

УДК 00.004.4

Ю.Э. Парфенов, В.Н. Федорченко

Харьковский национальный экономический университет, Харьков

РАЗРАБОТКА «НАСЫЩЕННЫХ» ИНТЕРНЕТ-ПРИЛОЖЕНИЙ С ПОМОЩЬЮ JAVAFX

Рассматривается новая платформа создания «насыщенного» графического интерфейса пользователя – JavaFX, основанная на технологии Java. Представлены основные характеристики указанной платформы, преимущества ее использования разработчиками. Обсуждаются отличия данной платформы от инструментария Swing. Рассмотрен состав архитектуры и описана среда JavaFX. Рассмотрены особенности Java API для JavaFX, особенности реализации графического интерфейса, принципы реализации графической системы JavaFX. Выполнен анализ реализации многопоточности и системы событий. Рассматривается функциональность платформы JavaFX с точки зрения использования медиа, 2D/3D анимации и основных элементов управления. Дается характеристика среды разработки приложений JavaFX.

Ключевые слова: графический пользовательский интерфейс, платформа, граф сцены, 2D/3D анимация, Java, API, JDK, SDK, веб-компонент, элемент управления, язык разметки, потоки, события, каскадные таблицы стилей, программный рендеринг, контейнерные классы, схемы расположения.

Постановка проблемы

В течение более десяти лет, разработчики приложений считали Swing весьма эффективным инструментарием для создания графического пользовательского интерфейса (ГПИ) и добавления интерактивности в Java-приложения. Тем не менее, некоторые из самых популярных в настоящее время элементов ГПИ не могут быть легко реализованы с использованием Swing.

JavaFX является инструментарием следующего поколения для создания ГПИ, который позволяет разрабатывать «насыщенные» кроссплатформенные приложения, которые ведут себя одинаково на разных платформах.

JavaFX предназначен для обеспечения приложений с графическим интерфейсом такими сложными функциями как плавная анимация, веб-представления, воспроизведение аудио и видео, стили на основе CSS.

JavaFX 2.0 основан на парадигме графа сцены. Граф сцены JavaFX представляет собой древовидную структуру данных, которая поддерживает векторные графические узлы.

Цель JavaFX - использование во многих типах устройств, таких как мобильные устройства, смартфоны, телевизоры, планшетные компьютеры и десктопы [1].

Что такое JavaFX?

Платформа JavaFX является развитием клиентской платформы Java и позволяет разработчикам приложений легко создавать и развертывать «насыщенные» интернет-приложения. Построенная на технологии Java, платформа JavaFX предоставляет бога-

тый набор графических и медиа-API к высокопроизводительным графическим и мультимедиа-«движкам» с аппаратным ускорением, которые упрощают разработку управляемых данными клиентских приложений. Инвестирование в платформу JavaFX предоставляет следующие преимущества для Java-разработчиков и компаний, которые являются частью экосистемы Java:

- поскольку платформа JavaFX написана на Java, Java-разработчики могут использовать имеющиеся у них навыки и инструменты для разработки приложений JavaFX;
- поскольку Java широко используется, легко найти опытных Java-разработчиков, которые могут быстро стать продуктивными, разрабатывая приложения JavaFX;
- с помощью однородного набора Java-технологий для серверных и клиентских платформ, платформа JavaFX уменьшает риск инвестиций за счет снижения сложности бизнес-решений;
- затраты на разработку также снижаются из-за вышеупомянутых преимуществ;
- платформа JavaFX предоставляет разработчикам среду разработки и выполнения для создания корпоративных- и бизнес-приложений, работающих на разных платформах, которые поддерживают Java.

