# 特別研究報告

題目

# プログラムで多用される動詞と目的語の関係を利用したメソッド名提案ツール

指導教員
井上 克郎 教授

報告者
鬼塚 勇弥

平成 24 年 2 月 16 日

大阪大学 基礎工学部 情報科学科

平成 23 年度 特別研究報告

# プログラムで多用される動詞と目的語の関係を利用したメソッド名提案ツール

鬼塚 勇弥

**内容梗概**

プログラム理解のコストは，ソフトウェア保守にかかるコストのうち大きな割合を占めるが，理解の過程で識別子が果たす役割は大きい．そのため，もし不適切な名前が識別子に付けられている場合，開発者がプログラム要素の役割や機能を識別子名から正しく推測できず，理解に長い時間を要してしまったり，間違って推測してしまうなどの問題が起こる．

本研究では，様々な種類の識別子が存在する中で，オブジェクト指向プログラムにおけるメソッド名に着目する．メソッド名は，それ自身の内部や，関連する識別子との間に，複雑な動詞と目的語の関係を持つ．さらに，動詞と目的語にどのような単語を採用するべきか，プログラム言語やアプリケーションドメイン，利用するフレームワークなどの環境で異なっている．そのため，開発者が適切なメソッド名を名付けるには，長い経験と豊富な知識が必要となる．

そこで，メソッドを命名しようとしている開発者に対して，メソッド名の候補リストを提示することで，メソッドの命名を支援する手法を提案する．候補リストを生成するために，開発者が編集中のソースコードと，既存研究で作成された動詞-目的語関係の辞書を利用する．提案した手法は，統合開発環境 Eclipse 上に実装した．

提案手法を評価するため，手法を用いる場合と用いない場合とで命名がどのように変化するかを，被験者実験を行って調査した．被験者に与えた課題は，メソッド名などの情報を削除した実際のソースコードから，削除されたメソッド名を推測するものである．実験の結果，手法を用いることで，削除されたメソッド名と同一のメソッド名を解答する割合や，動詞が一致するメソッド名を解答する割合が高まったものの，統計的に有意な差は見られなかった．一方で，被験者に対して行なったアンケート調査では，メソッド名の候補を提示するというユーザインタフェースや，提示された候補リストの内容に対して好意的な意見が多いことが分かり，本研究のアプローチが有効であることが確認できた．

**主な用語**

プログラム理解
動詞-目的語関係

メソッド名
命名支援

# 目 次

# 1　まえがき

ソースコードの読解では，識別子の名前からメソッドや変数の役割・機能を類推することが多いため，識別子に対して不適切な命名がされている場合，適切な命名が行われたソースコードに比べてソースコードの理解に，より多くの時間がかかってしまうことが知られている [6]． 近年，ソフトウェアの規模が大きくなるに伴い，ソフトウェアの保守コストの増大が問題となっているが，保守作業者のソフトウェア理解の時間の増加は保守コストの増大につながる一つの要因である．

メソッドの命名は，動詞や名詞，前置詞といった複数の品詞の単語を組み合わせる必要があり，他の識別子の命名とは異なる特有の難しさがある．そのため，開発者の類似プログラムの作成経験や開発対象のドメインの知識が不足している場合，メソッドに対して不適切な命名をしてしまう場合がある．

そこで，本研究ではこのメソッド名特有の命名の難しさに着目し，悪いメソッド名がつけられることによって可読性が下がり，ソースコードの理解に時間がかかるという問題を解決するため，メソッド名の候補を開発者に提示することで開発者のメソッドの命名を支援する手法を提案し，ツールを実装した．

提示するメソッド名の候補を生成するにあたって，プログラム中の動詞-目的語関係に着目する．オブジェクト指向プログラムでは，あるオブジェクト A に対して操作 B を行うという動詞-目的語関係があり，メソッド名はその操作 B を示す動詞を先頭に置く命名が一般的である．既存研究には，そのような動詞-目的語関係を収録した辞書の生成手法がある [4]．本研究では，編集中のソースコードから取り出した目的語の候補を，既存研究に拡張を加えた動詞-目的語関係の辞書から検索し，取り出された動詞-目的語関係を組み合わせてメソッド名の候補を生成する．生成したメソッド名候補は，筆者が決めた基準を組み合わせることで並び替えて提示する．

ツールの実装は Eclipse のコード補完機能を用いて作成した．Eclipse の Java エディタでコード補完機能を呼び出すと，本手法で生成されたメソッド名の候補リストが表示される．

提案手法を実装したツールを使うことで，開発者が適切な命名が行えるかどうかを確認するために実験を行った．実験では，オープンソースプロジェクトのソースコードからメソッド名などを削除した課題を用意し，複数の被験者に削除したメソッド名を解答してもらった．そして，課題への解答時に，ツール使用の有無で正解率が変化するか比較を行った．また，提示されるメソッド名候補の一覧やツールの使用感に関するアンケートを行い，被験者の主観的な意見を聞いた．実験では，約 70 万の三つ組を収録した辞書を用いて，メソッド名の候補リストを作成した．また，被験者への課題は辞書作成に用いたものとは別のソースコードで作成した．

実験を行った結果，正解率はツールを使用して解答した場合の方が高かったが，統計的に有意な差は見られなかった．一方，アンケートでは，メソッド名の候補を提示するというユーザインタフェースや，提示された候補リストの内容に対して被験者から好意的な評価を得ることができた．以上から，メソッド名の候補を提示する本手法は命名支援として良いアプローチであると考えた．

以後，2 節では本研究の背景となった概念に関して述べる．3 節では，ユーザにメソッド名を提示する手法について説明する．4 節では，実装したツールの機能を述べる． 5 節では，実験とその考察を述べる．6 節では，本研究と関連がある研究について述べる． 7 節では，本研究のまとめと今後の課題に関して述べる．

## 2 背景

本研究の提案手法の背景として，まず識別子に対して不適切な命名がされている場合の問題について説明する．次に，メソッド名の命名がなぜ難しいかを，オブジェクト指向プログラミング言語の 1 つである Java のメソッドの命名規則と，オブジェクト指向プログラムに出現する動詞-目的語の関係を踏まえて説明する．その後，本提案手法で利用する動詞-目的語関係を収録した辞書について説明し，最後にツールの実装に用いた Eclipse のコード補完機能について説明する．

### 2.1 識別子に対して不適切な命名がされている場合の問題

保守作業において保守作業者は，対象のプログラムを理解するために，識別子の名前からメソッドや変数の役割・機能を推測しながらソースコードを読み進めるため，識別子に対して不適切な命名がされているとソースコードの可読性が下がり，理解により多くの時間がかかってしまう．

例えば，クラスのフィールドから名前を取得するメソッド名に “aaa” という名前がつけられている場合，読者はこのメソッドが使用されている場所まで読み進めた後，そのメソッド名の処理内容を理解するために，メソッド “aaa” が定義されている部分を探し出して，その中身を読む必要がある．もし，このメソッドに “getName” といった適切な命名がされている場合，読者はどのような処理を行っているかを推測することができる．また，このメソッドに “deleteName” という間違った名前がつけられている場合はさらに問題であり，読者は名前を削除するものだと考えて読み進めるが，途中で異なる処理が行われていると気づいても，どこに問題があるかわからず，原因を調べるのに長い時間がかかってしまう．

このように，不適切な命名は読者のソースコード理解の妨げとなるという問題がある．

### 2.2 メソッドの命名について

本節では，オブジェクト指向プログラムのメソッドの命名規則として，まず Java の例を挙げ，オブジェクト指向プログラムに出現する動詞-目的語関係を説明した上で，メソッドの命名にはどのような問題があるのかを説明する．

#### 2.2.1 Java のメソッドの命名規則

Java のメソッド名は，Oracle が公開している Code Conventions for the Java Programming Language [1] で示されているように，一般的には小文字で始まる動詞，もしくは 2 つ目以降の単語が大文字で始まる複数の単語で構成される．また，ここに書かれている以外にも，

Java のメソッド名には様々な守られるべきルールが存在する．それらのルールの一部を以下に列挙する．

- フィールドに値を設定するメソッド (Setter) は，フィールド名の前に set を付け加えた名前とする．
- フィールドの値を返り値とするメソッド (Getter) は，フィールド名の前に get を付け加えた名前とする．
- boolean 型の値を返すメソッドは，is，can，has で始まる名前とする．

### 2.2.2 オブジェクト指向プログラムに出現する動詞-目的語の関係

オブジェクト指向プログラムでは，メソッドがオブジェクトに対して何らかの操作を行う処理が多く出現する．操作と対象のオブジェクトの関係は，自然言語に出現する動詞と目的語の関係に類似しており，メソッド名に含まれる動詞の後に目的語が続く場合や，メソッドを定義しているクラスの名前に目的語が出現することがしばしばある．この特徴に着目し，Fry らは動詞をメソッド名から，直接目的語をメソッド名中の動詞の後ろの語と引数，及びクラス名から抽出する方法を提案している [3]． ただし，オブジェクト指向プログラムで出現する動詞と目的語の関係は，自然言語では出現しないような動詞と目的語の関係になっている場合も多い．

### 2.2.3 命名の難しさ

メソッドの命名には，様々な品詞が含まれる複数の単語を組み合わせてメソッドの振る舞いを表現するという，他の識別子とは異なる特有の難しさがある．メソッド名は他の識別子とは違い，動詞を先頭にした名前にするという特徴があり，その動詞はメソッド名の振る舞いを示す最も重要な部分となるため，処理内容を正確に表現した動詞を使用する必要がある．

しかし，メソッドの動詞の部分は，自然言語では処理内容を正確に表現しているようでも，プログラミング言語の慣習などによって正しく表現できていない場合がある．例えば，クラスのフィールドのリストから，ある 1 つの要素を取り出すメソッド名を，Java では一般に動詞 “find” を用いて表現する．自然言語では，フィールドからある要素を探すという処理から，動詞 “search” を用いても問題がないように思われるが，Java 言語ではそのような目的で “search” を使用しないため，ソースコードの読者は意図した処理を読み取ることができない．

