

FUJITSU Software
SIMPLIA MF-STEP COUNTER
Javaソフトウェアメトリクス計測機能
V60L14

ユーザーズガイド

Windows

SIMPLIA-STJ-JP60(07)
2018年11月

まえがき

SIMPLIA MF-STEP COUNTER Javaソフトウェアメトリクス計測機能は、Java開発資産の計測をおこなうアプリケーションです。当機能を使用することにより、クラスやメソッドの開発進捗状況や設計品質を把握することができます。

新機能

本機能の機能追加などはありません。

ヘルプを読むために

ユーザガイドの参照用ブラウザとしては、各オペレーティングシステムでサポートする最新版のInternet Explorerを推奨します。

登録商標について

本ユーザガイドで使われている登録商標及び商標は、以下のとおりです。

- Microsoft, Windows, MS-DOS, MSは、米国Microsoft Corporationの米国およびその他の国における登録商標です。
- OracleとJavaは、Oracle Corporation およびその子会社、関連会社の米国およびその他の国における登録商標です。文中の社名、商品名等は各社の商標または登録商標である場合があります。

略記について

本ユーザガイドでは、各製品を次のように略記しています。

「Windows(R) 10 Home」または、 「Windows(R) 10 Pro」、 「Windows(R) 10 Enterprise」、 「Windows(R) 10 Education」	→	「Windows 10」
「Windows(R) 8.1」または、 「Windows(R) 8.1 Pro」、 「Windows(R) 8.1 Enterprise」	→	「Windows 8.1」
「Windows(R) 7 Home Premium」または、 「Windows(R) 7 Professional」、 「Windows(R) 7 Enterprise」、 「Windows(R) 7 Ultimate」	→	「Windows 7」
「Microsoft(R) Windows Server(R) 2016 Datacenter」または、 「Microsoft(R) Windows Server(R) 2016 Standard」、 「Microsoft(R) Windows Server(R) 2016 Essentials」	→	「Windows Server 2016」
「Microsoft(R) Windows Server(R) 2012 R2 Datacenter」または、 「Microsoft(R) Windows Server(R) 2012 R2 Standard」、 「Microsoft(R) Windows Server(R) 2012 R2 Essentials」、 「Microsoft(R) Windows Server(R) 2012 R2 Foundation」	→	「Windows Server 2012 R2」
「Microsoft(R) Windows Server(R) 2012 Datacenter」または、 「Microsoft(R) Windows Server(R) 2012 Standard」、 「Microsoft(R) Windows Server(R) 2012 Essentials」、 「Microsoft(R) Windows Server(R) 2012 Foundation」	→	「Windows Server 2012」
「Microsoft(R) Windows Server(R) 2008 R2 Datacenter」または、 「Microsoft(R) Windows Server(R) 2008 R2 Standard」 「Microsoft(R) Windows Server(R) 2008 R2 Enterprise」 「Microsoft(R) Windows Server(R) 2008 R2 Foundation」	→	「Windows Server 2008 R2」
「Windows 10」または、 「Windows 8.1」、 「Windows 7」、 「Windows Server 2016」、 「Windows Server 2012 R2」、	→	「Windows」

「Windows Server 2012」、 「Windows Server 2008 R2」		
「Java(TM)」	→	「Java」
「Java(TM) 2 Platform Standard Edition Development Kit 8.0」	→	「JDK」または、「JDK8.0」

目次

第1章 背景と目的	1
第2章 概要説明	2
第3章 導入手順	4
3.1 インストール/アンインストール	4
3.2 動作環境	4
第4章 機能説明	5
4.1 計測内容説明	5
4.1.1 クラス単位の計測	5
4.1.2 メソッド単位の計測	6
4.2 アラーム指定	7
4.2.1 測定値の判定	7
4.2.2 アラーム指定基準値設定例	8
4.2.3 パーセンタイル形式のアラーム表示	11
第5章 操作説明	14
5.1 起動/終了方法	14
5.2 画面処理	14
5.2.1 新規計測対象一覧ファイル指定方法	14
5.2.2 対象資産の指定方法	15
5.2.3 計測対象一覧ファイルの読み込み方法	15
5.2.4 クラスパスの指定方法	15
5.2.5 計測単位の指定方法	16
5.2.6 アラーム表示の指定方法	16
5.2.7 計測結果の表示方法	17
5.2.8 計測対象一覧ファイルの保存方法	17
5.2.9 計測方法	18
5.3 バッチ処理	18
5.3.1 ファイル指定による計測方法	18
5.3.2 フォルダ指定による計測方法	18
5.3.3 計測対象一覧による計測方法	19
5.3.4 計測単位の指定方法	19
5.3.5 計測結果出力先の指定方法	19
5.4 コマンドライン	20
5.4.1 コマンドライン引数	20
5.4.1.1 起動形態パラメタ	20
5.4.1.2 対象資産パラメタ	20
5.4.1.3 クラスパスパラメタ	21
5.4.1.4 オプションパラメタ	21
5.4.1.5 CSVパラメタ	22
5.4.1.6 ERRパラメタ	22
5.4.2 パラメタ排他関係	22
第6章 画面説明	24
6.1 メイン画面	24
6.2 計測中断ダイアログボックス	25
6.3 計測結果表示画面	26
6.4 クラスパス指定ダイアログボックス	28
6.5 オプション指定ダイアログボックス	28
6.6 アラーム指定ダイアログボックス	29
6.7 バージョン情報ダイアログボックス	30
第7章 補足事項	31
7.1 注意事項	31

第8章 メッセージ.....	33
8.1 メッセージ一覧.....	33
第9章 付録.....	40
9.1 計測結果説明(CSVファイル).....	40
9.2 サンプルの使い方.....	41
9.2.1 サンプルソースを計測するには.....	41
9.2.2 計測方法を変更するには.....	41
9.2.3 アラーム表示するには.....	42
9.2.4 計測対象を保存するには.....	42
9.2.5 再計測するには.....	42
9.2.6 計測結果の判定について.....	43

第1章 背景と目的

ソフトウェア開発の中ではバグの収束率が悪い、要件変更などが度々行われデグレートを引き起こすといった状況は多くあります。

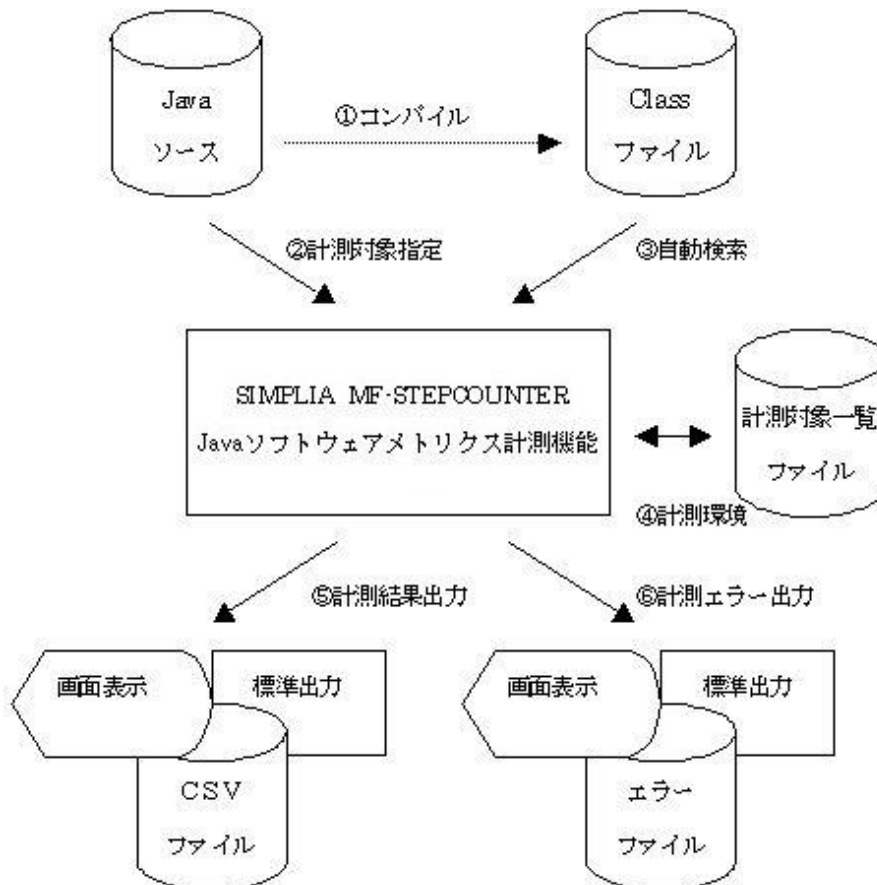
度重なる改修を行うことで、ソースコードの肥大化が要因となった場合は、処理を理解するために多くの労力が必要となり、限られた時間の十分なテストによってバグが存在しないことを示すのは難しくなります。また、要件変更に迅速に応えられるような柔軟な設計を十分に行うことも難しくなっていきます。

ソースコードの品質を数値化して定量的に評価することは品質向上の面で重要です。

メトリクス計測機能では、Javaソースとクラスファイルから、クラスやメソッドの品質管理/評価の指標となる項目を計測します。当機能を使用することにより、クラスやメソッドの開発進捗状況や設計品質を把握することができます。また、テスト時にはどのクラス/メソッドを重点的にテストすべきか簡単に把握できます。

第2章 概要説明

当ツールは、Java開発資産であるJavaソースと、Javaソースのコンパイル結果であるClassファイルを入力とし、計測をおこなうツールです。計測は、クラス単位の計測とメソッド単位の計測が可能で、計測結果は指定の計測単位ごとに出力されます。計測結果は、画面上への表示の他、CSVファイルにも出力されます。一度計測した資産は、計測対象一覧ファイルとして保存することで、何度でも同じ環境で計測することができます。ただし、SQLJを使用したJavaソースは計測できません。



1. コンパイル

当ツール実行前に、事前にJavaソースをコンパイルする必要があります。

2. 計測対象指定

計測対象となるJavaソースをファイルまたはフォルダ単位に指定します。指定資産のうち、拡張子が.Javaのファイルのみを計測対象とし、計測をおこないます。

3. 自動検索

Classファイルの計測は、Javaソースの解析によりファイルを特定し、環境変数CLASSPATHに指定されるパス名とツール内に指定されるクラスパス名から自動的に検索し、計測をおこないます。

4. 計測環境:

指定の環境(出力先csvファイル名/計測対象資産名/計測単位/クラスパス名/アラーム指定形式)を保存できます。保存した計測対象一覧ファイルを使用することで同一環境で何度でも計測することが可能です。

5. 計測結果出力

指定の計測単位ごとに計測結果を出力します。

6. 計測エラー出力

計測エラー発生時は、エラー内容を出力します。

計測結果として出力される項目は以下のとおりです。

(1)クラス単位の計測を指定した場合

No	基本計測	情報取得元	内容
1	クラス名	Javaソース	計測単位となるクラス名を出力します。
2	文数	Javaソース	クラス内の実行文の数を計測します。
3	継承数	Classファイル	クラスの階層数を計測します。
4	フィールド変数の数	Classファイル	クラス内のフィールド変数の数を計測します。
5	インスタンス変数の数	Classファイル	クラス内のインスタンス変数の数を計測します。
6	パブリック変数の数	Classファイル	クラス内のパブリック変数の数を計測します。
7	メソッド数	Classファイル	クラス内に定義されるメソッドの数を計測します。
8	インスタンスメソッド数	Classファイル	クラス内に定義されるインスタンスメソッド数を計測します。
9	パブリックメソッド数	Classファイル	クラス内に定義されるパブリックメソッド数を計測します。
10	定義ファイル名	Javaソース	「1.クラス名」が定義されるJavaソースのファイル名を出力します。Javaソース内に複数のクラスが定義される場合、Javaソース上の先頭クラスにのみ出力され、その他のクラスには出力されません。