Основные характеристики JavaFX 2

JavaFX 2 обладает следующими характеристиками и особенностями [2]:

- полная интеграция с JDK 7. С момента выпуска JavaFX SDK 2.2 и Java SE 7 update 6, JavaFX SDK полностью интегрирован с Java SE 7 Runtime Environment (JRE) и Development Kit (JDK). Эта интеграция с JDK 7 устраняет необходимость в загрузке и

установке JavaFX 2 SDK отдельно. Автономная же загрузка JavaFX 2 SDK для Windows, будет доступна для пользователей JDK 6 до тех пор, пока компания Oracle будет поддерживать Java SE 6;

- Java API для JavaFX, обеспечивающие все знакомые черты языка (например, типизированные коллекции, аннотации и многопоточность), которые привыкли использовать Java-разработчики. Эти API могут использоваться в альтернативных JVM-языках, таких как JRuby и Scala. Поскольку возможности JavaFX доступны через Java API, вы можете продолжать использовать ваши любимые средства Java-разработчика (например, интегрированные среды разработки, средства рефакторинга кода, отладчики и профайлеры) для разработки приложений JavaFX;

- новый графический «движок» для использования современных графических процессоров. В основе этого нового «движка» лежит графический конвейер с аппаратным ускорением, который называется Prism. Он сочетается с новым «оконным» инструментарием «Glass». Этот графический движок обеспечивает основу для нынешних и будущих усовершенствований, делая «насыщенную» графику простой, плавной и быстрой;

- новый декларативный язык разметки FXML, который основан на XML и используется для определения пользовательского интерфейса в приложениях JavaFX. Это не компилируемый язык и, следовательно, не требуется перекомпиляция кода каждый раз, когда вносятся изменения в расположение элементов управления;

- новый медиа-«движок», который поддерживает воспроизведение мультимедийного веб-контента. Это обеспечивает стабильную среду с низкой латентностью, которая основана на мультимедийном фреймворке GStreamer;

- веб-компонент, который дает возможность встраивания веб-страниц в приложения JavaFX, используя технологию WebKit для отображения HTML. Отображение с использованием аппаратного ускорения доступно с использованием Prism;

- широкий выбор встроенных элементов управления пользовательского интерфейса, которые включают в себя диаграммы, таблицы, меню и панели. Кроме того, предоставляется API, позволяющий сторонним разработчикам разрабатывать элементы управления, которые могут использоваться сообществом пользователей;

- упаковщик приложения, который позволяет создать простое в развертывании автономное настольное приложение, содержащее все Java-библиотеки времени выполнения, необходимые для установки и запуска приложений JavaFX;

- доступна на платформах Windows, Mac OS X, и Linux. С выходом JavaFX 2.2, JavaFX доступен на

всех основных десктопных платформах, обеспечивая согласованную рабочую среду для разработчиков и конечных пользователей. Oracle обеспечивает синхронизированные релизы и обновления на всех трех платформах, а также предлагает обширную программу поддержки для компаний, использующих критически важные приложения.

Архитектура и среда JavaFX

Платформа JavaFX 2 содержит следующие архитектурные компоненты [3]:

- общедоступный API и граф сцены;
- инструментарий «Quantum»;
- графический конвейер с аппаратным ускорением Prism;
- «оконный» инструментарий «Glass»;
- медиа-«движок»;
- веб-«движок»;
- Java 2D API;
- Open GL API;
- D3D API.

«Движок», на котором работает ваш код JavaFX, состоит из компонентов, которые включают высокопроизводительный графический движок Prism, «оконный» инструментарий «Glass», медиа-«движок» и веб-«движок».

Граф сцены является отправной точкой для построения приложений JavaFX и является иерархическим деревом узлов, которое представляет все визуальные элементы пользовательского интерфейса приложения.

Он может обрабатывать пользовательский ввод и может быть отображен.

Любой элемент графа сцены называется узлом. Каждый узел имеет идентификатор, класс стиля и поверхность. За исключением корневого узла, каждый узел графа сцены имеет единственного «родителя» и ноль или более «детей». Он также может иметь эффекты, такие как «размытие» и «тени»; прозрачность; преобразования; обработчики событий мыши, клавиатуры и т.д.; состояние, специфическое для приложения.

