Графический пользовательский интерфейс на Java

Давыдов Антон Валериевич
Студент ТГУ, Россия, г. Тольятти
Научный руководитель: **Ерофеева Е.А.**

Пользовательский интерфейс на Java прошел весьма тернистый путь становления и развития. Его долгое время обвиняли в жадности к ресурсам системы, медленной работе и ограниченной функциональности. Появление .NET с более быстрыми графическими компонентами еще больше пошатнуло позиции Java. Но подобная конкуренция лишь подстёгивала разработчиков Java к развитию и улучшению графических библиотек. И в этой статье мы посмотрим, что из этого получилось.

Abstract Window Toolkit

Abstract Window Toolkit (сокращённо AWT) впервые была выпущена в 1995 году компанией Sun Microsystems. Это была первая попытка создать графический интерфейс для Java. AWT выступал в качестве прослойки, вызывающей методы из библиотек, написанных на C. А эти методы, в свою очередь, использовали графические компоненты операционной системы [2]. С одной стороны, программа, построенная таким образом, внешне была похожа на все остальные программы в используемой операционной системе, но с другой, одна и та же программа может выглядеть совершенно по-разному на разных операционных системах, что осложняло разработку. К тому же, ради мультиплатформенности пришлось унифицировать интерфейсы вызовов компонентов, что привело к несколько урезанной функциональности. Набор компонентов также довольно скромный. Например, отсутствуют таблицы, а в кнопки нельзя поместить иконки. AWT старается автоматически освобождать использованные ресурсы. Это влияет на производительность и усложняет архитектуру. AWT прост для освоения, но написание чего-то сложного вызывает затруднения. Сейчас AWT используется в основном для аплетов. Oracle в данный момент поощряет переход разработчиков на Swing, как более безопасный.

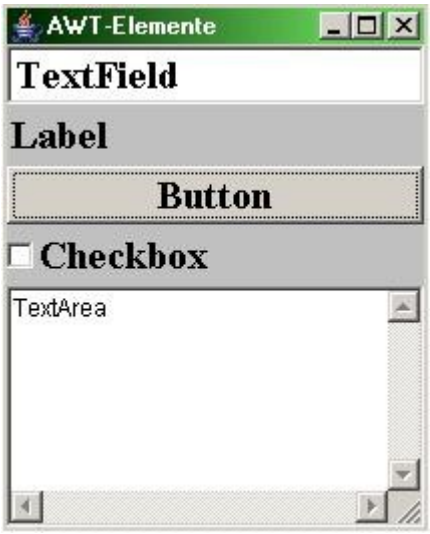

Рис.1 – Образец программы, написанной с использованием AWT в среде Windows

Swing

После AWT, в 1998 году, Sun выпустила Swing. Он полностью написан на Java и для отрисовки использует 2D. В Swing гораздо больше разнообразных компонентов, чем в AWT. Сами компоненты стало гораздо проще создавать, наследуя их от существующих [1]. Также была введена

возможность использования различных стилей и скинов. Однако, скорость работы ранних версий Swing была довольно низкой, а ошибки в написании программы могли и вовсе привести к зависанию операционной системы.

Однако, благодаря лёгкому освоению и наличию большого количества документации, Swing стал самым популярным графическим интерфейсом в Java. На его основе появилось множество расширения, например SwingX и JGoodies, которые ещё больше упрощают создание визуально сложных приложений. Все современные среды программирования на Java включают в себя графические редакторы Swing. Даже не смотря на то, что сейчас существуют более современные фреймворки, Swing остаётся самым популярным.