このような暗黙のルールは，各プログラミング言語だけでなく，各種ドメインにも存在する場合がある．そのため，たとえ熟練した開発者であっても，類似プログラムの作成経験や

開発対象のドメインの知識が不足している場合には，不適切な命名を行ってしまう可能性がある．

### 2.3 動詞-目的語関係を収録した辞書

2.2.2 節で説明したオブジェクト指向プログラムに出現する動詞-目的語の関係に着目し，この関係を収録した辞書を生成する手法が早瀬らの研究 [4] で提案されている．本節ではこの概要について説明する．

早瀬らの研究では，ソースコード集合を入力として，そのソースコード中の動詞-目的語関係を収録した辞書を生成する手法が提案されている．動詞-目的語関係は，<動詞，直接目的語，間接目的語> の三つ組で表現される．

辞書作成の流れを図 1 に示す．辞書作成では，まずソースコード集合の解析を行って，全メソッドの情報が抽出される．そして，事前に生成した三つ組抽出パターンを用いて，そのメソッド情報に対してパターンマッチを行い，三つ組が抽出される．<動詞, 直接目的語, 間接目的語> の三つ組のうち，動詞はメソッド名中の動詞から，直接目的語及び間接目的語はメソッド名中の名詞，仮引数の型名と名前，メソッドを定義しているクラスの名前から抽出する．間接目的語は空の場合がある．

メソッド情報から三つ組を抽出するパターンは，表 1 の一覧の通りである．表中の * は任意の語句とマッチすることを示す記号であり，表中の void は void 型を示す値である．

### 2.4 Eclipse のコード補完機能

統合開発環境 Eclipse[2] は，コードアシストと呼ばれるコード補完機能を有している．これは，ソースコード編集中に Ctrl + Space で起動することにより，カーソル位置に挿入可能なコード片をポップアップウィンドウで一覧表示するものである．Eclipse のユーザは，方向キーでそのリストを選択し，Enter キーでその内容をソースコードに挿入することができる．

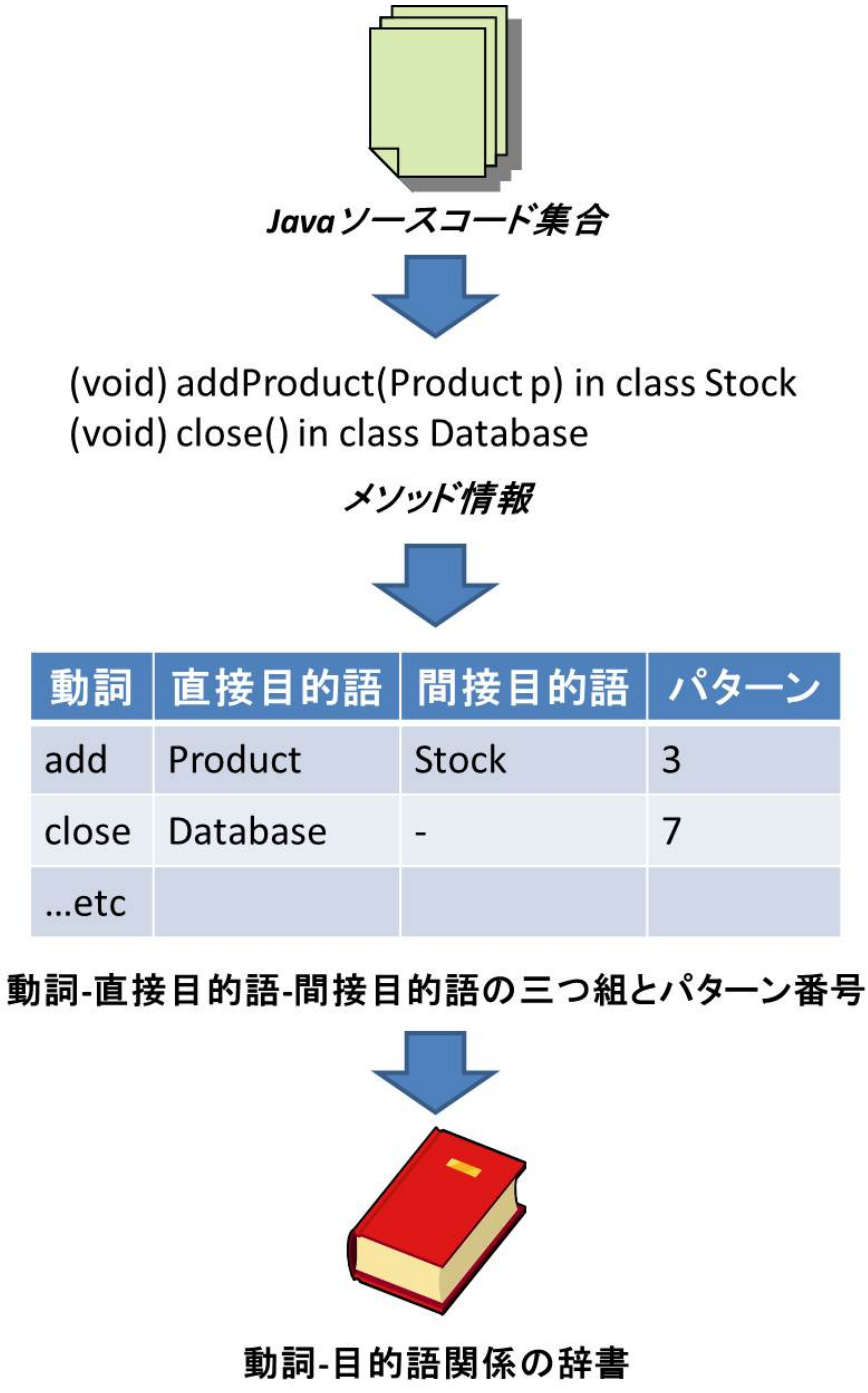
Javaソースコード集合
(void) addProduct(Product p) in class Stock
(void) close() in class Database
メソッド情報
動詞 直接目的語 間接目的語 パターン
add Product Stock 3
close Database - 7
...etc
動詞-直接目的語-間接目的語の三つ組とパターン番号
動詞-目的語関係の辞書

図 1: 辞書作成の流れ

表 1: 抽出パターン一覧 (参考文献 [4] TableI より)

| return type | method name | parameters | class name | verb | DO | IO |
|---|---|---|---|---|---|---|
| * | verb1 noun2 prepos3 noun4 | * | * | verb1 | noun2 | noun4 |
| * | verb1 prepos2 noun3 | * | noun4 | verb1 | noun4 | noun3 |
| * | verb1 noun2 | * | noun3 | verb1 | noun2 | noun3 |
| * | verb1 prepos2 noun3 | noun4 | * | verb1 | noun4 | noun3 |
| * | verb1 noun2 prepos3 | noun4 | * | verb1 | noun2 | noun4 |
| void | verb1 | (empty) | noun2 | verb1 | noun2 | (empty) |
| void | verb1 prepos2 noun3 | (empty) | noun4 | verb1 | noun4 | noun3 |
| void | verb1 noun2 | noun2 | noun3 | verb1 | noun2 | noun3 |
| void | verb1 noun2 prepos3 noun4 | * | noun5 | verb1 | noun2 | noun5 |
| void | verb1 | noun2 | noun3 | verb1 | noun2 | noun3 |
| void | verb1 noun2 | noun3 | noun4 | verb1 | noun3 | noun4 |
| void | verb1 noun2 | noun3 | noun2 | verb1 | noun2 | noun3 |
| void | verb1 noun2 | (empty) | noun2 | verb1 | noun2 | (empty) |
| void | verb1 noun2 | (empty) | noun2 | verb1 | noun2 | (empty) |
| void | noun1 verb2 | noun3 | noun4 | verb2 | noun1 | noun3 |
| void | noun1 verb2 | noun3 | noun4 | verb2 | noun1 | noun4 |
| void | noun1 verb2 | noun1 | noun3 | verb2 | noun1 | noun3 |
| noun1 | verb2 noun1 | noun3 | noun1 | verb2 | noun1 | noun3 |
| noun1 | verb2 noun1 prepos3 noun4 | noun4 | noun3 | verb1 | noun1 | noun4 |
| noun1 | verb2 noun3 prepos4 noun5 | (empty) | noun6 | verb2 | noun3 | noun 3 |
| noun1 | verb2 prepos3 noun4 | noun5 | noun6 | verb2 | noun6 | noun4 |
| noun1 | verb2 noun1 | (empty) | noun3 | verb2 | noun1 | noun3 |
| noun1 | verb2 | noun3 | noun4 | verb2 | noun4 | noun3 |
| noun1 | verb2 prepos3 | noun4 | noun5 | verb2 | noun5 | noun4 |
| noun1 | verb2 prepos3 noun4 | * | * | verb2 | noun1 | noun4 |
| noun1 | verb2 prepos3 noun4 | (empty) | noun1 | verb2 | noun1 | noun4 |
| noun1 | verb2 prepos3 | noun4 | noun4 | verb2 | noun4 | noun4 |
| noun1 | verb2 noun3 | (empty) | noun1 | verb2 | noun3 | noun1 |
| noun1 | verb2 noun1 | (empty) | noun3 | verb2 | noun3 | (empty) |
| noun1 | verb2 noun3 | (empty) | noun3 | verb2 | noun3 | noun1 |
| noun1 | verb2 noun1 | (empty) | noun1 | verb2 | noun1 | (empty) |

## 3　提案手法

本節では，返り値の型，メソッドの名前，引数の型の組 (以下，これをメソッド名と呼ぶ) の候補を開発者に提示することでメソッドの命名を支援する手法を提案する．メソッド名の候補の生成には，背景で述べた動詞-目的語関係の辞書を拡張して作成した辞書を利用する．拡張は，抽出パターンの追加であり，辞書の三つ組それぞれにその三つ組がどのように抽出されたかの情報を加えるものである．メソッド名候補の生成を行うために，この辞書を事前に作成しておく必要がある．メソッド名の候補は，開発者がメソッド名を記述する場所でツールを起動すると提示される．