В отличие от Swing и AWT, граф сцены JavaFX также имеет графические примитивы, такие как прямоугольники и текст, в дополнение к элементам управления, контейнерам расположения, изображений и медиа.

Для большинства применений, граф сцены упрощает работу с интерфейсом пользователя, особенно, когда используются «насыщенные» пользовательские интерфейсы. В графе сцены может быть быстро выполнена графическая анимация, используя javafx.animation API, и декларативные методы, такие как XML doc.

API javafx.scene позволяет создавать и описывать несколько типов контента:

- Узлы: Формы (2-D и 3-D), изображения, медиа, встроенный веб-браузер, текст, элементы управления, диаграммы, группы и контейнеры.

- Состояние: Преобразования (позиционирование и ориентация узлов), визуальные эффекты и другие визуальные состояния контента.

- Эффекты: простые объекты, которые изменяют внешний вид узлов графа сцены, такие как размытие, тени и настройки цвета.

Особенности Java API для JavaFX

Платформа JavaFX 2 включает полный набор общедоступных API. Эти интерфейсы обеспечивают беспрецедентную свободу и гибкость для построения насыщенных клиентских приложений. Платформа JavaFX объединяет лучшие возможности платформы Java и всеобъемлющую многонаправленную медиа-функциональность в интуитивно понятной и универсальной среде разработки.

Задачи новых Java APIs для JavaFX:

- разрешить использование мощных возможностей Java, таких как типизированные коллекции, аннотации и многопоточность;

- упростить для веб-разработчиков использование JavaFX из других популярных динамических языков, таких как JRuby, Groovy и JavaScript;

- разрешить Java-разработчикам использовать другие языки системы, такие как Groovy, для написания больших или сложных приложений JavaFX;

- разрешить использование привязки, так же как это было в языке JavaFX Script. Это включает поддержку для обеспечения высокопроизводительной «ленивой» привязки, выражений привязки, связанной последовательности выражений, и частичной переоценки связывания;

- расширить библиотеку коллекций Java новыми структурами данных, которые позволяют приложениям связывать пользовательские интерфейсы с моделями данных, наблюдать за изменениями в этих моделях, и обновлять соответствующие элементы управления.

JavaFX 2 API и модель программирования являются продолжением линейки продуктов JavaFX 1.x. Большинство JavaFX API были портированы непосредственно на Java. Некоторые API, такие как расположение и медиа, наряду со многими другими, были улучшены и упрощены на основе отзывов, полученных от пользователей релиза JavaFX 1.x. JavaFX 2 опирается в большей степени на различные веб-стандарты, например, CSS для оформления элементов управления.

Графическая система

Графическая система JavaFX является низкоуровневой реализацией слоя графа сцены. Она поддерживает как 2-D, так и 3-D графы сцены. Графическая система JavaFX обеспечивает программный рен-

деринг, если возможности графического оборудования в системе недостаточны для поддержания рендеринга с аппаратным ускорением. На платформе JavaFX 2 реализованы два канала рендеринга с аппаратным ускорением:

- Prism - может работать как с использованием аппаратной так и программной визуализации, в том числе 3-D. Он отвечает за растеризацию и рендеринг сцен JavaFX. Следующие несколько возможных способов визуализации возможны на основе конкретного используемого устройства:

- DirectX 9 на Windows XP и Windows Vista;

- DirectX 11 на Windows 7;

- OpenGL на Mac, Linux, Embedded;

- Java2D, когда аппаратное ускорение не представляется возможным.

По возможности используется способ с полностью аппаратным ускорением, но когда это не доступно, используется способ визуализации на основе Java2D, так как этот способ визуализации поддерживается во всех средах JRE. Это особенно важно при обработке 3-D сцен. Тем не менее, производительность лучше, когда используется способ аппаратного рендеринга.

- Инструментарий Quantum связывает Prism и «оконный» инструментарий Glass вместе и делает их доступными для вышестоящего JavaFX-слоя. Он также управляет правилами многопоточности, связанными с рендерингом.

