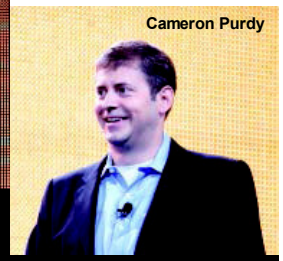

Java™
magazine
By and for the Java community
//NOVEMBER/DECEMBER 2011 /
2
「進化し続けるJava」
JavaOne 2011
Javaコミュニティによる
世界最大のイベントを
レポート
五感への
衝撃
JavaFX 2.0により再活性化するクライアントサイドのJava開発 6

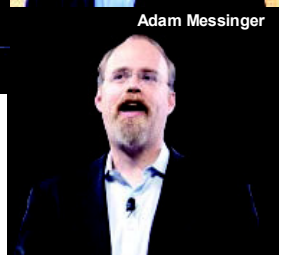
JavaOne 2011

目標への集中
高まる評価

10月2日から6日にかけて、「Moving Java Forward（進化し続けるJava）」をテーマに**JavaOne 2011が開催されました。**イベントが終盤に近づく頃には、参加者のほとんどが、10年以上前Javaが最初に広まった頃のような新時代の幕開けを感じていました。
本年のJavaOneでは、オラクルがJavaプラットフォームを管理下に置いた意味が明確となりました。重点目標に注力して、最終的にすべての構成要素を1つにまとめた統合的なJavaプラットフォームの構築を目指すという、一貫性のある新たな方向性が提示されたのです。JavaOneで大きく取り上げられた話題は、「Java/JDK 8」、「Java Platform, Enterprise Edition（Java EE）」、「Java Platform, Standard Edition（Java SE）」、「Java Platform, Micro Edition（Java ME）」、さらには「Java Community Process（JCP）」、「Javaユーザー・グループ」、そしてもちろん「Javaコミュニティ」です。特に重要な話題について、いくつかご紹介しましょう。

写真：HARTMANN STUDIOS


Cameron Purdy


Adam Messinger

Java/JDK 8ロードマップ

JavaOneでは、Javaへの投資活動や技術開発についてのオラクルの長期的な構想が示されました。Javaのストラテジー・キーノートでは、オラクルの**Hasan Rizvi**、**Adam Messinger**、**Cameron Purdy**がJava 8の明確なロードマップについて説明しました。この講演によると、オラクルのJavaチームはJavaコミュニティからのフィードバックに対応する形で、Javaのメジャー・リリースを（2000年代初頭に機能していた）2年ごとのサイクルに戻すという決定に至ったということです。Java 8のリリースは2013年夏まで延期されますが、その対象範囲は拡大されています。

Java 8は、OpenJDKへのOracle JRockit機能の組込み、Project Jigsaw（モジュール化）、Project Lambda（クロージャ）、JavaFX 3.0、高度なJavaScript相互運用などが含まれる、完全一体型のJavaプラットフォームとなります。そのすべての機能がWindows、Mac OS X、Linux、Oracle Solaris、組込みプラットフォームで完全にサポートされ、さらに新しいデバイスのサポートも追加されます。また、NetBeansの次バージョンでは、これらのすべての新機能がサポートされる予定です。

Java EEとJava ME

Java EE 7開発では、今後クラウド・プラットフォームが標準化されることを見据えて、マルチテナント、キャパシティ・オンデマンド、自動プロビジョニングの3点に重点を置いています。また、Java MEでは、統合された小型端末向け/組込みJavaプラットフォームを目指して、Java ME、Java SE、Java SE Embedded（組込みデバイス向けJava SE）の一体化を重点項目としています。

JCP、Javaユーザー・グループ、コミュニティ

JavaOne 2011では、コミュニティの意見とオープン性を重要視する姿勢も顕著に現れていました。たとえば、JCP Executive Committeeでの議論やJCP.next（JSR-348）の承認プロセスにJavaユーザー・グループ（JUG）が関与することで、JCPの活性化につながっています。JUGが主導する「Adopt-a-JSR」構想では、それぞれのJSRにJUGが積極的に参加することで、今後のJavaプラットフォームの決定事項にコミュニティがより深く関わることを目標としています。

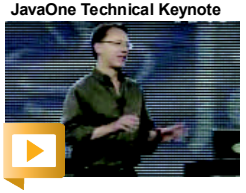
Twitterも仲間に

TwitterのRob Benson氏による発表には誰もが驚きました。Javaのステークホルダの中でも有力者であり、高性能Javaプラットフォームに強い関心を持つBenson氏は、OpenJDKプロジェクトに加えて今後はJCPにもTwitterが参加することを表明しました。Twitterの公式発表はこちらから。

さあ前へ！

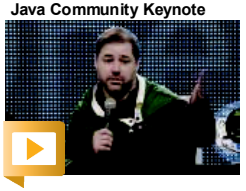
総括すると、JavaOne 2011は、Javaが全方面に向かって確実に前に進んでいることを示す極めて重大なイベントとなりました。

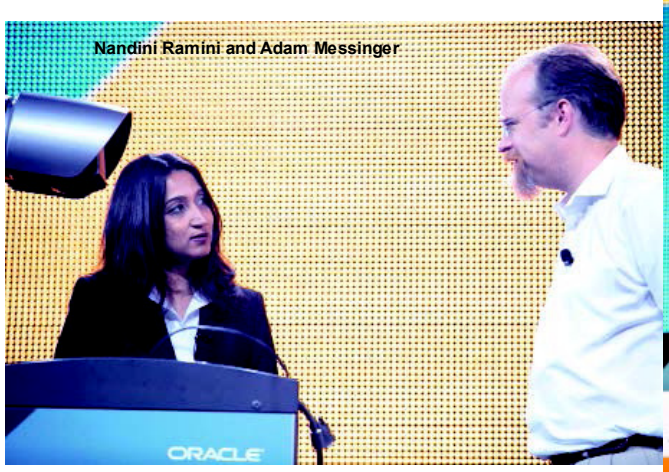
JavaOne Technical Keynote



Java Strategy Keynote

Java Community Keynote





JavaFX 2.0の誕生

JavaOne 2011では、エンタープライズ・ビジネス・アプリケーション向けの高度なJava UIプラットフォーム「JavaFX 2.0」のリリースについて大きく取り上げられました。JavaFX 2.0は、最高レベルのリッチ・クライアント・プラットフォームとして、Javaが進化していく中で次のステップを担うものとなります。

