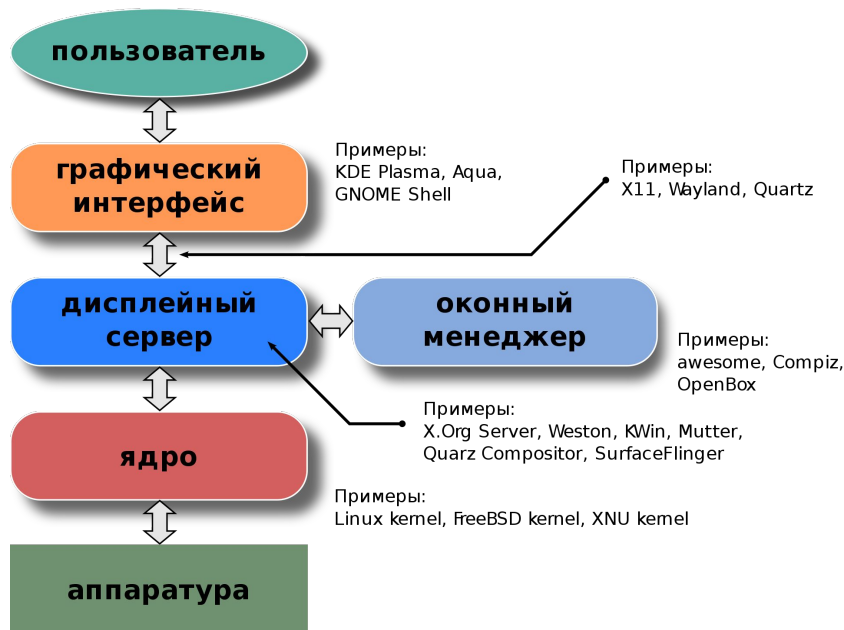


# Java для начинающих

Графический интерфейс

# Графический интерфейс

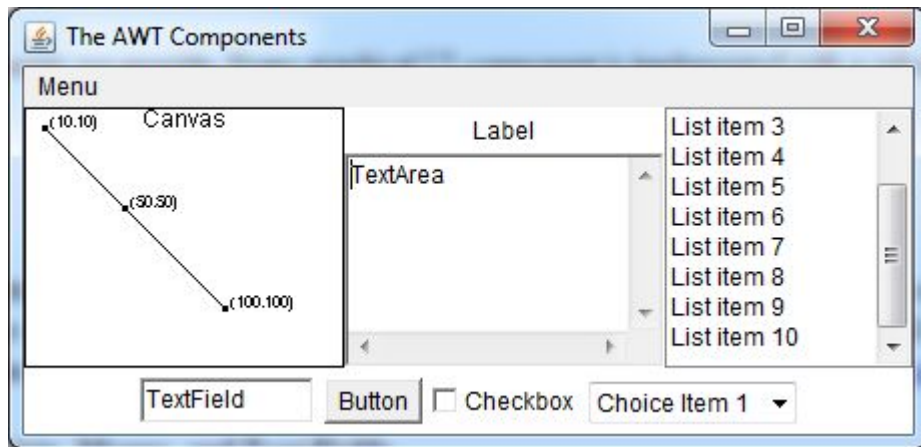
**Графический пользовательский интерфейс (ГПИ)** (graphical user interface, **GUI**) — система средств для взаимодействия пользователя с электронными устройствами, основанная на представлении всех доступных пользователю системных объектов и функций в виде графических компонентов экрана (окон, значков, меню, кнопок, списков и т. п.).



# GUI Java

- Abstract Window Toolkit (AWT)
- Swing
- JavaFX

# Abstract Window Toolkit



Компоненты Java AWT зависят от платформы, что означает, что они отображаются в соответствии с представлением операционной системы.

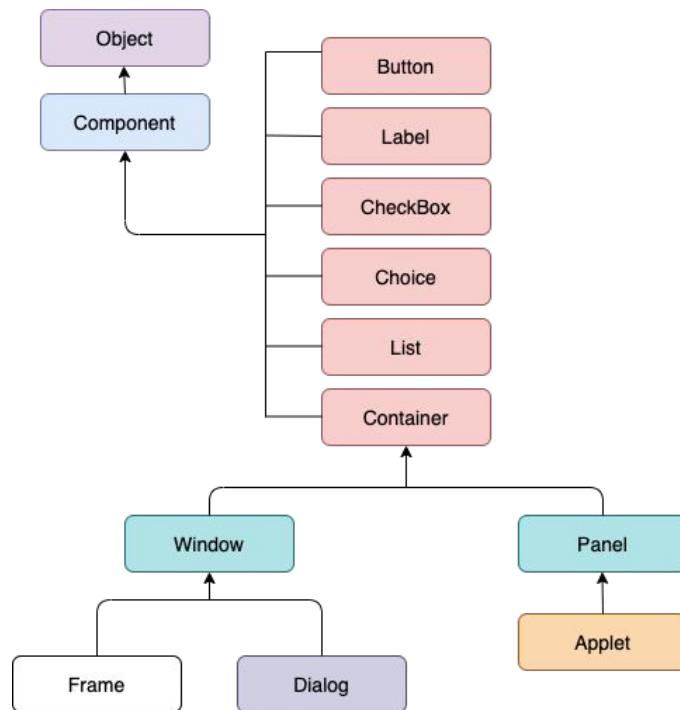
Библиотечные методы создают и используют графические компоненты операционной среды.

