



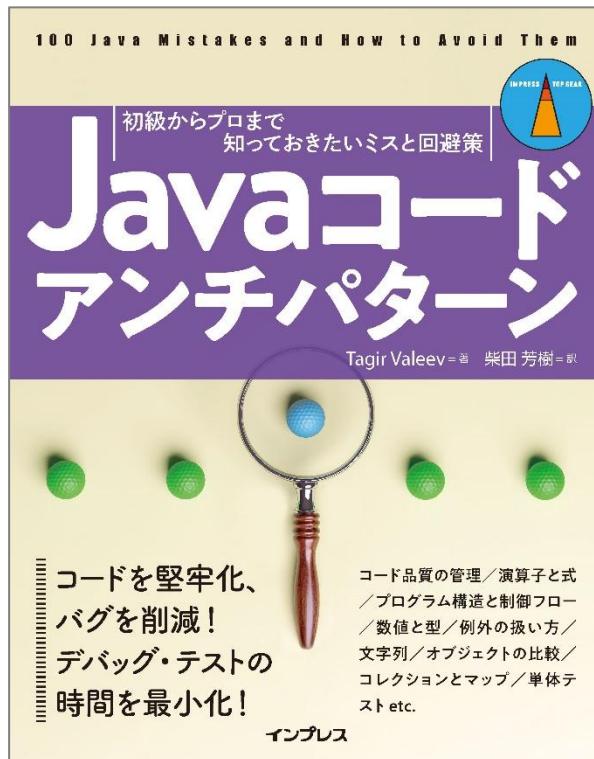
An **impress** Group Company

各 位

2026年2月12日
株式会社インプレス

間違いややすいミス、潜在的なバグを摘み取る！
100の失敗から学ぶJavaコーディングを解説
『Javaコードアンチパターン 初級からプロまで知っておきたいミスと回避策』を
2月12日に発売
発売を記念して第1章を2週間、無料公開

インプレスグループでIT関連メディア事業を展開する株式会社インプレス（本社：東京都千代田区、代表取締役社長：高橋隆志）は、Javaソフトウェアの開発に携わる方々に向けて、陥りやすいミス（失敗）をまとめ、効果的な回避策を提供する書籍『Javaコードアンチパターン 初級からプロまで知っておきたいミスと回避策』を2026年2月12日（木）に発売いたします。



■大きな問題になる前にコードのミスを修正する

小さなバグであっても開発中に見逃してしまうと、運用環境において大きな問題になる可能性があります。問題が深刻化する前に、できるだけ早めにコードの間違いを見つけて修正したいところです。そこで本書では、Javaコードの品質向上が図れるように、プログラマーが陥りやすいミスとその回避策を解説します。

■陥りやすい 100 個のミスと、その回避策、ミスの検出について解説

本書の著者は、Java を使った商用ソフトウェア開発で 15 年、静的解析ツールの開発で 10 年の経験があり、さまざまなコードのミスに遭遇し、多くの開発者が犯しやすい間違いがあることに気付きました。本書では、プログラマーが陥りやすい 100 個の典型的なミスについて抽出し、それらを回避するための具体的な方法を提供します。まず第 1 章ではコードの品質を管理してバグを回避するための一般的なソフトウェアエンジニアリングのアプローチについて説明します。第 2 章以降では、コードのミスと回避策のほか、静的解析ツールによるミスの検出について各章ごとにカテゴリーに分けて解説していきます。

■本書は以下のような方におすすめです

- ・ 初中級レベルからプロまでの Java プログラマーの方
- ・ 特に十分な実務経験がないソフトウェア開発者の方

■紙面イメージ

第 2 章 式

2.2.3 條件演算子と null 検査

同じ問題の別のパターンは、文字列の結合後に参照を null と比較し、条件演算子の分岐で文字列の値を生成する場合に発生します。たとえば、次のメソッドを考えてみてください。

```
String format(String value) {
    return "Value: " + value != null ? value : "(unknown)";
}
```

ここで、value が null の場合に "Value: (unknown)" を返し、null ではない場合に実際の値を結合して返す意図でした。しかし、return の式は次のように構文解析されます（図 2.4 参照）。

```
(("Value: " + value) != null) ? value : "(unknown)"
```

