

# Java アーキテクチャー

## ～ 誕生からの10年と今後 ～

サン・マイクロシステムズ株式会社  
山口 浩

## アジェンダ

- Java の歴史
- パフォーマンス向上の歴史
- 開発者にとっての Java

# 「役に立つものは美しい」

花森安治

# 「美しさ」の要素は？

シンプルさ

リズム

バランス

## これらの共通点は？

emacs

NeWS

Java

## Java の歴史

## December, 1990



Java, whose original name was Oak, was developed as a part of the Green project at Sun. It was started in December '90 by Patrick Naughton, Mike Sheridan and James Gosling and was chartered to spend time (and money!) trying to figure out what would be the "next wave" of computing and how we might catch it. We quickly came to the conclusion that at least one of the waves was going to be the convergence of digitally controlled consumer devices and computers.

## Green Project

- 『ステルス』 プロジェクト
  - > 家電分野のアプリケーションに関する研究
- ビジョン
  - > ハンディコントローラから「スマート」な家電機器をプログラム/集中制御できる
- プロジェクトのメンバー
  - > Mike Sheradin: ビジネスプラン
  - > Patrick Naughton: グラフィックスシステム
  - > James Gosling: 開発用言語の検討
  - > Ed Frank: ハードウェア

## 開発用言語の検討

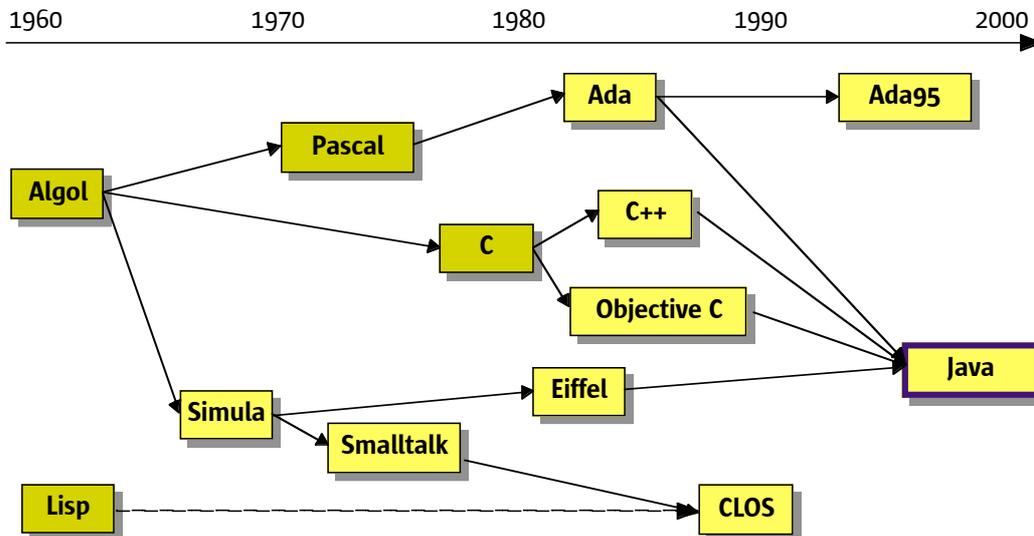
- 開発用言語としては C++ が有力候補
- ネットワーク対応の家電向けアプリケーション用開発言語に対する要件
  - > 多数のメーカーによる開発が可能（プラットフォームを選ばない）
  - > 高い信頼性（定期的なリブートは論外！）
  - > セキュリティの確保
  - > 習得が容易
  - > 開発が容易
- C++ では不可能と判断 => 新言語の開発

## Oak(Java) の言語の特徴

- C/C++ ベースのオブジェクト指向言語
  - > 言語習得の容易性
  - > 冗長な仕様を削除（プログラマのミス削減）
- コンパイラとインタプリタの併用
  - > バイトコード（中間コード）
  - > プラットフォームに非依存
  - > 実行時ではなくコンパイル時のエラー発見
  - > GC の導入（メモリーバグを撲滅）
  - > セキュアな実行環境の実現
- いくつかの新しい試み
  - > スレッド対応, Unicode, ...



## 他言語との関係



## 1992: Star 7

- UNIX/SPARC ベースの PDA
  - > 5" カラー LCD
  - > オーディオサポート
  - > タッチパネル入力
  - > ワイヤレスネットワーク
  - > フラッシュメモリ
  - > インタプリタ
  - > マルチスレッド
  - > ガーベッジコレクション
  - > ...



## Green Project の成果

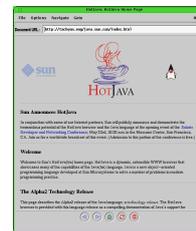
- 数名のエンジニア × 18 か月
  - > Green OS
  - > Oak 言語と実行環境の開発
  - > ユーザーインタフェース
  - > Star 7 ハードウェア
- Star 7 を片手に各地でマーケティング活動
  - > 反応はイマイチ
  - > 時期尚早だった

## 1993: FirstPerson Inc.

- 1992年 11 月, Green Project のメンバーは FirstPerson, Inc. を設立
- Time-Warner 社の STB 用 OS と対話型のVoD 技術に関する RFP をターゲットに活動を始める  
=> SGI 社に破れる
- 3DO からの Oak ベースの OS に関する技術提供依頼の商談 => 不成立
- ...

