



# Javaの未来

2007年4月23日  
稚内北星学園大学  
東京サテライト校  
浅海智晴



# 内容

- Javaの12年、プログラマの12年
- これからの10年
- Javaの未来を考える

A decorative graphic on the left side of the slide features three balloons in shades of green, blue, and purple, each with a yellow streamer and small yellow triangular accents. The text is positioned to the right of these elements.

Javaの12年、プログラマの12年



# Javaの歴史・黎明期

- 1994年 Netscape
- 1995年 Netscape 2.0, JavaScript
- 1995年 Windows 95
- 1995年  $\beta$  版, HotJava
- 1996年1月 JDK1.0
- 1996年5月 JavaBeans, Java 3D
- 1996年8月 Servlet (Jeeves alpha 1.0)
- 1996年9月 Javaカンファレンス設立
- 1997年1月 UML1.0
- 1997年2月 JDK1.1
- 1997年5月 EJB, JFC, Personal Java, Embedded Java, SmartCard Java
- 1998年1月 RMI-IIOP (OMG CORBA IIOP)
- 1998年1月 XML1.0
- 1998年3月 Jini
- 1998年5月 Javaコンソーシアム設立



# Javaの歴史・成長期

- 1998年12月 Java2 SDK 1.2, JCP
- 1999年2月 『The Unified Software Development Process』
- 1999年6月 J2SE, J2EE, J2ME
- 1999年10月 『Extreme Programming Explained: Embrace Change』
- 1999年11月 Tomcat (Apache Jakartaプロジェクト)
- 2000年3月 JAXP 1.0
- 2000年5月 J2SE 1.3
- 2000年6月 Javaカンファレンス日本インターネット協会(現財団法人インターネット協会)と合併
- 2000年7月 JavaコンソーシアムXML部会を母体の一つにしてXMLコンソーシアム設立
- 2000年9月 MIDP 1.0 (Mobile Information Device Profile)
- 2000年9月 UDDI



# Javaの歴史・成長期(続き)

- 2001年1月 iアプリ
- 2001年2月 JXTA
- 2001年3月 WSDL, MDA
- 2001年4月 W3C XML Schema
- 2001年5月 Javaコンソーシアム解散
- 2001年6月 Struts 1.0 (Apache Jakartaプロジェクト)
- 2001年9月 J2EE 1.3
- 2001年11月 AspectJ 1.0
- 2001年12月 OASIS RELAX NG 1.0
- 2001年12月 ECMA-334 *C# Language Specification*
- 2002年3月 J2SE 1.4
- 2002年8月 BPEL4WS
- 2002年9月 J2ME Personal Profile
- 2003年3月 JXTA 2.0
- 2003年3月 UML 2.0



# Javaの歴史・転換期

- 2003年6月 JBoss 4.0, Hibernate 2.0
- 2003年8月 WS-I Basic Profile 1.0
- 2003年11月 J2EE 1.4
- 2004年3月 Spring Framework 1.0, Seasar2
- 2004年6月 Eclipse 3.0
- 2004年9月 J2SE 5.0
- 2005年2月 Google Maps
- 2005年3月 JIUL(Java Internal Use License), JDL(Java Distribution License)
- 2005年3月 JIS X 3015 プログラミング言語C#
- 2005年12月 AspectJ 1.5.0 (eclipse.org)
- 2006年3月 NetBeans 5.0
- 2006年5月 JEE 5
- 2006年11月 JavaがGPLでオープンソース化
- 2006年12月 JSE 6



# Javaの歴史・これから

- 2007年4月日本Javaユーザグループ(JJUG)発足
- 2008年? JEE 6
- 2009年? JSE 7

# 視点

- JavaはWebとともにある
  - Applet, Servlet, XML, Web Service
  - SOA, SaaS
- 技術革新はJavaから
  - Applet, Servlet, XML, Web Application Server (Java EE)
  - apache.org, eclipse.org, Junit, Maven
  - AspectJ (AOP), DIコンテナ (JBoss, Spring, Seasar)
- 応用の広がり
  - 企業システム
  - 携帯機器/組込みシステム
  - プログラマ・コミュニティ
- 競争・連携
  - Windows(.NET)
  - LAMP (Linux, Apache HTTP Server, MySQL, Perl, PHP, Python)
  - スクリプト言語、動的型付言語 (JavaScript, Ruby, Python)