С одной стороны, это хорошо, так как программа на Java похожа на остальные программы в рамках данной ОС.

Но с другой стороны, нет никакой гарантии, что различия в размерах компонентов и шрифтах не испортят внешний вид программы при запуске ее на другой платформе.

Кроме того, чтобы обеспечить мультиплатформенность, пришлось унифицировать интерфейсы вызовов компонентов, из-за чего их функциональность получилась немного урезанной.

# AWT дерево компонентов



# AWT достоинства и недостатки

## Достоинства:

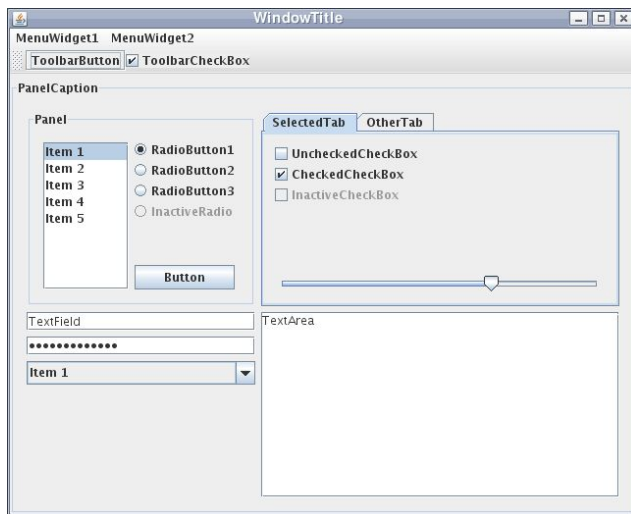
- часть JDK;
- скорость работы;
- графические компоненты похожи на стандартные.

## Недостатки:

- использование нативных компонентов налагает ограничения на использование их свойств. Некоторые компоненты могут вообще не работать на «неродных» платформах.
- некоторые свойства, такие как иконки и всплывающие подсказки, в AWT отсутствуют.
- стандартных компонентов AWT очень мало.
- программа выглядит по-разному на разных платформах.

В настоящее время AWT используется крайне редко — в основном в очень старых проектах. Хотя, крайне редко, но при работе с Swing или JavaFX приходится импортировать классы из AWT.

# Swing



Вслед за AWT Sun разработала набор графических компонентов под названием **Swing**.

Он построен на основе API AWT и полностью написан на java.

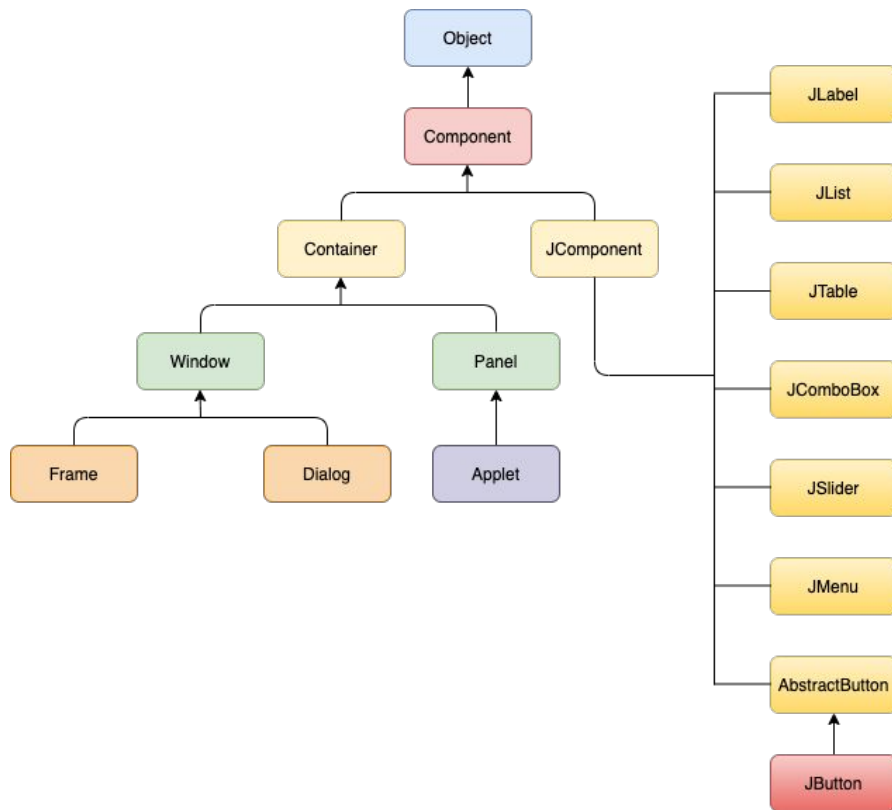
В отличие от AWT, Java Swing предоставляет платформенно-независимые и легкие компоненты.