その後、文字列の結合が先に実行され、その後に null 検査が行われます。文字列の結合は null を生成しないため、検査は常に成功します。したがって、このメソッドは必ず接頭辞なしの value 値を返すことになり、null の場合はそのまま返されます。デスクタはこれが意図された動作だと想うかもしれないが、この動作は手動テストでは見逃されやすくなります。私自身も、内部診断情報を生成する手続きの中でこのようなハックを見つかったことがあります。

プログラマが条件をどのように解釈するか



コンパイラが条件をどのように解釈するか



図 2.4 文字列の結合は、null 検査や条件演算子よりも高い優先順位を持つ

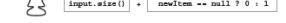
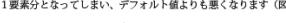
同様の問題が、Elasticsearch プロジェクトでも発見されました。それは、初期容量を指定して ArrayList オブジェクトを初期化するコードでした。元のコードは、次のメソッドとほぼ同じです。

```
List<String> trimAndAdd(List<String> input, String newItem) {
    List<String> result = new ArrayList<>();
    input.size() + newItem -- null ? 0 : 1; // ← 誤った初期容量計算
    for (String s : input) {
        result.add(s.trim());
    }
    if (newItem != null) {
        result.add(newItem.trim());
    }
    return result;
}
```

30

2.2 No.2: 条件式での括弧の欠落

ここでは、結果として得られるリストの要素数が既もってわかっているため、あらかじめ容量を確保しておくのがよいと考えるに見えます。しかし、括弧がなければ、加算演算子が数値の加算ではなく文字列結合として解釈されます。文字列結合の結果は null にはならないので、ArrayList の初期容量は常に 1 要素分として計算され、デフォルト値よりも悪くなります（図 2.5 参照）。



文字列の結合結果は null にはならない

図 2.5 加算が結合として解釈される

静的解析

この状況は、文字列結合の結果を null と比較しているのは怪しいため、静的解析ツールで簡単に検出できます。IntelliJ IDEA では「条件は常に false」警告が表示され、SonarQube では「S2288 Conditionally Executed Code Should Be Reachable」警告が表示されます。

回避方法

- 異なる演算子を含む複雑な式は、次のように中间変数（この場合は displayedValue）を使って分割してください。

```
String displayedValue = value != null ? value : "(unknown)";
return "Value: " + displayedValue;
```

同様に、trimAndAdd() の例における ArrayList の初期化も、次のように書き換えられます。

```
int additionalElements = newItem -- null ? 0 : 1;
List<String> result = new ArrayList<>(input.size() + additionalElements);
```

・文字列結合を避け、次のようにフォーマット呼び出しを使うようにしてください。

```
return String.format("Value: %s",
                      value != null ? value : "(unknown)");
```

あるいは、Java 15 以降なら、次のように書けます。

```
return "%s".formatted(value != null ? value : "(unknown)");
```

残念ながら、今日の JVM では、フォーマット呼び出しは文字列結合よりもかなり遅いため、

31

ミスやバグとなりうる各種コードについてその例と背景を解説した後、その「回避方法」を説明。静的解析ツールでそうしたコードを検出できるか否かなどは「静的解析」コラムで示しています。

■本書の第 1 章を 2 週間限定で無料公開

書籍の発売を記念して、第 1 章の内容を 2 週間限定で無料公開いたします。それらの内容は、弊社開発の「インプレス Web ブックビューアー」を使って、登録不要かつ Web ブラウザで紙面を閲覧できるので、場所や時間を選ばずに PC やスマートフォンなどから利用できます。使用するビューアーは SNS で書籍の情報を共有したり、書籍の購入や詳細を確認したりできる機能も備えています。

- ・『Java コードアンチパターン 初級からプロまで知っておきたいミスと回避策』

無料公開へのリンクを掲載したページ：<https://book.impress.co.jp/books/1125101013>

【公開期間：2026年2月12日（木）から2月25日（水）まで】

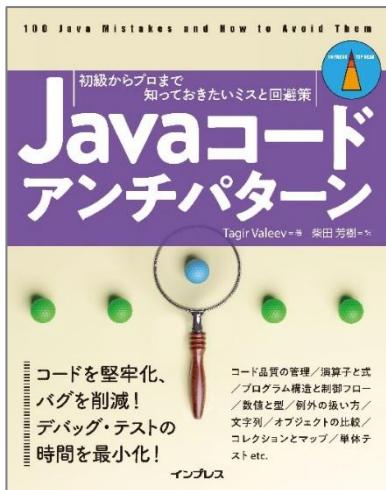
■本書の構成

- 第1章 コード品質の管理
- 第2章 式
- 第3章 プログラム構造
- 第4章 数値
- 第5章 代表的な例外
- 第6章 文字列
- 第7章 オブジェクトの比較
- 第8章 コレクションとマップ
- 第9章 その他のライブラリメソッド
- 第10章 単体テスト