# 別の視点では

- JavaはWebとともにある
  - Ajax
- 技術革新はJavaから
  - ASP(Active Server Pages)
  - EoD(Ease of Development)
  - Ruby on Rails
- 応用の広がり
  - Windows
  - LAMP



# Javaによって得られたもの

- オブジェクト指向
- 業界標準プラットフォーム
- 業界標準API
- 新技術の市場
- コミュニティ



# オブジェクト指向

- 正統派オブジェクト指向プログラミング言語
  - 癖のない正統的な言語仕様
  - 堅牢なシステム構築には静的型付け言語が向いている
  - 実務向けのオブジェクト指向言語として最も広く普及している
  - C++の反省
    - GCがない, 言語仕様が複雑, 共通ライブラリが貧弱, 可搬性に問題
- オブジェクト指向技術への広がり
  - UML
  - オブジェクト指向分析/設計
    - Unified Process
    - ソフトウェア・パターン
  - オブジェクト指向開発プロセス
    - Unified Process
    - アジャイルプロセス



# 業界標準プラットフォーム

- プラットフォーム独立
  - PC
  - ビジネス・サーバ
  - 携帯機器
  - 電子デバイス
- 実行環境
  - Java VM
  - Java Platform SE/ME/EE



# 業界標準API

- 各種技術分野の標準API
  - JSRとして多くの技術分野で標準APIが定義され、実装が提供されている
  - JCPによるオープンな活動
  - 単なる標準仕様だけでなく(フリーで利用できる)実装が提供されていることが重要
- たとえば...
  - JSR 1: Real-time Specification for Java
  - JSR 94: Java Rule Engine API
  - JSR 223: Scripting for the Java Platform
    - JSE6に取り込み
  - JSR 310: Date and Time API
    - JSE 7に取り込み?
  - JSR 313: Java Platform, Enterprise Edition 6(Java EE 6) Specification
  - JSR 901: Java Language Specification

# 新技術の市場

- Java VM
- javadoc
- Applet
- Servlet
- JITコンパイラ
- Tomcat
- XML
- Web Application Server (Java EE)
- iアプリ
- apache.org
- eclipse.org
- Ant
- Junit
- CruiseControl
- Maven
- Hibernate
- AspectJ (AOP)
- DIコンテナ (JBoss, Spring, Seasar)



# コミュニティ

- オープンソース・コミュニティ
- ビジネス・コミュニティ
- 勉強会コミュニティ
- 地域コミュニティ
- プログラマとビジネスをつなぐコミュニティ
  - ここは弱いかもしれない

これからの10年





# 視点

- Web 2.0
- CBD (Component-Based Development)
- モデル駆動開発
- UI (User Interface)
- IDE
- プログラミング言語



