

Java 5.0

～止まらぬ進化, オブジェクト指向の更なる完全性～

平岩 健一郎, 佐藤 史隆

Kenichiro HIRAIWA, Fumitaka SATO

1 はじめに

近年, インターネットの普及に伴い, Java は急速に認知され, 普及した. 1995 年に, 実用性を持たない試作品レベルの言語として姿を現した Java は, わずか数年で急速に進化し完成した. 現在では Web アプリケーションのようなインターネットの分野だけでなく, データベース連携, 財務管理などの広範囲のタスクにおいて Java プログラミングが必要とされている. その理由は Java の持つ完全なオブジェクト指向性や, どんな環境でも動作する利便性, そしてオープンなシステムであることによるものが大きい.

本稿では, 2004 年 9 月 30 日に開発され, 劇的にバージョンアップした Java2 Standard Edition 5.0(J2SE) について述べていく.

2 これまでの Java の歴史

Java のバージョンアップの変遷を Fig. 1 にまとめる. 1998 年, 試作品といえる JDK1.1 から, Java のプログラミング言語としての完成を持って, 目的別に J2SE と J2EE(Java 2 Enterprise Edition) に分かれた. J2SE とは Java の基本的な機能を提供することを目的とし, J2EE は J2SE にサーバ上で Java を動かすための API や諸機能の付加を目的として開発が進められている.

Java1 から Java2 という名称変更後も, Java は常に新しい言語仕様の概念を取り込んできたが, J2SE 1.2 から 1.4 まで, Java に性急な変更を加えるものではなかった. しかし, J2SE 5.0 はその名称だけでなく, 言語仕様としてもそれまでに例がないほどの変更があり, すべてが驚愕に値するものである.

3 J2SE 5.0 の 4 つの目的

J2SE 5.0 の開発には, 次に示すような 4 つの大きな目的がある.

- 開発の容易性
- 拡張性と完成度
- 監視と管理可能性
- デスクトップアプリケーションの開発

1996	JDK 1.0	
1997	JDK 1.1	
1998	JDK 1.1	
1999	JDK 1.2 =J2SE	J2EE 1.1
2000	J2SE 1.3	J2EE 1.2
2001	J2SE 1.3	J2EE 1.2
2002	J2SE 1.4	J2EE 1.3
2003	J2SE 1.4 :Merlin	J2EE 1.3
2004	J2SE 1.4	J2EE 1.4
2005	J2SE 5.0 :Tiger	J2EE 1.4
2006	J2SE 5.0	J2EE 1.4

Fig. 1 Java のバージョンの変遷

4 つの目的の中でも「開発の容易性」は特に重要視されており, 追加された新機能によってコードの品質や可読性の大きな向上を実現している. その代表的なものとしては Generics, Metadata が挙げられる. 以下に, 「開発の容易性」の新機能について説明する.

3.1 Generics

Generics とは, クラス, インタフェース, メソッドなどの「型」を指定する役割を持つ. J2SE1.4 では利用する側とされる側の「型」の異なる際にもコンパイルが通り, 実行時にエラーが起こる. しかし, J2SE5.0 では, このような場合, コンパイル時にエラーを見つけることが出来るため, 問題の早期の発見が可能である. 以下の例は 3 行目でコンパイルエラーが起こる.

— J2SE 5.0 において —

```
List <String> list = new ArrayList <String>();
list.add(new String("aaa"));
Integer value = list.get(0);
```

3.2 Metadata

J2SE 5.0 における Metadata とは, クラスやメソッドなどに対するデータ (情報) のことを指す. Metadata を用いる利点は「定義したメタデータに応じて, その対象となるクラスやメソッドに何らかの処理を施せること」である.

J2SE 5.0 のメタデータは, アノテーション (注釈) に

より実装される。アノテーションはメタデータを定義したいクラスやメソッドの前に「@[アノテーション名]」と記述して使用する。Table 1 に標準で用意されているアノテーションと、その役割について示す。また、自ら定義することも可能である。

Table 1 標準で用意されているアノテーション

Deprecated	推奨されないクラスやメソッドとして定義し、コンパイル時に警告を表示する。
Override	サブクラスでオーバーライドするメソッドに定義する。誤ってオーバーロードした場合に警告を出す。
Inherited	アノテーションの定義に、このアノテーションを付けると、アノテーションが利用される際に、そのクラスのサブクラスにもアノテーションが引き継がれる。
Retention	アノテーションの利用範囲を指定する。
Target	アノテーションの対象(クラス、メソッド、パラメータなど)を定義する。
Documented	アノテーションの定義に、このアノテーションを付けると、Javadocでアノテーションが表記されるようになる。

3.3 その他の追加要素

以下に、その他に追加された要素を示す。

1) Enum

Enum とはプリミティブ型と参照型の 2 種類とは異なる、新しい第 3 のデータ型「列挙型」のことである。

Enum で定義されたクラスから生成されたオブジェクトは、要素の文字列を用いて扱うことができる上、コンパイラによるチェックを可能とする。これにより、そのクラスの使い勝手とともに、コードの可読性も向上する。

2) Autoboxing と Auto-Unboxing

プリミティブ型をラッパークラス (Integer, Boolean など) に自動交換する Autoboxing, その逆の自動変換を行う Auto-Unboxing が追加された。これによって、以下のような書き方が許されるようになった。

J2SE 5.0 において

```
Integer value = new Integer(1);
int x = value;
```

3) Varargs(可変引数)

可変引数とは、コンストラクタやメソッドの引数の個数がいくつあるかあらかじめ限定できない場合のために使用するものである。

J2SE 5.0 について

```
public void varArgs(String... arg)  (※ arg は配列)
```

4) Formatter クラスと Scanner クラス

可変引数が追加されたことに伴い、C 言語の「printf() 関数」に相当する java.util.Formatter クラス、「scanf() 関数」に相当する java.util.Scanner クラスが追加された。

5) for/in ステートメント

新しい for 文の記述方法が可能となった。

— 拡張 for 文 —

```
for(要素を表す整数:コレクション or 配列)
```

6) Static インポート

J2SE 5.0 では新しく static 変数や static メソッドの import が可能になった。以下に例を示す。

— Static import 適用例 —

```
import static java.lang.Math.PI
```

この場合、Math.PI を「PI」と記述するだけで使用でき、可読性が向上する。Static メソッドに関しても同様である。

4 Java の今後

Java は現在も開発が進んでおり、2006 年には J2SE 6.0:Mustang, 2007 年には J2SE 7.0:Dolphin が公開される予定である。J2SE 6.0 および 7.0 に向けて開発される新技術のテーマとしては、以下のものが掲げられている。

- 互換性, 安定性, 高品質
- 問題解決を容易にするための診断可能性, 監視と管理機能
- XML と Web サービス
- 開発容易性
- デスクトップ Java
- さらなるトランスペアレント

Java は開発環境が優れており、私自身 C や C++ よりも楽にプログラミングできた経験がある。Java はプログラマーに対してシンプルかつ多機能な言語なのである。

今後 Java は、さらに開発の容易性が高まり、人間がマニュアル車ではなく、オートマチック車を選ぶように、今まで以上にプログラマーに強い支持を受ける言語になっていくだろう。

参考文献

- 1) Java プログラミング同好会. 進化する Java J2SE 5.0. 株式会社 エクスメディア, 2004.
- 2) Brett McLaughlin, David Flanagan. 開発ノートシリーズ Java 5.0 Tiger. 株式会社 オーム社, 2004