

2008年12月8日
日立ソフト

日立ソフト、集中型Javaコード診断ツール「anyWarp CodeDirector」に新機能 ～バグに直結するコードを指摘するルール追加により品質の大幅向上～

日立ソフト（本社：東京都品川区、執行役社長：小野 功）は、Java開発におけるコーディングルール違反をプロジェクト全体で組織的にチェックできる集中型Javaコード診断ツールを機能拡大した「anyWarp CodeDirector（エニーワープ・コードディレクター）V5.2」を2008年12月10日より発売します。今回の製品では、日立ソフトが社内で運用してきた実績や既存ユーザからの要望を元に、開発者が間違いやすくバグに直結しやすいコードを指摘できるように、新たに31のルールを追加しました。従来のルールに加え今回追加されたルールを適用することにより、品質の大幅な向上を図ることができます。

大規模ソフトウェアの品質、保守性、生産性を維持するためには、プロジェクトで定めた標準化規約（ルール）に従った実装（コーディング）を行い、誰でもメンテナンスできるようなソースコードを書くことが重要となってきます。開発者固有の癖やムラをなくし、ソースコードのメンテナンス性を高めるためには、ルールの逸脱を開発者に早期にフィードバックし、工程中も繰り返しチェックを行うことが有効です。しかし、通常コード診断は人手によるレビューによって行われるため、レビューアーや開発者など多くの人手がかかり、また目視による確認では見落としも多くなる状況でした。このような状況を打破するため、「anyWarp CodeDirector」は、開発者からプロジェクトマネージャに至る、開発プロジェクトに携わる全員にとって、負担の少ない効率的なコード診断を実施する仕組みを提供しています。

「anyWarp CodeDirector」が提供するコード診断の実行方式には二通りあります。一つは、開発者向けに、自分の担当しているソースコードを隨時チェックできるように、Java統合開発環境「Eclipse」上でコード診断を実行する方式です。もう一つは、主にプロジェクトマネージャ向けに、「CVS」「Visual SourceSafe」「Subversion」といった現在普及している構成管理ツールと連携して自動的にシステム全体のソースコードを取り出し、コード診断を実行する方式です。後者の実行結果は、コード診断結果レポートとしてWeb上で公開されますので、規約違反の状況、推移、集中箇所などを開発者、リーダ、プロジェクトマネージャ、品質管理部門等で共有、比較することができます。これらにより、プロジェクトマネージャは開発者に負担をかけずに、品質改善状況と推移を日々客観的に管理することができます。同時に開発者もJavaの標準的なコーディングスタイルを学習できるため、スキルアップや品質の向上を図ることができます。

「anyWarp CodeDirector」は、日立ソフトが手がけた数多くの開発プロジェクトでの経験や既存ユーザからの要望に基づき、診断の準備、実施、結果の確認を省力化するための仕組みや、コード診断に用いるルールを充実してきました。今回の機能強化では、開発者が間違いやすくバグに直結しやすいコードを指摘するルールを拡充しました。例えば、条件式の結合ミス、リソースのクローズ漏れ、マルチスレッドに関する考慮漏れなどは、開発者のスキル不足あるいは不注意によりソースコードに紛れ込みやすい間違いと言えます。しかも、こうした間違いは致命的かつ比較的再現が難しいバ

© 日立ソフトウェアエンジニアリング株式会社

〒140-0002 東京都品川区東品川4丁目12番7号（日立ソフトタワーA）
TEL. (03) 5780-2111 (大代)

ホームページ <http://hitachisoft.jp/>

グに直結することがほとんどです。こうした間違いを指摘するルールの充実により、より早い段階で重大なバグを検出することができ、少ない作業で品質の安定したシステム構築が実現できます。

今回新たに追加された 31 のルールのうち、3 つの具体例を参考資料に示します。

日立ソフトでは、「anyWarp CodeDirector」をソフトウェア開発ソリューションの一つとして位置づけ、年間 500 ユーザライセンスを目標に、積極的に拡販していきます。

また当社では、

- (1) 「anyWarp CodeDirector」：コーディング段階のテストにて品質の維持とリスク低減を実現
- (2) 回帰テスト自動実行ツール「anyWarp Capture/Replay」：回帰テストの効率化を支援
- (3) Web システム性能測定ツール「Assam WebBench」：総合テストから運用段階での性能テストをカバー

を組み合わせたテストスイート製品「anyWarp TestSuite」を提供しており、プログラム開発から総合テストにわたり、テストの効率化・品質向上を支援します。

<出荷開始時期> 2008 年 12 月 10 日

<製品紹介ホームページ> <http://hitachisoft.jp/products/anywarp/codedirector/index.html>

<「anyWarp CodeDirector」製品価格>

製品名	製品内容	標準価格（税別）
anyWarp CodeDirector 10 ユーザ版（*）	ライセンス	1,000,000 円
anyWarp CodeDirector 30 ユーザ版（*）	ライセンス	2,600,000 円
anyWarp CodeDirector 50 ユーザ版（*）	ライセンス	4,000,000 円
anyWarp CodeDirector 100 ユーザ版（*）	ライセンス	6,000,000 円
anyWarp CodeDirector サイトライセンス版（*）	ライセンス	30,000,000 円
anyWarp CodeDirector メディアキット	CD	5,000 円