# 1994: WWW 向けアプリケーション

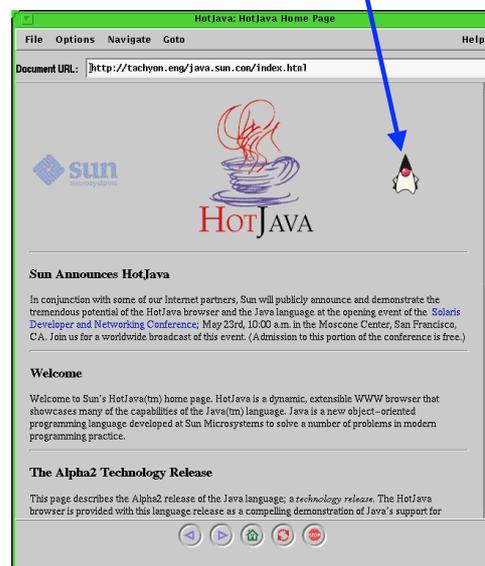
- WWW ブラウザを Oak で開発
  - > WebRunner
  - > 後の HotJava
- 生のデータではなく，その上で動くアプリケーションを提供できる
- WWW は情報家電に変わる適用分野
  - > プラットフォーム非依存
  - > セキュリティ
  - > 高い信頼性



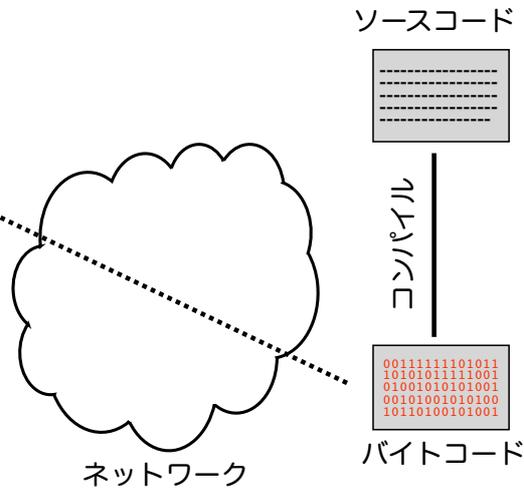
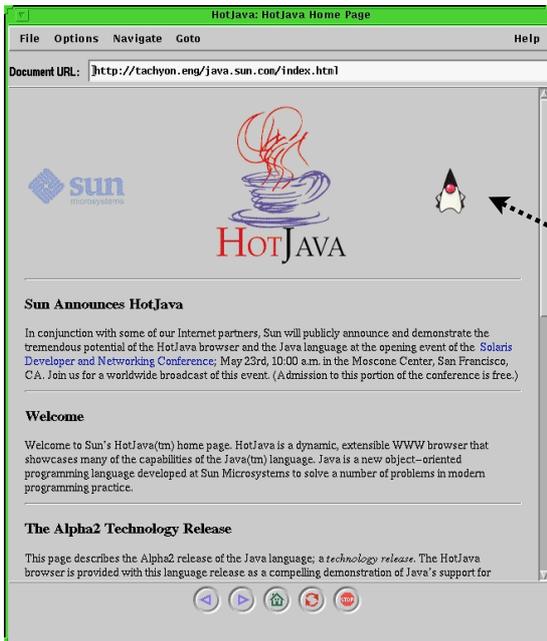
# May 23, 1995: Java 誕生



## Java Applet

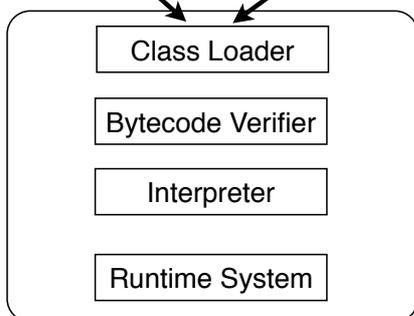
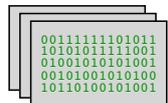