メソッド名候補の生成は図 2 のように，編集中のソースコードの目的語の候補抽出，動詞-目的語関係の辞書の検索，メソッド名生成，並び替えの 4 ステップで行われる．メソッド名候補の生成は，起動した時のソースコードの状態によって異なる．1 つ目はメソッドの返り値が書かれていない場合であり，2 つ目は返り値の型として void が書かれている場合であり，3 つ目は返り値の型として void 以外の型が書かれている場合である．返り値が書かれている場合には，提示されるメソッド名候補は必ずその返り値を持つものとなる．

ツールが起動されたら，まず編集中のソースコードから目的語の候補抽出を行う (Step1)．この候補は，動詞-目的語関係の辞書における目的語の候補であり，カーソル位置から参照可能な名詞である．具体的には，インポートクラス名とその親クラス名，定義クラス名とその親クラス名，フィールド変数の型名と名前である．また，記述したいメソッドの返り値の型名が記述されている場合は，それも目的語の候補として抽出する．

次に，抽出した目的語の候補で動詞-目的語関係の辞書を検索し，三つ組や検索条件などの情報を得る (Step2)．動詞-目的語関係の辞書生成手法で用いられた三つ組の抽出パターンを基に，メソッド名生成パターンを事前に作成しておき，この生成パターンに基づいて辞書検索を行う．得る情報は三つ組の他に，三つ組の抽出パターンと，どの識別子で三つ組を検索したかの情報が含まれる．

辞書検索で情報を取得したら，三つ組を組み合わせてメソッド名を生成する (Step3)．メソッド名の生成は，辞書検索で使用したメソッド名生成パターンに従う．

最後に，複数生成されたメソッド名を，適切だと考えられる順に並び替えて，開発者に提示する (Step4)．並び替えは，筆者があらかじめ決めた基準に重みを与えたものを組み合わせて行った．この重みは，既存のソースコードでツールを使用したときに，定義されているメソッド名ができるだけ上に表示されるように決定した．

以降では，まず事前準備として，辞書作成と拡張内容について説明する．次に，手法の説明の準備として，メソッドの生成パターンとは何か説明する．その後，各ステップの詳細を説明し，最後にパラメータチューニングについて説明する．

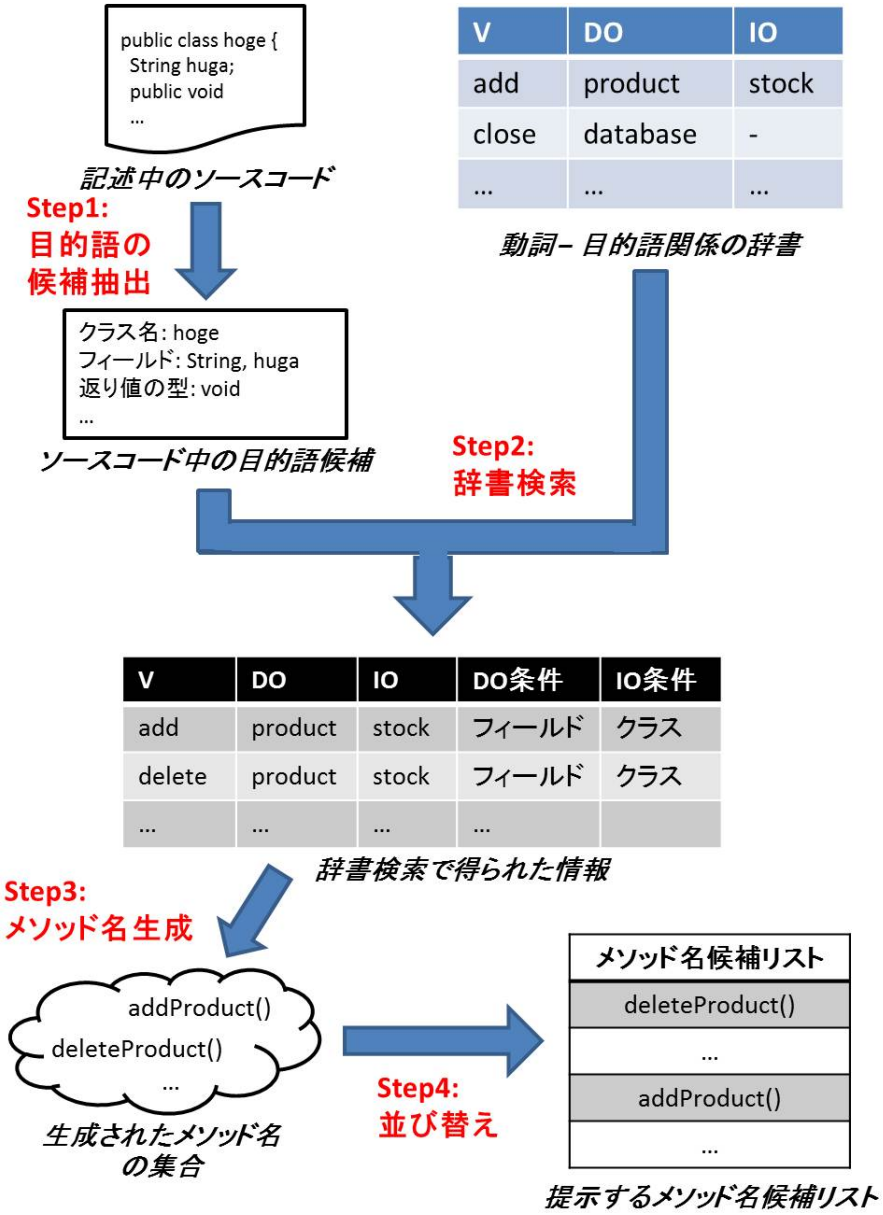


図 2: メソッド名候補の生成の流れ

### 3.1 事前準備

メソッド名候補の提示の前に，事前準備として辞書を作成する．辞書は背景で述べた動詞-目的語関係の辞書生成手法に拡張を行い，ソースコード集合から辞書を生成する．

拡張内容は辞書に収録する情報の追加である．並び替えのときに，辞書生成で抽出した三つ組を復元するような組み合わせ方でメソッド名候補を生成したかどうかの情報を利用するため，抽出に用いた抽出パターンを各三つ組に対して追加する．

### 3.2 生成パターン

目的語の候補として抽出した名詞を動詞-目的語関係の辞書から検索する際，直接目的語 (DO) と間接目的語 (IO) をどの識別子の組み合わせで検索したかによって生成されるメソッド名が異なる．検索時の識別子の組み合わせと生成されるメソッド名 (以下，これを生成パターンと呼ぶ) は，記述したいメソッドの返り値の型が書かれていない場合，返り値の型として void が書かれている場合，返り値の型として void 以外が書かれている場合の 3 つの場合で異なる．返り値の型が書かれていないときの生成パターンを表 2，返り値の型が void のときの生成パターンを表 3，返り値の型が void 以外のときの生成パターンを表 4 に示す.「生成されるメソッド名」における “verb”, “DO”, “IO” は 3 つ組の動詞，直接目的語，間接目的語である.「IO の検索条件」の “-” は三つ組の内容が空であることを意味する．“フィールド名” は取得したフィールド変数の名前それぞれでメソッド名を生成することを意味する．“Type” は何らかのクラス名が入ることをユーザに示す文字列であり，実際に “Type” という文字列でメソッド名が生成される．これらの生成パターンは，表 1 の抽出パターンに基づいて筆者が作成した．

### 3.3 Step1．目的語の候補抽出

Step1 では，開発者が編集しているファイルのソースコードに構文解析を行い，目的語の候補としてカーソル位置から参照可能な名詞である識別子を抽出する．抽出は，その識別子の種類と名前の組<識別子の種類，識別子の名前>で抽出を行う．抽出する識別子の種類と名前は以下の 4 つである．

**インポートクラス名** インポートクラスの名前とその親クラスの名前

**クラス名** 定義クラスの名前とその親クラスの名前

**フィールド** フィールド変数の型名と名前

**返り値の型** 返り値の型名

表 2: 返り値が書かれていないときのメソッド名生成パターン

| DO の検索条件 | IO の検索条件 | 生成されるメソッド名 |
|---|---|---|
| クラス名 | - | void verb() |
| クラス名 | - | void verb+DO() |
| クラス名 | - | DO verb+DO() |
| クラス名 | インポートクラス名 | void verb+DO(IO) |
| クラス名 | インポートクラス名 | DO verb+DO(IO) |
| クラス名 | インポートクラス名 | Type verb(IO) |
| クラス名 | インポートクラス名 | IO verb+DO() |
| インポートクラス名 | クラス名 | void verb+DO(DO) |
| インポートクラス名 | クラス名 | void verb(DO) |
| インポートクラス名 | クラス名 | void verb+フィールド名 (DO) |
| インポートクラス名 | クラス名 | DO verb+DO() |
| インポートクラス名 | インポートクラス名 | DO verb+DO(IO) |
| フィールド | クラス名 | IO verb+DO() |

表 3: 返り値が void のときのメソッド名生成パターン

| DO の検索条件 | IO の検索条件 | 生成されるメソッド名 |
|---|---|---|
| クラス名 | - | verb() |
| クラス名 | - | verb+DO() |
| クラス名 | インポートクラス名 | verb+DO(IO) |
| インポートクラス名 | クラス名 | verb(DO) |
| インポートクラス名 | クラス名 | verb+DO(DO) |
| インポートクラス名 | クラス名 | verb+フィールド名 (DO) |

### 3.4 Step2. 辞書検索

Step1 で抽出した名前で動詞-目的語関係の辞書を検索し，検索結果情報を生成する．検索は辞書の直接目的語 (DO) と間接目的語 (IO) に対して行われ，返り値が未記入のときは表 2，返り値が void のときは表 3，返り値が void 以外のときは表 4 の各検索条件に従う．辞書検索によって生成される検索結果情報は，<検索にヒットした三つ組，事前準備で追加した三つ組の抽出パターン，DO の検索条件，IO の検索条件>の組である．DO と IO の検索はどちらも前方一致で行われ，インポートクラスと定義クラスについては CamelCase のルールで単語ごとに分割し，そのそれぞれの単語でも前方一致検索を行う．