(2)メソッド単位の計測を指定した場合

No	基本計測	情報取得元	内容
1	メソッド名	Javaソース	計測単位となるメソッド名を出力します。
2	文数	Javaソース	メソッド内の実行文の数を計測します。
3	分岐文数	Javaソース	メソッド内に定義される条件分岐文 (if文/switch内のcase文) の数を計測します。
4	ループ数	Javaソース	メソッド内に定義されるループ文 (for文/while文/do-while文) の数を計測します。
5	メソッドCALL数	Javaソース	他メソッドの実行数を計測します。
6	定義ファイル名	Javaソース	「1.メソッド名」が定義されるJavaソースのファイル名を出力します。Javaソース内に複数のメソッドが定義される場合、Javaソース上の先頭メソッドにのみ出力され、その他のメソッドには出力されません。

第3章 導入手順

MF-STEP COUNTER Javaソフトウェアメトリクス計測機能の導入手順について説明します。

3.1 インストール/アンインストール

MF-STEP COUNTER Javaソフトウェアメトリクス計測機能のインストール/アンインストールについては、製品に付属するソフトウェア説明書を参照してください。

3.2 動作環境

動作環境については以下の通りとなります。

ハードウェア要件

操作マシン
Windows 10、Windows 8.1、Windows 7、Windows Server 2016、Windows Server 2012 R2、Windows Server 2012、またはWindows Server 2008 R2が動作するマシン

ソフトウェア要件

必須ソフトウェア		
ソフトウェア名称	適用要件(バージョン/レベル)	備考
OS	Windows 10 Windows 8.1 Windows 7 Windows Server 2016 Windows Server 2012 R2 Windows Server 2012 Windows Server 2008 R2	左記のいずれかのOSが必要です。
JDK	8.0	JDKのbinフォルダが環境変数PATHに追加されている必要があります。

第4章 機能説明

本章では計測内容及び、アラーム指定について説明します。

4.1 計測内容説明

クラス単位及び、メソッド単位での計測内容について説明します。

4.1.1 クラス単位の計測

クラス単位の計測を実行した場合、指定の対象資産に含まれるJavaファイルをクラス単位に計測します。Javaファイル計測後、Javaファイルに定義されるクラス名をもとにClassファイルの計測をおこないます。Classファイルは、環境変数CLASSPATHに設定される検索パスと、当ツール内にクラスパス指定された検索パスから自動的に検出をおこない、計測されます。

注意) Classファイルがフォルダに展開されている場合のフォルダ構成や、jarファイルの内のフォルダ構成は、通常のJava実行環境でのclasspath指定と同じく、パッケージと同じ構成になっている必要があります。

計測対象のクラスに、利用者独自またはシステムのクラスライブラリ(例えばservlet.jarなど)が含むクラスを使用/継承している場合、そのパッケージが存在するフォルダ名/Jarファイル名についても、環境変数CLASSPATHか当ツール内のクラスパス指定に検索パスとして設定されている必要があります。

また、ClassファイルはJavaファイルと同期が取れていて、最新状態になっていなければ、正常に計測をおこなうことができません。クラス単位の計測をおこなう前には、Javaファイルがコンパイル済であることを確認してください。

出力項目説明

出力項目	説明
クラス名	計測単位となるクラス名です。「パッケージ名. クラス名」の形式で出力されます。Javaファイルへの定義順で出力されます。Javaファイルに内部クラス・匿名クラスが定義される場合、以下の規則によりクラス名を生成します。 規則1) クラス直下に定義される内部クラスは、 <u>定義元クラス\$内部クラス</u> でクラス名を生成します。 規則2) メソッド内に定義される内部クラスは、 <u>定義元クラス\$連番\$内部クラス</u> でクラス名を生成します。 規則3) 匿名クラスは、 <u>定義元クラス\$連番</u> でクラス名を生成します。 ※連番は、出現順を指し命名後のクラス名の重複を避けるための連番とします。
文数	クラス内の文数です。
継承数	クラスの継承数です。
フィールド変数の数	クラス内に定義されるフィールド変数の数です。
インスタンス変数の数	クラス内に定義されるインスタンス変数の数です。
パブリック変数の数	クラス内に定義されるパブリック変数の数です。
メソッド数	クラス内に定義されるメソッドの数です。
インスタンスメソッド数	クラス内に定義されるインスタンスメソッドの数です。
パブリックメソッド数	クラス内に定義されるパブリックメソッドの数です。
定義ファイル名	クラスが定義されるJavaファイル名です。Javaファイルの先頭に定義されるクラス名にのみ出力されます。

計測基準

計測項目	説明	計測ファイル
文数	実行文の数です。行数ではありません。また、コメント行/ <code>package</code> 文/ <code>import</code> 文/クラス定義文/メソッド定義文/フィールド変数定義文は、含まれません。	Java
継承数	上位クラスを継承する階層数です。	Class
フィールド変数の数	インスタンスとStatic両方のすべてを含む変数の数です。	Class
インスタンス変数の数	Static修飾子のない変数の数です。	Class
パブリック変数の数	<code>public</code> 修飾子される変数の数です。	Class
メソッド数	全メソッド数です。	Class
インスタンスメソッド数	Static修飾子のないメソッド数です。	Class
パブリックメソッド数	<code>public</code> 修飾子されるメソッド数です。	Class

Classファイルの計測に失敗した場合、計測項目には全角ハイフン(-)が出力されます。

計測基準の判定例を以下に示します。

Javaファイルの記述	判定
<code>package TEST. Package;</code>	
<code>class TEST01 extends TEST00 {</code>	→ TEST00が最上位クラスの場合、継承数は2
<code>int nDec;</code>	→ インスタンス変数/フィールド変数
<code>static public String str = "TEST01";</code>	→ パブリック変数/フィールド変数
<code>public TEST01() {</code>	
<code> nDec = 0;</code>	→ 文数
<code>}</code>	
<code>int getDec() {</code>	→ インスタンスメソッド数/メソッド数
<code> return nDec;</code>	→ 文数
<code>}</code>	
<code>static public void printStr() {</code>	→ パブリックメソッド数/メソッド数
<code> System.out.println(str);</code>	→ 文数
<code>}</code>	
<code>private String toDecString() {</code>	→ インスタンスメソッド数/メソッド数
<code> return Integer(nDec).toString();</code>	→ 文数
<code>}</code>	
<code>};</code>	

4.1.2 メソッド単位の計測

メソッド単位の計測を実行した場合、指定の対象資産に含まれるJavaファイルをメソッド単位の計測に計測します。当計測では、Classファイルの計測はおこなわれません。

出力項目説明

出力項目	説明
メソッド名	計測単位となるメソッド名です。「パッケージ名. クラス名. メソッド名(メソッドの引数)」の形式で出力されます。
文数	メソッド内の文数です。この計測項目は、行数を指すものではありません。
分岐文数	メソッド内の条件分岐文の数です。if文(else if文)/switch文内のcase文の数、または起動オプションで-tオプションを指定した場合は、これらのAnd/Or条件の数です。単独のelse文とswitch文内のdefault文は、対象外とします。
ループ数	メソッド内のループ文の数です。for文/while文/do~while文の数、またはこれらのAnd/Or条件の数です。
メソッドCALL数	他メソッドの実行数です。

出力項目	説明
定義ファイル名	メソッドが定義される計測対象となったJavaファイル名です。Javaファイルの先頭に定義されるメソッド名にのみ出力されます。

計測基準

計測項目	説明	計測ファイル
文数	実行文の数です。行数ではありません。また、コメント行は含まれません。	Java
分岐文数	if文 (else if文) / switch文内のcase文の数、または起動オプションで-tオプションを指定した場合は、if文内のAnd/Or条件の数です。単独のelse文とswitch文内のdefault文は、対象外とします。	Java
ループ数	for文 / while文 / do~while文の数、またはこれらのAnd/Or条件の数です。	Java
メソッドCALL数	他メソッドの実行数です。	Java

計測基準の判定例を以下に示します。

Javaファイルの記述	判定
<pre> package TEST.Package; class TEST01 extends TEST00 { int nDec; static public String str = "TEST01"; static public void printStr() { String strOut = ""; for(int i=0; i<10; i++) { if(i==0 && nDec > 0) { strOut += "CLASS"; } else { strOut += str; } } print(strOut); } static private void print(String str) { System.out.println(str); } }; </pre>	<p>→ 文数</p> <p>→ ループ数 / 文数</p> <p>→ 分岐文数 (And/Or条件計測する場合は2、しない場合は1) / 文数</p> <p>→ 文数</p> <p>→ 文数</p> <p>→ メソッドCALL数 / 文数</p> <p>→ 文数</p>

4.2 アラーム指定

測定値の判定、基準値の設定例、パーセンタイル形式の表示方法について説明します。

4.2.1 測定値の判定

概要

当ツールでの計測内容は、クラスやメソッドの文数により目標値と実際の開発量から、開発進捗状況を把握することができます。その他に、クラスやメソッドの設計品質の判定に活用することができます。判定内容を以下に示します。

判定内容

1. メソッド数が大きすぎる場合

メソッド数が大きすぎる場合、クラスに機能が詰め込まれ過ぎている恐れがあります。全体としてメソッド数が大きい場合は、内容の理解が難しくなり、保守や再利用がしづらくなる可能性もあります。この場合、クラスの単位の考え方を見直し、適切な単位に細分でき

ないか検討する必要があります。また、一部に極端に大きなクラスやクラス群(パッケージ)があつて平均値を大きくしている場合は、他の部分との機能分担を見直す必要があると考えられます。

2. 変数の数が大きすぎる場合

基本的にはメソッド数の場合と同様の問題があります。但し、定数フィールドについては、多少大きくても問題にする必要はありません。また、画面やDBエンティティなどのクラスは、他の値よりもかなり大きくなります。

3. 文数が大きすぎる場合

上記同様、クラスに機能が詰め込まれ過ぎている可能性があります。特に、メソッド数に比較して、文数が大きいようであれば、手続き中心のプログラムになっている可能性が高いと考えられます。この場合、オブジェクトの持つ機能として、メソッドが適切に分割されているか考え直す必要があります。特定の大きな処理のみをメソッドとしていたのでは、部品としては使いづらく再利用性が低くなります。このような傾向が全体に見られる場合は、大きなレベルの設計/モデリングから見直す必要があります。

4. メソッド数・フィールド変数の数・文数が小さすぎる場合

クラスが小さ過ぎれば、部品としての価値は低くなり、再利用性は低下します。また、分割によるオーバーヘッドが大きくなり、実行性能の点でも好ましくありません。

5. 継承数が深過ぎる場合

継承の使い過ぎは、プログラムの可読性を低下させ、保守を困難にします。適度な深さになるように設計を見直すべきです。継承を多用した場合、上位のクラスの変更が困難になることもあり、委譲などによって機能を分割することも考える必要があります。但し、継承階層の深いシステムクラスを継承して用いている場合は、ユーザ定義の階層部分を判定対象とします。

6. 継承がほとんど使われていない場合

継承がほとんど使われていない場合、開発者がオブジェクト指向の考えを理解できていない可能性を示しています。クラスサイズの情報と合わせて、機能をサブクラス化できるような箇所がないか検討する必要があります。

4.2.2 アラーム指定基準値設定例

概要

計測結果表示画面でのアラーム指定基準値の設定例を、ビジネスソフトの事例をもとに以下に示します。

基準値

Javaプログラムのアラーム指定基準値の検討のために、以下のようなビジネスソフトの事例について測定をおこないました。

- **A** ソフト開発発注の管理システム(A社殿)
ファイル数 1132, 行数 約318K, クラス数 1177
- **B** 部品表システム(B社殿, クライアントのみ)
ファイル数 709, 行数 約59K, クラス数 799
これはフェーズ1とフェーズ2に分かれている。
- **C** 営業支援システム(C社殿)
ファイル数 132, 行数 約36K, クラス数 186
- **D** システム(D社殿, クライアントのみ)
ファイル数 44, 行数 約8K, クラス数 44
- **E** 製造業向け設計業務 部門間ワークフロー(E社殿)
ファイル数 136, 行数 約12K, クラス数 161

その結果を、事務処理ソフトの場合に大きな特徴を示すと思われる画面(Panel)クラス、エンティティクラス(DBテーブルに相当)と、それぞれ他のクラスと分類し集計しました。分類は、パッケージ及びクラスの名称をもとにおこないました。一般には、分布の特徴を見るには標準偏差がよく用いられますが、当ツールの測定で得られるデータは、通常は正規分布でないため、有効ではありません。ここでは、四分位偏差をもとに分布の性質を調べ、基準平均値と上限値を設定しました。Javaの基準値を下表に示します。