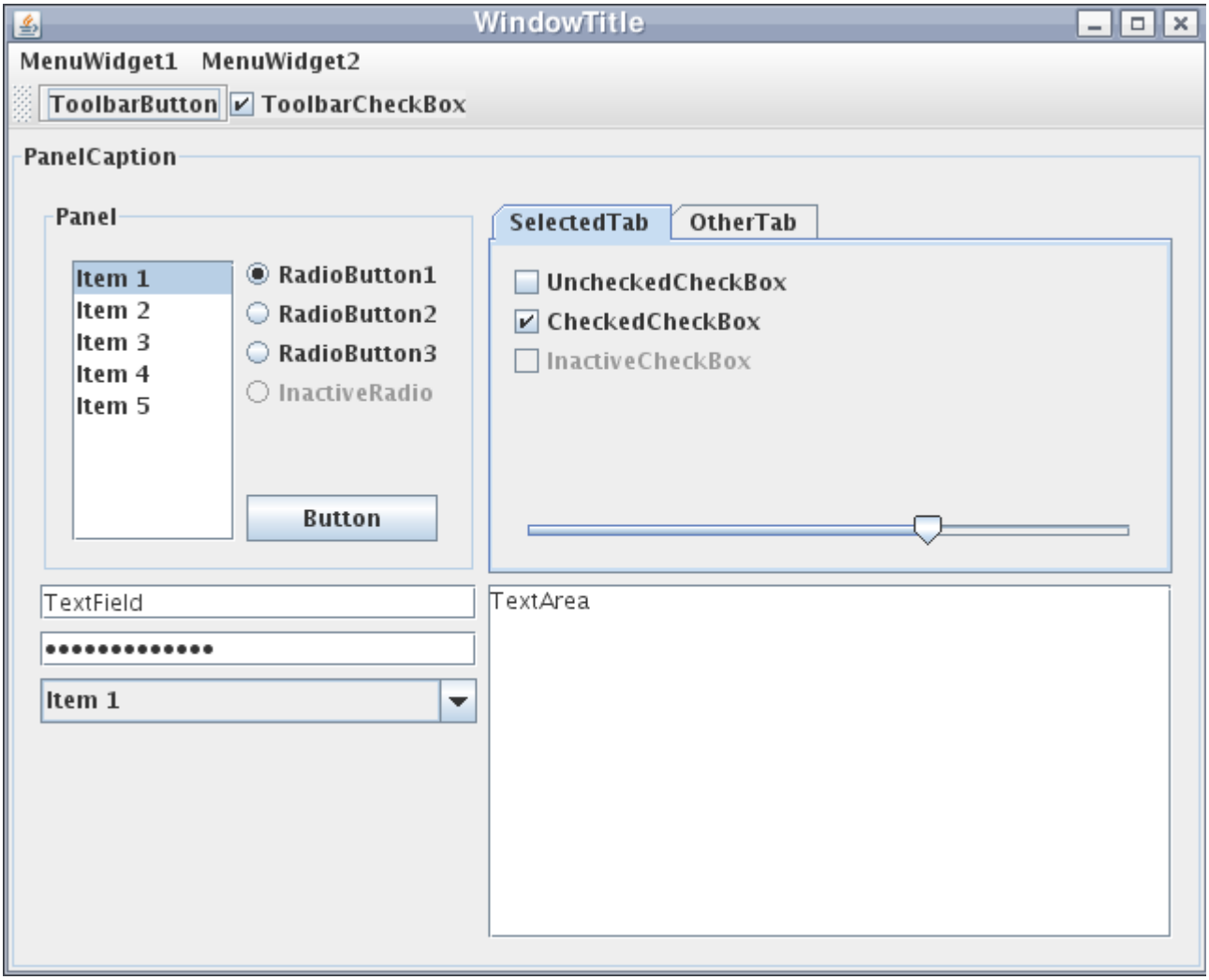

Рис.2 – Образец программы, написанной с использованием Swing

Standard Widget Toolkit

SWT был выпущен компанией IBM во времена, когда Swing был ещё медленным, и в основном для продвижения среды программирования Eclipse. Как и AWT, SWT использует компоненты ОС, но для различных платформ используются различные интерфейсы взаимодействия [4]. Таким образом для каждой операционной системы необходимо поставлять отдельную JAR-библиотеку. Это позволяет более полно использовать функции, соответствующие различным операционным системам. А недостающие компоненты были реализованы с помощью 2D. Тем не менее, SWT получилась более сложной для освоения, чем Swing. Кроме того, программист должен сам реализовывать освобождение ресурсов приложением.

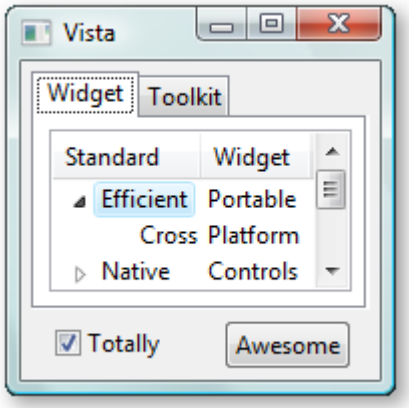

Рис.3 – Образец программы, написанной с использованием Swing

JavaFX

JavaFX была выпущена в 2008 году компанией Oracle. Она позиционируется как платформа для создания насыщенного интернет-приложения. Для отрисовки используется графический конвейер, что значительно ускоряет работу приложения. Имеется большой набор встроенных компонентов. Также имеются отдельные компоненты для построения графиков. Реализована поддержка мультимедийного контента, анимации и даже множественное касание. Внешний вид компонентов настраивается при помощи CSS-стилей [3]. Кроме того, в набор утилит JavaFX входит возможность сделать родной инсталлятор для самых популярных платформ: exe или msi для Windows, deb или rpm для Linux, dmg для Mac. На сайте Oracle имеется подробная документация и большое число готовых примеров.

Таким образом, описав основным особенности и недостатки вышеперечисленных графических пользовательских интерфейсов, мы можем решить, для каких задач они лучше подходят. Abstract Window Toolkit больше подойдёт для создания аплетов. Новичку можно порекомендовать Swing в виду того, что для него можно найти огромное количество документации в интернете, в том числе и на русском языке. Для создания насыщенных интернет-приложений отлично подойдёт JavaFX.

Список использованных источников

1. Рыженко А. В. Объектно-ориентированное программирование: Учебно-методический комплекс по дисциплине для специальности 010501 – "Прикладная математика и информатика". – 2007.

2. Хабибуллин И. Ш. Java 7 (4-е изд.). – БХВ-Петербург, 2012.

3. Clarke J., Connors J., Bruno E. J. JavaFX: Developing Rich Internet Applications. – Pearson Education, 2009.

4. Northover S., Wilson M. Swt: the standard widget toolkit, volume 1. – Addison Wesley Professional, 2004.