付録 A 静的解析用アノテーション

付録 B 静的解析ツールの拡張

■書誌情報



書名：『Java コードアンチパターン 初級からプロまで知っておきたいミスと回避策』（impress top gear シリーズ）

著者：Tagir Valeev

訳者：柴田 芳樹

発売日：2026年2月12日（木）

ページ数：376ページ

サイズ：B5変型判

定価：3,960円（本体3,600円+税10%）

電子版価格：3,960円（本体3,600円+税10%）※インプレス直販価格

ISBN：978-4-295-02377-7

◇Amazonの書籍情報ページ：<https://www.amazon.co.jp/dp/4295023779/>

◇インプレスの書籍情報ページ：<https://book.impress.co.jp/books/1125101013>

■著者プロフィール

Tagir Valeev（タギール・ヴァーレーエフ）

統合開発環境などを開発・提供する JetBrains GmbH のテクニカルリード。IntelliJ IDEA における Java 言語サポートを担当する Java チームに所属。主な関心分野は、静的解析とコードリファクタリング。 IntelliJ IDEA に内蔵されている静的解析機能の多くのコードインスペクションを設計・開発。商用の Java ソフトウェアの開発で 15 年、静的解析の分野で 10 年の経験を有する。2006 年にコンピュータサイエンス博士号を取得、2020 年に Java テクノロジー／コミュニティのリーダーの称号 Java Champion に認定された。

■翻訳者プロフィール

柴田 芳樹（しばた よしき）

1959年11月生まれ。九州工業大学情報工学科で情報工学を学び、1984年同大学大学院で情報工学修士課程を修了。パロアルト研究所を含む米国ゼロックス社での5年間のソフトウェア開発も含め、UNIX（Solaris/Linux）、C、Mesa、C++、Java、Goなどを用いたさまざまなソフトウェア開発に従事してきた。2017年9月以降、Goによるウェブサービスのバックエンドソフトウェア開発に携わっている。Java や Go の技術教育も行っており、私的な時間に技術書の翻訳や講演など多く行っている。訳書は

『Effective Java 第3版』『プログラミング言語 Go』『プログラミング言語 Java 第4版』（以上、丸善出版）、『Go 言語で学ぶ並行プログラミング 他言語にも適用できる原則とベストプラクティス』『Go 言語 100Tips ありがちなミスを把握し、実装を最適化する』（以上、インプレス）、『ベタープログラマ』『アプレンティスシップ・パターン』（以上、オライリー・ジャパン）など20タイトル以上。

■impress top gear シリーズについて



impress top gear シリーズは、21世紀のIT時代に求められる新たなトレンドを取り上げることに加えて、これまでの技術・知識を時代に合わせた形で提供する技術解説書です。読者がITのフィールドで「トップギア」へとシフトチェンジできるような技術・知識を書籍として提供していきます。

以上

【株式会社インプレス】 <https://www.impress.co.jp/>

シリーズ累計8,000万部突破のパソコン解説書「できる」シリーズ、「デジタルカメラマガジン」等の定期雑誌、IT関連の専門メディアとして国内最大級のアクセスを誇るデジタル総合ニュースサービス「Impress Watch シリーズ」等のコンシューマ向けメディア、「IT Leaders」をはじめとする企業向けIT関連メディアなどを総合的に展開・運営する事業会社です。IT関連出版メディア事業、およびデジタルメディア&サービス事業を幅広く展開しています。

【インプレスグループ】 <https://www.impressholdings.com/>

株式会社インプレスホールディングス（本社：東京都千代田区、代表取締役：塙本由紀）を持株会社とするメディアグループ。「IT」「音楽」「デザイン」「山岳・自然」「航空・鉄道」「モバイルサービス」「学術・理工学」を主要テーマに専門性の高いメディア&サービスおよびソリューション事業を展開しています。さらに、コンテンツビジネスのプラットフォーム開発・運営も手がけています。

【本件に関するお問合せ先】

株式会社インプレス 広報担当：丸山

E-mail: pr-info@impress.co.jp URL: <https://www.impress.co.jp/>

※弊社はテレワーク推奨中のため電話でのお問合せを停止しております。メールまたはWebサイトからお問合せください。