«Оконный» инструментарий Glass

Glass является фреймворком самого низкого уровня в графической системе JavaFX. Его основная задача заключается в предоставлении низкоуровневых услуг, таких как управление окнами, таймеры и поверхности. Он служит в качестве слоя, независимого от платформы, который соединяет платформу JavaFX и используемую операционную систему.

Инструментарий Glass также отвечает за управление очередью событий. В отличие от AWT, который управляет своей собственной очередью событий, Glass применяет для планирования использования потоков функциональность очередей событий операционной системы. Также в отличие от AWT, Glass работает в том же потоке, что и приложение JavaFX. В AWT часть функциональности, которая использует операционную систему, работает в одном потоке, а Java-уровень работает в другом потоке. Это приводит к серьезным проблемам, многие из которых решены в JavaFX путем использования единственного потока приложения JavaFX.

Потоки

Система выполняет два или более из следующих потоков в любой момент времени:

- поток приложения JavaFX: Это первичный поток, который используется разработчиками при-

ложеній JavaFX. Любая "живая" сцена, то есть сцена, которая является частью окна, должны быть доступна из этого потока. Граф сцены может быть создан и управляется в фоновом потоке, но когда его корневой узел присоединяется к любому «живому» объекту в сцене, этот граф сцены должны быть доступен из потока приложения JavaFX. Это позволяет разработчикам создавать сложные графы сцены в фоновом потоке, выполняя анимацию «живых» сцен плавно и быстро. Поток приложения JavaFX отличается от потока диспетчеризации событий Swing и AWT, поэтому необходимо соблюдать осторожность при внедрении кода JavaFX в Swing-приложения;

- поток визуализации Prism: Этот поток обрабатывает рендеринг отдельно от диспетчера событий. Это позволяет кадру N воспроизводиться, в то время как кадр N+1 обрабатывается. Эта способность выполнять параллельную обработку является большим преимуществом, особенно на современных системах с несколькими процессорами. В потоке визуализации Prism также могут выполняться несколько потоков растеризации, что помогает разгрузить работу, которую необходимо выполнить при рендеринге;

- медиа-поток: этот поток работает в фоновом режиме и синхронизирует последние кадры через граф сцены с помощью потока приложения JavaFX.

Событие Pulse

Pulse является событием, которое указывает графу сцены JavaFX, что наступило время для синхронизации состояния элементов на сцене графа с Prism. Pulse генерируется со скоростью максимум 60 кадров в секунду, когда на графе выполняется анимация. Даже если анимация не выполняется, событие Pulse планируется, когда что-то в графе сцены изменилось. Например, если положение кнопки изменилось, планируется событие Pulse. Когда генерируется Pulse, состояние элементов на графе сцены синхронизируется до слоя рендеринга. Pulse предоставляет разработчикам приложений способ асинхронной обработки событий. Эта важная функция позволяет системе группировать и выполнять события.

Схемы расположения элементов управления и CSS также связаны с событиями Pulse. Многочисленные изменения в графе сцены могут привести к множественным обновлениям схемы расположения или CSS, что может серьезно ухудшить производительность. Система автоматически выполняет проход CSS и схемы расположения один раз на событие Pulse, чтобы избежать снижения производительности. Разработчики приложений могут также вручную запустить проходы схемы расположения по мере необходимости, чтобы проводить измере-

ния до Pulse.

«Оконный» инструментарий Glass отвечает за выполнение событий Pulse. Он использует для этого таймеры высокого разрешения.

Медиа и изображения

Медиа-функциональность JavaFX доступна через API `javafx.scene.media`. JavaFX поддерживает визуальные и звуковые медиа. Поддержка предоставляется для MP3, AIFF, WAV аудио-файлов и видео-файлов FLV. Медиа-функциональность JavaFX обеспечивается как три отдельных компонента: объект Media представляет собой мультимедийный файл, объект MediaPlayer проигрывает медиа-файл, а объект MediaPlayer - это узел, который отображает медиа.

