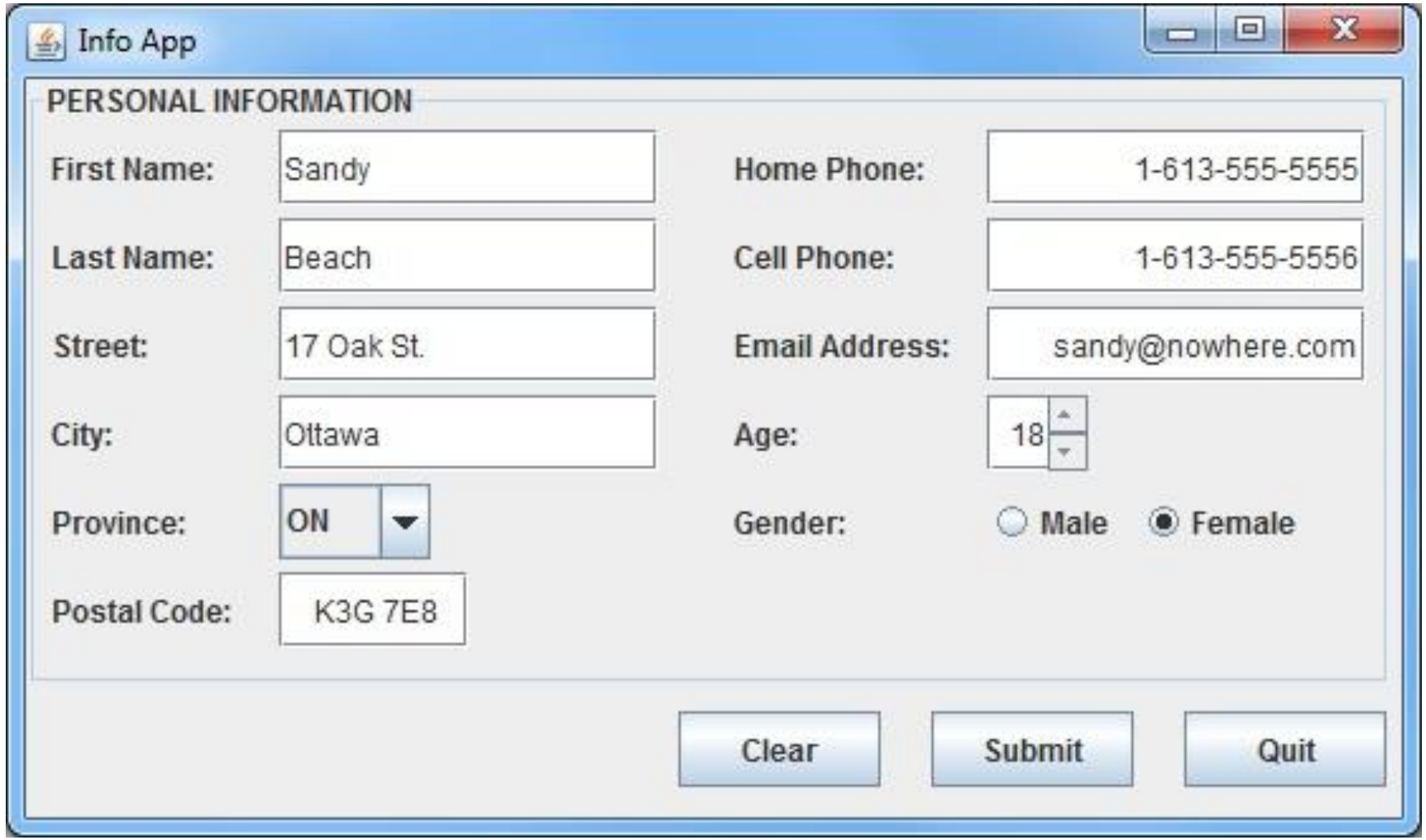
✓ **JPanels** are containers to which usually we add components.