表4.1 Javaの基準値

項目	画面(平均,上限)	DBエンティティ(平均,上限)	一般(平均,上限)
フィールド数	15 - 50,100	20,80	3 - 10,20
メソッド数	10 - 35,80	40,160	5 - 10,30
文の数	100 - 400,700	100,300	25 - 80,300
非private非定数フィールド数	0	0	0

・ 平均値

一定の単位の平均値の基準であり、プログラム全体やパッケージ単位の特徴を調べる場合に用います。これを大きく外れていれば、その部分の設計に問題がある可能性があります。

・ 上限値

クラスや関数が異常であるかどうかを判定するための目安です。これを越えるクラスや関数は、構造に問題があると考えられるため、見直した方が良いでしょう。なお、これらの基準は、おおよその目安として設定したものであり、絶対的な基準ではありません。実際の適用には、プロジェクトの必要に応じて、基準を設定することが望ましいでしょう。例えば、通常のアプリケーションを設計する場合と再利用を強く意識し汎用性を高めたライブラリを設計する場合とでは、基準は当然異なってきます。上記の基準は、通常のアプリケーションを意識したものです。再利用ライブラリの場合は、メソッド数等がより大きくなる傾向があります。

以下に分類毎の特徴を述べます。

画面クラスは、全体にサイズが大きくなる傾向があります。画面設計上の要件や使用するGUIビルダが生成するコードの特徴など種々の制約はありますが、プログラムの理解性を保つためには、上記の基準程度にとどめておくとい良いでしょう。

DBエンティティクラスは、DBテーブルのサイズをもとにするので、フィールド数やメソッド数は大きくなります。通常は、データアクセスのメソッドが中心になるので、文の数は比較的小さくできます。これについて、基準は暫定値であり、はっきりしたことが言えない点が多々あります。

その他のクラスについての基準値が、一般的なJavaのクラス設計基準と考えています。

参考として、C++での基準値を以下に示します。

表4.2 C++の基準値

項目	基準平均	上限
データメンバの数(=フィールド数)	2 - 4	10
メソッド関数の数(=メソッド数)	3 - 13	30
文の数	20 - 120	500

分類毎のフィールド数/メソッド数/文数の集計結果を以下に示します。

1. 画面

画面クラスについては、比較的規模の大きいA、Bの結果を主に採用し、基準値を設定しました。

表4.3 A(クラス数 131, インタフェース数 1)

測定項目	合計値	平均値	最大値	最小値	標準偏差	中央値	四分位偏差	飛び離れ値
フィールド数	6503	49.27	253	0	59.90	29.0	57	235
メソッド数	3413	25.86	120	2	21.45	19.5	23	104
文数	62749	475.37	2897	0	521.22	307.0	609	2534

表4.4 B(Phase1)(クラス数 50,インタフェース数 0)

測定項目	合計値	平均値	最大値	最小値	標準偏差	中央値	四分位偏差	飛び離れ値
フィールド数	715	14.30	46	0	14.90	11.5	26	104
メソッド数	524	10.48	24	1	6.69	11.0	12	52
文の数	4958	99.16	309	3	89.21	78.5	146	600

表4.5 B(Phase2)(クラス数 7,インタフェース数 0)

測定項目	合計値	平均値	最大値	最小値	標準偏差	中央値	四分位偏差	飛び離れ値
フィールド数	197	28.14	48	14	11.95	29.0	22	103
メソッド数	157	22.43	33	16	7.61	17.0	15	76
文の数	2034	290.57	463	162	97.00	282.0	119	710

2. DBエンティティ

データベースエンティティについては、方式によって様々なバラツキがありました。データベースエンティティは、AのXXXEntityImplという名をもつクラスの値をベースに基準値を設定しました。

表4.6 A(クラス数 275)

測定項目	合計値	平均値	最大値	最小値	標準偏差	中央値	四分位偏差	飛び離れ値
フィールド数	5657	20.57	164	1	26.86	10.0	18	78
メソッド数	12695	46.16	334	4	53.58	24.0	35	157
文の数	24723	89.90	682	5	110.94	48.0	64	286

3. その他

その他一般のクラスについては、これらの事例に加えて、Java標準ライブラリの測定結果、他の2つのミドルウェアソフトでの測定結果、及びC++について定めた値を参考にして基準を設定しました。

表4.7 A(クラス数 238,インタフェース数 11)

測定項目	合計値	平均値	最大値	最小値	標準偏差	中央値	四分位偏差	飛び離れ値
フィールド数	2169	7.66	254	0	21.10	3.0	6	25
メソッド数	2543	8.99	78	0	9.78	7.0	7	31
文の数	17558	62.04	996	0	103.80	33.0	39	173

表4.8 B(Phase1)(クラス数 646,インタフェース数 23)

測定項目	合計値	平均値	最大値	最小値	標準偏差	中央値	四分位偏差	飛び離れ値
フィールド数	4301	6.66	145	0	11.05	3.0	7	29

メソッド数	6503	10.0 7	138	0	11.82	7.0	10	43
文の数	5035 9	77.9 6	1319	0	143.67	33.5	72	300

表4.9 B(Phase2)(クラス数 96,インタフェース数 0)

測定項目	合計値	平均値	最大値	最小値	標準偏差	中央値	四分位偏差	飛び離れ値
フィールド数	699	7.28	99	0	13.54	2.0	8	33
メソッド数	843	8.78	116	0	16.00	5.0	9	37
文の数	7863	81.9 1	1496	0	181.70	31.0	84	340

表4.10 C(クラス数 74,インタフェース数 0)

測定項目	合計値	平均値	最大値	最小値	標準偏差	中央値	四分位偏差	飛び離れ値
フィールド数	243	3.28	66	0	9.12	1.0	0	1
メソッド数	507	6.85	137	0	18.60	3.0	2	10
文の数	1958	26.4 6	469	0	79.37	3.0	2	11

表4.11 D(クラス数 30,インタフェース数 3)

測定項目	合計値	平均値	最大値	最小値	標準偏差	中央値	四分位偏差	飛び離れ値
フィールド数	295	9.83	64	0	14.24	4.0	12	48
メソッド数	190	6.33	31	0	7.20	4.0	7	29
文の数	2402	80.0 7	398	0	97.57	49.0	123	500

表4.12 E(クラス数 125,インタフェース数 8)

測定項目	合計値	平均値	最大値	最小値	標準偏差	中央値	四分位偏差	飛び離れ値
フィールド数	802	6.42	53	0	10.15	3.0	6	25
メソッド数	562	4.50	18	0	3.63	4.0	4	18
文の数	3352	26.8 2	145	0	31.43	18.0	35	144

4.2.3 パーセンタイル形式のアラーム表示

概要

計測結果表示画面で表示される計測値について、パーセンタイル形式によるアラーム表示を行います。

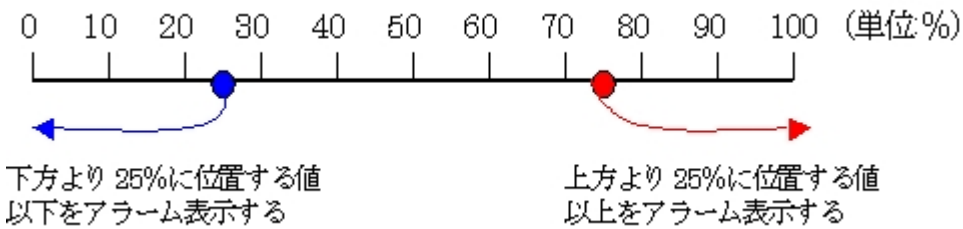
表示説明

パーセンタイルは、計測したデータの分布を大きさの順に並べて、何パーセント目の値がどれくらいかを評価する統計的な表示法のひとつです。

パーセンタイル形式によるアラーム表示では、アラーム指定ダイアログボックスで指定された下限のパーセント点より、計測クラスのpパーセンタイル(下限) / 100-pパーセンタイル(上限)を算出し、下限以下 / 上限以上の値について、アラーム表示を行います。

以下にパーセンタイルによるアラーム表示範囲の例を示します。

アラーム表示範囲の例) 25パーセンタイル

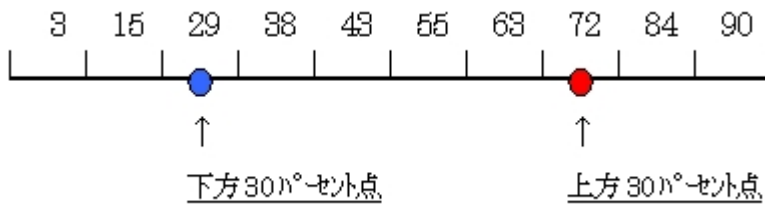


以下にパーセンタイルによるアラーム表示の例を示します。

例) クラスが10個あって、それぞれの文数が以下のようにだったとします。パーセント点=30(下限) 指定時のパーセンタイル(上限/下限)を求め、アラームを表示します。

[各クラスの文数] 15,63,84,29,43,55,90,3,72,38

- 上記データを大きさの順に並べ替えます
3,15,29,38,43,55,63,72,84,90
- パーセンタイル(上限/下限)を求めます
下限は、10クラス中の3番目 … 文数=29
上限は、10クラス中の8番目 … 文数=72

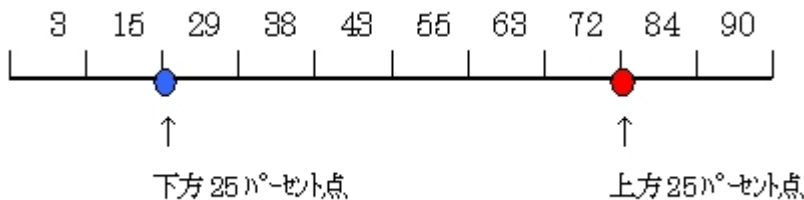


- アラームを表示します
文数が29(下限)以下を青色でアラーム表示します。
文数が72(上限)以上を赤色でアラーム表示します。

[各クラスの文数] 15,63,84,29,43,55,90,3,72,38

例) 上記例にて、パーセント点=25(下限) 指定時のパーセンタイル(上限/下限)を求め、アラームを表示します。

- パーセンタイル(上限/下限)を求めます
下限は、10クラス中の2番目(15)と3番目(29)の間 … 文数=22
上限は、10クラス中の8番目(72)と9番目(84)の間 … 文数=78



- アラームを表示します
文数が22(下限)以下を青色でアラーム表示します。
文数が78(上限)以上を赤色でアラーム表示します。

[各クラスの文数] 15,63,84,29,43,55,90,3,72,38

例) 重複するデータが存在する場合のパーセント点=30(下限)を求め、アラームを表示します。

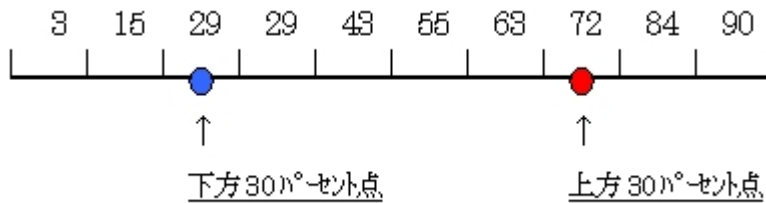
[各クラスの文数]3,15,29,29,43,55,63,72,84,90

- 上記の重複する29をそれぞれカウントし、全件で10件として割合を求めます

- パーセンタイル(上限/下限)を求めます

下限は、10クラス中の3番目 … 文数=29

上限は、10クラス中の8番目 … 文数=72



- アラームを表示します

文数が29(下限)以下を青色でアラーム表示します。

文数が72(上限)以上を赤色でアラーム表示します。

[各クラスの文数]3,15,29,29,43,55,63,72,84,90

第5章 操作説明

MF-STEP COUNTER Javaソフトウェアメトリクス計測機能の操作方法について説明します。

5.1 起動/終了方法

当ツールの起動/終了は、以下の手順で実施してください。

起動方法

1. 当ツールのインストールフォルダに格納されているmfstpjm.jar(JDK8.0版)ファイルを環境変数CLASSPATHに追加します。

```
追加例 (JDK8.0版) :  
set CLASSPATH= [インストールフォルダ] ¥mfstpjm.jar;%CLASSPATH%
```