検索手順は以下の通りである．

1. DO と IO それぞれの検索条件である識別子の名前を Step1 で得た目的語候補から抽出し，それぞれを「DO で検索する名前の集合」,「IO で検索する名前の集合」とする．
2. どちらかの集合が空であれば，次の検索条件に移る．
3. 抽出した集合の名前がクラス名，もしくはインポートクラス名であれば，その集合のそれぞれを CamelCase のルールで単語に分割し，各単語を元の名前とは別として集合に加える．
4. DO で検索する名前の集合と IO で検索する名前の集合の全ての組み合わせで辞書を前方一致検索する．
5. 辞書の項目がヒットするたびに，<ヒットした三つ組，その三つ組の抽出パターン，DO の検索条件，IO の検索条件>の組を出力する情報に加える．

### 3.5 Step3. メソッド名生成

Step2 で得た情報を入力として，その情報から<生成されたメソッド名，生成に用いた三つ組の抽出パターン>の組の集合を出力する．入力の情報を，ユーザが返り値の型を書いていなければ表 2 ，ユーザが書いた返り値の型が void であれば表 3 ，返り値の型が void 以外であれば表 4 の生成パターンに従ってメソッド名を生成する．

### 3.6 Step4. 並び替え

Step3 で出力されたメソッド名を，編集中のソースコードで出現する可能性が高いものが上位に来るように並び替え，ユーザに提示する．並び替えは，複数の基準に対してそれらを表す変数を定義し，各変数に適当な重みを与え，それらを組み合わせた値が高い順にメソッド名を配置することで行う．

編集中のソースコードに出現する可能性が高いメソッド名の基準として以下の 4 つを考えた．

**基準 1** 多くのプログラムで出現する三つ組を使用して生成されている

**基準 2** 辞書作成で抽出した三つ組を復元するような組み合わせで生成されている

**基準 3** ソースコード特有の語を使用して生成されている

**基準 4** 検索で完全一致した三つ組を使用して生成されている

これらの基準から，以下の 6 つの変数を用意した．

**C** 使用した 3 つ組がいくつのプロジェクトで使用されているか．値はプロジェクト数．(基準 1 より)

**P** 三つ組を作成するときに使用した識別子の組み合わせを復元するように生成されたメソッド名であるかどうか．組み合わせが一致していれば 1，一致していなければ 0 の値を取る．(基準 2 より)

**T** 基本型が使われているかどうか．基本型が使用されていなければ 2，直接目的語と間接目的語のどちらかが基本型であれば 1，どちらも基本型でなければ 0 の値を取る．(基準 3 より)

**V** メソッド名生成に使用した動詞を持つ三つ組が辞書中にいくつ存在するか．値は辞書中に存在する同じ動詞の三つ組数．逆数を使用する．(基準 3 より)

**M1** 直接目的語と間接目的語のどちらか片方のみが完全一致したかどうか．どちらかが完全一致したなら 1，それ以外なら 0 の値を取る．(基準 4 より)

**M2** 直接目的語と間接目的語の両方が完全一致したかどうか．両方が完全一致したなら 1，それ以外なら 0 の値を取る．(基準 4 より)

この各変数に重みとして，3.7 節で後述するパラメータチューニングによって決定した $[0, 1]$ の範囲の定数 cw，pw，tw，m1w，m2w，vw を与え，以下の計算式に適用し，値が大きいものから順に上から列挙する (以後，この値を優先度と呼ぶ)．この計算式は，各基準となる変数を足し合わせるものであり，V の値は他の値に比べて非常に大きくなるため自然対数をとった．

$$(cw \times C) + (pw \times P) + (tw \times T) + \frac{vw}{\ln(V+1)} + (m1w \times M1) + (m2w \times M2)$$

### 3.7 パラメータチューニング

3.6 節での並び替え優先度の計算に用いる各変数の重みを決定するためにパラメータチューニングを行った．このパラメータチューニングの手法には Simulated Annealing(以下 SA) を用いる．SA は金属加工の焼きなましを模した大域的最適化問題への確率的メタアルゴリズムであり，広大な探索空間内の与えられた関数の大域的最適解の近似を求める方法である [8]．

Step4 で使用すると決定した 6 つの変数のパラメータに対応する 6 次元の空間で SA を行う．パラメータチューニングには，辞書作成に用いたものとは別のソースコード集合を用いる．このソースコードそれぞれでメソッド名候補を表示したときに，そのソースコードで定義されているメソッド名ができるだけ上に表示されるようにパラメータチューニングを行う．大きいほどソースコードで定義されているメソッド名が上に表示されていることを示す評価値は，辞書作成に用いたものと別のソースコード集合それぞれの評価値の和であり，各ソースコードの評価値は以下の手順で求める．

1. ソースコードに Step3 までを実行して，メソッド名の集合を得る．
2. パラメータで並び替える．
3. 元のソースコードで定義されているメソッド名が，メソッドリスト内で何位に出現するか調査．
4. 各順位の逆数の和を得る．

パラメータチューニングの手順は以下の通りである．まず，初期パラメータとして各パラメータを $[0,1]$ の範囲の乱数としたものと近傍を確実に移動する確率を用意する．次に，このパラメータの近傍として，6 つのパラメータのどれかに絶対値が小さい正か負の値を加える．そして，その近傍のパラメータにおける評価値が前のパラメータの評価値より高い場合，パラメータを近傍に変更する．また，一定の確率でもパラメータを近傍に変更する．最後に，確率に $[0,1)$ の範囲の値をかけて確率を小さくする．この近傍移動から最後までを，確率がある一定の値以下になるまで繰り返すことで近似解が求まる．

# 4　ツールの実装

前節で述べた提案手法を Eclipse のコードアシスト機能を拡張することで実装した．本節では，そのツールの使用方法，機能について説明する．

## 4.1　メソッド名候補リストの提示機能

本ツールは Eclipse の Java エディタ内で動作し，ユーザがソースコード編集中にメソッドの命名に悩んだとき，メソッド名を記述する位置にカーソルがある状態で本ツールを起動すると，メソッド名の候補リストが Eclipse のコードアシストを通して表示される．「メソッド名を記述する位置」とは，クラス定義の内部かつ他のメソッド定義の外部であり，カーソル位置から行頭までの間に文字がない，もしくはメソッドの修飾子や型名が正しい順序で記述されているような位置のことである．ツールの起動はコードアシスト同様 Ctrl + Space キーを入力することで行われる．

ツールが起動されたときのコードの状態は，以下の 2 つのどちらかである．

1. 記述したいメソッドの返り値の型がまだ書かれていない状態．
2. 記述したいメソッドの返り値の型を書いた状態．

1 のときは，返り値の型を含んだメソッド名候補リストが表示される．図 3 の状態でツールを起動すると，図 4 のように本手法を利用した返り値の型を含むメソッド名候補リストがコードアシストに表示される．表示された一覧は図 5 のように方向キーで選択することができ，Enter キーを押すと図 6 のようにソースコードのカーソル位置に選択したメソッド名が挿入される．なお，メソッド名候補リストの各項目の先頭に表示されている数値は，並び替えで生成した優先度の最大値から各生成メソッドに対応する優先度を引いた値である．コードアシストの内容は辞書順に並べられるため，優先度が大きいほど，すなわち優先度の最大値から優先度を引いた値が小さいほど上に表示される．

2 のときは，返り値の型を含まないメソッド名候補リストが提示される．このときツールを起動すると図 7 のように，本手法を利用した返り値の型を含まないメソッド名候補リストがコードアシストに表示される．この場合も 1 のときと同様に，図 8 のように選択後に Enter キーを押すと選択したメソッド名が挿入される．

## 4.2　メソッド名候補リストの絞り込み機能

本ツールは，返り値の型が記述されている場合，入力した文字列でメソッド名候補リストを絞り込む機能を有する．その様子を図 9 に示す．返り値の型の後ろに文字列を記入した状

表 4: 返り値が void 以外のときのメソッド名生成パターン

| DO の検索条件 | IO の検索条件 | 生成されるメソッド名 |
|---|---|---|
| クラス名，返り値の型 | インポートクラス名 | verb+DO(IO) |
| クラス名，返り値の型 | - | verb+DO() |
| クラス名 | - | verb+フィールド名 () |
| クラス名 | 返り値の型 | verb+DO() |
| クラス名 | インポートクラス名 | verb(IO) |
| 返り値の型 | クラス名 | verb+DO() |
| フィールド | クラス名，返り値の型 | verb+DO() |

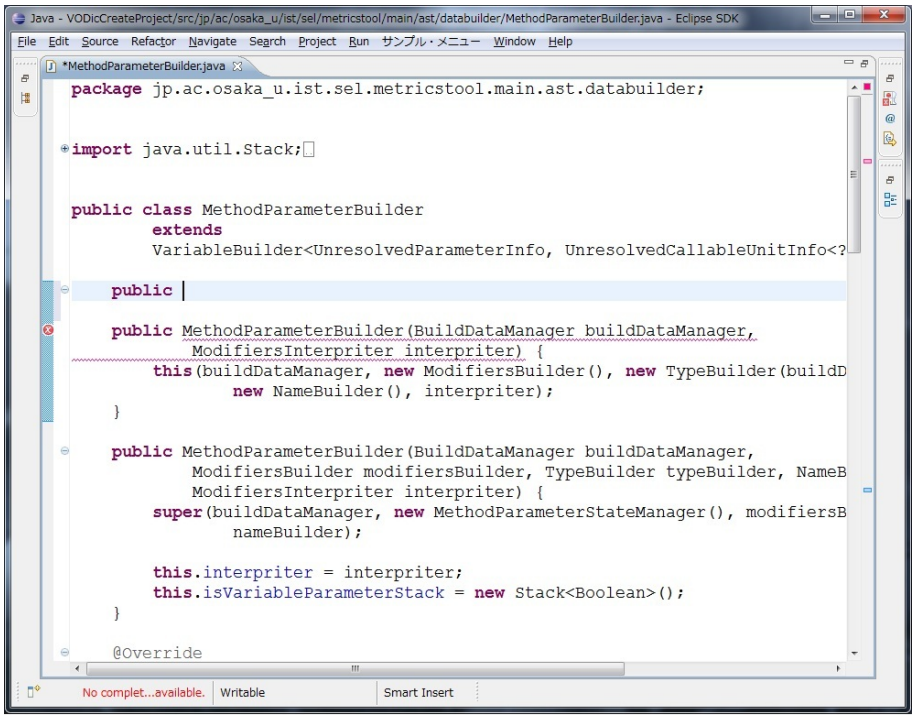


図 3: Eclipse の Java エディタでメソッド定義位置にカーソルがある状態 (返り値未記入時)