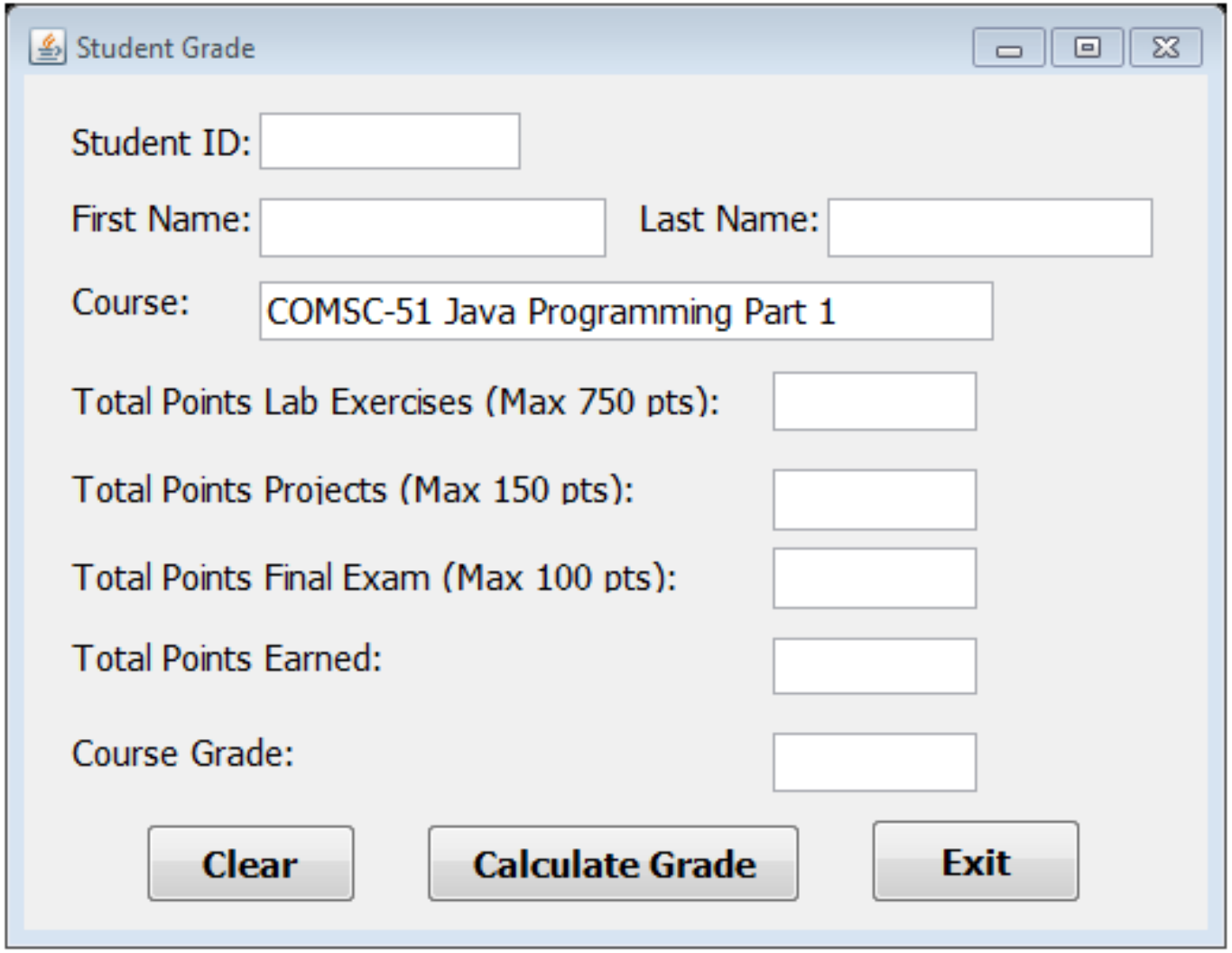
JFrame


Icon
Title
Window-Buttons
java.awt.Frame
File Edit
Menu Bar
(Optional)
Content Pane