10月4日月曜日に行われたJavaOneテクニカル・キーノートでは、オラクルのクライアントJavaプラットフォーム部門のチーフ・アーキテクトであるRichard Bairが登壇し、Windows（32ビット版Windows XP、32ビット版/64ビット版Windows Vista、Windows 7）向けのJavaFX 2.0の機能概要について詳しく説明しました。また、NetBeans 7.1 Beta（JavaFX 2.0を完全サポート）や、Mac OS X向けJavaFX 2.0開発者プレビュー、JavaFX Scene BuilderのEarly Access版の発表がありました。

JavaFX 2.0のおもな機能は、100% Java API、新しいUIマークアップ言語「FXML」、Swingとの完全統合などです。また、JavaFX 2.0にはWebKitベースのWebコンポーネントが用意されており、開発者はこれを利用してネイティブなJava機能とWeb技術の動的な機能をシームレスに組み合わせることができます。

写真：HARTMANN STUDIOS

10月5日火曜日に開かれたJavaのストラテジー・キーノートでは、オラクル開発部門の統括責任者である**Adam Messinger**が登壇し、OpenJDKコミュニティ内の新しいプロジェクトとして、JavaFXプラットフォームのオープンソース化を提案していることを発表しました。また、オラクルは今後JCPと連携して、JavaFXをJavaプラットフォームの公式標準とする提案を行う予定であると述べました。まずはJavaFXのUIコントロールや関連ライブラリを提供し、その後複数の段階に分けて他のJavaFXコンポーネントをリリースする計画を立てています。

オラクルのJavaクライアント・グループ開発部門の統括責任者である**Nandini Ramani**は同基調講演で、Apple iPad 2.0、Android 3.1対応Samsung GALAXY Tab 10.1、Acer Windows 7タブレットでJavaFX 2.0を実行するデモを続けて行い、この中でRamaniは次のように語りました。「コミュニティからのご意見をお待ちしています。皆様の求めるものを優先したいと思っています」（Nandini Ramaniへのインタビューの詳細は、32ページの「五感への衝撃」をご覧ください。）

将来は、Mac向けJava 7リリースと同時にMac OSを完全にサポートし、さらに他のプラットフォーム（Linux、Solarisなど）をサポートする予定です。Nicolas LorainのJavaOne 2011セッション「Introduction to JavaFX 2.0（JavaFX 2.0の紹介）」やその他のJavaFXセッションについては、parleys.comで視聴できます。

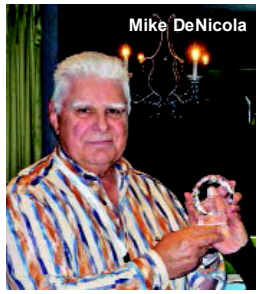
Duchess、討論会で見解を語る

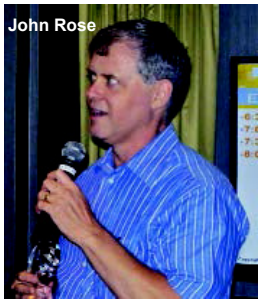
Duchessは、世界中にいる女性のJava開発者を結ぶネットワークです。JavaOne 2011では、同ネットワークの現プレジデントであるRegina ten Bruggencate氏と共同創業者のClara Ko氏による自由討論セッションにおいて、IT業界における女性の役割が議論されました。

Duchessのネットワークで結ばれている女性エンジニアは、地域のJavaユーザー・グループ（JUG）ミーティングやグローバルなカンファレンス、またはオンラインでそのつながりを保っています。オランダで誕生した同ネットワークは、積極的な女性たちと、そうした女性を支援するJUGリーダーおよびメンバーの貢献によって、昨年メンバーの数が倍増しました。現在、同ネットワークには57か国400人のメンバーが所属しています。

「Duchessは、典型的なITプロフェッショナルに対する『オタクっぽい男性』というイメージを変えつつある」とten Bruggencate氏が語れば、「IT業界は、より良質なソフトウェア開発のための協力を惜しまず、また協調性に長けた開発者の存在により前進している」とKo氏も続けます。DuchessにはオンラインでDevoxxでもDuchessの活躍をご覧になれます。

Duchess プレジデント
Regina ten Bruggencate





JAVA Community Process 受賞者を発表

JavaOneで執り行われたJCPのセレモニーでは、**今年で9回目となるJava Community Process（JCP）各賞**の受賞者が発表されました。「JCP Member/Participant of the Year（最優秀JCPメンバー/参加者）」、「Most Innovative JSR（革新性に最も優れたJSR）」、「Outstanding Spec Lead（最優秀仕様リード）」の3賞が対象となります。

JCP Member of the Year賞には、JCP.next（JSR-348）ワーキング・グループ会長として貢献した富士通の**Mike DeNicola**氏が選出されました。DeNicola氏はJCP改革プランの策定に尽力し、重要なJSRの発展の速度を加速させました。

また、Most Innovative JSR賞を獲得したのは、JSR-292、「Supporting Dynamically Typed Languages on the Java Platform」（Javaプラットフォームにおける動的型付け言語のサポート）です。審査委員会より、「JSR-292はJava以外の言語のサポートを目的とした初めてのJSRであり、Java VMの長期的な成功を確約するもの」という評価を受けての受賞です。

Outstanding Spec Lead賞は、JSR-292を担当したオラクルのコンサルティング・エンジニア、**John Rose**の手に渡りました。Roseは、専門家グループや、より規模の大きいJava仮想マシン言語コミュニティを含めて、JSR-292の多様な参加者の間で着実に合意を形成した点が評価されました。Roseはこの受賞を受けて、受賞できたことへの深い感動と仲間への感謝の意を表しました。彼は次のように記しています。「私たちは一丸となって、根本的にまったく新しい命令とデータ型をJava仮想マシンに導入しました。目標としては大きく、手間のかかる作業でした。JSR-292専門家グループ・メンバーの皆さん、同僚のエンジニア、企業のスポンサー、そしてDa Vinci Machineコミュニティの献身的な努力があったからこそ、この取組みを実行に移すことができました」