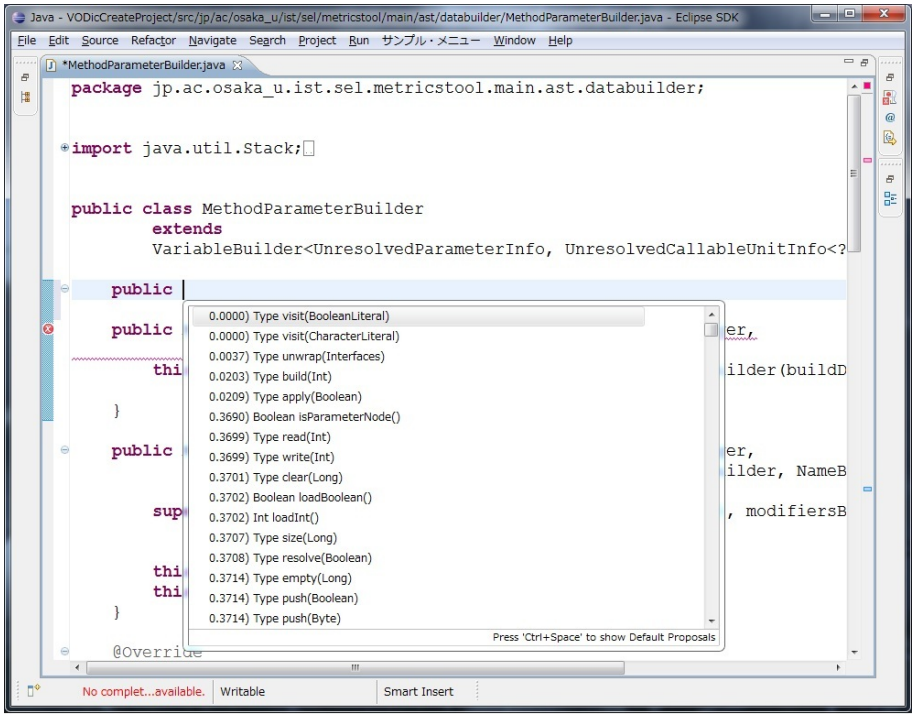


図 4: ツール起動直後 (返り値未記入時)

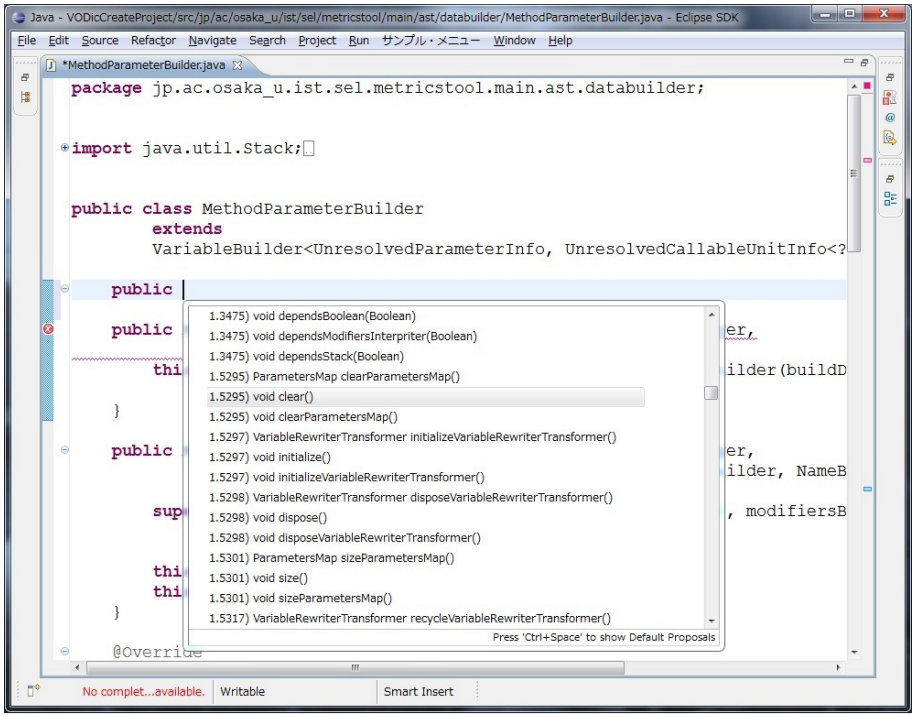


図 5: 一覧からメソッド名を選択 (返り値未記入時)

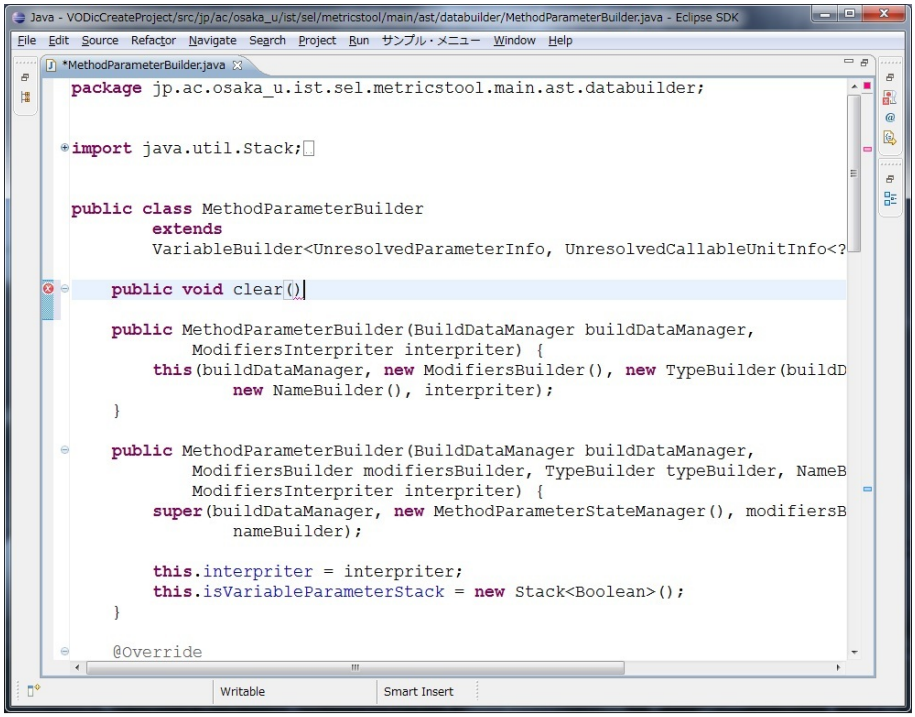


図 6: メソッド名の挿入 (返り値未記入時)

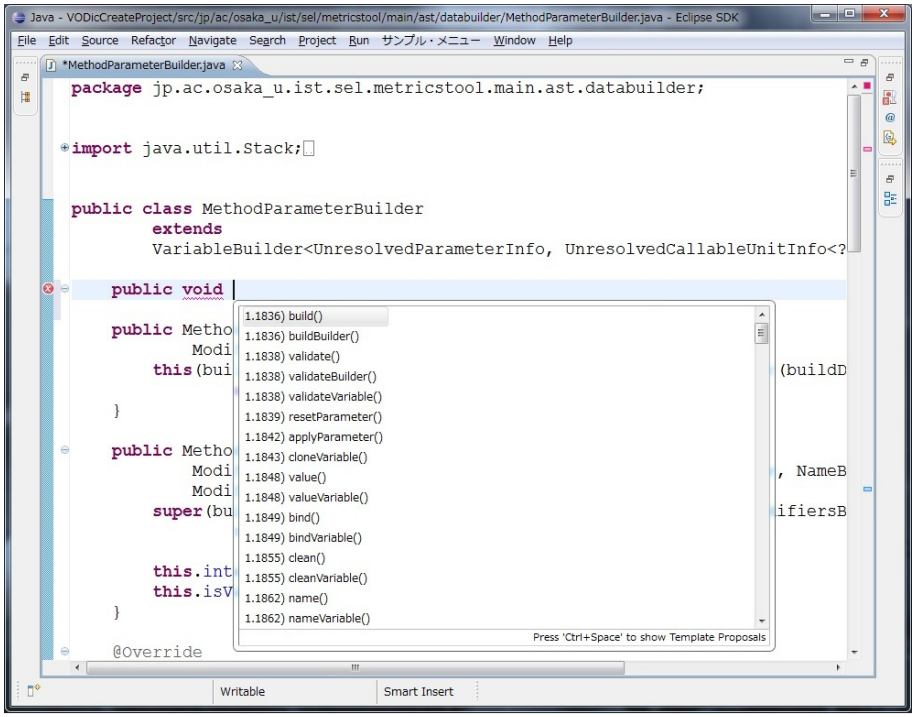


図 7: ツール起動直後 (返り値記入時)