JFrame Example 1



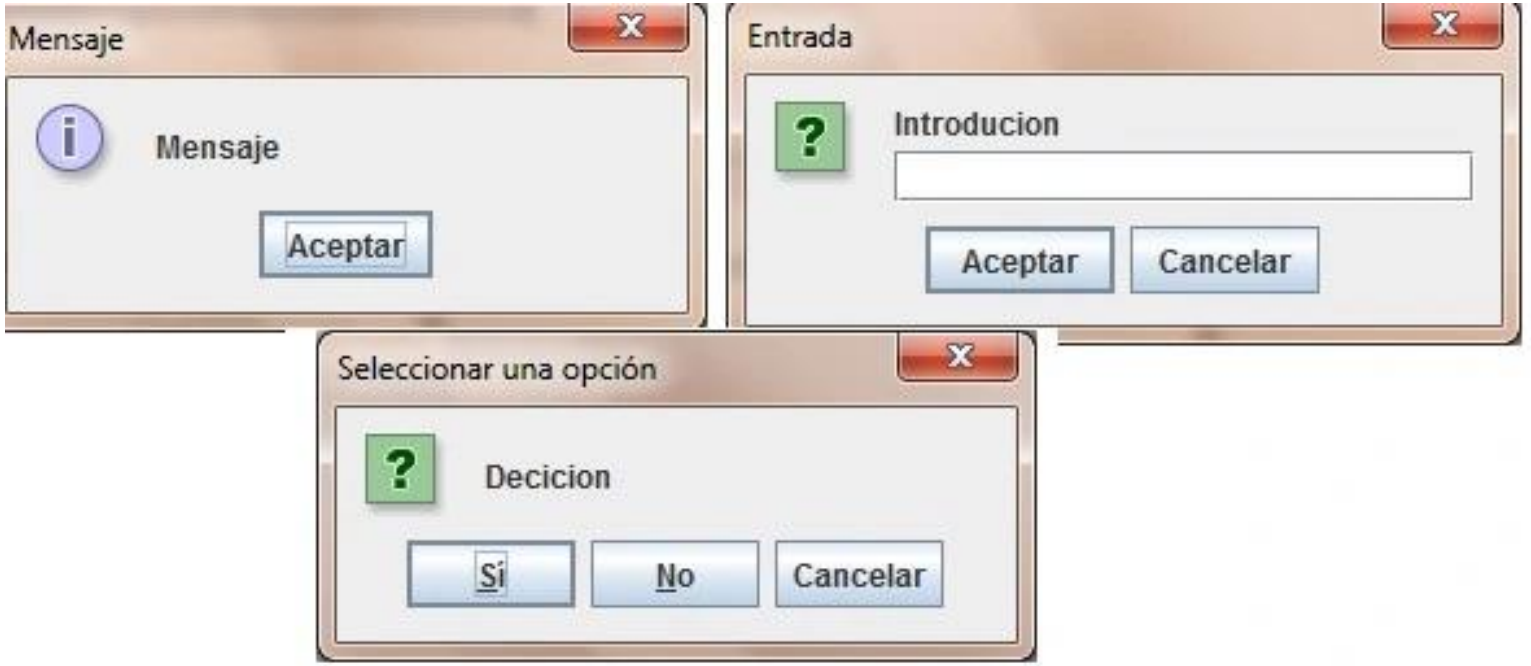
JFrame Example 2



JOptionPane

JOptionPane can be used to inform the user about something or to ask for some input. The class has many public constructors and many static methods to show dialogs.

- ✓ **showMessageDialog()**
- ✓ **showConfirmDialog()**
- ✓ **showInputDialog()**
- ✓ **showOptionDialog()**



Parameters:

- parentComponent
- message
- messageType
- optionType
- options
- icon
- title
- initialvalue

JOptionPane: Message Types



Example

```
import javax.swing.JOptionPane;
import static javax.swing.JOptionPane.showConfirmDialog;
import static javax.swing.JOptionPane.showInputDialog;
import static javax.swing.JOptionPane.showMessageDialog;

private static void demo() {
    String name = showInputDialog(null, "What's your name");
    int result = showConfirmDialog(null, "Your name is: " + name + "\n Is it right?");
    if (result == JOptionPane.OK_OPTION) {
        showMessageDialog(null, "Hi " + name + "!");
    } else {
        showMessageDialog(null, "Try again", "Incorrect name", JOptionPane.ERROR_MESSAGE);
    }
}
```

Software Development Life Cycle Cycle

This presentation will explore the Software Development Life Cycle (SDLC), its phases, and popular methodologies. We'll learn how to efficiently design, develop, and deploy high-quality software.