# Java Applet の実行

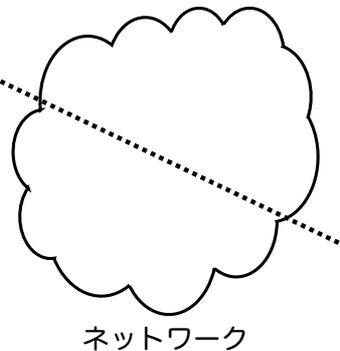


# Java Applet の実行

Java Class Library



Java VM



ネットワーク

ソースコード



コンパイル

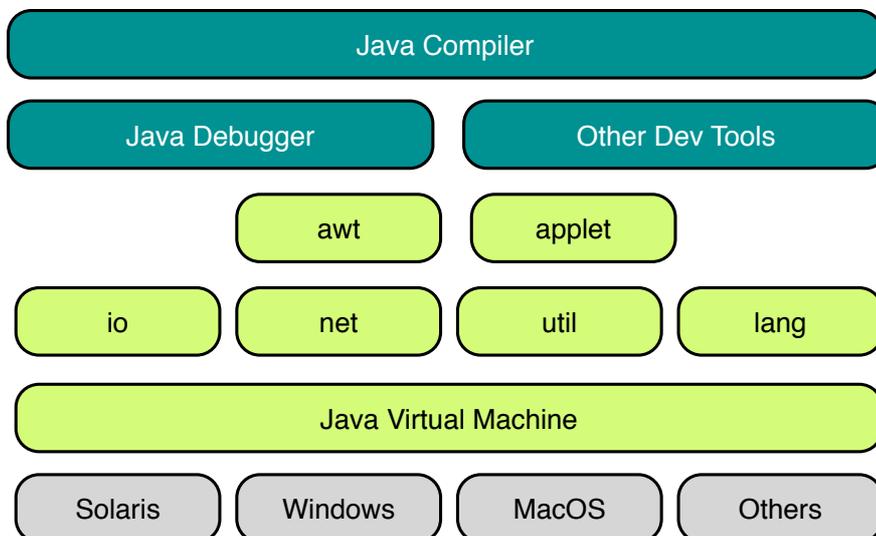


バイトコード

OS

Hardware

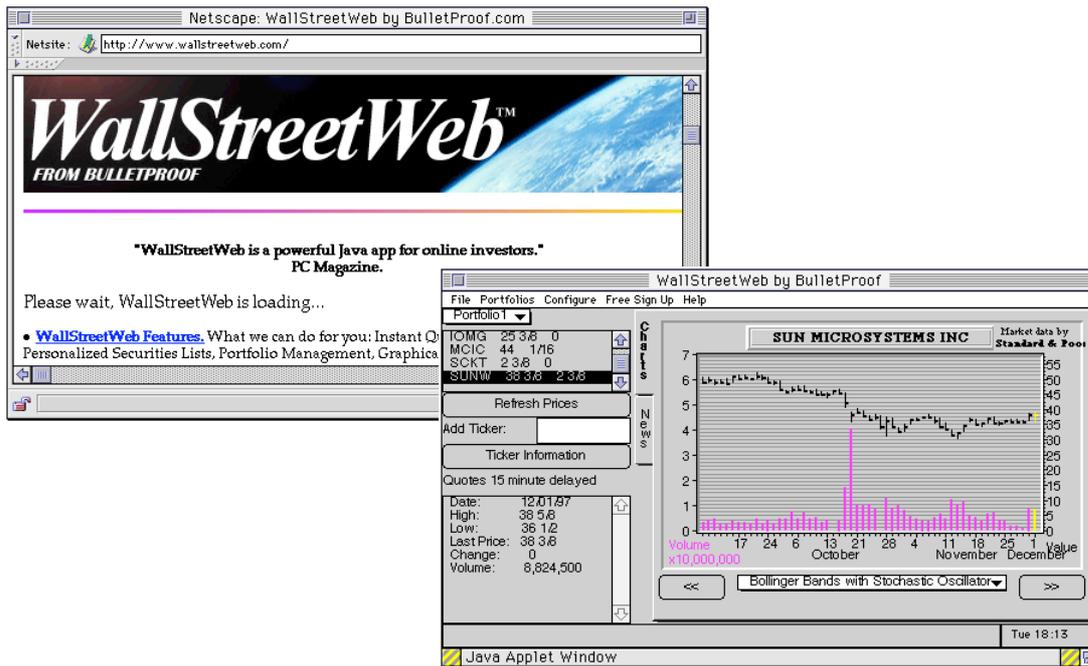
# Java Development Kit 1.0



# Java アプレット



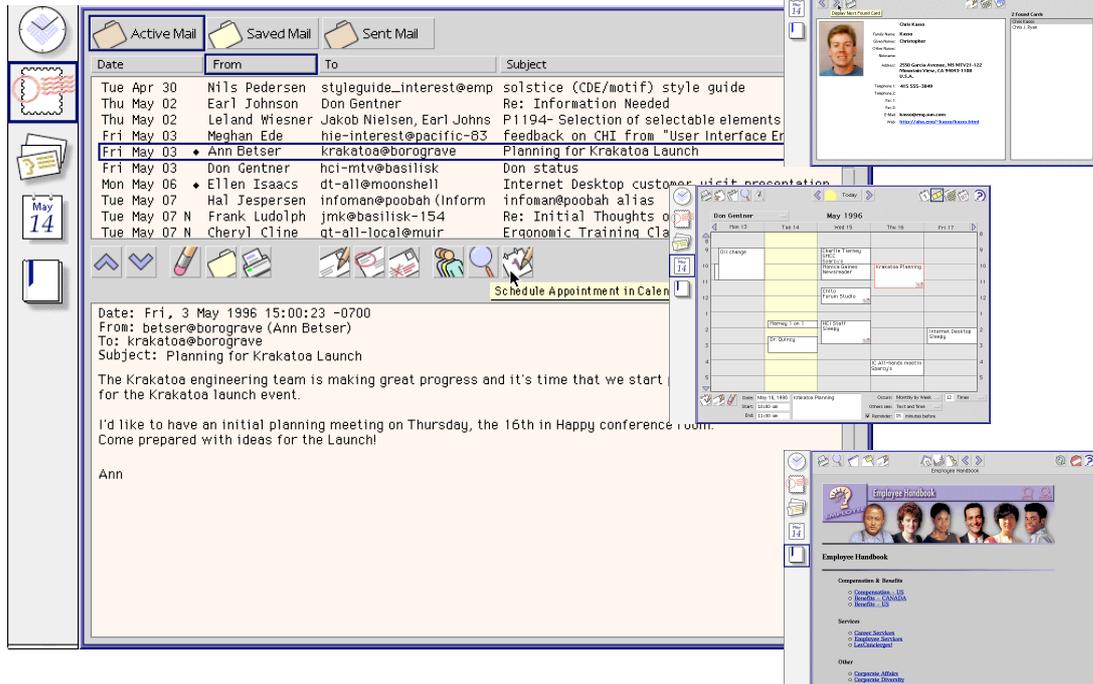