# Web 2.0

- 技術的には

- Webがプラットフォームになる

- HTTP、HTML(XML)のプロトコルスタックでサービスを結びつける
- 企業システムとの本格的な連携

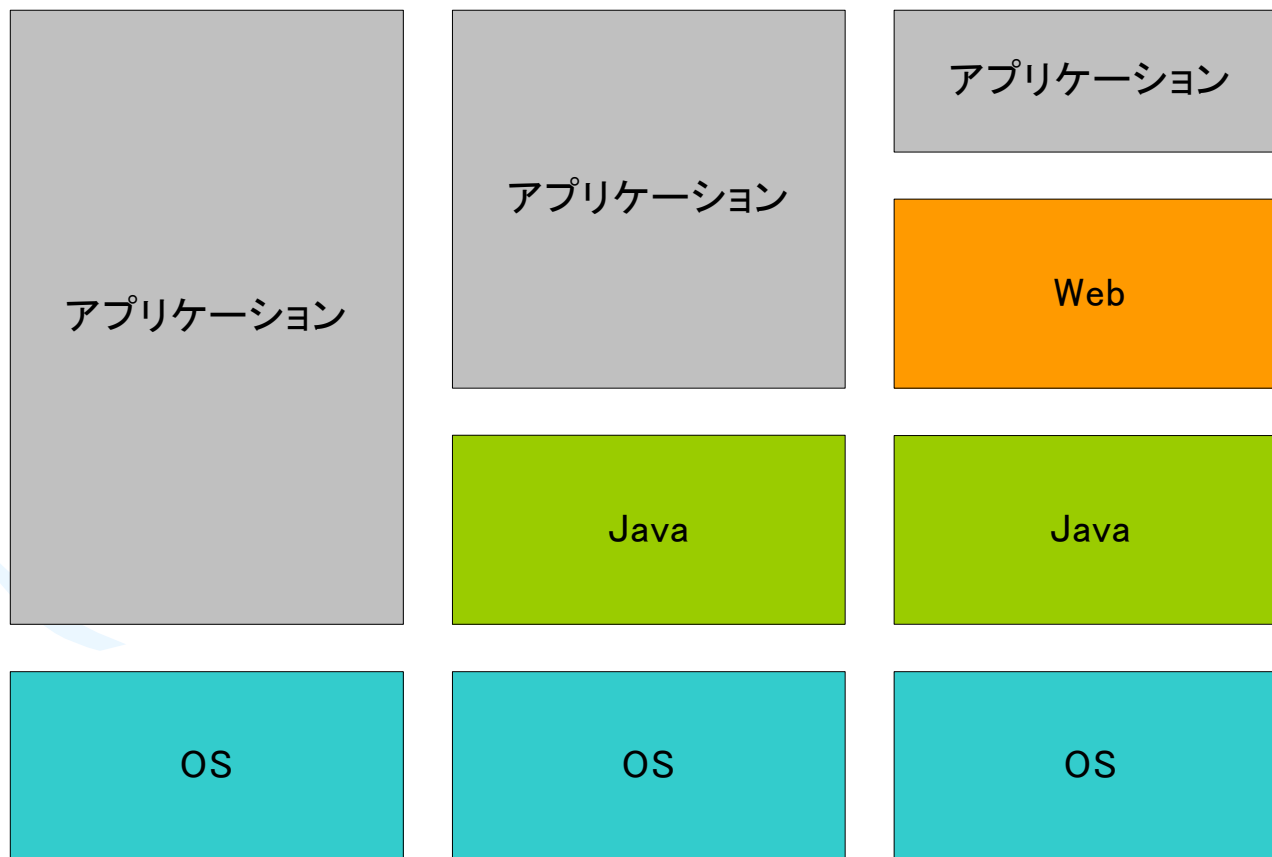
- 社会的には

- 新しいビジネス・プラットフォーム

- CGM
- その他もろもろ?