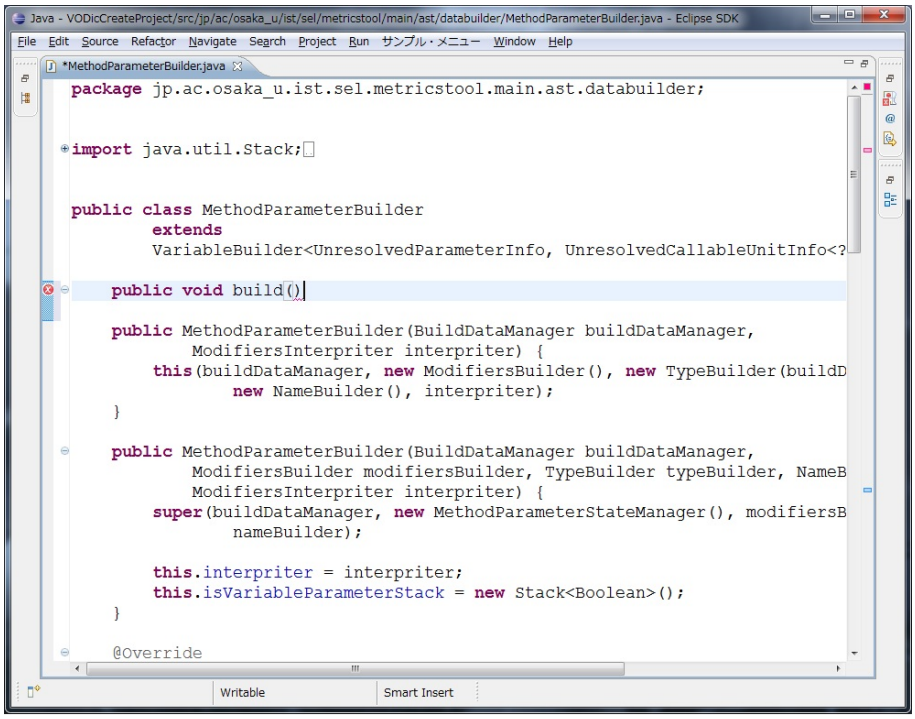


図 8: メソッド名の挿入 (返り値記入時)

態でコードアシストを起動すると，生成されたメソッド名候補リストのうち，その文字列が含まれているメソッド名のみを表示する．また，コードアシスト起動中に文字列を入力すると，文字を入力するたびに絞り込みが行われる．

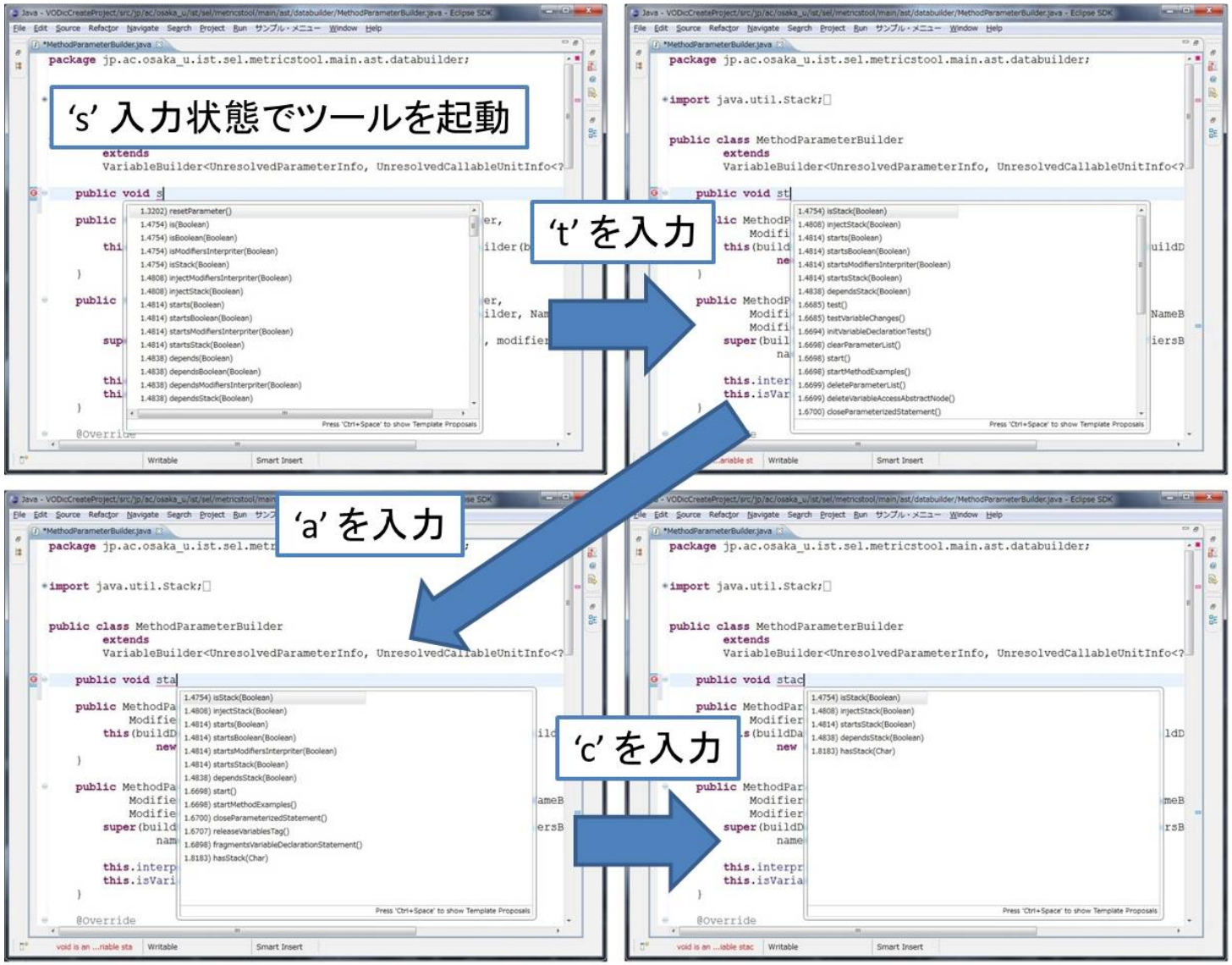


図 9: 絞り込みの様子

# 5　実験

本提案手法を実装したツールを使用することで，開発者がメソッドに対して適切な命名を行えるか調査するために実験を行った．そこで，実際に広く使われているアプリケーションのソースコードからメソッド名などを削除した課題を多数作成し，複数の被験者にツールを使用した場合と使用していない場合のそれぞれで，削除されたメソッド名を推測して解答してもらうという方法を用いた．評価はツールの有無で課題の正解率に変化があるかの比較，及びアンケートで被験者の主観的な意見の収集で行った．完全に正しい名前がつけられることと，メソッド名の振る舞いを表現する重要な動詞部分が正しい名前がつけられることのそれぞれを確認するために，課題の正解基準は元のメソッドとの完全一致と動詞一致の 2 つを用意した．

## 5.1　方法

実験では，メソッド名が記述されていないメソッドに対して開発者が適切な命名を行えるかを確認するため，複数の被験者にメソッド名が記述されていないソースコードを読んでもらい，そのメソッド名を考えてもらう方法を用いた．そこで，実際に広く使われているアプリケーションのソースコードからメソッド名やそのメソッド名の直接のヒントになり得る情報を削除した課題を複数作成し，各被験者に偏りのないよう割り当てたものを解答してもらった．

また，ツールに関する被験者の主観的な意見を聞くために，実験の際には被験者に対してアンケートも行った．アンケートは被験者の知識を尋ねるもの，各課題でリストに表示されたメソッド名について尋ねるもの，ツールを使用した感想について尋ねるものの 3 つを用意した．

### 5.1.1　課題作成

メソッド名が記述されていないメソッドに対して開発者が適切な命名を行えるかを確認するため，ソースコードからメソッド名などを削除した課題を手作業で複数作成した．課題は適切な名前がつけられたメソッド名とするため，実際に広く使われているアプリケーションのソースコードから作成した．また，特定の場面でのみツールが有効である場合に，実験結果が偶然良くなることを避けるため，課題作成には複数のドメインに属するアプリケーションのソースコードを使用し，各ドメインは課題となるメソッド名に同じ動詞は 2 つまでと決めた．本ツールはメソッドの命名に悩んだときに使用するため，既にどのような名前をつけるかが決まっている標準クラスのオーバーライドメソッドは課題には含めなかった．本課題は，被験者が課題のメソッドのボディや周りのフィールドなどの情報からメソッドの処理内

容を考慮して，その課題のメソッド名を推測して解答してもらうことを目的とするものであり，他の場所で課題のメソッドが使用されている部分や課題のメソッドが説明されている箇所を探すものではない．そのため，メソッド名の直接的なヒントになり得る他のメソッド，解答のメソッド名の動詞を含んだコメントについては削除した．同様の理由で，各課題のローカル変数名はその変数の型名の先頭を小文字にした名前に統一し，各ファイルに課題は最大 1 つとした．辞書作成やパラメータチューニングで用いたソースコードを課題とすることで正解率が高くなることを避けるため，課題作成には辞書作成とパラメータチューニングに使用したものとは別のソースコード集合を用意した．辞書作成に用いたソースコード一覧は付録に添付する．

以上の基準で課題を作成した結果，全 44 問の課題が完成した．課題は Database(DB)，GUI，Web Application(Web)，XML の 4 つのドメインに属するアプリケーション全 6 つのソースコードから作成した．全課題の一覧を表 5 に示す．

### 5.1.2 評価方法

本実験は，ツールの有無による課題の正解率の比較とアンケートの集計結果によって評価する．

課題は正解を元のメソッド名とし，正解基準は以下の 2 つとした．

- 元のメソッド名と完全一致した解答
- 元のメソッド名と動詞が一致した解答

ただし,「動詞が一致した解答」では，元のメソッド名で省略形となっている部分を省略せずに書いた場合も正解とした．

完全一致の他に動詞一致の基準を用意したのは，メソッド名の動詞はその振る舞いを示す，メソッド名で最も重要な部分だと考えたためである．

アンケートは，実験前に行う被験者の知識に関するアンケート，実験中に行うリストに関するアンケート，実験終了後に行うツールの使用感に関するアンケートの 3 つを行う．実験前のアンケートは，各被験者が用意した問題に使用した 4 つのドメインに対してどの程度の知識があるかを尋ねるものである．実験中のアンケートは，ツールありで課題を解答する際に表示された各リストが役に立ちそうかなどを尋ねるものである．実験終了後のアンケートは，そのツールを実験で使用してみた感想などを尋ねるものである．