# Java アプレット



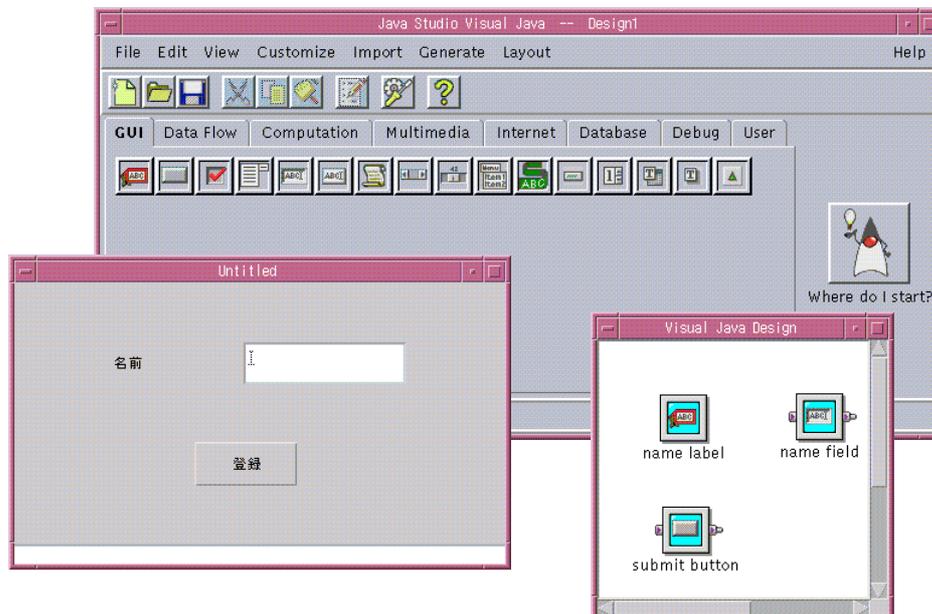
# 1996-7: JavaOS/JavaStation



# HotJava Views



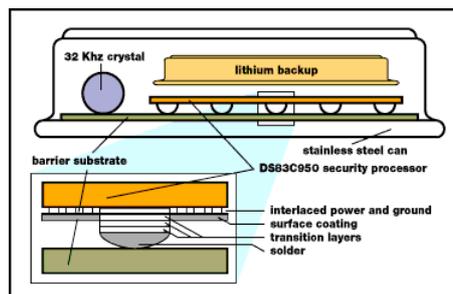
# 1997: JavaStudio



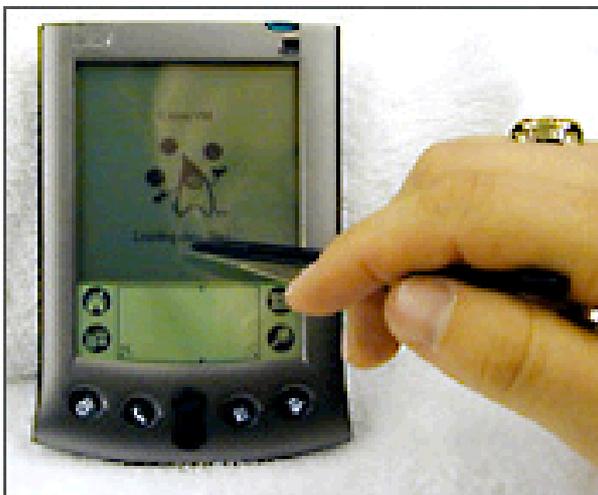
# 1997: JavaCard/JavaRing



ISO 7816 スマートカード規格準拠  
 128KB ROM  
 64KB EEPROM  
 4KB RAM  
 JavaCard VM (Virtual Machine)  
 Java 言語仕様のサブセットをサポート  
 複数アプリケーション対応