- 組織 > 個人 ⇒ 個人(の集まり) > 組織

# Webがプラットフォームになる




# CBD (Component-Based Development)

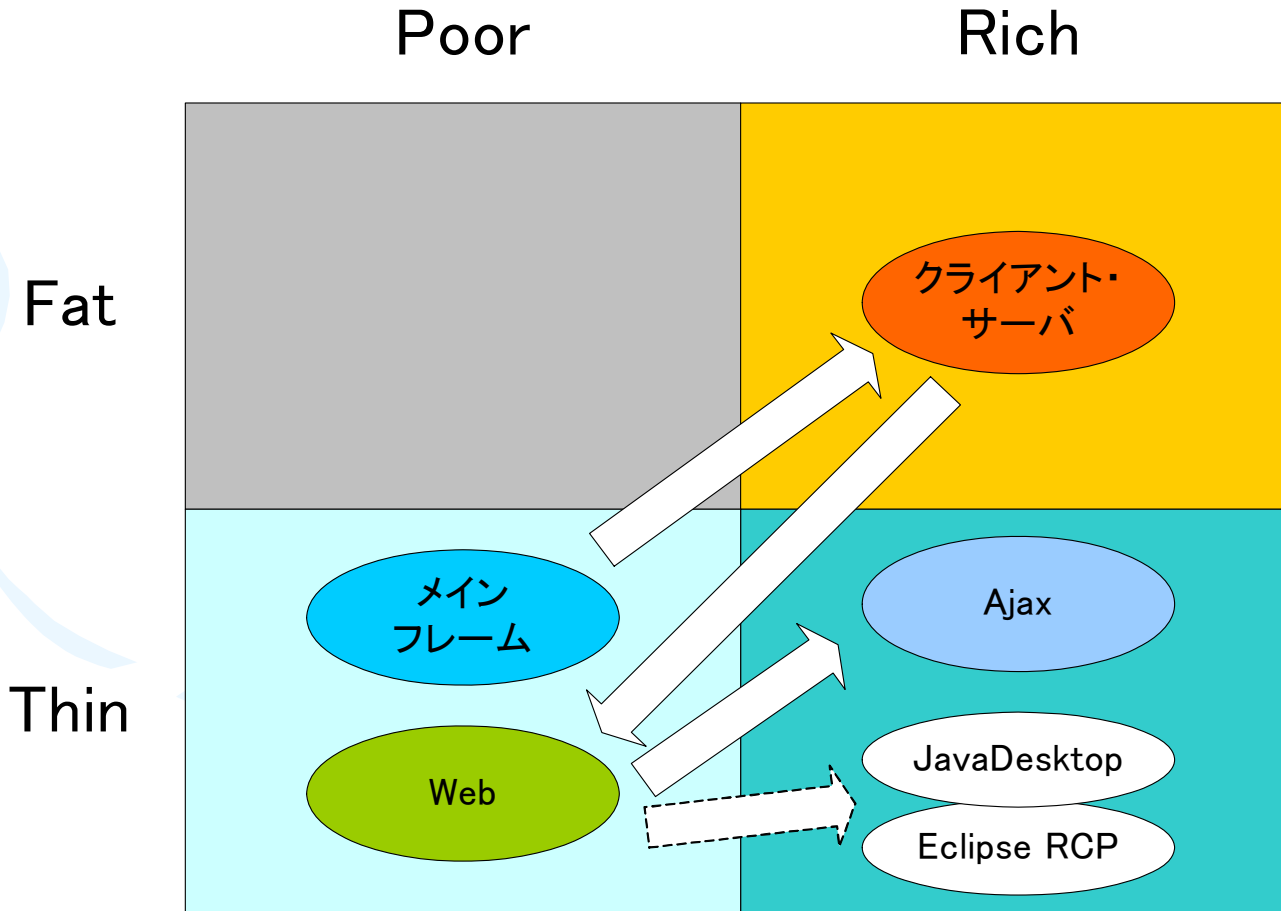
- オブジェクトは枯れた技術
  - もちろん、常に普及の問題はある
- 技術の焦点はコンポーネントへ移行している
  - 開発の単位
  - プロジェクト管理の単位
  - 流通の単位
- コンポーネント技術から派生する技術
  - JEE
  - SOA(Service Oriented Architecture)
  - SCA(Service Component Architecture)
  - SaaS(Software as a Service)
  - プロダクトライン
- ソフトウェア開発をCBD(Component-Based Development)の枠組みで実行できるスキルがビジネス上のアドバンテージになる



# モデル駆動開発

- Javaよりも上位の抽象度でプログラミングが行われる
    - モデル作成がプログラミング
  - モデルからプログラム(の一部)を自動生成
    - Java言語はアセンブラ的な位置付け
    - Javaプラットフォーム(Java VM, ソフトウェア資産)が重要
  - 今のところUMLが有力だが、技術革新があるかも
    - DSL (Domain Specific Language)
- 

# UI (User Interface)



# IDE

- Eclipseの登場で新しいフェーズに入った
  - 旧来のIDEは、テキストエディタ+コマンド・シェルにUIをつけたレベル
    - これならEmacsの方が便利
  - プラグインによる拡張
  - プログラミング言語の構文木をメタ・レベルで取り扱う
    - 入力補完、リファクタリング
  - ソフトウェアライフサイクル
    - 構成管理、配備
  - IDEビジネスが変わった
    - 本格的な普及が始まる
- NetBeansが追走、老舗のVisualStudioとも競合



# プログラミング言語

- スクリプト言語

- 自転車

- 動的型付け＋メタ・プログラミング

- オートバイ

- 静的型付け

- 乗用車

- DSL (Domain Specific Language)

