

5カ条で学ぶ 堅牢性を考慮したJavaコードを書く技術

Acroquest Technology

谷本 心

この連載では、Javaプログラミングの改善方法やテクニックについて、Javaの良いコード(イケてるコード)として「読みやすいコード」「拡張しやすいコード」「性能を考慮したコード」を書く技術を解説してきました。性能を考慮したコードが書けるようになったら次に進みましょう。次のステップは、プログラムの堅牢性を考慮することです。

5カ条で学ぶ堅牢性を考慮したコード

せっかく読みやすいコードや性能を考慮したコードが書けるようになって、プログラムが意図しない個所で終了してしまったり、問題発生時に原因を調査できなかったりするコードを書いてしまえば、障害などに耐性が低いコードが完成してしまいます。これらは、堅牢性がないコードと呼べるでしょう。

堅牢性を考慮していない例を図1に挙げます。あなたのコードは大丈夫でしょうか?当てはまるものがあったら、これから説明する問題点と、その改善方法を実践してください。まずは、System.exitメソッドを直接呼んでいる例から紹介しましょう。

1 System.exitは直接呼ばない

アプリケーションを終了させる処理には、System.exitメソッドが使えます。例えば、リスト1のようなコードを見たことがある人も多いでしょう。このSystem.exitメソッドは、Javaの入門書や簡単なサンプルプログラムでよく見かけます。しかし、System.exitメソッドを乱用すべきではありません。なぜでしょうか。

System.exitメソッドは、Java仮想マシン(JVM)自体を止めてしまうメソッドです。ごく簡単なテストプログラムならまだしも、もしサーバー上で動くアプリケーションを開発していたとしたら、サーバー自体を止めてしまいかねない

凶悪な処理になるのです。

この問題への対策としては、System.exitメソッドの利用を止めて、適切な例外処理を記述することです。例外処理については後述するので、ここでは割愛します。

2 System.out.printlnも直接呼ばない

先に挙げたSystem.exitメソッド以上に、入門書や簡単なサンプルプログラムで多用されるメソッドが、System.out.println(以降、長いのでsysoutと記述します)です。例えば、リスト2のようなコードはよく見かけるコードでしょう。実は、sysoutも深く考えずに多用することは、お勧めできません。

リスト2の問題点は、sysoutで出力するメッセージにあります。sysoutが出力するメッセージは、出力先や出力フォーマットの制御が難しいのです。例えば、WebアプリケーションやAndroidアプリケーションを開発していた場合、sysoutで表示するメッセージはどこに出力されるのでしょうか?おそらくWebアプリとAndroidアプリでは、全く別の場所に表示されてしまいます。sysoutを何も考えずに書いてしまうプログラマは、一体メッセージがどこに出

図1●プログラムの堅牢性を調べる5つのポイント

□ あなたのコードは大丈夫か?

1. System.exitを直接呼んでいないか
2. System.out.printlnを直接呼んでいないか
3. toStringメソッドを実装しているか
4. 例外処理を行なっているか
5. throws Exceptionだけを書いていないか