Компонент Media Engine был полностью переработан для JavaFX 2, чтобы увеличить стабильность, повысить производительность и обеспечить одинаковое поведение на разных платформах.

Встроенный браузер

Встроенный браузер JavaFX представляет собой новый компонент пользовательского интерфейса JavaFX, который обеспечивает веб-просмотрщик и полную браузерную функциональность посредством API. Этот компонент Web Engine основан на WebKit, который является «движком» веб-браузера с открытым исходным кодом, поддерживающим HTML5, CSS, JavaScript, DOM и SVG. Это позволяет разработчикам реализовывать следующие функции в своих Java-приложениях:

- Визуализация HTML контента с локального или удаленного URL.
- Поддержка «истории» и обеспечение навигации «Назад» и «Вперед».
- Обновление контента.
- Применение эффектов к веб-компонентам.
- Редактирование HTML-контента.
- Выполнение команд JavaScript.
- Обработка событий.

Этот встроенный компонент-браузер состоит из следующих классов:

- WebEngine предоставляет базовые возможности просмотра веб-страниц.
- WebView инкапсулирует объект WebEngine, включает HTML-контент в сцену приложения, обеспечивает поля и методы для применения эффектов и преобразований. Является расширением класса Node.

CSS

Каскадные таблицы стилей JavaFX CSS дают возможность применять настраиваемые стили для пользовательского интерфейса приложения JavaFX без изменения исходного кода этого приложения. CSS может быть применен асинхронно к любому узлу в графе сцены JavaFX. Стили JavaFX CSS также могут быть легко назначены сцене во время выпол-

нения, что позволяет динамически изменять внешний вид приложения.

JavaFX CSS основан на версии 2.1 спецификации W3C CSS с добавлениями от версии 3. Поддержка JavaFX CSS и расширения были разработаны, чтобы позволить корректно разбирать листы стилей JavaFX с помощью любого совместимого парсера CSS, даже не поддерживающего расширения JavaFX. Это позволяет смешивать стили CSS для JavaFX и для других целей (например, для HTML страниц) в единой таблице стилей. Все названия свойств JavaFX начинаются с расширения поставщика "-fx-", в том числе те, которые, казалось бы, совместимы со стандартными HTML CSS, потому что некоторые значения JavaFX имеют несколько другую семантику.

Визуальные элементы управления

Визуальные элементы управления JavaFX, доступные через JavaFX API, строятся с использованием узлов в графе сцены [4]. Они могут в полной мере обладать визуально «насыщенными» возможностями платформы JavaFX и переносимы между различными платформами. JavaFX CSS позволяет использовать темы и скины для элементов управления пользовательского интерфейса. Классы для построения элементов управления пользовательского интерфейса находятся в пакете `javafx.scene.control`.

Список элементов управления включает в себя типичные компоненты пользовательского интерфейса, которые вы можете знать из предыдущего опыта разработки клиентских приложений на Java. Тем не менее, JavaFX 2.0 SDK вводит новые элементы управления, например `TitledPane` и `TableView`.

Классы элементов управления обеспечивают дополнительные переменные и методы, помимо существующих в классе `Control`, для поддержки типичных взаимодействий пользователей в интуитивно понятной форме. Вы можете назначить определенный стиль вашему компоненту пользовательского интерфейса с помощью каскадных листов стилей. Для некоторых необычных задач, возможно, потребуется расширить класс `Control` для создания пользовательских визуальных компонентов или использовать интерфейс `Skin`, чтобы определить новый скин для существующего элемента управления. Поскольку элементы управления из пакета `javafx.scene.control` являются расширениями класса `Node`, то они могут быть интегрированы с анимацией, преобразованиями и анимированными переходами графа сцены.