Набор стандартных компонентов значительно превосходит AWT по разнообразию и функциональности.

Стало легко создавать новые компоненты, наследуясь от существующих.

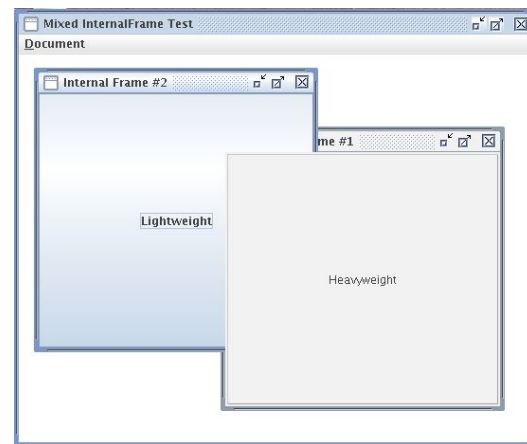
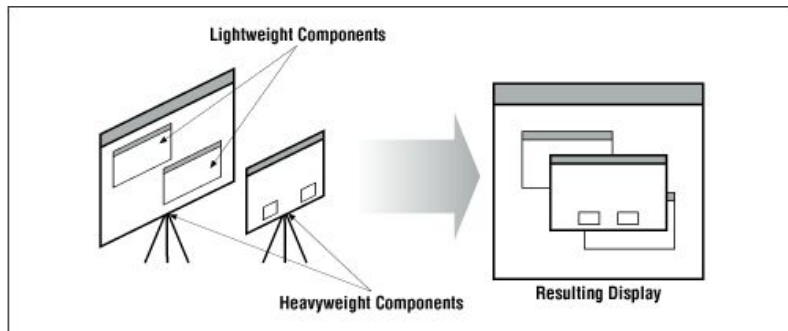
Благодаря простоте использования, богатой документации и гибкости компонентов Swing стал, пожалуй, самым популярным графическим фреймворком в Java.

# Swing дерево компонентов





# Принцип Lightweight



**Lightweight** означает, что компоненты Swing отрисовываются самими компонентами на поверхности родительского окна, без использования компонентов операционной системы.

В отличие от «тяжёлых» компонентов AWT, в приложении Swing может иметься только одно окно, и все прочие компоненты отрисовываются на ближайшем родителе, имеющем собственное окно (например, на JFrame).

В приложении могут сочетаться Swing- и AWT-элементы, хотя это может порождать некоторые проблемы — в частности, компоненты AWT всегда перекрывают Swing-элементы.

# Swing достоинства и недостатки

## Достоинства:

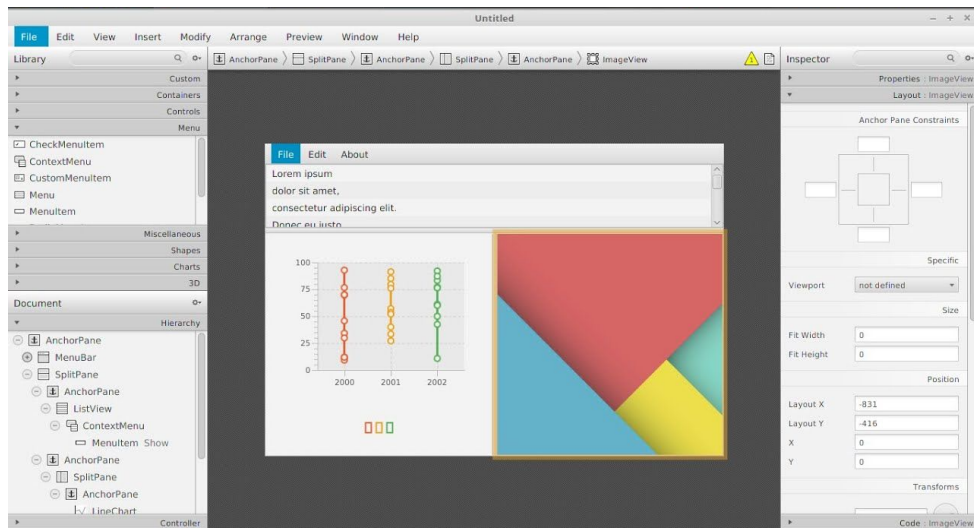
- часть JDK, не нужно ставить дополнительных библиотек.
- по Swing гораздо больше книжек и ответов на форумах. Все проблемы, особенно у начинающих, легко находятся в интернете.
- встроенный редактор форм почти во всех средах разработки.
- на базе свинга есть много расширений типа SwingX, JGoodies.
- поддержка различных стилей (Look and feel).