# 1999: K Virtual Machine



# 1999: Java2 プラットフォーム



Java Enabled Devices



Desktop WS/PCs



Workgroup Servers



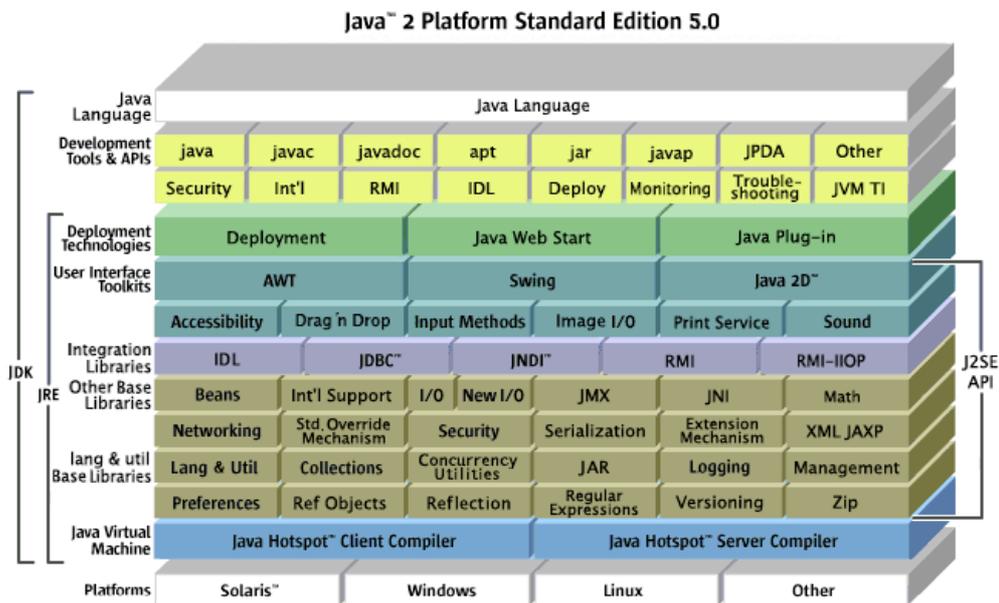
High-End Servers



## 2000 ~ 2005

- J2EE Application Servers
- Java + XML
- JAX-RPC/Web Services
- Java3D ~ Project Looking Glass
- Real-time Java
- Groovy
- ...

# 2004: Java SE 5.0



# 10 Years of Features

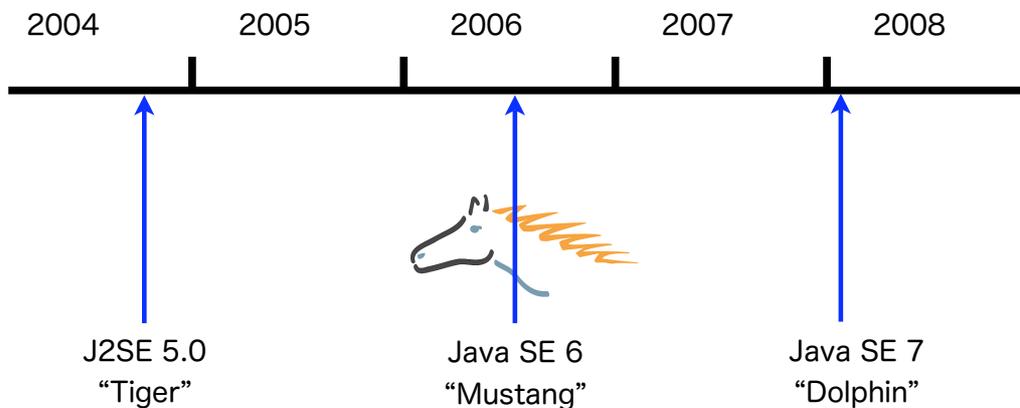
A small sampling of Java features

Feature	1.0	1.1	1.2	1.3	1.4	Java 5
GC	■	■	■	■	■	■
Threading	■	■	■	■	■	■
Inner Classes	□	■	■	■	■	■
Reflection	□	□	■	■	■	■
JIT	□	□	■	■	■	■
Swing	□	□	■	■	■	■
Webstart	□	□	■	■	■	■
Hotspot	□	□	■	■	■	■
assert	□	□	□	□	■	■
Regular Exp.	□	□	□	□	■	■
NIO	□	□	□	□	■	■
XML	□	□	□	□	■	■
Generics	□	□	□	□	□	■
Autoboxing	□	□	□	□	□	■
Annotations	□	□	□	□	□	■
Enumerations	□	□	□	□	□	■

# JDKリリース

バージョン	コード名	リリース日
J2SE 1.2	Playground	Dec 4, 1998
J2SE 1.2.1		March 30, 1999
J2SE 1.2.2	Cricket	July 8, 1999
J2SE 1.3	Kestrel	May 8, 2000
J2SE 1.3.1	Ladybird	May 17, 2001
J2SE 1.4.0	Merlin	Feb 13, 2002
J2SE 1.4.1	Hopper	Sept 16, 2002
J2SE 1.4.2	Mantis	June 26, 2003
J2SE 5.0 (1.5.0)	Tiger	Sept 29, 2004
J2SE 6.0 (1.6.0)	Mustang	Not yet released
J2SE 7.0 (1.7.0)	Dolphin	Not yet released

# Future Releases



18 ~ 24 か月のリリースサイクル

# Planned Features for Mustang...

**JSR-199 Compiler API**      **Longhorn Look & Feel**  
 Split Verifier      MBeans metadata      chmod  
 Parallelize Concurrent GC      JConsole upgrades  
**APT Pluggability API**      parallel old-space GC  
 JVM DTrace      XML digital signatures  
**Web Services Stack**  
 Docs in more Locales      **JDBC 4.0**      **JavaDoc Upgrade**  
 Windows system tray      **JAXB 2.0**      more gfx acceleration      http cookie manager  
 improved OOM diagnosability      **JVM & CLR Co-Existence**  
 more desktop integration      improved text rendering      Firefox support  
**Improved Native L&Fs**      **Scripting Language Support**  
**XAWT**      **More GC Ergonomics**      **JavaScript engine**