表 5: 被験者に出題した課題一覧

| ドメイン | アプリケーション | 使用ファイル名 | 課題メソッド名 |
|---|---|---|---|
| DB | tinySQL | dbfFileTable.java | updateCol |
| DB | tinySQL | tinySQLGlobals.java | readLongNames |
| DB | tinySQL | tinySQLParser.java | validateTable |
| DB | tinySQL | tinySQLWhere.java | clearValues |
| DB | tinySQL | tsColumn.java | update |
| DB | tinySQL | UtilString.java | findTableAlias |
| DB | myoodb | MyOodbDatabase.java | close |
| DB | myoodb | AbstructVessel.java | createTarget |
| DB | myoodb | BasicWork.java | acquireVessel |
| DB | myoodb | StreamHelper.java | readIdentifier |
| GUI | ArgoUML | ArgoJFontChooser.java | updatePreview |
| GUI | ArgoUML | DisplayTextTree.java | formatTransitionLabel |
| GUI | ArgoUML | ProjectBrowser.java | saveScreenConfiguration |
| GUI | ArgoUML | ProjectSettingsDialog.java | showDialog |
| GUI | ArgoUML | SettingTabLayout.java | createProperty |
| GUI | ArgoUML | SettingTabProfile.java | refreshLists |
| GUI | ArgoUML | ActionAdjustGrid.java | buildGridAction |
| GUI | ArgoUML | ExplorerPopup.java | buildDirectionalCreateMenuItem |
| GUI | ArgoUML | ExplorerTreeNode.java | remove |
| GUI | ArgoUML | PerspectiveConfigurator.java | loadPerspectives |
| GUI | ArgoUML | PerspectiveManager.java | loadPerspectives |
| GUI | ArgoUML | UMLToDoItem.java | select |
| GUI | ArgoUML | DiagramFactory.java | createRenderingElement |
| GUI | ArgoUML | FigObjectFlowState.java | updateClassifierText |
| GUI | ArgoUML | UMLActivityDiagram.java | setup |
| Web | BBS | RegexFilter.java | applyFilter |
| Web | BBS | RegFilterBuilder.java | buildFilter |
| Web | BBS | Util.java | encodeHex |
| Web | BBS | ForumUploadFileImp.java | delUploadFile |
| Web | BBS | SysConfig.java | saveConfigs |
| Web | BBS | JSONArray.java | join |
| Web | BBS | XML.java | parse |
| XML | Xbeans | XMLOutputStream.java | initAttrs |
| XML | Xbeans | Project.java | deleteAssignment |
| XML | Xbeans | AttributeList.java | copy |
| XML | Xbeans | DataNodeImpl.java | removeAssignment |
| XML | Xbeans | DOMModel.java | addNode |
| XML | piccolo | CharStringConverter.java | convert |
| XML | piccolo | DocumentEntity.java | reset |
| XML | piccolo | XMLReaderReader.java | skip |
| XML | piccolo | XMLStreamReader.java | fillByteBuffer |
| XML | piccolo | FactoryFinder.java | findClassLoader |
| XML | piccolo | AttributesImpl.java | clear |
| XML | piccolo | ParserAdapter.java | makeException |

### 5.1.3 作業環境

作業中は実験実施者が部屋に待機しており，実験やツールに関する質問を自由に受け付けた．

課題の解答では，開発者がメソッド名を考える環境に近づけるため，オンラインの英和・和英辞書サービスの使用を許可し，それ以外のインターネットの参照は全て禁止した．また，被験者が正解の直接的なヒントを得ることを避けるため，解答中の課題のソースコード以外の参照は全て禁止し，Eclipse の画面には Package Explorer，Outline，Problems 以外のビューの表示を固定とした．

最後に，各被験者が作業を行ったノート PC の環境は以下の通りである．

**CPU** Core 2 Duo L7800 及び L7700

**メモリ** 2GB

**OS** Windows Vista 32bit

**画面の大きさ (解像度)** 14.1 型 (1400x1050 ドット)

**Eclipse のバージョン** 3.7.1

### 5.1.4 被験者

被験者はコンピュータサイエンス専攻に所属する大学生 8 人であり，いずれも Java プログラミング経験者である．

各被験者には実験の前に，それぞれのドメインに関する知識について事前アンケートを行った．アンケートは，4 つのドメインそれぞれに対して 4 つ質問をするものであり，実際に行ったアンケートは付録の図 11 に添付する．以下は 4 つの質問の要約である．

**(ア)** そのドメインを用いた Java プログラムを記述したことがあるか．

**(イ)** (ア) で「ある」と答えた方は，最近ではいつ記述したか．

**(ウ)** (ア) で「ある」と答えた方は，そのドメインを用いて作成した最大のプログラムは約何行だったか．

**(エ)** その他，各ドメインについてどのような知識があるか．

このアンケートで集計した被験者の各ドメインに関する知識を表 6 に示す．

表 6: 被験者の各ドメインに関する知識

| 被験者 | ドメイン | (ア) | (イ) | (ウ) | (エ) |
|---|---|---|---|---|---|
| A | DB | ない | | | 3 |
| | GUI | ある | 1 年前 | 1000 | 1 |
| | Web | ある | 3ヶ月前 | 800 | |
| | XML | ない | | | 1 |
| B | DB | ある | 3 年前 | 500 | 3 |
| | GUI | ある | 3ヶ月前 | 1000 | |
| | Web | ない | | | |
| | XML | ない | | | 1 |
| C | DB | ある | 3ヶ月前 | 1000 | 3 |
| | GUI | ある | 6ヶ月前 | 100 | |
| | Web | ない | | | |
| | XML | ある | 3ヶ月前 | 1000 | 1 |
| D | DB | ない | | | |
| | GUI | ある | 1ヶ月前 | 400 | 1,2 |
| | Web | ない | | | 1,2 |
| | XML | ない | | | |
| E | DB | ない | | | 1,3 |
| | GUI | ある | 3 年前 | 300 | |
| | Web | ない | | | |
| | XML | ない | | | 3 |
| F | DB | ある | 5ヶ月前 | 1000 | 1,3 |
| | GUI | ある | 6ヶ月前 | 1200 | 2 |
| | Web | ない | | | |
| | XML | ある | 6ヶ月前 | 1200 | 2,3 |
| G | DB | ある | 1ヶ月前 | 2000 | 1,3 |
| | GUI | ある | 1ヶ月前 | 2000 | 1 |
| | Web | ある | 2ヶ月前 | 3000 | 1 |
| | XML | ない | | | 1 |
| H | DB | ある | 1 年前 | 4000 | 2,3 |
| | GUI | ある | 4ヶ月前 | 3000 | 1 |
| | Web | ある | 2ヶ月前 | 1000 | 1,2 |
| | XML | ない | | | 1,3 |

### 5.1.5 課題の割り当て

まず，全体の作業時間や課題数を考慮して，被験者 1 人の課題数を 22 問と設定し，各問の制限時間は 3 分とした．全課題に均等に人数を配分すると，各課題の解答人数は 4 人となる．各被験者にツールありとツールなしの両方で偏りなく解答してもらうため，各課題をツールありで 2 人，ツールなしで 2 人が解答し，各被験者はツールありで 11 問，ツールなしで 11 問解答する．また，8 人の被験者のうち 4 人はツールありで先に解答し，残りの 4 人はツールなしで先に解答を行う．

各被験者の課題数を決めたら，次に各被験者に偏りがないように課題を割り当てた．まず，どの被験者もツールありツールなしの両方の場合で 4 つのドメインそれぞれを最低 1 問解答するように，かつ同じ被験者の組み合わせの課題が同程度の数になるように各被験者に課題を割り当てた．次に，ランダムで半分の被験者はツールありを先に，半分の被験者はツールなしを先に行うと決め，各被験者のツールありツールなしそれぞれの課題解答順をシャッフルした．最後に，課題の解答順が決まったそれぞれの被験者を誰にするか，8 人からランダムで決定した．

### 5.1.6 実施手順

課題の解答を行う前にまず，実験説明とツールの説明と練習を被験者に行った．実験説明では，実験で被験者が何を行うか，実験で被験者が守るべきルール，その他注意事項を説明した．ツールの説明と練習では，ツールをどのように使用するかについて一通り説明した後，問題とは別に用意した練習用のソースコード上で実際にツールを使用してもらった．

説明が終わったら，被験者に課題の解答を行ってもらった．ここではツールありで先に解答する被験者と，ツールなしで先に解答する被験者がおり，それぞれ解答が終了したらツールの有無を逆にして再び解答を開始した．各問の制限時間は 3 分であり，3 分たった時点で解答が思いついていればそれをすぐに記述して次の問題に，解答が思いついていなければ何も記述せずに次の問題に移ってもらった．また，3 分以内に解答が終わったら，3 分が経過するまで待機してもらった．解答中はボディ内についてはローカル変数の改名，コメントの記述などのソースコードの編集は自由に行って良いとしたが，ボディの外に関しては，ツールで表示されるリストの内容が変わり得るため，コメントの記述以外の編集は禁止とした．

ツールありの課題では，各課題が終わるたびに，表示されたリストに関するアンケートを行った．実際に行ったアンケートを付録の図 10 に添付する．

被験者全員がツールありツールなしの両方で全課題の解答を終了したら，最後にツールの使用感に関するアンケートに回答してもらった．実際に行ったアンケートを付録の図 12 に添付する．

## 5.2 結果

本節では実験を行った結果を，課題の解答結果とアンケートの結果それぞれについて示す．課題の正解率については 2 つの視点から結果を調査したが，ツールがあるときとないときで差を見ることはできても，検定で有意な差を示すことはできないという結果だった．アンケートについては，ツールに対して好意的な意見が多数あるという結果だった．

### 5.2.1 課題の解答結果

ツールによって課題の正解率が上がったのかどうかを確認するために，ツールありでの課題とツールなしでの課題の間に有意な差があるかを調べる．そこで，既に説明した 2 つの基準それぞれで被験者の解答を採点し，その結果に検定を行って比率の差を確認する．検定は，対応のある場合と対応のない場合の両方で行う．

まず，ツールがないときに比べてツールがあるときの方が正解者数が多い課題がどれくらいあったか調べる．ツールありの正解者数とツールなしの正解者数に差がある課題の数を表 7 に示す．この表から，2 つの基準のどちらにおいても，ツールありの正解者数がツールなしの正解者数に勝る課題の方が多かったということがわかる．

この結果に有意な差があるか確認した．そこで，ツールありの方が解答数が高いかを調べる片側検定で符号検定を行った．帰無仮説 $H1$ と対立仮説 $H1'$ を以下に示す．