2. 当ツールのインストールフォルダをカレントフォルダとし、java.exeにより当ツールの起動クラスを実行します。当ツールの起動クラス名は、com.fujitsu.simplia.StepCounter.metricsです。起動に必要なパラメタがある場合は、起動時に合わせて指定してください。パラメタの内容は、「5.4 コマンドライン」を参照してください。

```
起動例 :  
cd [インストールフォルダ]  
java com.fujitsu.simplia.StepCounter.metrics [パラメタ]
```

注) 当ツールの実行前に事前にJDKのbinフォルダを環境変数PATHに追加する必要があります。

終了方法

当ツールを終了するときは、メニューバーから[ファイル(F)]を選択し、プルダウンメニュー内の[SIMPLIA MF-STEP COUNTERの終了(X)]を選択します。

5.2 画面処理

画面処理での各計測方法について説明します。

5.2.1 新規計測対象一覧ファイル指定方法

概要

新規に計測する計測対象への各種設定を行います。

操作手順

1. メイン画面の[ファイル]メニューの[新規作成]コマンドを選択します。
2. 各種設定(※)を設定します。
3. メイン画面の[ファイル]メニューの[名前を付けて保存]コマンドを選択し、計測対象一覧ファイルを保存します。

※設定内容は以下のとおりです。

設定画面	設定項目	内容
メイン画面	CSVファイル	計測結果の出力先ファイル名を指定します。
	対象資産形式	計測対象の形式を選択します。
	対象資産名	計測対象となるファイルまたはフォルダを指定します。
クラスパス指定ダイアログ	クラスパス名	Classファイルの検索パスを指定します。

設定画面	設定項目	内容
		(クラス単位の計測をおこなう場合、Classファイルが存在するパス名が環境変数CLASSPATHに設定されていない場合、設定します。メソッド単位の計測をおこなう場合、設定の必要はありません。)
オプション指定ダイアログ	計測単位	計測単位を選択します。 (初期状態ではクラス単位が選択されています。)
アラーム指定ダイアログ	アラーム表示内容	計測結果を表示した場合のアラーム表示内容を設定します。 (初期状態ではパーセンタイル形式、閾値25%が設定されています。)

5.2.2 対象資産の指定方法

概要

計測対象となる対象資産名を指定します。

操作手順

1. メイン画面の[対象資産形式]コンボボックスから計測対象となる資産の指定形式を選択します。
2. メイン画面の[対象資産名]横の[参照]ボタンの選択により、ファイル選択ダイアログボックスが表示されます。ファイル選択ダイアログボックスから対象資産名を選択します。対象資産は、1.で指定した対象資産形式の内容により、以下の形式で選択されます。

[対象資産形式]の選択内容	[対象資産名]横の[参照]ボタンでの選択内容
ファイル形式	計測対象となるJavaファイル名を選択します。ファイル選択ダイアログボックスでは、計測対象ファイルを直接選択します。
フォルダ形式 (サブフォルダを含まない)	計測対象となるフォルダ名を選択します。フォルダ選択ダイアログボックスでは、計測対象フォルダを直接選択します。
フォルダ形式 (サブフォルダを含む)	同上

3. メイン画面の[追加]ボタンを選択し、2.で選択した対象資産名を[対象資産一覧]リストに追加します。[対象資産一覧]リストに追加されない対象資産は、計測の対象とはなりません。

5.2.3 計測対象一覧ファイルの読み込み方法

概要

作成済の計測対象一覧ファイルを読み込みます。

操作手順

1. メイン画面の[ファイル]メニューの[計測対象一覧ファイルを開く]コマンドを選択します。
2. 表示されるファイル選択ダイアログボックスから、必要な計測対象一覧ファイルを選択します。

※旧ツールバージョン(V50L10)にて作成された計測対象一覧ファイルが選択された場合、[上限値/下限値]形式によるアラーム表示を行います。

5.2.4 クラスパスの指定方法

概要

当ツールでは、クラス単位の計測時に対象資産として指定されるJavaファイルからClass名を特定し、Classファイルを検索し計測します。このため、Classファイルの検索パス(フォルダ名/Jarファイル名)をクラスパスとして指定します。

環境変数CLASSPATHが存在する場合、環境変数CLASSPATHへの指定パスも検索対象となるため、環境変数CLASSPATHの内容は、クラスパスとして指定する必要はありません。また、環境変数CLASSPATHに必要な検索パスがすべて設定されている場合や、メソッド単位の計測をおこなう場合は、当パラメタの指定は必要ありません。

Classファイルは、環境変数CLASSPATHへの指定順、クラスパス指定での指定順で検索されます。

注意) Classファイルがフォルダに展開されている場合のフォルダ構成や、Jarファイルの内のフォルダ構成は、通常のJava実行環境でのclasspath指定と同じく、パッケージと同じ構成になっている必要があります。

注意) ClassファイルまたはJarファイルへのフルパスの長さが、OSの制限以上になっている場合、参照できないためにエラーが発生することがあります。

操作手順

1. メイン画面の[オプション]メニューの[クラスパス指定]コマンドを選択します。
2. クラスパス指定ダイアログボックスで、クラスパスとなるフォルダ名・Jarファイル名の入力を行います。[ファイル参照]ボタン・[フォルダ参照]ボタンを選択することにより、ファイル選択ダイアログ・フォルダ選択ダイアログから選択することも可能です。

a.ファイル選択ダイアログ

クラスパスとなるJarファイル名を選択します。選択されたJarファイル名が[クラスパス名]テキストフィールドに設定されます。

b.フォルダ選択ダイアログ

クラスパスとなるフォルダ名を選択します。選択されたフォルダ名が[クラスパス名]テキストフィールドに設定されます。

例1) クラス名がaaa.class、パッケージがcom.xxx.yyyの場合で、以下のフォルダ構成の場合

```
c:\classes\com\xxx\yyy\aaa.class → フォルダ選択でc:\classesを指定します
```

例2) aaa.classがjarファイルに圧縮されて以下のように配置されている場合

```
c:\classes\jar\abc.jar → ファイル選択でc:\classes\jar\abc.jarを指定します
```

例3) aaa.classがcom.xxx.bbbクラスを継承している場合

```
c:\classlib\com\xxx\bbb.class → フォルダ選択でc:\classlibを追加指定します
```

3. クラスパス指定ダイアログボックスで、[追加]ボタンを選択し、2.で選択したクラスパス名を、[クラスパス一覧]リストに追加します。[クラスパス一覧]リストに追加されない場合、検索パスとして設定されません。
4. 必要なクラスパスを設定後、クラスパス指定ダイアログボックスの[OK]ボタンを選択し、指定内容を保存し、ダイアログボックスを終了します。

5.2.5 計測単位の指定方法

概要

指定対象資産の計測単位を指定します。

操作手順

1. メイン画面の[オプション]メニューの[計測オプション]コマンドを選択します。
2. 表示されるオプション指定ダイアログボックスから、計測単位(クラス単位/メソッド単位)を選択します。[And/Or条件の計測]オプションも選択できます。
3. オプション指定ダイアログボックスの[OK]ボタンを選択し、指定内容を保存しダイアログボックスを終了します。

5.2.6 アラーム表示の指定方法

概要

計測結果表示画面で表示される計測値のアラーム表示指定をおこないます。

操作手順

1. メイン画面の[オプション]メニューの[アラーム指定]コマンドまたは、計測結果表示画面の[オプション]メニューの[アラーム指定]コマンドを選択します。
2. [パーセンタイル]ラジオボタン・[上限値/下限値]ラジオボタンより、アラーム表示形式を選択します。
3. アラーム表示の対象項目を設定します。
 - a.パーセンタイル選択時
すべての計測項目がアラーム表示対象となります。対象項目の設定はありません。
 - b.上限値/下限値選択時
アラーム指定ダイアログボックスの[計測項目]コンボボックスから、アラーム指定の対象となる計測項目を選択します。
4. アラーム範囲を設定します。
 - a.パーセンタイル選択時
アラーム指定ダイアログボックスの[閾値]のテキストフィールドに、アラーム範囲(下限の閾値※)を設定します。計測結果表示画面でのアラーム表示は、指定された閾値より項目毎のパーセント点(上限/下限)を算出し、パーセント点(上限)以上の計測値、および、パーセント点(下限)以下の計測値に対しておこなわれます。詳細については、「4.2.3 パーセンタイル形式のアラーム表示」を参照してください。
 - b.上限値/下限値選択時
アラーム指定ダイアログボックスの[上限値][下限値]のテキストフィールドに、アラーム範囲を設定します。計測結果表示画面でのアラーム表示は、上限値以上の計測値、または下限値以下の計測値に対しておこなわれます。
5. アラーム指定ダイアログボックスの[OK]ボタンを選択し、指定内容を保存しダイアログボックスを終了します。

※下限の閾値は、0%～50%の範囲内で設定します。

5.2.7 計測結果の表示方法

概要

対象資産の計測結果を表示します。

操作手順

計測と計測結果表示

1. メイン画面の[計測]メニューの[計測]コマンドを選択し、対象資産の計測をおこないます。
2. 計測終了後に、計測結果表示画面が起動され、計測結果が表示されます。

計測済結果表示

メイン画面の[表示]メニューの[計測結果表示]コマンドを選択し、計測結果表示画面を表示します。

※ただし、当操作は、当ツール起動後計測処理を実行した場合に有効です。当ツールの前回起動時の計測済結果を表示することはできません。

また、当操作では、直前の計測処理の結果のみ表示可能です。

5.2.8 計測対象一覧ファイルの保存方法

概要

計測対象一覧ファイルを保存します。

操作手順

新規作成及びファイル名変更

1. メイン画面の[ファイル]メニューの[名前を付けて保存]コマンドを選択します。

2. 表示されるファイル選択ダイアログボックスから、保存先ファイル名を指定し[OK]ボタンを選択します。

上書き保存

メイン画面の[ファイル]メニューの[上書き保存]コマンドを選択します。

5.2.9 計測方法

概要

指定の対象資産の計測をおこないます。

操作手順

メイン画面の[計測]メニューの[計測]コマンドを選択します。

計測結果は、メイン画面で計測前に指定するCSVファイルに出力されます。

5.3 バッチ処理

バッチ処理での各計測方法について説明します。

5.3.1 ファイル指定による計測方法

概要

コマンドライン引数の指定で、指定のJavaファイル/Classファイルをバッチ処理で計測をおこないます。

操作手順

コマンドライン引数に以下の指定をおこない、当ツールを起動します。

```
-b [Javaファイル名]
```

実行例:

c:¥Java¥test.javaを計測対象のJavaファイルとした場合

```
java com.fujitsu.simplia.StepCounter.metrics -b c:¥Java¥test.java
```

※計測結果は標準出力に出力されます。

5.3.2 フォルダ指定による計測方法

概要

コマンドライン引数の指定で、指定フォルダに含まれるJavaファイルと、Javaファイルに定義されるClassファイルを、バッチ処理で計測をおこないます。

操作手順

コマンドライン引数に以下の指定をおこない、当ツールを起動します。

```
-b -d [フォルダ名]
```

実行例:

c:¥Javaを計測対象のフォルダとした場合

```
java com.fujitsu.simplia.StepCounter.metrics -b -d c:¥Java
```

※ 引数内に-sを指定すると、指定フォルダに含まれるサブフォルダも計測対象となります。

5.3.3 計測対象一覧による計測方法

概要

コマンドライン引数の指定で、計測対象一覧ファイルに指定されるJavaファイルと、Javaファイルに定義されるClassファイルを、バッチ処理で計測をおこないます。

操作手順

コマンドライン引数に以下の指定をおこない、当ツールを起動します。

```
-b -l [計測対象一覧ファイル名]
```

実行例:

c:¥stj¥test.stjを対象の計測対象一覧ファイルとした場合

```
java com.fujitsu.simplia.StepCounter.metrics -b -l c:¥stj¥test.stj
```