受賞されたのはわずか3名でしたが、JCPへ多大な貢献をしていただいた本年の候補者すべての方々に感謝を申し上げます。

JSR-335専門家グループ、Java 8のラムダ式構文を決定

9月初旬に、JSR-335、「Lambda Expressions for the Java Programming Language（Javaプログラミング言語のラムダ式）」の専門家グループが、構文に関する重要な決定を行いました。この決定によると、Javaのラムダ式は次のような構造になります。

[パラメータ] [一重の矢印 ->] [式または文]

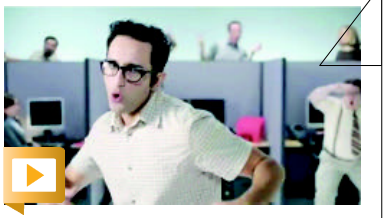
いくつかの例を紹介しましょう。

```
x -> x + 1
(x, y) -> x + y
(x, y) -> { System.out.printf("%d + %d = %d%n", x, y, x+y); }
```

C#やScalaのラムダ式と類似した構文ですが、これらの言語では矢印に「=>」が使用される点が異なります。Javaラムダ専門家グループは、Javaの式には「==」や「<=」という演算子も存在するため、一重の矢印「->」の方がわかりやすいと判断しました。

Javaラムダ構文の決定を受け、仕様リードのBrian Goetz氏は次のようにコメントしています。「ラムダ構文について膨大な調査を行ってきましたが、これぞという選択肢はありませんでした（それぞれの形式に良い面と悪い面があり、他の選択肢よりもはっきりと良いと言えるものは存在しなかったのです）。そのため、Java独自の新しい形式を考え出すよりは、Javaにもっともよく似た言語であるC#とScalaですでに成功している形式を採用する方が良いと考えました」

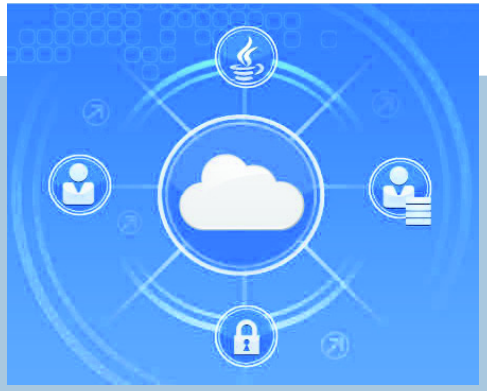
Javaのクロージャに関する背景については、Mark Reinholdのブログ記事「Closures for Java（Javaのクロージャ）」をご覧ください。ラムダ式については、JDK 8をダウンロードすることで実際にお試しいただけるようになります。詳細は、OpenJDKのProject Lambdaサイトをご覧ください。



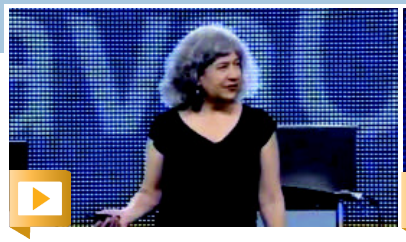
Java Life

仕事場でひたすらコーディングに打ち込んでいる皆さんへ。このオリジナルのラップ・ミュージック・ビデオはJavaOne 2011に向けて制作されたもので、「Java Life」を讃えています。Javaのある生活を送っている皆さん、ぜひご覧ください。

Oracle Public Cloudが実現するJavaエンタープライズ環境の相互運用性とスケーラビリティ



「Write once, run anywhere」（一度書けばどこでも実行できる）という概念はクラウドにも当てはまります。オラクルの新しいJava中心型インフラストラクチャ・サービス「**Oracle Public Cloud**」は、「業界標準の技術を基盤とし、他のクラウドや自社データセンターとの完全な相互運用をサポート」しています。その意味は仕様に現れています。オラクルのクラウド基盤は、Java Platform, Enterprise Edition 5（Java EE 5）スタック上で実行するOracle WebLogic Server 11gアプリケーション・サーバー・クラスタです。同サービスのツールキットは、Servlet 2.5、JSP 2.1、JSF 2.0、Enterprise JavaBeans 2.1/3.0、JPA 2.0、JAX-WS 2.1、JAX-RS 1.1（Jersey 1.9）で構成されています。また、トリプルミラーリングとオフサイト・テープ・バックアップを備えたシングル・テナント・アーキテクチャを採用しているため、データのセキュリティと整合性をより確実に確保できます。エンタープライズJavaアプリケーションの実行環境が整っており、可用性の向上や融通性の高いスケーラビリティを必要としている場合は、Oracle Public Cloudの利用をご検討ください。現在、このサービスは、機能が制限された試用モードでご利用いただけます。



Java EEをクラウドに移行することについて語るオラクルのJava EE 7仕様リード、Linda。Java EE 7のプラットフォームに関する基調講演にて

クラウドで実行するJavaアプリケーションの課題について語るIBMのクラウド・コンピューティング部門チーフ・アーキテクト、Jason McGee氏。IBMの基調講演にて

Xbox Kinectと連動するJavaFX

Xbox KinectとJavaFX 2.0を組み合わせたら、何を実現できるでしょうか。オラクルのJavaエバンジェリストであるSimon RitterのJavaOneセッション「Interfacing with the Interface:JavaFX 2.0, Wiimote, Kinect, and More」（インタフェースとの連動：JavaFX 2.0、Wiimote、Kinectなど）では、Javaクライアント部門の開発者エクスペリエンス・アーキテクトであるJasper Pottsがセッションの最後に演壇で実演し、参加者は「何を実現できるのか」についてのヒントをつかむことができました。KinectにはXboxゲームで利用するための、3D映像、音声、モーションを読み取るセンサーが備わっています。一方で、センサーが読み取ったデータにアクセスするためのC++ SDKも用意されています。