Рассмотрим задачу создания кнопки, применяя к ней эффект «отражение» и анимируя ее путем изменения прозрачности с максимального до минимального значения. С помощью JavaFX API можно реализовать эту задачу с помощью всего нескольких строк кода. Вы можете настроить внешний вид элементов управления путем определения собственных каскадных ли-

стов стилей. Использование CSS в приложениях JavaFX является почти таким же, как использование CSS в HTML, потому что они основаны на одной и той же спецификации CSS. Визуальные состояния элементов управления определяются в файле `.css`

В дополнение к типичным элементам пользовательского интерфейса, JavaFX SDK предоставляет различные диаграммы в пакете `javafx.scene.chart`. В настоящее время поддерживаются следующие типы диаграмм: диаграмма областей, гистограмма, пузырьковая диаграмма, график, круговая диаграмма, диаграмма рассеивания.

Любая диаграмма может содержать несколько рядов данных. В отличие от других клиентских Java-инструментариев, используя JavaFX SDK, вы можете создать диаграмму в вашем приложении, добавив всего несколько строк кода. Также можно определять различные цветовые схемы и стили, применять визуальные эффекты, обрабатывать события мыши и создавать анимацию.

Схемы расположения

Контейнеры схем расположения или панели могут быть использованы для обеспечения гибкого и динамичного расположения элементов управления пользовательского интерфейса в графе сцены приложения JavaFX. JavaFX Layout API включает в себя следующие контейнерные классы, которые автоматизируют общие модели схем расположения:

- Класс `BorderPane` размещает узлы его контента в верхней, нижней, правой, левой, или центральной области.
- Класс `HBox` организует узлы его контента горизонтально в одну строку.
- Класс `VBox` организует узлы его контента вертикально в один столбец.
- Класс `StackPane` размещает узлы его контента в стеке.
- Класс `GridPane` позволяет разработчику создать гибкую сетку из строк и столбцов, в которых размещаются узлы контента.
- Класс `FlowPane` организует узлы его контента в горизонтальный или вертикальный "поток", обтекая указанные границы по ширине или высоте.
- Класс `TilePane` размещает узлы его контента в ячейках одинакового размера.
- Класс `AnchorPane` позволяет разработчикам создавать узлы-якоря для привязки к верхней, нижней или левой стороне, или в центре макета.

Для достижения желаемой структуры расположения, различные контейнеры могут быть вложены.

2-D и 3-D преобразования

Каждый узел в графе сцены JavaFX с помощью классов пакета `followingjavafx.scene.transform` может быть преобразован в системе координат XY:

- перемещение – перемещение узла из одного

места в другое по отношению к его первоначальной позиции;

- масштабирование – увеличение или уменьшение размера узла в зависимости от коэффициента масштабирования;
- сдвиг – поворот одной из координатных осей так, чтобы ось X и ось Y перестали быть перпендикулярными;
- поворот – поворот узла относительно указанной опорной точки сцены;
- аффинное преобразование – выполнение линейного отображения из 2-D/3-D координат в другие 2-D/3-D координаты, сохраняя при этом параллельность прямых. Этот класс используется неявно в классах, которые отвечают за другие преобразования.

Визуальные эффекты

Разработка «насыщенных» клиентских интерфейсов в графе сцены JavaFX включает в себя использование визуальных эффектов или эффектов, улучшающих внешний вид приложений JavaFX в реальном времени.

Некоторые из визуальных эффектов, доступных в JavaFX, приведены ниже:

- Тень – отображает тень данного контента за контентом, к которому применяется эффект.
- Отражение – отображает отраженную версию контента под ним.
- Освещение – имитирует источник света, падающего на данный контент, и позволяет придать плоскому объекту более реалистичный трехмерный вид.

Развертывание

Существует три режима развертывания приложения JavaFX:

- Автономное приложение – приложение JavaFX устанавливается на локальный диск и запускается двойным щелчком по jar-файлу. Этот режим идеально подходит, когда пользователю не нужен онлайн-доступ или они не имеют его.
- Браузерное приложение – в этом режиме приложение JavaFX встраивается в веб-страницу и запускается автоматически при доступе к веб-странице. Оно может взаимодействовать с веб-страницей с помощью JavaScript.
- Web Start - в этом режиме приложение загружается с веб-сервера, а затем запускается на клиентском компьютере. Сразу после загрузки приложения, пользователи могут запустить его двойным щелчком JNLP-файлу.