# Mustang Snapshot

mustang: Java SE 6 Snapshot Releases  
 https://mustang.dev.java.net/

java.net The Source for Java Technology Collaboration  
 Logged in: yamaguch | Logout

My pages    **Projects**    Communities    java.net

Projects > jdk > mustang

**Get Involved**  
 • java-net Project  
 • Request a Project  
 • Project Help Wanted Ads  
 • Publish your Project  
 • Submit Content

**Project tools**  
 • Project home  
 • Announcements  
 • Discussion forums  
 • Mailing lists  
 • Documents & files  
 • Project FAQ  
 • How to contribute  
 • JDK Community

**Search**  
 This project

**mustang**  
 Project home  
[Request project membership/role](#) | [Watch project](#)

License [Java Research License \(JRL\)](#)  
 Owner(s) [bcbeck](#), [brinkley](#), [mreinhold](#), [peterkessler](#)

**Description**  

**Mustang Snapshot Releases**  
 Sun is releasing weekly snapshots of the complete [source](#), [binaries](#) and [javadoc](#) for Java SE 6 ("Mustang"). These raw snapshot releases let you review and contribute to Mustang as it is being developed.  
 For an overview of Mustang, see the [Mustang JSR](#) and the [desktop](#) and [core](#) feature overviews. Mustang is targeted for final release in summer of 2006.

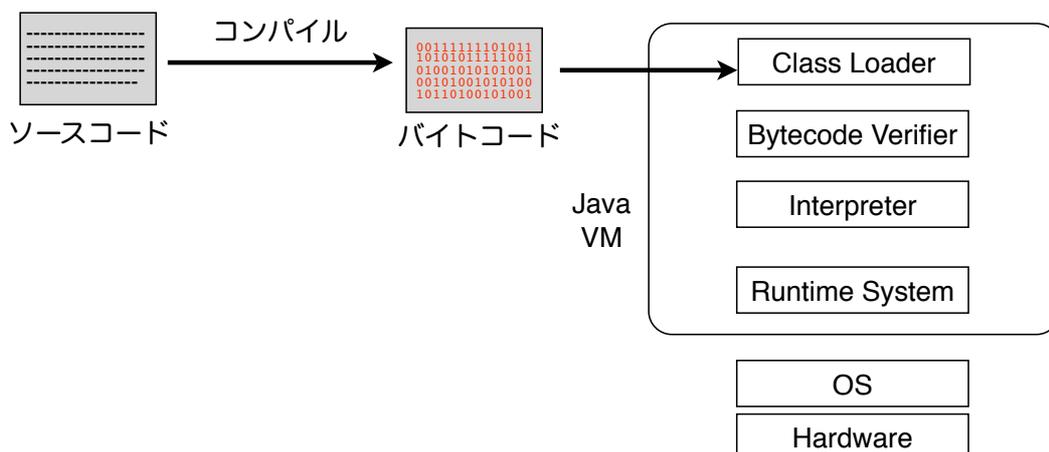
**Download Mustang**  
 • [Latest Mustang binary snapshots](#)  
 • [Latest Mustang source snapshots](#)

**We Want Contributions!**  
 Frustrated with a bug that never got fixed? Have a great idea for improving the JDK? See How to

## パフォーマンス向上の歴史

## 初期の Java は確かに遅かった

- Java は C 言語に比べ，数倍から 50 倍遅い
- GC (ガーベッジコレクション) 中は停止



# Java のプログラム

```
import java.math.BigDecimal;

public class Multiply {
    public static void main(String args[]) {
        BigDecimal x = new BigDecimal(args[0]);
        BigDecimal y = new BigDecimal(args[1]);
        BigDecimal z = x.multiply(y);
        System.out.println(x + " * " + y + " = " + z);
    }
}
```