RitterとPottsはOpenNIフレームワークを利用してKinect APIをラップし、JavaFXがJava Native Interface経由でKinect APIにアクセスできるコードを作成しました。作業にはやや時間がかかり、特殊なテクニックも必要でしたが、最終的には次のようなJavaFXコードでKinectのセンサーから3Dの深度データを取得できるようになりました。

```
context =
  Context.createFromXmlFile(config.getKinectXMLConfig(),
  scriptNode);
depthGenerator = DepthGenerator.create(context);
DepthMetaData depthMetaData = getDepthGenerator().getMetaData();
userGenerator = UserGenerator.create(context);
width = depthMetaData.getFullXRes();
height = depthMetaData.getFullYRes();
```

このデモや、その背景にあるJavaFXテクノロジーについて、Pottsは次のように述べました。「これはJavaFXラボが提供するクールな3Dアプリケーションです。Kinectコントローラで操作しながら、私自身がDukeを模して、3DのDukeをお見せしました。とても楽しめたうえに本番でうまく行ったので、大変嬉しく思っています」

Ritterがプレゼンテーションの冒頭で説明したように、人間は長年にわたってマシンと調和してきました（たとえば、タイプライターは偉大な発明でした）。今やソフトウェア開発者はJavaFXなどのオープンソース・テクノロジーを利用して、Kinectのような最新の制御デバイスとも連動します。Ritterの言葉どおり、今すぐ実践すべきことは「アイデアを出す、行動する、とにかく作る！」ことです。

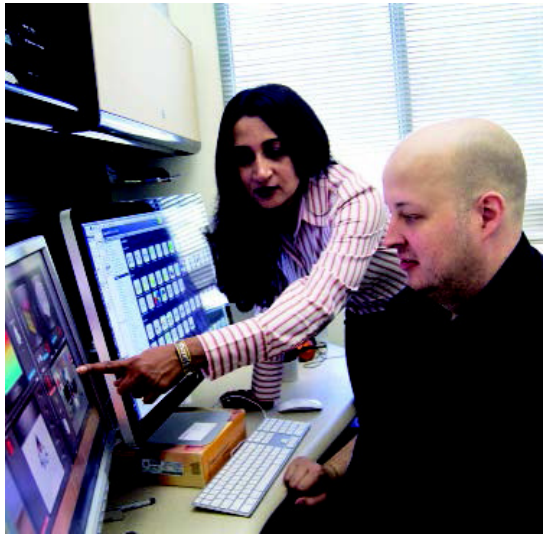
写真：BOB ADLER

JavaFX 2.0

五感への衝撃

オラクルの**Nandini Ramani**が*Java Magazine*に語る、JavaFX 2.0の主要な機能。**Michael Meloan**

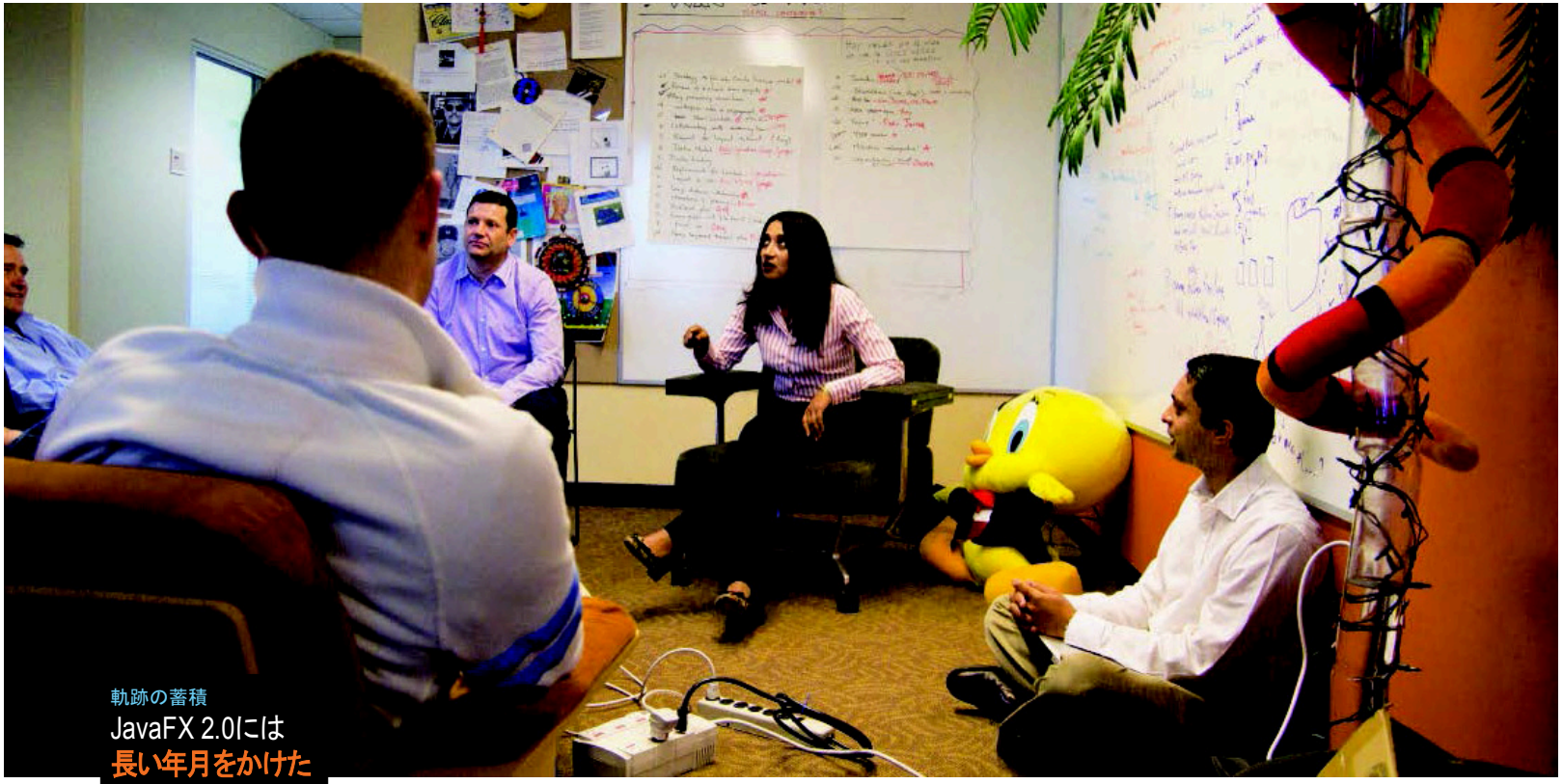
JavaFX 2.0は、オラクルが提供する最高レベルのリッチ・クライアント・アプリケーション向け開発環境です。JavaのAPIとクロスプラットフォームを柔軟に活用すべく設計されたJavaFX 2.0は、最適なUIを作成するための最先端のソリューションを提供します。オラクルのJavaクライアント開発部門の統括責任者であるNandini Ramaniが、高度なシステムを構築する開発者を支援するこの新しいJavaテクノロジーについて解説します。