※ 計測対象一覧ファイルとは、画面処理で指定した対象資産名／クラスパス／オプションを格納した保存ファイルを指します。

5.3.4 計測単位の指定方法

概要

コマンドライン引数の指定で、計測単位を指定し、バッチ処理で計測をおこないます。

操作手順

1. クラス単位で計測をおこなう場合、コマンドライン引数に以下の指定をおこない、当ツールを起動します。計測単位を指定しない場合、クラス単位に計測がおこなわれます。

```
-b [対象資産パラメタ]
```

実行例:

c:¥Java¥test.javaを計測対象のJavaファイルとした場合

```
java com.fujitsu.simplia.StepCounter.metrics -b c:¥Java¥test.java
```

2. メソッド単位に計測をおこなう場合、コマンドライン引数に以下の指定をおこない、当ツールを起動します。

```
-b [対象資産パラメタ] -m
```

実行例:

c:¥Java¥test.javaを計測対象のJavaファイルとした場合

```
java com.fujitsu.simplia.StepCounter.metrics -b c:¥Java¥test.java -m
```

※ 対象資産パラメタは、ファイル指定／フォルダ指定を指します。対象資産パラメタに計測対象一覧ファイルが指定される場合、計測単位の指定はできません。この場合、計測対象一覧ファイルに指定される計測単位が有効になります。

5.3.5 計測結果出力先の指定方法

概要

コマンドライン引数の指定で、計測結果の出力先を指定し、バッチ処理で計測をおこないます。

操作手順

コマンドライン引数に以下の指定をおこない、当ツールを起動します。

```
-b [対象資産パラメタ] -csv [CSVファイル名]
```


実行例:

c:¥Java¥test.javaを計測対象のJavaファイルとし、c:¥csv¥test.csvを出力先のCSVファイルとした場合

```
java com.fujitsu.simplia.StepCounter.metrics -b c:¥Java¥test.java -csv c:¥csv¥test.csv
```

※対象資産パラメタは、ファイル指定/フォルダ指定を指します。対象資産パラメタに計測対象一覧ファイルが指定される場合、出力先の指定はできません。この場合、計測対象一覧ファイルに指定される出力先が有効になります。

※CSVファイル名には、書き込み可能なファイル名を指定します。

※-csvパラメタが指定されない場合、計測結果は標準出力に出力されます。

5.4 コマンドライン

コマンドラインの各引数について説明します。

5.4.1 コマンドライン引数

コマンドライン引数には、以下のパラメタが存在します。各パラメタの指定順は任意とします。

- ・ 起動形態パラメタ
- ・ 対象資産パラメタ
- ・ クラスパスパラメタ
- ・ オプションパラメタ
- ・ CSVパラメタ
- ・ ERRパラメタ

パラメタ間には、排他関係にあるパラメタも存在するため、指定には注意が必要です。

各パラメタの指定方法に記述される記号は、以下の内容を指します。

記号	説明
[]	省略可能です。
{}	複数のパラメタから1つ選択し、指定します。
下線	デフォルト値です。省略時は、デフォルト値が指定されたものとします。

5.4.1.1 起動形態パラメタ

指定方法

```
[[ -b  
-f ]]
```

説明

当ツールの起動方法を指定します。起動形態パラメタ省略時は、-fが指定されたものとみなします。

指定内容	説明
-b	バッチ処理を起動します。当パラメタ指定時は、対象資産パラメタの指定が必要です。
-f	画面処理を起動します。当パラメタ指定時は、対象資産/クラスパス/オプション/CSVパラメタは、指定できません。

5.4.1.2 対象資産パラメタ

指定方法

```
[{-l 計測対象一覧ファイル名
  -d フォルダ名
  ファイル名
}]
```

説明

計測対象となる対象資産を指定します。

指定内容	説明
-l(エル) 計測対象一覧ファイル名	計測対象一覧ファイルに格納される内容を計測対象とします。計測対象一覧ファイルは、対象資産/クラスパス/CSV出力先/計測オプションを格納したファイルです。当パラメタが指定される場合、クラスパス/オプション/CSVパラメタは指定できません。
-d フォルダ名	指定フォルダ名配下に存在する全ファイルを計測対象とします。 (サブフォルダを含めて計測する場合はオプションパラメタに-sを指定します。)
ファイル名	計測対象となるJavaファイル名を指定します。当パラメタには、ワイルドカード指定(*や?を使ったファイル指定)が可能です。ワイルドカード指定時は、該当するJavaファイルの計測をおこないます。

5.4.1.3 クラスパスパラメタ

指定方法

```
[ -c { フォルダ名 } [ ; { フォルダ名 } ] ... ]
    jar ファイル名 jar ファイル名
```

説明

当ツールでは、クラス単位の計測時に対象資産として指定されるJavaファイルからClass名を特定し、Classファイルを検出し計測します。Classファイル検出時の検索パス(フォルダ名またはJarファイル名)を指定します。

指定内容	説明
フォルダ名	Classファイル検索対象となるフォルダ名を指定します。
Jarファイル名	Classファイルが格納されるJarファイル名を指定します。

複数の検索パスを指定する場合、パス名間をセミコロン(;)で区切り、継続して指定します。環境変数CLASSPATHが存在する場合、環境変数CLASSPATHへの指定パスも検索対象となるため、環境変数CLASSPATHの内容は、当パラメタに指定する必要はありません。また、環境変数CLASSPATHに必要な検索パスがすべて設定されている場合や、メソッド単位の計測をおこなう場合は、当パラメタの指定は必要ありません。

Classファイルは、環境変数CLASSPATHへの指定順、当パラメタでの指定順で検索されます。

5.4.1.4 オプションパラメタ

指定方法

```
[ -m [ -t ] ] [ -s ]
```

説明

計測オプションを指定します。

指定内容	説明
-m	メソッド単位の計測を指定します。当パラメタ省略時は、クラス単位の計測を実行します。当パラメタが指定される場合、クラスパスパラメタの指定を無効とします。

-t	if文などでAnd/Or条件ごとに分岐数を計測します。当パラメタ省略時は、従来通り複数の条件があっても1つとして計測されます。このパラメタの指定は-mパラメタの指定が必要です。
-s	当パラメタは、対象資産パラメタにフォルダ名が指定される場合に有効となります。当パラメタが指定される場合、対象資産(フォルダ)のサブフォルダに含まれるJavaファイルも計測対象とします。対象資産パラメタにファイル名が指定される場合、当パラメタの指定を無効とします。

5.4.1.5 CSVパラメタ

指定方法

[-csv CSVファイル名]

説明

計測結果の出力先を指定します。

指定内容	説明
CSVファイル名	計測結果の出力先となるCSVファイル名を指定します。当パラメタ省略時は、計測結果を標準出力に出力します。

5.4.1.6 ERRパラメタ

指定方法

[-efile]

説明

計測エラーの出力先を指定します。

指定内容	説明
-efile	当パラメタ指定時は、計測中に発生するエラーをエラーファイルに出力します。エラーファイルは、当ツールのインストールフォルダ配下のTempフォルダに「mfstpjm.err」のファイル名で出力されます。当パラメタを省略した場合、計測中に発生したエラーは、標準出力に出力されます。また、当パラメタを指定した場合でも、エラーファイルへの出力異常を検出した場合は、計測中に発生したエラーを、標準出力に出力します。

5.4.2 パラメタ排他関係

各パラメタの排他関係を以下に示します。

パラメタ		ERR	CSV	オプション		クラスパス	対象資産			起動形態	
		-efile	-csv ~	-s	-m	-c ~	ファイル名	-d ~	-l ~	-f	-b
起動形態	-b	○	○	○	○	○	● (どれか1つ)			×	×
	-f	○	×	×	×	×	×	×	×	×	×
対象資産	-l ~	○	×	×	×	×	×	×	×	-	-
	-d ~	○	○	○	○	○	×	×	-	-	-

パラメタ		ERR	CS V	オプション		クラスパス	対象資産			起動形態	
		-efile	-csv ~	-s	-m	-c ~	ファイ ル名	-d ~	-l ~	-f	-b
	ファイル名	○	○	△	○	○	×	-	-	-	-
クラスパス	-c ~	○	○	○	○	×	-	-	-	-	-
オプション	-m	○	○	○	×	△	-	-	-	-	-
	-s	○	○	×	-	-	-	-	-	-	-
CSV	-csv ~	○	×	-	-	-	-	-	-	-	-
ERR	-efile	×	-	-	-	-	-	-	-	-	-

●:指定必須、○:指定可、△:指定無効、×:指定不可

第6章 画面説明

MF-STEP COUNTER Javaソフトウェアメトリクス計測機能の画面について説明します。

6.1 メイン画面

レイアウト



説明

計測処理に必要な以下の項目の指定をおこないます。

必須指定内容	説明
CSVファイル	計測結果の出力先ファイル名を指定します。
対象資産一覧	計測対象となる資産(フォルダ/Javaファイル)名を指定します。

メニューコマンド説明

表6.1 (1)ファイルメニュー

メニューアイテム	説明
新規作成(N)	新しく計測をおこない、新規の計測対象一覧ファイルを作成します。
計測対象一覧ファイルを開く...(O)	既存の計測対象一覧ファイルを読み込みます。
上書き保存(S)	計測対象一覧ファイルの内容を更新して保存します。
名前を付けて保存...(A)	計測対象一覧ファイルの名前を指定して保存します。
SIMPLIA MF-STEP COUNTERの終了(X)	当ツールを終了します。

表6.2 (2)計測メニュー

メニューアイテム	説明
計測(K)	計測処理を実行し、計測結果を表示します。

表6.3 (3)表示メニュー

メニューアイテム	説明
計測結果表示(D)	計測結果を画面表示します。

表6.4 (4)オプションメニュー

メニューアイテム	説明
クラスパス指定...(P)	Classファイルの検索パスを指定します。
計測オプション指定...(I)	計測時のオプションを指定します。
アラーム指定...(L)	計測結果のアラーム表示内容を指定します。

表6.5 (5)ヘルプメニュー

メニューアイテム	説明
MF-STEP COUNTER ヘルプ(T)	ヘルプ画面を表示します。
バージョン情報...(A)	当ツールのバージョンレベルを表示します。

画面コントロール説明

画面コントロール	説明
[CSVファイル]テキストフィールド	計測結果の出力先CSVファイル名を入力します。[CSVファイル参照]ボタンにより、ファイルの選択が可能です。
[CSVファイル参照]ボタン	計測結果の出力先CSVファイル名を選択します。
[対象資産形式]コンボボックス	計測対象資産の資産形式を選択します。
[対象資産名]テキストフィールド	計測対象となる資産名を入力します。対象資産形式によっては、フォルダの選択も可能です。
[対象資産参照]ボタン	計測対象となる対象資産名を選択します。選択内容は、[対象資産形式]コンボボックスでの資産形式の選択内容により異なります。詳細は、「 5.2.2 対象資産の指定方法 」を参照してください。
[対象資産一覧]リスト	選択済の対象資産名を表示します。
[追加]ボタン	[対象資産名]テキストフィールドに指定された内容を、[対象資産一覧]リストの末尾に追加します。
[削除]ボタン	[対象資産一覧]リストの選択資産を、[対象資産一覧]リストから削除します。
[全て削除]ボタン	[対象資産一覧]リストに表示される全資産を、[対象資産一覧]リストから削除します。
[計測]ボタン	計測処理を実行し、計測結果を表示します。

6.2 計測中断ダイアログボックス

レイアウト



説明

計測状況を表示します。

画面コントロール説明

画面コントロール	説明
[計測率]ラベル	メイン画面で指定する対象資産の計測率を表示します。
[ファイル名]ラベル	計測中のJavaファイル名を表示します。
[計測状況]キャンパス	計測状況が描画されます。
[中断]ボタン	計測中のJavaファイルの計測終了後に処理を中断します。

6.3 計測結果表示画面

レイアウト

The screenshot shows two windows from the SIMPLIA MF-STEP-COUNTER application. The top window displays a summary table of measurement results for various classes. The bottom window displays a detailed table of method-level measurement results.