Обзор JavaFX Scene Builder

JavaFX Scene Builder является ключевым средством разработки для платформы JavaFX [5]. Он обеспечивает среду визуальной разработки,

которая позволяет быстро создавать графические пользовательские интерфейсы для приложений JavaFX без необходимости написания кода. Когда вы создаете макет пользовательского интерфейса, автоматически генерируется соответствующий код на языке FXML.

JavaFX Scene Builder предоставляет простой, интуитивно понятный интерфейс, который может помочь даже непрограммистам быстро создать прототип интерактивного приложения, который подключает компоненты пользовательского интерфейса к логике приложения.

Целевая аудитория

Целевая аудитория JavaFX Scene Builder включает:

- Java-разработчики: они могут быстро создавать прототипы пользовательского интерфейса приложения и отдельно разрабатывать логику приложения;
- дизайнеры: они могут быстро создавать прототипы пользовательского интерфейса без написания какого-либо кода. Они могут проектировать и просматривать макет визуального интерфейса и определить его вид с помощью таблиц стилей.

Основные характеристики

JavaFX Scene Builder включает в себя такие основные характеристики:

- drag-and-drop WYSIWIG интерфейс позволяет быстро создавать пользовательский интерфейс без необходимости написания кода. Вы можете добавлять, комбинировать и редактировать элементы управления JavaFX с помощью библиотеки элементов управления и панели содержания.
- Тесная интеграция с интегрированной средой разработки NetBeans обеспечивает оптимальный рабочий процесс разработки.
- Простая интеграция с любой интегрированной средой разработки Java, так как это - автономный инструмент разработки.
- Автоматическая генерация кода FXML происходит при создании и изменении элементов пользовательского интерфейса. Сгенерированный код FXML хранится в отдельном файле.
- Редактирование и просмотр в реальном времени позволяет быстро визуализировать изменения в пользовательском интерфейсе без необходимости компиляции. Эти функции позволяют минимизировать время разработки приложения. Вы также можете применять каскадные таблицы стилей к элементам пользовательского интерфейса и просматривать полученный внешний вид этих элементов.
- Обеспечивается полный доступ к библиотеке элементов управления JavaFX 2.2.
- Поддержка CSS позволяет гибко управлять

внешним видом пользовательского интерфейса приложения.

- Поддержка кроссплатформенности обеспечивается для операционных систем Windows и Mac OS X.
- Бесплатность.

Выводы

В результате проведенного анализа JavaFX можно отметить следующие «за» и «против»:

«За»:

- Предлагает Java API – можно использовать ваши навыки в Java и использовать существующие возможности и библиотеки JVM.
- Предлагает DSL, для JVM-языков, например, Groovy (GroovyFX) и Scala (ScalaFX).
- Опциональность: можно выбирать между «макетированием» с использованием языка разметки FXML и программированием с использованием языка Java.
- Одна и та же среда разработки для различных задач (в том числе отладки, рефакторинга и т.д.).
- Отсутствие межбраузерных проблем.
- Поддержка CSS (как в HTML).
- HTML и/или JavaScript могут быть интегрированы в приложения JavaFX.
- Swing и JavaFX можно использовать в одном приложении, поэтому существующие приложения Swing могут быть расширены.
- JavaFX 2 обеспечивает единую архитектуру для написания приложения один раз и последующего развертывания его в различных контекстах (как отдельного приложения, приложения, встроенного в веб-браузер, или запускаемого с помощью Java Web Start). В будущем будут добавлены дополнительные контексты (например, запуск того же приложения на мобильном устройстве).