写真：BOB ADLER
画像：PAULINA MCFARLAND

***Java Magazine*：** JavaFX 2.0の目標と、システムアーキテクトや開発者にとってのJavaFX 2.0のメリットを教えてください。

Ramani： JavaFX 2.0はクライアント側の開発者にとって大きな進化です。スタックの最上層にある6,500個以上の新しいAPIによって、色彩豊かなグラフィックで強力なユーザー・インタフェースを構築できます。音声/映像の再生機能、UIコントロー



新たな機能について議論する、Nandini Ramani（中央）とJavaFX 2.0チームのメンバー

軌跡の蓄積

JavaFX 2.0には**長い年月をかけた**厳密な開発、テスト、クロスプラットフォームの機能が反映されます。

ルやレイアウト・コンテナ、シーン・グラフやアニメーション・ライブラリ、監視可能なコレクションやUIバインディングなど、新しいAPIにはあらゆる機能が含まれています。また、スタックの最下層に、新しいウィンドウ・ライブラリやハードウェア・アクセラレーションを利用するグラフィック・ライブラリを用意しました。

IT業界は今、GPU（グラフィック・プロセッシング・ユニット）を搭載したマルチコア/マルチスレッド・プラットフォームに移行しています。JavaFX 2.0ではこの特性が活かされ、実行効率とUI設計の柔軟性の向上につながっています。最初の目標は、良質なビジネス・アプリケーション、ダッシュボード、リッチなアニメーションUI、さまざまなWebベースのデータソースを統合するマッシュアップの構築を支援するツールセットをエンタープライズ・アプリケーションのアーキテクトや開発者に提供することです。

***Java Magazine*：** JavaFX 2.0で開発したソフトウェアはスタンドアロンでもブラウザ内でも、さらにはJava Web Startを使用しても実行できますが、これらのオプションのそれぞれについて、どのようなアプリケーション分野がもっとも適していますか。

Ramani： スタンドアロン・モードはランタイム環境の緻密な制御が必要な場合に適しています。アプリケーションはクライアント側のJARファイルから起動するか、もしくは既存のサードパーティ製ツールを利用して従来のOS固有の実行可能ファイルにバンドルできます。ブラウザで実行するJavaFXアプリケーションは、従来のプラグイン・モデルによってWebページに組み込むことができます。さらに、Java Web Startを利用したアプリケーションはデスクトップ上で実行されますが、その前にWebからダウンロードされるため、Webアプリケーションと同様にアプリケーションを実行するたびにアプリケーションを更新できます。このように、一元管理されたWeb上の場所からダウンロードすることで、配備やアップデート配信が単純化できるだけでなく、アップデートも自動的に行われます。

***Java Magazine*：** Adobe FlashやMicrosoft Silverlightと比較して、JavaFX 2.0にはどのような利点がありますか。

Ramani： 何よりもまず、JavaFX 2.0はJavaプラットフォームの最上層に位置づけられているということが最大の利点です。長い年月をかけた厳密な開発、テスト、Javaが得意とするクロスプラットフォームの機能がすべて反映されます。JavaFX 2.0は広範に利用されている効率的なJava仮想マシンにホストできます。また、JavaFX 2.0のユーザーの多くはすでに使用しているJava EEをバックエンド処理に利用するため、フロントエンドのJavaFXでも既存のスキル・セットやツール、インフラストラクチャを活用できます。バックエンド開発に適用されるあらゆるものを、フロントエンドでも適用できるのです。さらに、使用しているサードパーティ製のラ

JavaFX 2.0の紹介

イブラリという観点では、Javaは最大かつ最強のエコシステムです。クライアント・アプリケーションの開発にそのようなライブラリのサポートを活用できるのも大きな利点となります。

***Java Magazine*：** JavaFX 2.0とJRuby、Scala、Groovyなどの言語との間に相互運用性はありますか。

Ramani： もちろんあります。それらの言語がJavaプラットフォームの最上層で実行できるのと同様に、JavaFX 2.0とも相互運用性があります。実際Groovyコミュニティには、JavaFXアプリケーションのBuilderデザイン・パターンを利用するための「GroovyFX」というプロジェクトが存在します。JavaFXに関連する作業のほとんどはオープンソース化されており、そのような言語を実装しているユーザーの多くはJavaFX 2.0の技術を喜んで利用しています。一部のJava開発者にとっては、動的スクリプト言語は開発者用ツールキットの重要な部分を占めるものです。JavaFXはそのような動的言語ともシームレスに連携します。

JavaFXは歩み始めたばかりですが、JavaBeansの命名規則に従ったPOJOのような広く知られたデザイン・パターンを採用しているため、そうした動的言語の多くではすでに、JavaFXのサポートを強化しています。これに加えて、JavaFXはイベント処理などにおいて一貫性のあるAPIイディオムを基に慎重に設計されているため、他の言語でもAPIにシンタックス・シュガーを簡単に取り入れることができます。