クラス名	文数	継承数	フィールド変数の数	インスタンス変数の数	パブリック変数の数	メソッド数	インスタンスメソッド数	パブリックメソッド数	定義ファイル名
Sample2	6	1	1	1	0	1	1	1	1C:\Program Files (...)
Sample22	16	1	2	2	0	2	2	2	2
Sample23	27	1	3	3	0	3	3	3	3
Sample24	42	1	4	4	0	4	4	4	4
Sample25	60	1	5	5	0	5	5	5	5
Sample26	17	1	6	6	0	1	1	1	1
Sample27	31	1	7	7	0	2	2	2	2
Sample28	50	1	8	8	0	3	3	3	3
Sample29	75	1	9	9	0	4	4	4	4
Sample210	77	1	10	10	0	5	5	5	5

メソッド名	文数	分岐文数	ループ数	メソッドCALL数	定義ファイル名
Sample2.Sample2()	1	0	0	0	0C:\Program Files (\x86)\SIMPLIA\MF-STEP-COUNTER\SAMPLE\Sample2.java
Sample2.method1()	5	0	2	1	
Sample22.Sample22()	2	0	0	0	
Sample22.method1()	6	0	2	2	
Sample22.method2()	8	0	2	2	
Sample23.Sample23()	3	0	0	0	
Sample23.method1()	7	0	2	3	
Sample23.method2()	8	0	2	3	
Sample23.method3()	9	0	2	3	
Sample24.Sample24()	4	0	0	0	
Sample24.method1()	8	0	2	4	
Sample24.method2()	9	0	2	4	
Sample24.method3()	10	0	2	4	
Sample24.method4()	11	0	2	4	
Sample25.Sample25()	5	0	0	0	
Sample25.method1()	9	0	2	5	
Sample25.method2()	10	0	2	5	
Sample25.method3()	11	0	2	5	
Sample25.method4()	12	0	2	5	
Sample25.method5()	13	0	2	5	
Sample26.Sample26()	6	0	0	0	
Sample26.method1(int)	11	1	2	3	
Sample27.Sample27()	7	0	0	0	
Sample27.method1(int)	11	1	2	4	
Sample27.method2(int)	13	2	2	4	
Sample28.Sample28()	8	0	0	0	
Sample28.method1(int)	12	1	2	5	
Sample28.method2(int)	14	2	2	5	
Sample28.method3(int)	16	3	2	5	
Sample29.Sample29()	9	0	0	0	
Sample29.method1(int)	13	1	2	6	
Sample29.method2(int)	15	2	2	6	
Sample29.method3(int)	17	3	2	6	
Sample29.method4(int)	21	5	2	6	
Sample210.Sample210()	10	0	0	0	
Sample210.method1(int)	9	1	2	2	
Sample210.method2(int)	11	2	2	2	
Sample210.method3(int)	13	3	2	2	
Sample210.method4(int)	17	5	2	2	
Sample210.method5(int)	17	5	2	2	

説明

計測結果を表形式で表示します。表示データ内容は、計測単位の指定により異なります。

表6.6 (1) クラス単位の計測時

表示データ項目	説明
クラス名	計測単位となるクラス名を表示します。
文数	クラス内の文数を表示します。

表示データ項目	説明
継承数	クラスの継承数を表示します。
フィールド変数の数	クラス内に定義されるフィールド変数の数を表示します。
インスタンス変数の数	クラス内に定義されるインスタンス変数の数を表示します。
パブリック変数の数	クラス内に定義されるパブリック変数の数を表示します。
メソッド数	クラス内に定義されるメソッド数を表示します。
インスタンスメソッド数	クラス内に定義されるインスタンスメソッド数を表示します。
パブリックメソッド数	クラス内に定義されるパブリックメソッド数を表示します。
定義ファイル名	クラスが定義されるJavaファイル名を表示します。Javaファイル内の先頭に定義されるクラス名にのみ出力されます。

表6.7 (2)メソッド単位の計測時

表示データ項目	説明
メソッド名	計測単位となるメソッド名を表示します。
文数	メソッド内の文数を表示します。
分岐文数	メソッド内に定義される条件分岐文数を表示します。
ループ数	メソッド内に定義されるループ文数を表示します。
メソッドCALL数	メソッド内に定義されるメソッドCALL文数を表示します。
定義ファイル名	メソッドが定義されるJavaファイル名を表示します。Javaファイル内の先頭に定義されるメソッド名にのみ出力されます。

メニューコマンド説明

表6.8 (1)ファイルメニュー

メニューアイテム	説明
終了(X)	計測結果表示画面を終了します。

表6.9 (2)オプションメニュー

メニューアイテム	説明
アラーム指定...(L)	計測結果のアラーム表示内容を指定します。

表6.10 (3)ヘルプメニュー

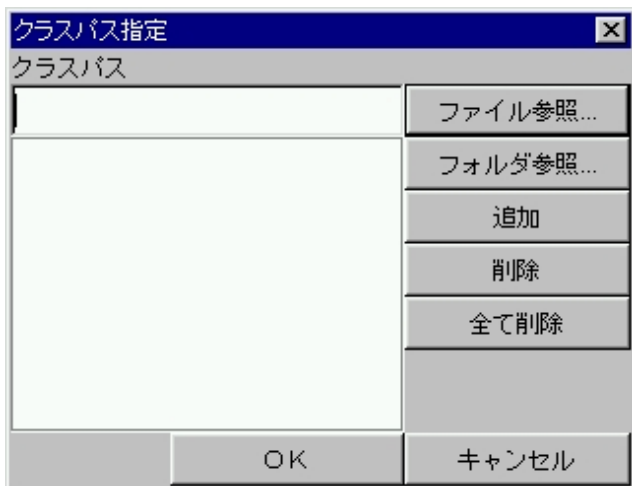
メニューアイテム	説明
MF-STEP COUNTER ヘルプ(T)	ヘルプ画面を表示します。
バージョン情報...(A)	当ツールのバージョンレベルを表示します。

画面コントロール説明

画面コントロール	説明
[計測結果表]テーブル	計測結果を表示します。アラーム指定ダイアログボックスでのアラーム表示指定により、以下のフォントカラーで表示されます。 - 上限値以上の計測値: 赤 - 下限値以下の計測値: 青 - 上記以外: 黒

6.4 クラスパス指定ダイアログボックス

レイアウト



説明

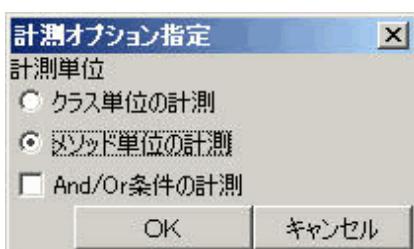
Classファイル計測時の検索パスとなるクラスパス名(フォルダ名/Jarファイル名)を指定します。

画面コントロール説明

画面コントロール	説明
[クラスパス名]テキストフィールド	クラスパスとなるフォルダ名・Jarファイル名の入力をおこないます。[ファイル参照]ボタン・[フォルダ参照]ボタンにより、フォルダ・Jarファイルの選択が可能です。
[ファイル参照]ボタン	クラスパスとなるJarファイル名を選択します。
[フォルダ参照]ボタン	クラスパスとなるフォルダ名を選択します。
[クラスパス一覧]リスト	選択済のクラスパス名を表示します。
[追加]ボタン	[クラスパス名]テキストフィールドに指定された内容を、[クラスパス一覧]リストの末尾に追加します。
[削除]ボタン	[クラスパス一覧]リストの選択資産を、[クラスパス一覧]リストから削除します。
[全て削除]ボタン	[クラスパス一覧]リストに表示される全パス名を、[クラスパス一覧]リストから削除します。
[OK]ボタン	指定した内容を保存し、当ダイアログボックスを終了します。
[キャンセル]ボタン	指定した内容を無効とし、当ダイアログボックスを終了します。

6.5 オプション指定ダイアログボックス

レイアウト



説明

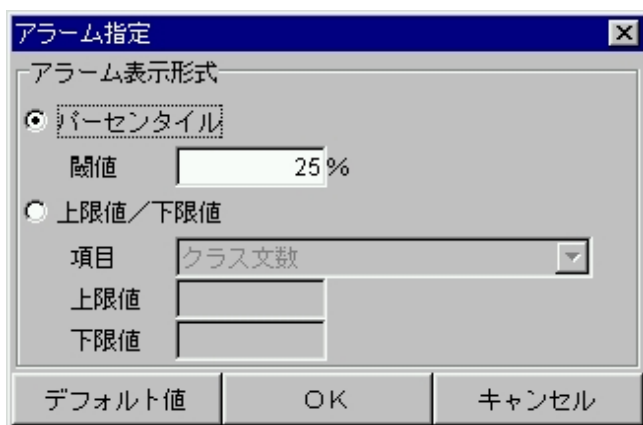
計測単位(クラス単位の計測/メソッド単位の計測)を指定します。

画面コントロール説明

画面コントロール	説明
[クラス単位の計測]ラジオボタン	対象資産となるJavaファイルに定義されるクラス単位に計測処理を実行します。
[メソッド単位の計測]ラジオボタン	対象資産となるJavaファイルに定義されるメソッド単位に計測処理を実行します。
[And/Or条件の計測]チェックボックス	if文などでAnd/Or条件ごとに分岐数を計測します。[メソッド単位の計測]の場合にのみ有効です。
[OK]ボタン	指定した内容を保存し、当ダイアログボックスを終了します。
[キャンセル]ボタン	指定した内容を無効とし、当ダイアログボックスを終了します。

6.6 アラーム指定ダイアログボックス

レイアウト



説明

計測結果表示画面で表示するアラーム範囲の指定をおこないます。当ダイアログボックスで指定される上限値以上の計測値/下限値以下の計測値をアラーム表示の対象とします。

画面コントロール説明

画面コントロール	説明
[パーセンタイル]ラジオボタン	パーセンタイル形式でアラームを表示します。 詳細は「 4.2.3 パーセンタイル形式のアラーム表示 」を参照してください。
[閾値]テキストフィールド	アラーム表示対象となる下限の閾値設定を行います。下限の閾値は(0%~50%)の範囲で指定します。当フィールドが空白の場合、および、0%が指定された場合、アラーム表示は行いません。当指定は、[パーセンタイル]ラジオボタンが選択された場合に有効となります。
[上限値/下限値]ラジオボタン	上限値/下限値形式でアラームを表示します。
[項目]コンボボックス	アラーム指定の対象となる計測項目を選択します。当指定は、[上限値/下限値]ラジオボタンが選択された場合に有効となります。
[上限値]テキストフィールド	アラーム表示対象となる上限値の設定をおこないます。下限値より大きい内容でなければなりません。当フィールドが空白の場合、上限値以上の計測値に対するアラーム表示は行いません。当指定は、[上限値/下限値]ラジオボタンが選択された場合に有効となります。

画面コントロール	説明
[下限値]テキストフィールド	アラーム表示対象となる下限値の設定をおこないません。上限値より小さい内容でなければなりません。当フィールドが空白の場合、下限値以下の計測値に対するアラーム表示はおこないません。当指定は、[上限値／下限値]ラジオボタンが選択された場合に有効となります。
[デフォルト値]ボタン	パーセンタイル形式が選択され、閾値として25%が設定されます。
[OK]ボタン	指定した内容を保存し、当ダイアログボックスを終了します。
[キャンセル]ボタン	指定した内容を無効とし、当ダイアログボックスを終了します。

6.7 バージョン情報ダイアログボックス

レイアウト



説明

当ツールのツール名／機能名／バージョンレベル／著作権を表示します。

画面コントロール説明

画面コントロール	説明
[OK]ボタン	当ダイアログボックスを終了します。

第7章 補足事項

MF-STEP-COUNTER Javaソフトウェアメトリクス計測機能での補足事項について説明します。

7.1 注意事項

Javaソフトウェアメトリクス計測機能を使用するにあたっての注意事項を以下に示します。

1. 計測対象となるJavaファイル/Classファイルについて

対象となるJavaファイルは、正常コンパイル可能なファイルで、シフトJISで記述されたファイルでなければなりません。また、対象となるClassファイルは、JDK8.0で作成されたファイルでなければなりません。

2. 計測処理の中断について

画面処理で計測処理を中断する場合、中断時点まで計測した内容を計測結果として出力します。バッチ処理や画面処理起動時に、コマンドプロンプトからアプリケーションを中断する場合、計測結果は出力されません。