MA



Introduction to the SDLC

What is the SDLC?

The SDLC outlines the structured process of building software, from initial planning to final deployment and ongoing maintenance. It's a roadmap for software development.

Why is it Important?

It ensures efficient project management, reduces risk, promotes collaboration, and ultimately leads to successful software development.

Phases of the SDLC

1. Requirement Gathering: Identifying and documenting the specific needs and functionalities of the software.
2. Design: Creating blueprints for the software's architecture, user interface, and data structures.
3. Development: Writing the actual code that brings the software to life, using programming languages and frameworks.
4. Testing: Rigorously evaluating the software's functionality, performance, and security to ensure it meets the defined requirements.
5. Deployment: Releasing the software to end-users, making it accessible and operational.
6. Maintenance: Ongoing updates, bug fixes, and enhancements to ensure the software remains functional and meets evolving needs.

Waterfall Model

1

Requirement Gathering

Requirements are thoroughly documented at the start.

2

Design

Detailed design specifications are created.

3

Development

Coding and development proceed sequentially.

4

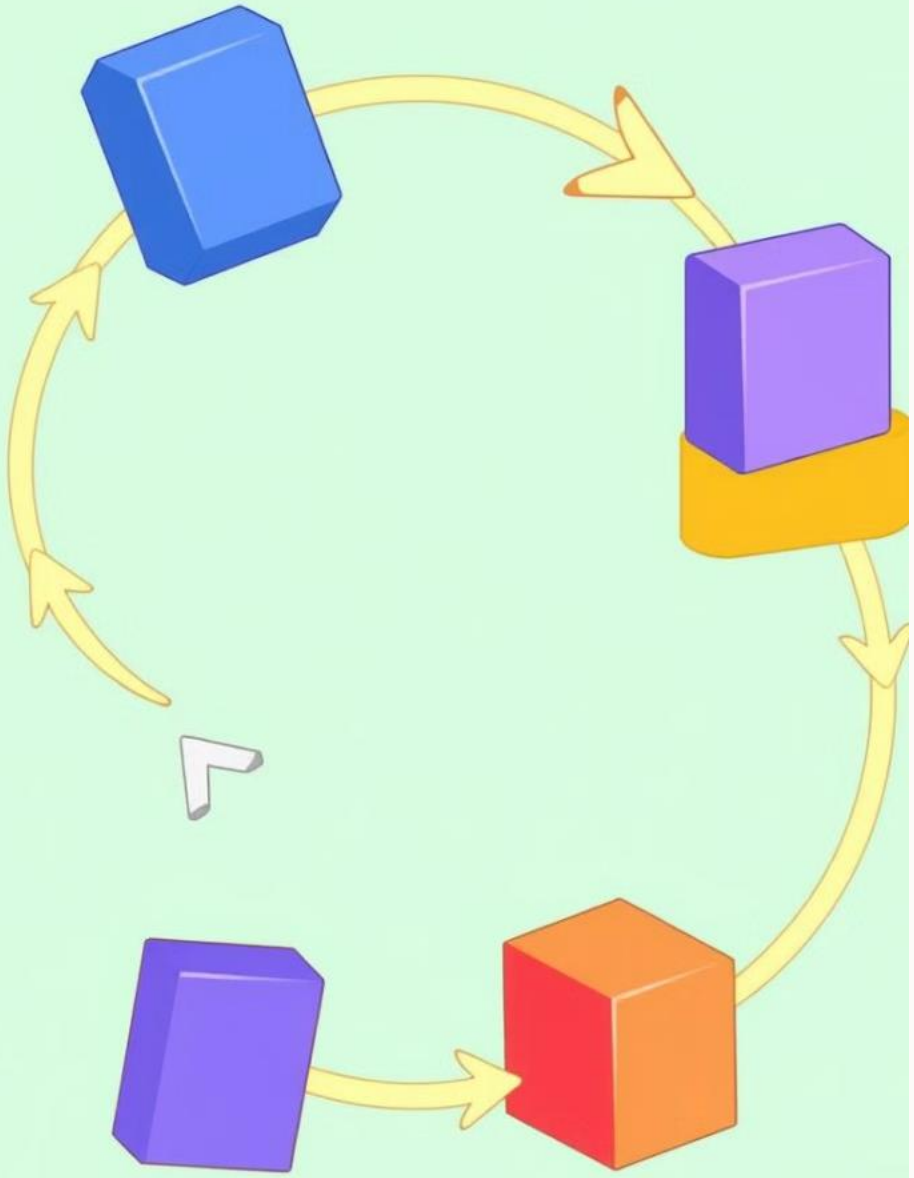
Testing

Testing is conducted after development is completed.

5

Deployment

The software is deployed and maintained.



Agile Methodology

Iterative Development

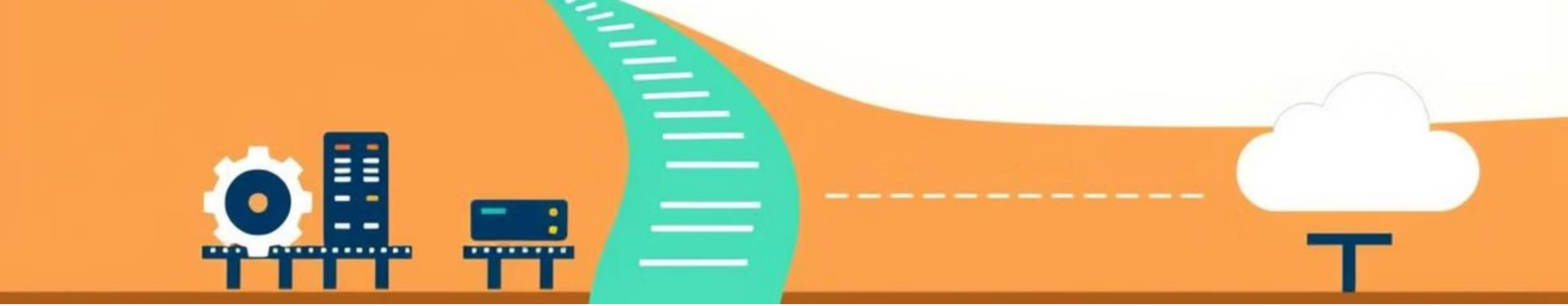