***Java Magazine*：** Prism（ハードウェア・アクセラレーションを利用したグラフィック・パイプライン）や新しいGlassウィンドウ・ツールキットを利用するメリットが明らかにあると言えるアプリケーション・シナリオはどのようなものですか。

Ramani： PrismはJavaFXシーンのラスタライズとレンダリングを処理し、ハードウェア・レンダラとソフトウェア・レンダラの両方を実行できるものです。Glassウィンドウ・ツールキットはJavaFX 2.0グラフィック・スタックの最下位レベルのフレームワークであり、タイマー、サーフェイス、ウィンドウを管理します。

開発者がグラデーションや網掛け、もしくは高度なグラフィックを必要とする場合に、ハードウェア・アクセラレーションを利用すればシステムの応答性が飛躍的に向上します。さらに、アプリケーションで表の組立てや大きなデータ・セットの表示が必要な場合も、Prismの価値が発揮されます。PrismとGlassを組み合わせることで、Swing/AWTのクラスのインスタンス化や読込みを行う必要がなくなるため、JavaFX 2.0はパフォーマンス面において確実に有利と言えます。

***Java Magazine*：** 開発者にとってFXML（JavaFX 2.0のXMLベース・マークアップ言語）を利用するメリットと、Scene BuilderのFXML生成がどのように役立つかについて説明してください。

Ramani： NetBeansやEclipseなどのJava IDEはXMLと容易に連携できるため、FXMLがXMLベースであるという点はツールの観点から非常に大きなメリットとなります。エンタープライズ・アプリケーションのレイアウト設計はテクニックと多大な労力をもっとも要する作業ですが、視覚的なJavaFX Scene Builderツールを活用すれば手作業によるコーディングが減少するため、生産性が確実に向上します。また、FXMLは次に挙げる2通りの方法でJavaFX Scene Builderに組み込まれます。ひとつは、開発者がプログラムに変更を加えてからFXMLを生成する方法で、もうひとつは、JavaFX Scene BuilderでJavaFX UIコンポーネントを設計してからFXMLマークアップとして保存する方法です。JavaScriptもこのシナリオにシームレスに統合できるため、非常に柔軟性が高いと言えます。これらのツールではCSSも利用できます。アプリケーションの作成後は、スタイル・シートかFXMLコードを変更することで、再コンパイルせずにまったく新しい外観を生成できます。このような傾向は業界全体に見られるため、JavaFX 2.0でもこのようなツールセットを用意しています。

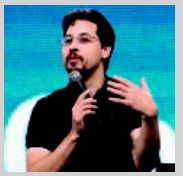
***Java Magazine*：** HTML5はJavaFX 2.0の今後の展望に含まれていますか。

Ramani： HTML5テクノロジーは現在もW3Cによって開発中ですが、現在定義されているHTML5のサポート内容については、JavaFX 2.0に組み込まれています。Java FXのWebコンポーネントは、HTML5をサポートしているWebKitをベースにしています。多くの開発者がすでにWebKitベースのブラウザを利用しています。オフライン・ストレージ、Web SQLデータベース、ネイティブ・ドラッグ・アンド・ドロップ、ジオロケーションなども、今後開発が進むとともにJavaFXの対象となるでしょう。JavaFXはHTML5と常に同期を取りながら開発する見込みです。

***Java Magazine*：** JavaFX UIコントロール・ライブラリはJavaFX 2.0ツールセットにどのように組み込まれていますか。

Ramani： 私自身は長い間Swing開発に携わってきており、優れたプログラム手法であるSwingが本当に好きなのですが、Swingのコンポーネントはすべて形状を長方形にする傾向があります。しかし今はもっと柔軟でなけ

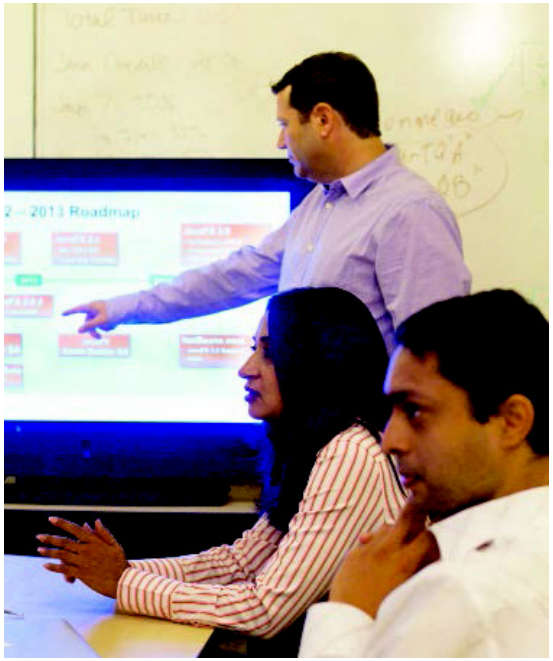
コミュニティの声



「JavaFXのオープンソース化は、このテクノロジーが正しい方向に向かうための大きな一歩です。企業はベンダー・ロックインを気にせずにJavaFXを利用でき、コミュニティは開発に参加できるため、革新的な新しい方法でプラットフォームが成長します。私は1.0のリリースの頃からこのような動きを提唱してきました。オラクルがコミュニティに耳を傾け、オープンソース化が可能になったことを嬉しく思います」

- RIAテクノロジーのテクニカル・エキスパート、GXSのチーフ・アジャイル・メソドロジスト、Stephen Chin氏

写真：HARTMANN STUDIOS



製品のロードマップについて議論する、Nandini RamaniとJavaFX 2.0チームのメンバー

飛躍的な成長

JavaFX 2.0は**飛躍を遂げた**テクノロジーです。開発者はWebコンテンツやメディア・コンテンツをJavaアプリケーションにシームレスに**統合**できます。