3. 複数起動について

当ツールは、複数起動をサポートしていません。このため、複数起動時の動作は保証されません。

4. 起動時のフォルダについて

当ツールを起動する場合、ツールのインストールフォルダがカレントフォルダになっていなければ、動作は保証されません。

5. ツールの構成について

当ツールがインストールされる、インストールフォルダには、Temp/Helpサブフォルダが存在します。インストールフォルダ内のこれらのサブフォルダを変更・削除すると、当ツールの動作は保証されません。

6. SQLJ使用ソースについて

当ツールでは、SQLJを使用したJavaソースの計測を対応していません。SQLJを使用したJavaソースを計測対象とした場合、そのソースは計測されません。

7. パス名長について

当ツールでは、使用するファイルを260バイトまで指定可能です。ファイルシステムによっては、260バイトまでの指定ができません。この場合、使用可能なバイト数分のみパス名として採用され、使用可能バイトを超える部分は切り捨てられます。最大長のパス名を指定する場合は、注意が必要です。

8. 列挙型と注釈計測について

JDK8.0では以下の注意点ががあります。

●列挙型について

enumクラスを宣言した場合、valueOf(), values()という2つのpublic staticなメソッドが追加されるため、記述したメソッド数より+2されます。

enumクラスを宣言した場合で、メソッドを宣言している場合、内部的なメソッドが保持されるためさらに+1されます。

通常のクラス等でenum型フィールドを宣言した場合でも、クラスとして別に計測されます。

●注釈について

ステップ数計測機能では記述された量を計測するという観点から計測対象としますが、注釈行(コメント)ではなく有効行として計測します。一方、本メトリクス機能ではソースの複雑度/特性を見るという観点から、文数の計測対象とはなりません。

注釈型を定義したソースをメソッド計測した場合、処理が存在しないため計測結果が出力されません。

クラス/インタフェース宣言内で注釈型を定義したソースは計測できません。

9. 同一クラス名を定義するJavaファイルの計測について

計測対象として、同一クラス名(パッケージ名を含む)を定義する複数のJavaファイルを一度に指定する場合、計測するJavaファイルに定義されるクラスを正常に計測できません。Classファイルを環境変数CLASSPATHに指定されるパス/当ツール内のクラスパス指定で指定するパスの順で検索しているため、常に同一のClassファイルが検出されます。

同一クラス名(パッケージ名を含む)を定義する複数のJavaファイルを計測する場合は、計測処理を複数回に分割し実行する必要があります。

10. ユーザ定義のパッケージを使用するClassファイルの計測について

ユーザ定義のパッケージ内のクラスを使用するClassファイルの計測をおこなう場合、事前に環境変数CLASSPATHに、使用するパッケージが存在するパス名(フォルダ/Jarファイル)を指定する必要があります。指定されない場合、ユーザ定義のパッケージに含まれるクラスを使用するClassファイルを正常に計測できません。

11. ClassファイルとJavaファイルの同期について

クラス単位の計測では、ClassファイルはJavaファイルと同期が取れていて最新状態になっている必要があります。最新状態になっていない場合、正常に計測することができません。計測前にJavaファイルがコンパイル済であることを確認してください。

12. CSVファイルの出力について

当ツールでは、計測処理を実行した場合、指定したCSVファイルの内容が計測結果で上書きされます。指定したCSVファイルの既存情報が必要な場合は、出力先となるCSVファイル名を変更するか、CSVファイルのバックアップを作成後、計測処理を実行してください。

コマンドライン引数に計測対象一覧ファイルを指定し、バッチ処理で計測処理を実行する場合、計測対象一覧ファイル内に指定されるCSVファイル名に計測結果が出力されます。この場合も、指定のCSVファイルに既存情報がある場合は、計測結果で上書きされるため、注意が必要です。

13. ヘルプ表示フレームの動作について

ヘルプ表示フレームでは、リンク先を選択するとHTMLページロードのため一時的に選択不可状態になります。選択不可状態が解除後もHTMLページロードが継続している時に、頻繁にリンクが発生するとJavax.swingの例外メッセージがコマンドプロンプト上に表示されることがありますが、処理上問題はありません。

14. JDKの動作環境について

当ツールの動作には、事前にJDKのbinフォルダが環境変数PATHに追加されている必要があります。

15. ファイル選択・フォルダ選択時のネットワークドライブについて

CSVファイルの選択時、対象資産ファイル/対象資産フォルダの選択時、クラスパスファイル/クラスパスフォルダ選択時に起動されるファイル選択ダイアログ/フォルダ選択ダイアログでは、直接ネットワークコンピュータをアクセスすることができません。ネットワークコンピュータをアクセスする際は、あらかじめ、エクスプローラ等にてネットワークドライブを割り当てておく必要があります。

16. 画面操作について

画面操作時において連続した同一操作(同一ボタンを連続して押下)を行うと、意図しない動作となる場合があります。

17. 使用文字について

JAVAファイルやCLASSファイルの格納パスに、UNICODEの環境依存文字は使用しないでください。

第8章 メッセージ

Javaソフトウェアメトリクス計測機能で出力されるエラー内容を説明します。

8.1 メッセージ一覧

STJ-001 当ツールの実行には、JDK7. 0以降が必要です。

【意味】

当ツールの実行には、JDK8.0が必要です。

【対処】

JDK8.0をインストール後、再度実行してください。

STJ-002 アラーム指定の閾値の内容が不当です。

【意味】

アラーム指定ダイアログでの設定で、アラーム範囲(閾値)の設定に誤りがあります。

【対処】

アラーム範囲の設定を修正します。

STJ-003 アラーム指定の上限値/下限値の内容が不当です。

【意味】

アラーム指定ダイアログでの設定で、アラーム範囲(上限値/下限値)の設定に誤りがあります。

【対処】

アラーム範囲の設定を修正します。

STJ-004 CSVファイル名が指定されていません。

【意味】

メイン画面でCSVファイル名が指定されていません。

【対処】

CSVファイル名を指定します。

STJ-005 計測対象のクラスまたは継承元クラスを検出できません。

JAVA=javaファイル名

CLASS=クラス名

【意味】

Classファイルの計測時に、計測対象となるクラスまたは、継承元クラスを検出できません。

【対処】

計測対象となるクラスまたは、継承元クラスが存在するパスを、当ツールの[クラスパス指定](#)に設定します。

例1) クラス名がaaa.class、パッケージがcom.xxx.yyyの場合で、以下のフォルダ構成の場合

```
c:\classes\com\xxx\yyy\aaa.class → フォルダ選択でc:\classesを指定します
```

例2) aaa.classがjarファイルに圧縮されて以下のように配置されている場合

```
c:\classes\jar\abc.jar → ファイル選択でc:\classes\jar\abc.jarを指定します
```

例3) aaa.classがcom.xxx.bbbクラスを継承している場合

c:\%classlib%\com%\xxx%\bbb.class → フォルダ選択でc:\%classlibを追加指定します

注意)フォルダに展開されている場合のフォルダ構成や、Jarファイルの内のフォルダ構成は、通常のJava実行環境でのclasspath指定と同じく、上記例1)のようにパッケージと同じ構成になっている必要があります。

また、ClassファイルまたはJarファイルへのフルパスの長さが、OSの制限以上になっている場合、参照できないためにこのエラーが発生することがあります。

STJ-006 クラスパスの指定が不当です。

【意味】

クラスパス指定ダイアログボックスで、クラスパス名の指定内容に誤りがあります。

【対処】

クラスパス名への指定内容を修正します。

STJ-007 クラスパスへの指定が重複しています。

【意味】

クラスパス指定ダイアログボックスで、クラスパス名の指定内容が重複しています。

【対処】

クラスパス名に指定した内容が、すでにクラスパス一覧に設定されているため、設定する必要はありません。

STJ-008 計測処理で異常が発生しました。

【意味】

計測処理中に異常が発生しました。

【対処】

標準出力または、当ツールのインストールフォルダ配下のTempフォルダに出力されるエラーファイル「mfstpjm.err」に出力されるエラーメッセージを確認し、メッセージごとに対処します。

STJ-009 計測処理を中断します。 よろしいですか？

【意味】

計測中に、中断ダイアログボックスで[中断]ボタンが選択されました。

【対処】

計測処理を中断する場合は[はい]ボタンを、中断しない場合は[いいえ]ボタンを選択します。

STJ-010 計測処理が正常終了しました。

【意味】

計測処理が正常終了しました。

【対処】

なし。

STJ-011 指定ファイルを読み込むことができません。 TYPE=ファイル種別 FILE=ファイル名

【意味】

指定ファイルの読み込みに失敗しました。

【対処】

指定ファイルが、読み込み可能な指定ファイル種別のファイルであることを確認します。必要であれば、指定ファイル名を修正し再度実行します。

STJ-012 指定ファイルに書き込むことができません。

TYPE=ファイル種別

FILE=ファイル名

【意味】

指定ファイルの書き込みに失敗しました。

【対処】

指定ファイルが、書き込み可能な指定ファイル種別のファイルであることを確認します。必要であれば、指定ファイル名を修正し再度実行します。

STJ-013 計測結果を表示中です。

計測結果表示フレームを終了してください。

【意味】

計測結果表示画面を表示中のため、選択した処理を実行できません。

【対処】

計測結果表示画面を終了し、再度実行してください。

STJ-014 エラー情報の表示に失敗しました。

標準出力またはエラーファイル(mfstpjm.err)を参照してください。

【意味】

エラー情報表示画面の表示に失敗しました。

【対処】

エラー出力先を標準出力に設定し、再度計測するか、当ツールのインストールフォルダ配下のTempフォルダに出力されるエラーファイル「mfstpjm.err」の内容を確認します。

STJ-015 計測処理で異常が検出されました。

エラーファイルを表示しますか？

【意味】

計測処理で異常が発生しました。

【対処】

エラー情報表示画面によりエラーファイルの内容を表示する場合は[はい]ボタンを、表示しない場合は[いいえ]ボタンを選択します。

STJ-016 Javaファイルの形式が正しくありません。

FILE=Javaファイル名

ERR=異常箇所

【意味】

指定のJavaファイルの形式が正しくありません。

【対処】

Javaファイル内の異常箇所を確認し、修正します。指定するJavaファイルは、正常コンパイルされるファイルでなければなりません。

STJ-017 対象資産に計測対象となるJavaファイルが存在しません。

PATH=対象資産名

【意味】

指定の対象資産内に計測対象となるJavaファイルが存在しません。

【対処】

計測対象となるJavaファイルが存在する、対象資産名を指定します。

STJ-018 ファイルの読み込みに失敗しました。

TYPE=ファイル種別

FILE=ファイル名

【意味】

指定ファイルの読み込みに失敗しました。

【対処】

指定ファイル名が、ファイル種別に合った読み込み可能なファイルであることを確認します。

STJ-019 ファイルの書き込みに失敗しました。

TYPE=ファイル種別

FILE=ファイル名

【意味】

指定ファイルの書き込みに失敗しました。

【対処】

指定ファイル名が、ファイル種別に合った書き込み可能なファイルであることを確認します。

STJ-020 ファイルの形式が正しくありません。

TYPE=ファイル種別

FILE=ファイル名

【意味】

指定ファイルの形式が、指定ファイル種別の形式と一致しません。

【対処】

指定ファイル名が、ファイル種別に合った形式のファイルであることを確認します。

STJ-021 ファイルが存在しません。

TYPE=ファイル種別

FILE=ファイル名

【意味】

指定ファイルが存在しません。

【対処】

ファイル種別に合った既存ファイル名を指定します。

STJ-022 フィールド情報を検出できません。

JAVA=Javaファイル名CLASS=クラス名

【意味】

Classファイルの計測時に、フィールド情報の検出に失敗しました。

【対処】

計測対象となるクラス内にユーザが定義するパッケージが使用されているかどうか確認します。使用している場合は、ツール内のクラスパス指定により、使用しているパッケージが存在するパス(フォルダ/Jarファイル)名を指定します。指定されていない場合、計測対象となるClassファイルが正常ファイルであることを確認します。