（*）ユーザ数は、本製品のレポート出力を参照するユーザおよび Eclipse プラグインを使用するユーザを意味しています。
例えば 10 ユーザ版の場合、5 人のユーザがレポート参照を、残りの 5 人のユーザが Eclipse プラグインを使用することができます。レポート参照を行うユーザと Eclipse プラグインを使用するユーザが同じ場合両方にカウントされます。

<「anyWarp CodeDirector」サービス体系>

サービス名	サービス内容	標準価格
年間保守サービス (@Service24)	サービス内容については以下をご覧下さい http://service24.hitachisoft.jp/	製品標準価格の 16%/年
コーディングルール開発 コンサルティングサービス	コーディングルール設計や運用環境の構築等 の支援	個別見積

<前提稼動環境>

OS	Microsoft® Windows 2003 Server Standard Edition Microsoft® Windows 2003 Server Enterprise Edition Microsoft® Windows 2000 Server (SP4 以降) Microsoft® Windows 2000 Advanced Server (SP4 以降) Microsoft® Windows 2000 Professional (SP4 以降) Microsoft® Windows XP Professional (SP1 以降)
Java	JDK1.4, 5.0
Eclipse	Eclipse 3.2, 3.3
ブラウザ	Internet Explorer 6, 7
ハードウェア	上記OSが稼動するPCサーバ CPU : Pentium4 2.4GHz以上を推奨 実装メモリ : 1GB 以上を推奨

- * Java および全ての Java 関連の商標およびロゴは、米国およびその他の国における米国 Sun Microsystems, Inc.の商標または登録商標です。
- * Microsoft、Windows、Internet Explorer、Visual SourceSafe は、米国およびその他の国における米国 Microsoft Corp.の登録商標です。
- * Pentium4 は、米国およびその他の国における米国 Intel Corp.の登録商標です。
- * Apache Struts は、Apache Software Foundation の商標または登録商標です
- * anyWarp CodeDirector は日立ソフトの登録商標です。
- * その他記載の会社名、製品名はそれぞれの会社の商標または登録商標です。
- * 日立ソフトの正式名称は、日立ソフトウェアエンジニアリング株式会社です。

<本件に関するお問い合わせ先>

担当部署：日立ソフト @Sales24

Tel: 03-5479-8831 ホームページ: <http://sales24.hitachisoft.jp/>

<報道機関のお問い合わせ先>

日立ソフト CSR 本部広報 IR 部 担当：竹橋、高野

Tel: 03-5780-6450 Email: press@hitachisoft.jp

©日立ソフトウェアエンジニアリング株式会社

〒140-0002 東京都品川区東品川4丁目12番7号(日立ソフトタワーA)
TEL. (03) 5780-2111(大代)

ホームページ <http://hitachisoft.jp/>

【参考資料】

<「anyWarp CodeDirector V5.2」で新たに追加されたルールの例>

今回新たに追加された31のルールのうち、具体例を3つ挙げます。

(1) 条件式の結合ミスを指摘するルール(ConditionWithKnownNullValue ルール)

このルールでは、条件式を誤って結合したため、値が null の変数を参照し実行時にエラーが発生するバグを指摘することができます。このようなバグは正常系の単体テストでは発生せず、またコンパイルエラーにならないため見過ごされがちです。以下の例では、4行目において、二つの条件式が誤って AND 演算子（正しくは OR 演算子であるべき）で結合されていることを指摘しています。