ればなりません。JavaFX 2.0 UIコントロールでは、円の形状作成機能、より幅の広いコンポーネント、CSSでのスキンが提供されます。このような強力で新しいコンポーネント群を利用すれば、どのような開発者でも高度なエンタープライズ・アプリケーション・インタフェースを構築できます。また、JavaFX UIコントロールは、JavaFXでは初めてOpenJDKプロジェクトとしてオープンソース化されるコンポーネントです。そのため、Swingコミュニティが長年実施してきたように、開発者は独自のカスタム・コンポーネントを構築できるようになります。

***Java Magazine*：** JavaFX 2.0のコードを既存のSwingアプリケーションと容易に統合できますか。

Ramani： はい、SwingとJavaFXの連携はスムーズです。既存のSwingアプリケーション内で、JavaFXをアドオンすることも組み込むこともできます。JavaFXにはSwingの全機能が引き継がれていますが、JavaFXでレンダリングされる部分については、すべて新しい機能を活用しています。たとえば、WebブラウザをSwingアプリケーションに組み込む場合は、JavaFX WebViewを組み込みます。他にも、Swingのコードの一部をJavaFXに合わせて書き直すという方法もあります。この方法は、既存のSwingアプリケーションの移行パスとして優れています。

***Java Magazine*：** JavaFX 2.0ではどのオペレーティング・システムがサポートされますか。

Ramani： 最初のリリースでは、32ビット版および64ビット版のMicrosoft Windows XP、Windows Vista、Windows 7がサポートされ、次にMac OS X向けのJavaFX 2.0 SDK開発者プレビュー、さらに後にLinuxバージョンがリリースされます。

***Java Magazine*：** JavaFXは今後どのように進化していくと思われますか。

Ramani： クライアント側では、Javaプラットフォームの進化と歩調を合わせようとしています。**Invokedynamic**や**NIO.2**などの新機能を、そのリリースに合わせて統合する予定です。Java SE 8で計画されているJavaプラットフォームの2大強化機能としてモジュール化とラムダ式がありますが、JavaFX APIではこれらの両方の機能を特に考慮して設計されています。たとえば、イベント処理APIはすべてSAM（シングル抽象メソッド）インタフェースにのっとって設計されています。これはまさに、Java SE 8のラムダ案で必要とされている種類のインタフェースです。

しかし、面白いものを未来に求める必要はありません。JavaFX 2.0はすでに飛躍を遂げたテクノロジーなのです。開発者は、コーディングやレイアウトに関する強力なオプションの数々を利用して、Webコンテンツやメディア・コンテンツをJavaアプリケーションにシームレスに統合できます。そのようなJavaアプリケーションをブラウザ内部で、あるいはスタンドアロン・アプリケーションとして実行できるのです。広範なエンタープライズ・アプリケーションで、多方面の機能を備えたこの新しいテクノロジーの恩恵を受けられることと思います。

***Java Magazine*：** 今後のJavaFXで、オープンソースはどのような役割を果たしますか。

Ramani： 私たちはJavaOneで、JavaFXプラットフォームをOpenJDKの新プロジェクトとしてオープンソース化するという提案計画があると発表しました。まずは、JavaFXのUIコントロールや関連ライブラリを提供し、その後複数の段階に分けて他のJavaFXコンポーネントをリリースする予定です。この記事が発行されるまでには、OpenJDKコミュニティがこの案を承認しているのではないでしょうか。JavaFXプロジェクトのステータスについてはOpenJDKサイトでご確認ください。`</article>`

Michael Meloan: IBMメインフレームとDEC PDP-11のアセンブリ言語のコーディングに関するプロとしてキャリアをスタートさせ、PL/I、APL、C、Javaのコーディング分野でも高い専門性を備えている。また、*WIRED*、*BUZZ*、*Chic*、*LA Weekly*には小説が掲載され、National Public Radioへの出演経験もある。さらに、**Huffington Post**のブロガーでもある。

参考情報

- JavaFXホームページ
- JavaFX 2.0ダウンロード
- FX Experience
- JavaFXの概要

SIMON RITTER

BIO

JavaFXとSwingの統合

JavaFX 2.0でSwingインタフェースをJavaFXに移行する

2011年10月、JavaFX 2.0がJavaOneでデビューを果たしました（3ページ「JavaFX 2.0の誕生」を参照）。ビジネス/エンタープライズ・アプリケーション向けの高度なJava UIプラットフォーム「JavaFX 2.0」は、最高レベルのリッチ・クライアント・プラットフォームとして、Javaが進化していく中で次のステップを担うものとなります。

JavaFX 2.0で必要となるのは新しいAPIセットの使用方法の習得と、バインディング、タイムラインベースのアニメーション、シーン・グラフなどの理解だけです。経験豊かなJavaプログラマーであればすぐに習得できます。さらに、JavaFX 2.0より、SwingとJavaFXの統合アプローチが逆転します。

本記事は、インタフェースをSwingからJavaFXに移行するために既存のSwingベースのアプリケーションを修正する方法を紹介するシリーズの第1回目です。このシリーズでは、アプリケーションの外観とユーザビリティを向上させるために、アプリケーションのパーツを段階的にJavaFXに置き換える方法について説明します。

注：この記事で紹介するサンプルのソース・コードはこちらからダウンロードできます。

次のステップ

JavaFX 2.0は、最高レベルのリッチ・クライアント・プラットフォームとして、Javaを**進化させていく中で次のステップ**を担うものとなります。

JavaFXに関する重要な知識

Java APIのセットであるJavaFX 2.0をIDE内部のプロジェクトに使用するということは、単純にクラスパスにJavaFXランタイム・ファイル（jfxrt.jar）を含めるということです。

これによって、Swing/JavaFX統合アプリケーションで使用したいコンポーネントの開発にすぐに取り組めます。

注：このシリーズでは、JavaFXの構造については深く取り上げません。JavaFXリソースの「参考情報」をご覧ください。

Swingサンプル・アプリケーション

JavaFXの能力を実証するために、ここでは比較的シンプルなSwingアプリケーションを取り上げます。このアプリケーションのコンポーネント内は適度に複雑ですが、理解しやすく修正しやすいものになっています。

