



ヒント:このプール・クラスのバグは、並行処理に関するものではありません。



前号のJava Magazine (May/June 2012)では、Angela Caicedoが、バインディングに関するJavaFXコードについて出題しました。コードの断片を見て、そのコードの出力を考えるとという問題でした。



正解は4番です。myBoundIntのsetメソッドを呼び出すところで例外が発生します。なぜでしょうか。myBoundIntはバインドされた変数であり、直接変更できないからです。myBoundIntの値を変更するには、myBoundIntがバインドされているプロパティであるmyIntを変更する必要があります。

さて今回は、Jason Hunter (左上) と Boris Shukhatからの問題です。Hunterは、『Java Servlet Programming, 2nd Edition』(O'Reilly Media)の著者であり、MarkLogicの副CTOです。Shukhatは、Bank of America Merrill Lynchのアプリケーション・プログラミング・マネージャー兼バイス・プレジデントです。

## 1 問題

『Java Servlet Programming』の第9章にあるコード例 ConnectionPool.javaには、ちょっとしたバグがあります。このコードは何年も問題なく動いていましたが、最近JDBCドライバがアップグレードされたため、動かなくなりました。

## 2 コード

このプール・クラスのコードの一部を次に示します。このコードの問題点を特定し、プログラム全体を再設計せずに修正してください。

```
private Hashtable connections = new Hashtable();
private void initializePool(...) ... {
    for (int i = 0; i < initialPoolSize; i++)
        connections.put(getNewConnection(...), Boolean.FALSE); //false=使用されていない
}
public Connection getConnection() ... {
    // ... フラグがFALSEのConnectionを検索 ...
    connections.put(con, Boolean.TRUE);
    return con;
}
public void returnConnection(Connection returned) {
    connections.put(returned, Boolean.FALSE);
}
```

## 3 正しい修正はどれ?

- 1) Connectionの使用をやめ、代わりにPooledConnectionを使う
- 2) Hashtableの使用をやめ、代わりにMapインタフェースの別の実装を使う
- 3) 問題のあるConnection実装を拡張し、いくつかのメソッドをオーバーライドする
- 4) 問題のあるConnectionをラップするようなConnectionの実装を作成する

わかりましたか?

正解は次号に掲載されます。もしくはチャレンジした内容を電子メールでお送りください。

画像: I-HUA CHEN