## Недостатки:

- окно с множеством компонентов начинает подтормаживать;
- работа с менеджерами компоновки (Layout) может стать настоящим кошмаром в сложных интерфейсах.

Даже сейчас, в 2021 году, Swing, вероятно, по-прежнему занимает значительную часть всех разработчиков GUI на Java, в основном в легаси проектах.

# JavaFX



JavaFX был предназначен для замены Swing, но он был исключен из новых JDK, в то время как Swing и AWT остались. Предположительно из-за того, что рыночная доля JavaFX была «подорвана ростом популярности “mobile first” и “web first” приложений.

С выпуском JDK 11 в 2018 году Oracle сделала JavaFX частью OpenJDK в рамках проекта OpenJFX.

Начиная с версии Java 11 больше не входит в Java SE и не разрабатывается компанией Oracle (как отдельный модуль поддерживается компанией Gluon).

Oracle будет поддерживать JavaFX в составе Java JDK 8 до марта 2025.

# Swing vs JavaFX



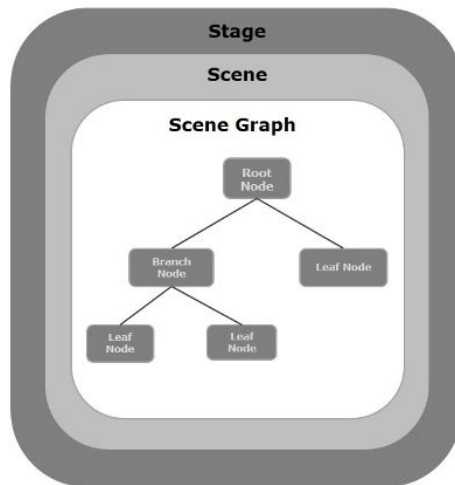
## Swing vs JavaFX

	 Верстка	 Стили	 Логика
Swing	Java code	Java code	Java code
JavaFX	FXML	CSS	Java code

# JavaFX особенности

- JavaFX изначально поставляется с большим набором частей графического интерфейса, кнопки, текстовые поля, таблицы, деревья, меню, диаграммы и т.д.
- JavaFX часто использует стили CSS, можно использовать FXML для создания GUI, а не делать это в коде Java.
- JavaFX имеет готовые к использованию части диаграммы, поэтому нам не нужно писать их с нуля в любое время, когда вам нужна базовая диаграмма.
- JavaFX дополнительно поставляется с поддержкой 3D графики, которая часто полезна, если мы разрабатываем игру или подобные приложения.

# JavaFX основные составляющие окна



**Stage** — окружающее окно, которое используется как начальное полотно и содержит в себе остальные компоненты. У приложения может быть несколько stage, но один такой компонент должен быть в любом случае. По сути Stage является основным контейнером и точкой входа.

**Scene** — отображает содержание stage (как матрёшка). Каждый stage может содержать несколько компонентов — scene, которые можно между собой переключать. Внутри это реализуется графом объектов, который называется — **Scene Graph** (где каждый элемент — узел, ещё называемый как Node).

**Node** — это элементы управления, например, кнопки, метки, макеты (layout), внутри которых может быть несколько вложенных компонентов. У каждой сцены (scene) может быть один вложенный узел (node), но это может быть макет (layout) с несколькими компонентами. Вложенность может быть многоуровневой, когда макеты содержат другие макеты и обычные компоненты. У каждого такого узла есть свой идентификатор, стиль, эффекты, состояние, обработчики событий.

# JavaFX достоинства и недостатки

## Достоинства:

- более удобный API
- поддержка стилей CSS
- встроенная система анимации
- система связки свойств (property binding)
- множество различных компонентов
- есть компонент WebView который может отображать современные веб-страницы

## Недостатки:

- фреймворк не получил широкого распространения
- требуется больше знаний для начала работы в сравнении с фреймворком Swing

# Домашнее задание

Создать пакет hw15.

Выполнить задание 14 (клиент-сервер на сокетах).

Для клиентской части программы разработать графический интерфейс пользователя (GUI).

Можно выбрать любую из библиотек AWT, Swing, или JavaFX.