まず初めのアプリケーション「StocksMonitor」は Hirondelle Systemsが作成したものです。株式市場取引を監視し、テーブルにデータを表示させるもので、ユーザーは株式のポートフォリオの作成、編集、削除を実行できます。StocksMonitorはjavapractices.comでダウンロードでき、クリエイティブ・コモンズ・ライセンスの下で配布されます。

統合の基本

まずは、JavaFXシーンをSwingコンポーネントとして扱うために必要となる項目について見ていきましょう。Swingでは、レイアウト・マネージャでコンテナ/コンポーネント階層を使用して、画面上での各コンポーネントの表示方法を整理しています。一方、JavaFXではシーン・グラフが利用されますが、Swingコンポーネントのような外観のJavaFXシーンを作成できるため、直接的な結果に違いはありません。

幸い、難しい作業の大部分をJavaFXプラットフォームが担ってくれます。JavaFX APIに含まれる**JFXPanel**クラスは、やや誤解を招く名称ですが、Abstract Windows Toolkit（AWT）の**Panel**クラスではなく、Swingの**JComponent**クラスを拡張しています。そこで、JFXPanelクラスはSwingアプリケーションで容易に利用できますが、純粋なAWTアプリケーションでは利用できません。

JavaFX 2.0のリリースについて語るJavaクライアント部門の開発者エクスペリエンス・アーキテクト、Jasper Potts

リスト1にこのクラスの単純な使用例を示します。不明確な点もいくつか見られるため、詳細を順に見ていきましょう。

まず8行目でJFXPanelオブジェクトをインスタンス化していますが、この時点ではオブジェクトに格納する情報を何も設定せず、後でその情報を設定します。このコンポーネントは、他のSwingコンポーネントと同様にSwingのコンテナに追加できます。

JavaFXシーンをカプセル化する方法を理解するために、SwingとJavaFXにおけるスレッドについて詳しく学ぶ必要があります。Swingではシングル・イベント・ディスパッチ・スレッド（EDT）が、JavaFXではアプリケーション・スレッドが使用されます。両方とも結果的には同じ操作、つまりUIに影響する非同期イベントの処理を実行します（ボタンを押す、表示された値を更新する、ウィンドウのサイズを変更するなど）。

SwingコンポーネントであるJFXPanelオブジェクトにアクセスできるのは、Swing EDTだけです。ただし唯一の例外にsetScene()メソッドがあります。これは、JavaFXアプリケーション・スレッドからアクセスする必要があります。

JFXPanelオブジェクトにシーンを設定するには、Runnableインタフェースを実装する別のオブジェクトを作成する必要があります。このオブジェクトをイベント・キューに送信すると、しばらく（不特定の）時間が経った後にオブジェクトを実行できます。Runnableオブジェクトのインスタンス化には、Platformユーティリティ・クラスに用意された静的ユーティリティ・メソッドの「runLater()」や非同期インナー・クラスを利用します。インナー・クラスの制約があるため、JFXPanelへの参照はfinalにする必要があります。

initFXComponent()メソッドで、実際のJavaFXの処理が実行されます。JavaFXではシーン・グラフが利用されるため、親ノードの参照を保持するSceneオブジェクトを作成する必要があります。このオブジェクトをインスタンス化する際に、ピクセル単位でシーンのサイズも設定します。そうすると、Swingレイアウト・マネージャがこのオブジェクトを利用して、JavaFXコンポーネントを配置します。さらに、表示対象オブジェクトのグラフを構成するシーンのルート・ノードを設定します。

このSwingアプリケーションに複数のJavaFXシーンを組み込む場合は、単純に複数のJFXPanelオブジェクトをインスタンス化します。複数のJavaFXシーンの利用を最適化するには、Platform.runLater()を1回呼び出して、インナー・クラスのrun()メソッドにすべてのシーンの初期化を実行させます。

SwingとJavaFXのイベント

SwingとJavaFXの統合について考慮すべき基本的事項がもう1点あります。それは、SwingコンポーネントがJavaFXシーンの更新をトリガーする場合（またはその逆の場合）のコンポーネント間の操作方法です。JavaFXシーンからSwingコンポーネントを操作する場合、そのコードをRunnableオブジェクトでラップし、SwingUtilities.invokeLater()メソッドを呼び出してEDT上にそのオブジェクトを配置する必要があります（**リスト2**）。

その逆のケース、つまりSwingコンポーネントからJavaFXシーンを操作する必要がある場合は、やはりそのコードをRunnableオブジェクトでラップした後、今度はPlatform.runLater()メソッドを呼び出してJavaFXアプリケーション上にそのオブジェクトを配置する必要があります（**リスト3**）。

まとめ

これまでの内容で、通常のSwingコンポーネントにさらに数行加えるだけで、JavaFXシーンをSwingコンテナに追加できることを確認しました。JavaFXシーンとSwingコンポーネント間の操作も簡単です。必要なコードをRunnableオブジェクトにラップして、適切な実行キューに配置すればよいということだけを覚えておいてください。

次号の記事では、この知識に基づいて、JavaFXで作成した楽しく新しいコンポーネントをこのSwingサンプル・アプリケーションに追加していきます。

リスト1 リスト2 ／ リスト3

```
1 public void init() {
2   setSize(300, 200);
3   setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
4   setLayout(new BorderLayout());
5
6   /* Put the JavaFX scene in the middle */
7   fxComponent = new SimpleFXComponent(this);
8   final JFXPanel fxPanel = new JFXPanel();
9   add("Center", fxPanel);
10
11  /* Construct the JavaFX scene on its own application thread */
12  Platform.runLater(new Runnable() {
13   @Override
14   public void run() {
15     Scene scene = new Scene(fxComponent, 200, 100);
16     scene.setFill(Color.BLUEVIOLET);
17     fxPanel.setScene(scene);
18   }
19  });
20
21  /* Put the Swing button at the bottom */
22  changeButton = new JButton("Change JavaFX Button");
23
24  ...
25
26  add("South", changeButton);
27  setVisible(true);
28 }
```

リスト全体をテキストで表示する

参考情報

- ドキュメントとチュートリアル
- APIドキュメント（Javadoc）
- JavaFX Showcase