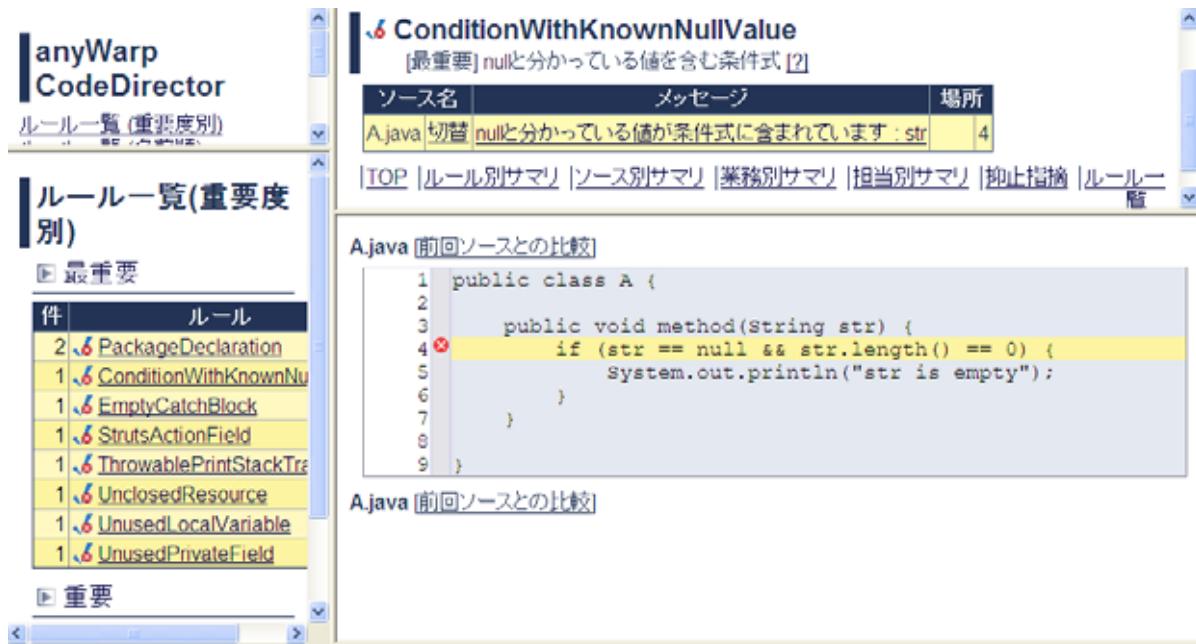


図1 . ConditionWithKnownNullValue ルールの指摘画面例

(2) リソースのクローズ漏れを指摘するルール(UnclosedResource ルール)

このルールでは、リソースのクローズ処理が漏れることにより、リソースが解放されずメモリリークなどの問題が発生するバグを検出します。以下の例では、9行目で生成した BufferedWriter オブジェクトに対するクローズ処理が漏れていることを指摘しています。

anyWarp
CodeDirector

ルール一覧(重要度別)

ルール一覧(重
要度別)

□ 最重要

件	ルール
1	EmptyCatchBlock
1	PackageDeclaration
1	StrutsActionField
1	ThrowablePrintStatement
1	UnclosedResource
1	UnusedLocalVariable
1	UnusedPrivateField

□ 重要

件	ルール
---	-----

TOP | ルール別サマリ | ソース別サマリ | 業務別サマリ | 担当別サマリ | 抑止指摘 | ルール一覧

UnclosedResource

[最重要] リソースのクローズ処理が抜けている可能性 [?]

ソース名 メッセージ 場所

Test.java 切替ソースのクローズ処理が抜けている可能性があります: bw 9

```

4
5 public class Test {
6
7     public static void main(String[] args) {
8         try {
9             BufferedWriter bw = new BufferedWriter(new FileWriter(args[0]));
10            bw.write("Hello");
11
12        } catch (IOException e) {
13            e.printStackTrace();
14        }
15    }
16 }
```

Test.java [前回ソースとの比較]

図 2 . UnclosedResource ルールの指摘画面例

(3) マルチスレッドに関する考慮漏れを指摘するルール(StrutsActionField ルール)

このルールでは、Struts の Action クラスを継承したクラスにおいて、インスタンスフィールドが宣言されているというように、マルチスレッドに関する考慮漏れがあることを指摘します。Action クラスのインスタンスは、複数のスレッドにより共有されるオブジェクトであるため、こうしたインスタンスフィールドを持っていると、同時にアクセスされた HTTP リクエストによって値が共有されてしまい、例えば別人になりすませるなどの重大な問題が発生する可能性があります。また、このようなバグは単体テストでは発生せず再現性も低いので、早期に発見することが重要です。以下の例では、Action クラスを継承したクラスの 14 行目にインスタンスフィールドが宣言されていることを指摘しています。

anyWarp
CodeDirector

ルール一覧(重要度別)

ルール一覧(重
要度別)

□ 最重要

件	ルール
1	EmptyCatchBlock
1	PackageDeclaration
1	StrutsActionField
1	ThrowablePrintStatement
1	UnclosedResource
1	UnusedLocalVariable
1	UnusedPrivateField

□ 重要

件	ルール
---	-----

TOP | ルール別サマリ | ソース別サマリ | 業務別サマリ | 担当別サマリ | 抑止指摘 | ルール一覧

StrutsActionField

[最重要] Struts の Action クラスにフィールドを定義している [?]

ソース名 メッセージ 場所

LoginAction.java 切替 Struts の Action クラスにフィールドを定義しています 14

example/LoginAction.java [前回ソースとの比較]

```

1 package example;
2
3 import javax.servlet.http.HttpServletRequest;
4 import javax.servlet.http.HttpServletResponse;
5
6
7 import org.apache.struts.action.Action;
8 import org.apache.struts.action.ActionForm;
9 import org.apache.struts.action.ActionForward;
10 import org.apache.struts.action.ActionMapping;
11
12 public class LoginAction extends Action {
13
14     private String name;
15
16     public ActionForward execute(ActionMapping mapping, ActionForm form,
17         HttpServletRequest req, HttpServletResponse res) {
```

マイコンピュータ 100%

図 3 . StrutsActionField ルールの指摘画面例