Work is broken down into small iterations, allowing for flexibility and adaptation to changing requirements.

Continuous Feedback

Stakeholders provide feedback throughout the development process, ensuring the software meets their needs.

Team Collaboration

Agile emphasizes collaboration, communication, and shared responsibility among team members.



Continuous Integration and Deployment

1

Automated Builds

The software is automatically built and tested with every code change.

2

Automated Testing

Automated tests are run to ensure the software functions as expected.

3

Frequent Deployment

New features and bug fixes are released to users regularly.



Testing and Quality Assurance Assurance

Bug Detection

Identifying and fixing software defects before deployment.

Functional Testing

Verifying that the software meets the defined requirements.

Performance Testing

Evaluating the software's speed, responsiveness, and stability under load.

Security Testing

Assessing the software's vulnerability to security threats.

Best Practices for Effective SDLC

1

Clear Communication

Ensure effective communication among team members, stakeholders, and customers.

2

Documentation

Maintain thorough documentation throughout the development process.

3

Code Reviews

Conduct regular code reviews to improve quality and identify potential issues.

4

Continuous Improvement

Regularly evaluate and improve the SDLC process to optimize efficiency and effectiveness.



Thanks

References:

https://www.w3schools.com & https://www.tutorialspoint.com & https://www.javatpoint.com