# バイトコード (Java VM マシン語)

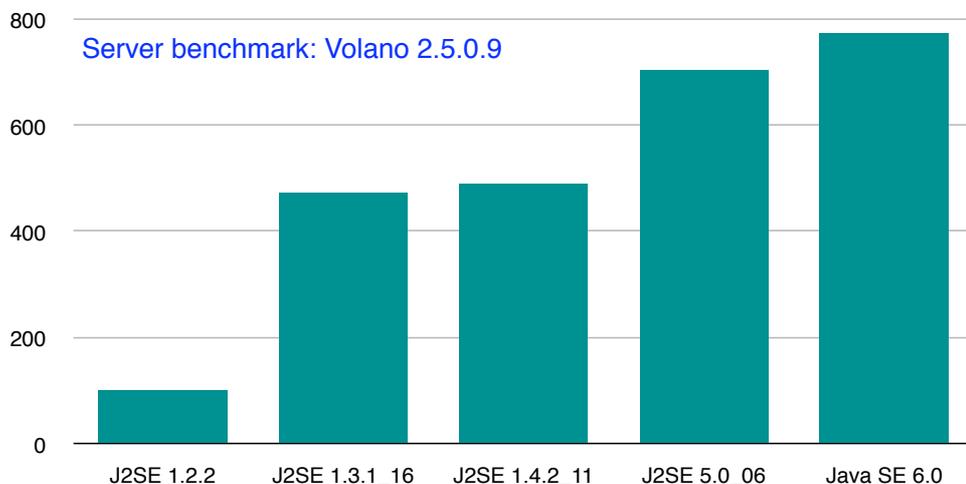
```
Method void main(java.lang.String[])
 0 new #2 <Class java.math.BigDecimal>
 3 dup
 4 aload_0
 5 iconst_0
 6 aaload
 7 invokespecial #3 <Method java.math.BigDecimal(java.lang.String)>
10 astore_1
11 new #2 <Class java.math.BigDecimal>
14 dup
15 aload_0
16 iconst_1
17 aaload
18 invokespecial #3 <Method java.math.BigDecimal(java.lang.String)>
21 astore_2
22 aload_1
23 aload_2
24 invokevirtual #4 <Method java.math.BigDecimal multiply(java.math.BigDecimal)>
27 astore_3
28 getstatic #5 <Field java.io.PrintStream out>
31 new #6 <Class java.lang.StringBuffer>
34 dup
...
60 invokevirtual #12 <Method java.lang.String toString()>
63 invokevirtual #13 <Method void println(java.lang.String)>
66 return
```

約200種類のマシン命令がある

## パフォーマンス向上の方法

- コンパイラによるパフォーマンス向上
  - > 生成するバイトコードを最適化
  - > インライン化
- インタプリタによるパフォーマンス向上
  - > JIT (Just in Time) コンパイラ
  - > 80:20 ルールと HotSpot Performance Engine
  - > GC アルゴリズムの改良
- ライブラリのコードの改良
  - > BigInteger
  - > Swing

## 10 Years of Performance



Sun JVMs are 32-bit  
 Sun Fire V490: Solaris 9 / SPARC  
 Volano Website: <http://www.volano.com>  
 Source: Java Performance Engineering

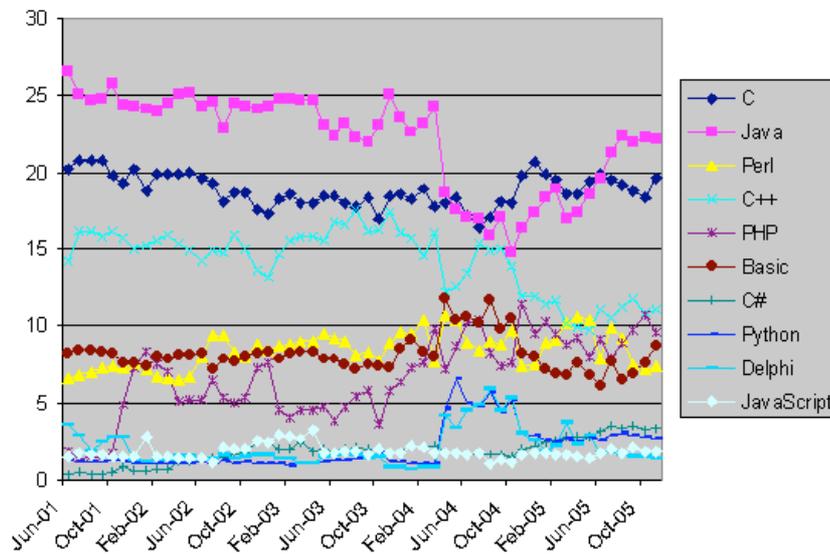
## 開発者にとっての Java

## TPC インデックス

Position 12/2005	Position 12/2004	Delta	Programming language	Rating	Delta	Status
1	2	↑	Java	22.129%	+5.80%	A
2	1	↓	C	19.632%	+0.06%	A
3	3	=	C++	11.026%	-0.89%	A
4	4	=	PHP	9.629%	-1.76%	A
5	5	=	(Visual) Basic	8.710%	+0.54%	A
6	6	=	Perl	7.393%	-0.03%	A
7	9	↑↑	C#	3.310%	+1.36%	A
8	7	↓	Python	2.678%	-0.35%	A
9	11	↑↑	JavaScript	1.838%	+0.24%	A
10	8	↓↓	Delphi/Kylix	1.426%	-1.55%	A

<http://www.tiobe.com/>

# TPC インデックス（5年間の推移）



<http://www.tiobe.com/>

# SourceForge の Java プロジェクト



## 加速するJava利用、オープンソースのSourceForgeサイトでC++を追い抜く

世界最大のオープンソースソフトウェア開発サイト、SourceForgeでJavaが、プロジェクトの数でC++を追い抜き、最有力の言語となった。Javaの父、ゴスリング氏は、Sunの無償化戦略が後押ししていると話す。

2005年12月06日 10時43分 更新

先週、「SourceForge」（世界最大のオープンソースソフトウェア開発サイト）でJavaが、プロジェクトの数でC++を初めて追い抜き、最有力の言語となった。12月1日時点で、1万6731のC++プロジェクトに対して、Javaのプロジェクトは1万6738と上回っている。

Javaを生み出し、今も後見人であるSun Microsystemsは、今回の首位到達を「力強い成長の明白な表れ」であり、Javaデベロッパーコミュニティの「参加（がもたらす力）だと呼ぶ。