- 産業用自動車 (トラクター、トレーラー、消防車)

# プログラミングからメタ・プログラミングへ

- プログラミングは枯れた技術
  - もちろん、常に普及の問題はある
- 技術の焦点はメタ・プログラミングへ移行している
  - ここでいうメタ・プログラミングとは、メタ・モデルを操作するプログラミング
- メタ・プログラミングの目的は：
  - モデルの再利用性の向上
  - 最終的にはプログラム(断片)の自動生成
- メタ・プログラミングの技術と適材適所で使いこなせるスキルがビジネス上のアドバンテージになる
  - マイDSL

# Javaの未来を考える





# Javaほどのコースを選ぶのか

- FORTRUN
- COBOL
- Lisp
- PASCAL
- Smalltalk
- C
- C++



# いかにもありそうなシナリオ

- 企業システム

- COBOLの後継、密室化

- 携帯機器

- 一部機器の主力言語(iアプリ、MIDLET)

- 組み込みシステム

- C、C++を補完

- 教育

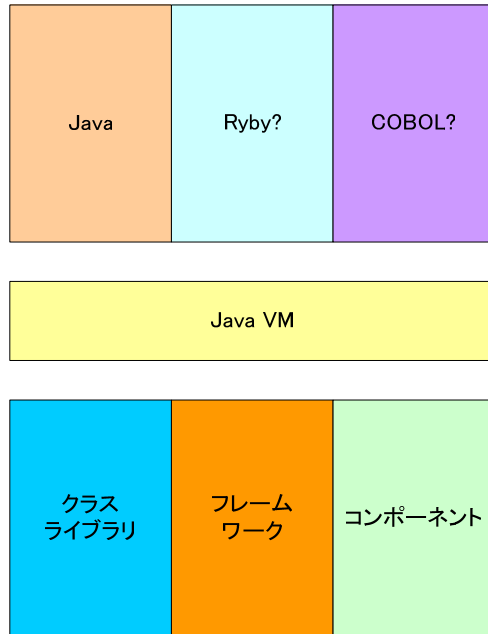
- Rubyに置き換え(Lisp, Pascalの後継として)

A decorative graphic on the left side of the slide features a large green balloon at the top, a blue balloon in the middle, and a purple balloon at the bottom. Yellow streamers and triangular flags are scattered around the balloons. The title text is positioned to the right of the green balloon.