«Против»:

- На клиентском компьютере требуется Java Runtime Environment.
- JavaFX только частично является программным обеспечением с открытым исходным кодом.

Oracle JavaFX runtime и SDK по-прежнему будут выпускаться под лицензией Java Binary Code License.

- Предлагает меньшее количество виджетов и других функций, чем HTML5.
- Хотя JavaFX в будущем заменит Swing, разработка приложений для JavaFX отличается от Swing из-за нескольких новых концепций. Конечно, это является следствием добавления «насыщенных» возможностей, таких как анимация и визуальные эффекты. Тем не менее, это не является существенным недостатком, и для Java-разработчиков проще изучить некоторые новые концепции, чем изучать HTML и JavaScript.

Список литературы

1. Carl Dea. *JavaFX 2.0: Introduction by Example*, apress, 2011. – 181 p.
2. Oracle. *JavaFX 2 Documentation. JavaFX Overview* [Электронный ресурс]. – Режим доступа : <http://docs.oracle.com/javafx/2/overview/jfxpub-overview.htm>.
3. Oracle. *JavaFX 2 Documentation. JavaFX Architecture and Framework* [Электронный ресурс]. – Режим доступа : <http://docs.oracle.com/javafx/2/architecture/jfxpub-architecture.htm>.
4. Oracle. *JavaFX 2 Documentation. Using JavaFX UI Controls Release 2.1* [Электронный ресурс]. – Режим доступа : <http://docs.oracle.com/javafx/2/layout/jfxpub-layout.pdf>.
5. Oracle. *JavaFX 2 Documentation. JavaFX Scene Builder Overview* [Электронный ресурс]. – Режим доступа : <http://docs.oracle.com/javafx/scenbuilder/1/overview/jsbpub-overview.htm>.

Поступила в редколлегию 29.09.2012

Рецензент: д-р техн. наук, проф. И.В. Рубан, Харьковский университет Воздушных Сил им. И. Кожедуба, Харьков.

«

РОЗРОБЛЕННЯ НАСИЧЕНИХ ІНТЕРНЕТ-ДОДАТКІВ ЗА ДОПОМОГОЮ JAVA FX

Ю.Е. Парфьонов, В.М. Федорченко

Розглядається нова платформа створення графічного інтерфейсу користувача - JavaFX, заснована на технології Java. Представлені основні характеристики платформи, переваги використання розробниками. Обговорюються відмінності даної платформи від інструментарію Swing. Розглянуто склад архітектури і описана середу JavaFX. Розглянуто особливості Java API для JavaFX, особливості реалізації графічного інтерфейсу, принципи реалізації графічної системи. Виконано аналіз реалізації багатопотоковості і системи подій. Розглядається функціональність платформи JavaFX з точки зору використання медіа, 2D/3D анімації та основних елементів управління. Дається характеристика середовища розробки додатків JavaFX.

Ключові слова: графічний користувальницький інтерфейс, платформа, граф сцени, 2D/3D анімація, Java API, JDK, SDK, веб-компонент, елемент управління, мова розмітки, потоки, події, каскадні таблиці стилів, програмний рендеринг, контейнерні класи, схеми розташування.

DEVELOPMENT OF RICH INTERNET APPLICATIONS USING JAVA FX

Y.E. Parfyonov, V.M. Fedorchenko

In the paper JavaFX - a new Java-based platform for GUI design is considered. The main characteristics of the platform and the benefits of its using are highlighted. The differences of the platform from Swing tools are discussed. We considered the architecture and described environment of JavaFX, considered the features of the Java API for JavaFX, implementation of the JavaFX GUI and the graphics system. Multi-threading and the event system were analyzed. The functionality of the JavaFX platform in terms of media, 2D/3D animation and basic controls, characteristic of the development environment are mentioned.

Keywords: *graphical user interface, platform, scene graph, 2D/3D animation, Java API, JDK, SDK, Web component, control, markup language, threads, events, cascading, style sheets, software rendering, container classes, layout.*