## 開発ツールは無償

- NetBeans 4.1 & 5.0
- Java Studio Creator 2
- Java Studio Enterprise 8
- Sun Studio 11



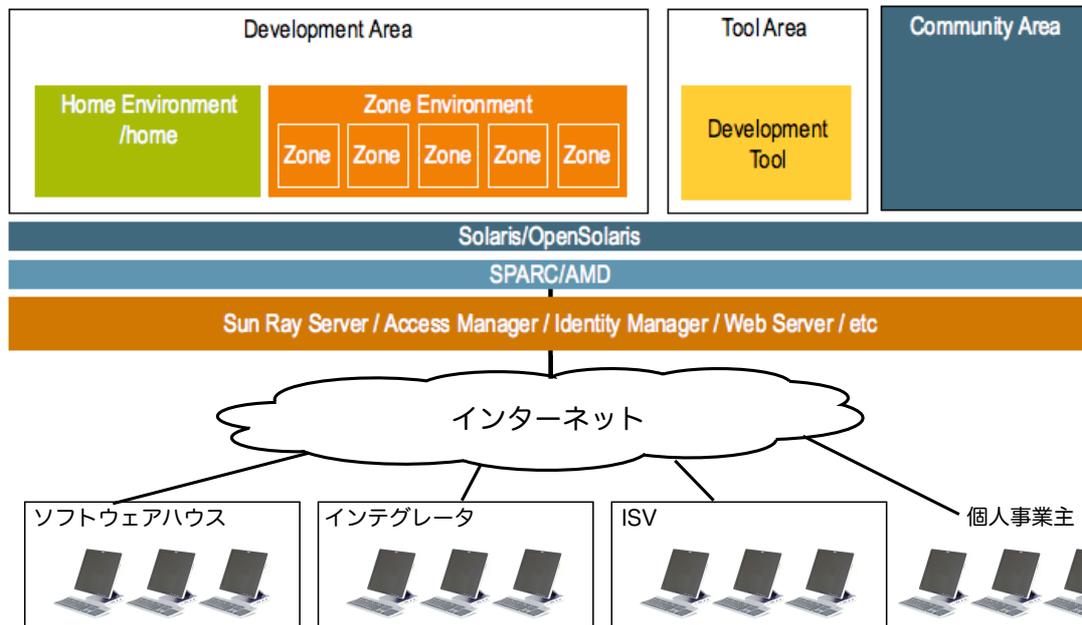
## Sun Japan Developer Grid

- ネットワーク型アプリケーション開発環境
  - > Java SE 5 等の最新の開発環境
  - > セキュアな環境
  - > モビリティ
  - > サブスクリプション・ライセンス
  - > 最適なサーバリソース

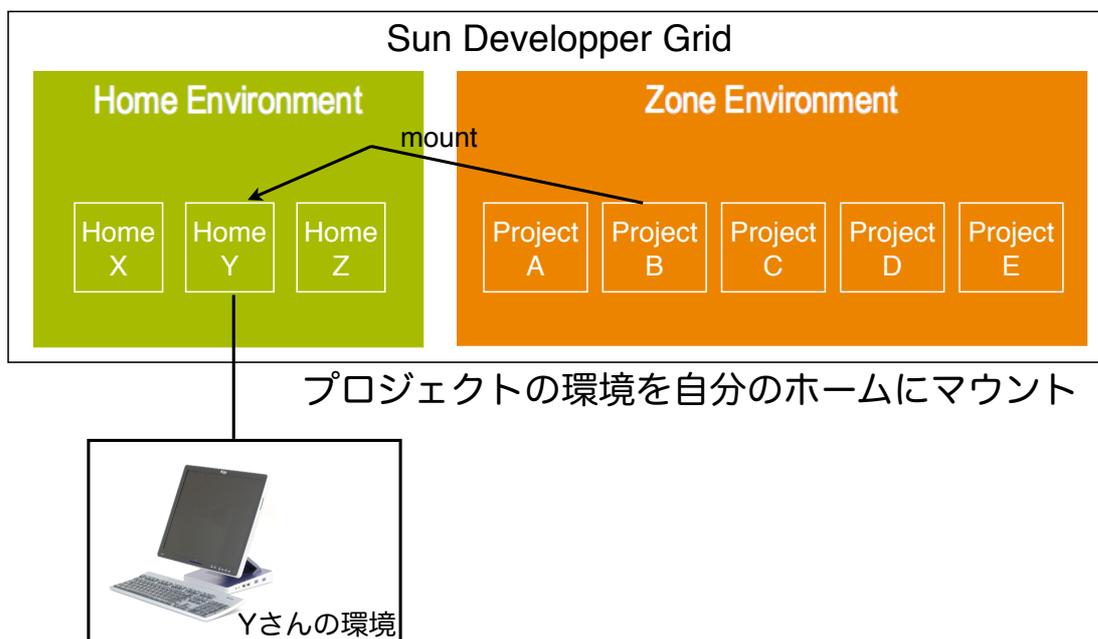


Sun Ray Ultra Thin Client

# システムの全体像



# Area Hopping





Thank you!

Hiroshi.Yamaguchi@Sun.COM