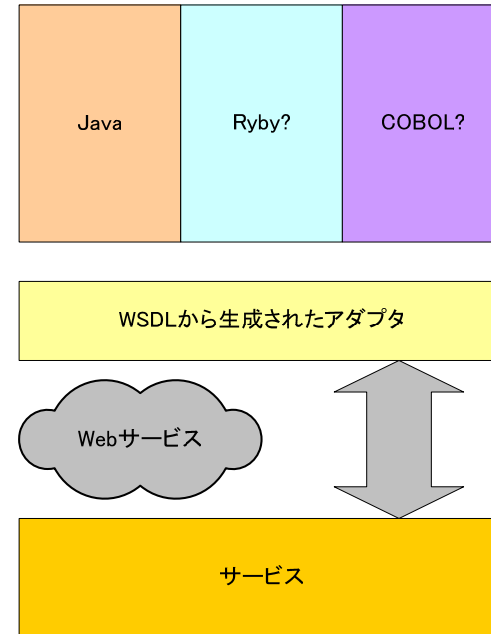
# Javaによって得られたもの

- オブジェクト指向
- 業界標準プラットフォーム
- 業界標準API
- 新技術の市場
- コミュニティ

# 2つのアーキテクチャ



(1) Java VMを中心にした  
アーキテクチャ



(2) WSDLを中心にした  
アーキテクチャ

- プログラミング言語が提供する生産性と信頼性
- プログラムの可搬性
- ソフトウェア資産の再利用性

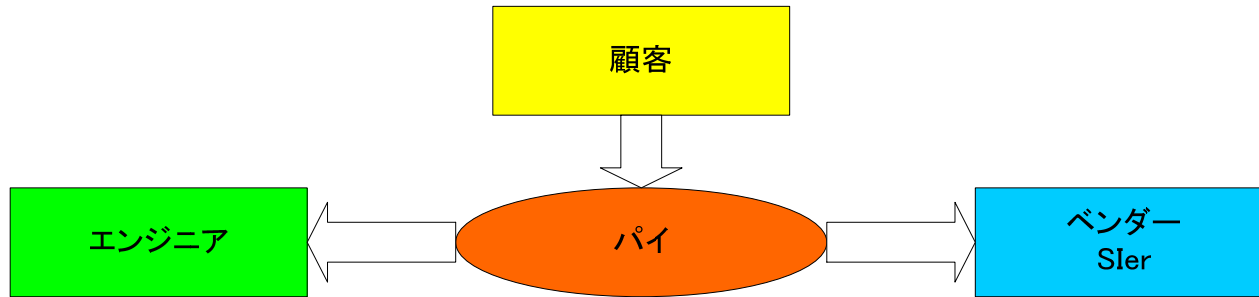
# Javaによって得られるはずのもの

- Javaをハブにして技術がつながる
  - 企業システムから携帯機器、電子デバイスまでフルセットの標準プラットフォームを提供
  - 各種技術分野のAPI+ソフトウェア資産が用意されている
- JavaをハブにしてWebがつながる
  - SOA系、Web系とも本格的なシステム構築のプラットフォームとしてJavaが最右翼
  - サーバサイドはJava、クライアントサイドは各種スクリプト言語という組み合わせ
- Javaをハブにしてエンジニアがつながる
  - オープンソースコミュニティ
  - 企業システムのサーバサイドはJava

# 懸念事項

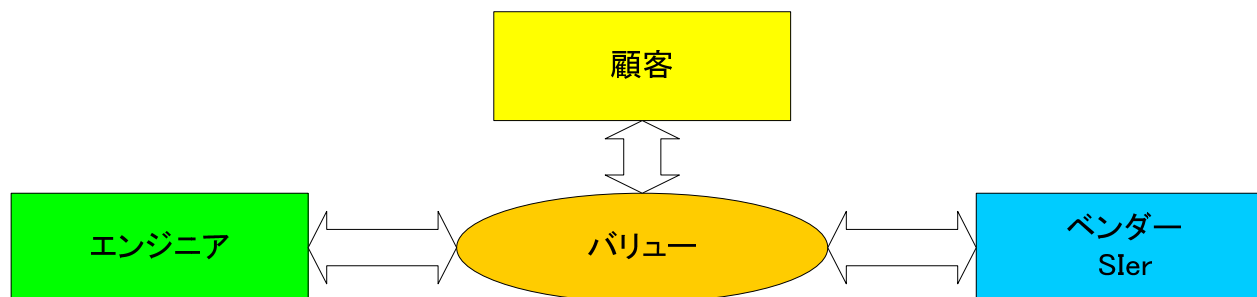
- Java技術誌の休刊
  - JAVA PRESS誌
  - Java World誌
- JavaのCOBOL化
  - Javaが次世代COBOLになることは決して悪いことではないが、これだけになると技術の活力が失われる
  - 案外使われているのにパブリックには注目されない状態
    - 若手技術者への誤ったメッセージ
    - 地方のIT業界への誤ったメッセージ
    - 顧客企業への誤ったメッセージ
    - 教育界への誤ったメッセージ
- 限定合理性による”合成の誤謬”

# エンジニアの QoL (Quality of Life)



- ソフトウェア開発ビジネスをゼロサムゲームにしてしまっはいけない
- 3K - 「きつい」「帰れない」「給料が安い」
- 7K - 「休暇が取れない」「規則が厳しい」「化粧がのらない」「結婚できない」

# バリュー創出のコラボレーション



- エンジニアと企業のコラボレーションの形
- IT業界の顧客に対してバリューを提供するための共同作業
- エンジニアと企業は持ちつ持たれつ



# JJUGに期待すること

- コミュニティへの多重所属
  - 生産の立場
  - 流通の立場
  - 消費の立場
- 色々な立場のエンジニアが、臨機応変にコラボレーションできる場の提供
- Javaエンジニアが幸せな技術者生活を行うためには
  - 技術上の交流
  - コミュニティの中での交流
  - ビジネス上の交流