STJ-023 メソッド情報を検出できません。**JAVA=Javaファイル名****CLASS=クラス名****【意味】**

Classファイルの計測時に、メソッド情報の検出に失敗しました。

【対処】

計測対象となるクラス内に、ユーザが定義するパッケージが使用されているかどうか確認します。使用している場合は、ツール内のクラスパス指定により、使用しているパッケージが存在するパス(フォルダ/Jarファイル)名を指定します。指定されていない場合、計測対象となるClassファイルが正常ファイルであることを確認します。

STJ-024 対象資産名の指定が重複しています。**【意味】**

メイン画面で、対象資産名の指定に重複があります。

【対処】

メイン画面で指定される対象資産名が、すでに対象資産一覧に存在するため、指定の必要がありません。対象資産形式がサブフォルダを含むフォルダの場合、上位フォルダを指定するよう修正します。

STJ-025 対象資産が存在しません。**【意味】**

メイン画面で指定する対象資産が存在しません。

【対処】

既存の対象資産を指定します。

STJ-026 対象資産が存在しません。**OBJECT=対象資産名****【意味】**

メイン画面で指定する対象資産が存在しません。

【対処】

既存の対象資産を指定します。

STJ-027 対象資産名が指定されていません。**【意味】**

メイン画面で対象資産一覧に追加するための対象資産名が指定されていません。

【対処】

既存の対象資産名を指定し、追加処理を実行します。

STJ-028 対象資産形式が変更されました。**対象資産一覧への選択内容は、消去されます。****処理を続行しますか？****【意味】**

メイン画面で対象資産形式が変更されました。対象資産一覧に指定される資産名は、形式が一致しないため、対象資産一覧から資産名を消去します。

【対処】

対象資産一覧から資産名を消去し、対象資産形式を変更する場合は[はい]ボタンを、対象資産一覧から資産名を消去しないで、対象資産形式を変更しない場合は[いいえ]ボタンを選択します。

STJ-029 対象資産の形式が不当です。

【意味】

メイン画面の対象資産名に指定する資産の形式が、対象資産形式で選択した形式と一致しません。

【対処】

対象資産形式の形式内容と一致する、対象資産名を指定します。

STJ-030 対象資産の形式が不当です。

OBJECT=対象資産名

【意味】

メイン画面の対象資産名に指定する資産の形式が、対象資産形式で選択した形式と一致しません。

【対処】

対象資産形式の形式内容と一致する対象資産名を指定します。

STJ-031 コマンドライン引数の指定に誤りがあります。

【意味】

当ツール起動時に指定するコマンドライン引数の指定に誤りがあります。

【対処】

コマンドライン引数の内容を修正し、再度実行します。

STJ-032 計測対象一覧ファイルが保存されていません。

保存しますか？

【意味】

メイン画面/クラスパス指定ダイアログボックス/オプション指定ダイアログボックス/アラーム指定ダイアログボックスで指定された内容が保存されていません。

【対処】

指定内容を保存する場合は[はい]ボタンを、保存しない場合は[いいえ]ボタンを選択します。

STJ-033 計測対象のJavaソースの更新日より古いクラスファイルです。

JAVA=Javaファイル名

CLASS=クラス名

【意味】

計測対象のJavaファイルの更新以前にClassファイルが作成されています。

【対処】

計測対象Javaファイルをコンパイルし、作成されたClassファイルと同期がとれていることを確認します。また、当ツールのクラスパス指定に設定して、Classファイルの検索パスが、正常であることを確認します。

STJ-034 エラーファイルの出力に失敗しました。

エラーメッセージは標準出力に出力されます。

【意味】

エラーファイル(mfstpjm.err)の出力に失敗しました。計測時のエラーメッセージは、標準出力に出力します。

【対処】

当ツールのインストールフォルダ配下のTempフォルダのエラーファイル(mfstpjm.err)が出力可能なファイルであることを確認してください。

STJ-035 JDKのbinフォルダを参照できません。

【意味】

JDKインストールフォルダのbinフォルダの参照に失敗しました。

【対処】

環境変数PATHにJDKのbinフォルダを追加し再度実行します。

**STJ-036 指定のパスは既に存在します。
上書きしますか？**

【意味】

指定のパスは既に存在します。

【対処】

上書きする場合は[はい]ボタンを、上書きしない場合は[いいえ]ボタンを選択します。

**STJ-037 指定のパスは存在しません。
既存のパスを指定してください。**

【意味】

指定のパスは存在しません。

【対処】

既存のパスを再度指定します。

STJ-038 メッセージの取得に失敗しました。

【意味】

表示すべきメッセージを取得できませんでした。

【対処】

当ツールが正常にインストールされていない可能性があります。インストール後再度実行してください。

第9章 付録

Javaソフトウェアメトリクス計測機能の出力結果及び、サンプルについて説明します。

9.1 計測結果説明(CSVファイル)

概要

計測内容は、CSVファイルに出力されます。画面起動時の計測で、計測処理を[中断]ボタンにより中断した場合、計測された部分までの内容が出力されます。CSVファイルは、計測単位により出力内容が異なります。

出力内容説明

(1) クラス単位計測時

```
" class-name" , 999 , 999 , 999 , 999 , 999 , 999 , 999 , 999 , "file-name"  
a          b   c   d   e   f   g   h   i   j
```

項	計測項目	説明
a	クラス名	計測対象となるクラス名を「パッケージ名. クラス名」の形式で出力されます。
b	文数	クラス内の文数が出力されます。
c	継承数	クラスの継承数が出力されます。
d	フィールド変数の数	クラス内に定義するフィールド変数の数が出力されます。
e	インスタンス変数の数	クラス内に定義するインスタンス変数の数が出力されます。
f	パブリック変数の数	クラス内に定義するパブリック変数の数が出力されます。
g	メソッド数	クラス内に定義するメソッド数が出力されます。
h	インスタンスメソッド数	クラス内に定義するインスタンスメソッド数が出力されます。
i	パブリックメソッド数	クラス内に定義するパブリックメソッド数が出力されます。
j	定義ファイル名	クラスが定義されるJavaファイル名がフルパスで出力されます。Javaファイル内で先頭に定義されるクラスにのみ出力されます。

(2) メソッド単位計測時

```
" method-name" , 999 , 999 , 999 , 999 , "file-name"  
a          b   c   d   e   f
```

項	計測項目	説明
a	メソッド名	計測対象となるメソッド名を「パッケージ名. クラス名. メソッド名(引数)」の形式で出力されます。
b	文数	メソッド内の文数が出力されます。
c	分岐文数	メソッド内の条件分岐文の数が出力されます。

d	ループ数	メソッド内のループ文の数が出力されます。
e	メソッド CALL数	メソッド内での他メソッドのCALL数が出力されます。
f	定義ファイル名	メソッドが定義されるJavaファイル名がフルパスで出力されます。Javaファイル内で先頭に定義されるメソッドにのみ出力されます。

9.2 サンプルの使い方

MF-STEP-COUNTERのインストールフォルダには"**SAMPLE¥jp**"フォルダが作成され、サンプルソースが格納されています。

ファイル名	説明
Sample2.java	サンプルのアプレットソース
SAMPLE2.jar	Sample2.javaから生成されるclassファイルを格納したjarファイル

9.2.1 サンプルソースを計測するには

1. Javaソフトウェアメトリクス計測機能をウィンドウズのスタートメニューから起動するか、起動されているJavaソフトウェアメトリクス計測機能の[ファイル]メニューから[新規作成]コマンドを実行します。
2. メイン画面の[CSVファイル]横の[参照]ボタンを押して、計測結果を出力するCSVファイル名を[ファイル選択]ダイアログから指定します。
3. メイン画面の[対象資産形式]のリストから、[ファイル指定]を選択します。
4. メイン画面の[対象資産名]横の[参照]ボタンを押して、"**SAMPLE¥jp**"フォルダにある"**Sample2.java**"ファイルを[ファイル選択]ダイアログから指定します。
5. メイン画面の[追加]ボタンを押して、[対象資産一覧]に追加します。
6. 計測対象となる"**Sample2.java**"に対応するclassファイルの場所を、次のどちらかの方法でクラスパスに登録します。
 - a. 環境変数のCLASSPATHに追加する
 - b. [オプション]メニューの[クラスパス指定]コマンドを実行して、[クラスパス指定]ダイアログボックスから追加する
- 尚、計測対象資産から参照されるクラス(JDKなど)のクラスパスは、必ず環境変数のCLASSPATHに追加しておいてください。
7. メイン画面の[計測]メニューから[計測]コマンドを実行します。
8. 計測結果は指定されたCSVファイルに保存され、計測結果が画面表示されます。

詳細については以下を参照してください。

- [5.2.1 新規計測対象一覧ファイル指定方法](#)
- [5.2.2 対象資産の指定方法](#)
- [5.2.4 クラスパスの指定方法](#)
- [9.1 計測結果説明\(CSVファイル\)](#)

9.2.2 計測方法を変更するには

初期設定では[クラス単位の計測]になっていますが、[メソッド単位の計測]に変更して、クラス内のより詳細な計測をすることができます。

1. メイン画面の[オプション]メニューから[計測オプション指定]コマンドを実行します。
2. [計測オプション指定]ダイアログボックスで[メソッド単位の計測]を選択します。
3. メイン画面の[計測]メニューから[計測]コマンドを実行します。

4. 計測結果は指定されたCSVファイルに保存され、計測結果が画面表示されます。

詳細については以下を参照してください。

- ・ [4.1.1 クラス単位の計測](#)
- ・ [4.1.2 メソッド単位の計測](#)

9.2.3 アラーム表示するには

パーセンタイル形式の閾値指定、または、各種計測項目の上限値／下限値を指定することにより、計測結果表示画面でアラーム表示することができます。

- ・ 上限値以上の値は**赤色**表示します
- ・ 下限値以下の値は**青色**表示します

[パーセンタイル指定]

1. メイン画面の[オプション]メニューから[アラーム指定]コマンドを実行します。
2. [アラーム指定]ダイアログボックスの[パーセンタイル]ラジオボタンを選択します。
3. 閾値を設定します。
4. メイン画面の[計測]メニューから[計測]コマンドを実行して再度計測するか、[表示]メニューから[計測結果表示]コマンドにより計測結果を再表示します。
5. 計測結果表示画面の[オプション]メニューの[アラーム指定]コマンドからも指定できます。

[上限値／下限値指定]

1. メイン画面の[オプション]メニューから[アラーム指定]コマンドを実行します。
2. [アラーム指定]ダイアログボックスの[上限値／下限値]ラジオボタンを選択します。
3. [項目]リストから任意の項目を選択します。
4. 上限値と下限値を設定します。
5. メイン画面の[計測]メニューから[計測]コマンドを実行して再度計測するか、[表示]メニューから[計測結果表示]コマンドにより計測結果を再表示します。
6. 計測結果表示画面の[オプション]メニューの[アラーム指定]コマンドからも指定できます。

新規計測時は、[パーセンタイル指定][閾値=25%]がデフォルトで設定されます。

アラームを表示しない場合は、閾値、および、上限値／下限値に空白(データなし)を設定します。

9.2.4 計測対象を保存するには

指定した計測対象や出力先CSVファイル、各種オプション設定などを[計測対象一覧ファイル]として保存することができます。

1. メイン画面の[ファイル]メニューから[名前を付けて保存]コマンドを実行します。
2. [名前を付けて保存]ダイアログボックスで保存先を指定します。
3. [名前を付けて保存]ダイアログボックスの[保存]ボタンにより保存されます。

9.2.5 再計測するには

保存された[計測対象一覧ファイル]を読み込むことにより、簡単に再計測が行えます。

1. メイン画面の[ファイル]メニューから[計測対象一覧ファイルを開く]コマンドを実行します。
2. [ファイルを開く]ダイアログボックスで読み込むファイルを指定します。
3. [ファイルを開く]ダイアログボックスの[開く]ボタンにより読み込まれます。
4. メイン画面の[計測]メニューから[計測]コマンドを実行します。

9.2.6 計測結果の判定について

計測結果の判定については、[4.2.1 測定値の判定](#)を参考にして下さい。

また、アラーム指定の基準値の設定については[4.2.2 アラーム指定基準値設定例](#)を参考にして下さい。