**帰無仮説** $H1$ 「ツールありの方が正解者が多い問題の数」と「ツールなしの方が正解者が多い問題の数」に差はない．

**対立仮説** $H1'$ 「ツールありの方が正解者が多い問題の数」と「ツールなしの方が正解者が多い問題の数」に差がある．

その結果の P 値は以下のようになった．

**完全一致を正解とする基準** 有意確率 $P = 0.500$

表 7: ツールありの正解者数とツールなしの正解者数に差がある課題の数

| | 完全一致を正解とする | 動詞一致を正解とする |
|---|---|---|
| 正解者がツールあり $>$ ツールなし | 2 | 7 |
| 正解者がツールなし $>$ ツールあり | 1 | 4 |

**動詞一致を正解とする基準** 有意確率 $P = 0.274$

有意水準 0.05 で検定を行った結果，いずれも $P \not< 0.05$ であり帰無仮説を棄却できないため，検定で差があると示すことができない．よって，ツールがないときに比べてツールがあるときの方が正解者数が多い課題の数に有意な差は見られなかった．

次に，ツールありでの正解率がツールなしの正解率に比べてどれだけ高かったかを調べる．ツールの有無それぞれの正解数と不正解数を，正解基準が完全一致の場合を表 8 に，正解基準が動詞一致の場合を表 9 に示す．この表から，2 つの基準のどちらにおいても，ツールありの方が正解率が高いということがわかる．

この結果に有意な差があるか確認した．そこで，ツールありの方が解答数が高いかを調べる片側検定でフィッシャーの正確確率検定を行った．帰無仮説 $H2$ と対立仮説 $H2'$ を以下に示す．

**帰無仮説** $H2$ 「ツールがあるときの正解率」と「ツールがないときの正解率」に差はない．

**対立仮説** $H2'$ 「ツールがあるときの正解率」と「ツールがないときの正解率」に差がある．

その結果の P 値は以下のようになった．

**完全一致を正解とする基準** 有意確率 $P = 0.500$

**動詞一致を正解とする基準** 有意確率 $P = 0.097$

有意水準 0.05 で検定を行った結果，いずれも $P \not< 0.05$ であり帰無仮説を棄却できないため，検定で母代表値に差があると示すことができない．よって，ツールがあるときとツールがないときの正解率の差に有意な差は見られなかった．

### 5.2.2 アンケートの結果

ツールを使用した被験者の主観的な意見を調べるために，アンケート結果の集計を行った．その結果，ツールで表示されるリストは見やすい，表示されるメソッド名も役に立ちそうで

表 8: 完全一致を正解とする場合のツールの有無それぞれの正解数

| | 正解 | 不正解 | 合計 |
|---|---|---|---|
| ツールあり | 2 | 86 | 88 |
| ツールなし | 1 | 87 | 88 |
| 合計 | 3 | 139 | 176 |

ある，ツールを開発で実際に使いたいなど好意的な意見が多かった．一方，メソッド名リストの表示時間が少し遅いという意見も見られた．

**表示されたリストに関するアンケート**　ツールで表示されたリストに対して被験者がどのように感じたかを見るため，ツールで表示されたリストに関するアンケートについて調べる．「ツールで表示されたメソッド名は、実際にそのクラスで使用されそうな名前だと思ったか」という質問に対する回答を表 10 ,「メソッド名リストに解答で用いた名前が表示されたか」という質問に対する回答を表 11 ,「解答に用いた，もしくは解答のヒントとなったメソッド名を、リストからすぐに見つけることができたか」という質問に対する回答を表 12 に示す．

「ツールで表示されたメソッド名が実際にそのクラスで使用されそうな名前だと思ったか」の意見でそう思う以上の回答が全回答数の 88 の過半数 45 であることから，表示するメソッド名の候補は編集中のソースコードにある程度適した内容であることがわかる．また，メソッド名候補リストに表示されたメソッド名を使用してくれた人が多く，表示される候補がメソッド名として正しい構造であろうことがわかる．

**ツールの使用感に関するアンケート**　被験者がツールを使用してどう思ったかを見るために，ツールの使用感に関するアンケートについて調べる．「ツールで表示されるメソッド名リストは見やすかったか」という質問に対する回答を表 13 ,「メソッド名リスト表示までの待ち時間はどう感じたか」という質問に対する回答を表 14 ,「プログラムを書いているときにこのツールを使用したいと思ったか」という質問に対する回答を表 15 に示す．

「ツールのよかったところはどこですか」の質問に対する回答として以下のような意見があった．

- 思ったよりも良さそうな候補が多数あり便利だと思った．
- たとえ一覧から選択しなくても，メソッドの命名の参考にはなる．

表 9: 動詞一致を正解とする場合のツールの有無それぞれの正解数

| | 正解 | 不正解 | 合計 |
|---|---|---|---|
| ツールあり | 16 | 72 | 88 |
| ツールなし | 9 | 79 | 88 |
| 合計 | 25 | 151 | 176 |

表 10: 「ツールで表示されたメソッド名が実際にそのクラスで使用されそうな名前だと思ったか」のアンケート結果

| 選択肢 | 回答数 |
|---|---|
| とてもそう思う | 13 |
| そう思う | 32 |
| どちらとも言えない | 15 |
| あまりそう思わない | 22 |
| 全くそう思わない | 6 |

表 11: 「メソッド名リストに解答で用いた名前が表示されたか」のアンケート結果

| 選択肢 | 回答数 |
|---|---|
| 解答に用いた完全なメソッド名が表示された | 44 |
| 解答に用いた動詞を使用したメソッド名が表示された | 17 |
| 解答のヒントとなるメソッド名は表示された | 3 |
| 表示されなかった | 24 |

表 12: 「解答に用いた，もしくは解答のヒントとなったメソッド名を、リストからすぐに見つけることができたか」のアンケート結果

| 選択肢 | 回答数 |
|---|---|
| すぐに見つけた | 27 |
| 少し探して見つけた | 19 |
| 少し苦労して見つけた | 7 |
| かなり苦労して見つけた | 1 |

表 13: 「ツールで表示されるメソッド名リストは見やすかったか」のアンケート結果

| 選択肢 | 回答数 |
|---|---|
| とても見やすかった | 1 |
| 見やすかった | 7 |
| どちらとも言えない | 0 |
| やや見にくかった | 0 |
| 非常に見にくかった | 0 |

表 14: 「メソッド名リスト表示までの待ち時間はどう感じたか」のアンケート結果

| 選択肢 | 回答数 |
|---|---|
| 十分に早かった | 0 |
| 実用に支障がない程度には早かった | 4 |
| 実用可能ではあるが少し遅かった | 4 |
| 実用に支障があるぐらい遅かった | 0 |

表 15: 「プログラムを書いているときにこのツールを使用したいと思ったか」のアンケート結果

| 選択肢 | 回答数 |
|---|---|
| とてもそう思う | 1 |
| そう思う | 4 |
| どちらとも言えない | 1 |
| あまりそう思わない | 1 |
| 全くそう思わない | 1 |

- 動詞が思いついたときに，絞り込み機能で目的語のバリエーションのヒントが得られるところは便利.

「ツールの悪かったところはどこですか」の質問に対する回答として以下のような意見があった.

- 実行時間が気になる.
- 並び替えの精度に疑問を感じた.
- クラスに全く関係なさそうなメソッド名が多数表示されている.

「その他自由な意見，感想を書いてください」には以下のような意見があった.

- 結果をどのように訂正したかを学習すると面白そう.

ツールで表示されるメソッド名は全被験者が見やすかったと回答しており，メソッド名の提示方法としては正しいものであったことがわかる．また，半分以上の被験者がプログラムを書いているときに本ツールを使用したいと回答しており，ツールの良かったところとして「良さそうな候補が多数ある」や「メソッドの命名の参考になる」と回答していることから，ツールの使用感は悪くないものであるとわかる．一方,「メソッド名リスト表示までの待ち時間はどう感じたか」のアンケートの半分が少し遅いと回答していることや，悪かった意見として実行時間が気になるという意見があることから，ツールの実行速度は改善の必要があるとわかる.

### 5.3 考察

ツールがあるときと無いときで有意な差があると言えなかった原因としては大きく分けて二通り考えられる．1 つは適切な実験が行えなかったこと，もう 1 つはツールが適切なメソッド名提示を行えなかったことである.

1 つ目は実験についてである．表 8 を見てわかるように，正解 15 問に対して不正解が 151 問と不正解の数が非常に多い．これは被験者に出題する問題が難しすぎたことが原因だと考えられる.

2 つ目はツールについての問題である.「解答に用いた名前が表示されたか」で表示されなかったという回答が全回答の約 $1/4$ であることや，ツールの悪かったところはどこかという自由記述アンケートに「並び替えの精度に疑問を感じた」とあることから，メソッドリストの並び替えが適切に行えておらず，被験者が適切なメソッド名を見つけることができなかったとわかる．これは並び替えの手法が適切に行われていないと考えられる．そのため，本手

法のメソッドの候補の並び替えの部分については再度考え直す必要がある．また，解答に用いた名前が表示されなかったという意見が多かったことに加え,「そのクラスで使用されそうな名前だと思ったか」であまりそう思わない以下が全回答の 1/4 であったことから，そもそも正解のメソッド名を生成する三つ組が辞書に収録されておらず，適切なメソッド名が表示されなかったことがわかる．これは辞書のさらなる大規模化によって解決できると考えている．

有意な差があると言えなかった一方で，ツールの使用感に関するアンケートでは，ツールを使用したいと思う人が 8 人中 5 人いるなど高評価を得ることができた．そのため，メソッド名の候補を開発者に提示して命名支援を行うという提案手法のアプローチは間違っておらず，上で述べたツールの改善を行うことで本手法はメソッドの命名支援として有効だと考える．

本実験ではツールの使用時間および実験の経過時間に伴うツール使用や命名への慣れが考えられるが，これは被験者への問題の割り当てや解答の順序を十分ランダム化することで解消した．

## 6　関連研究

ソースコード中の動詞や目的語を抽出していくつかの問題に適用する手法が提案されている．本節ではそれらの関連研究について述べる．

早瀬らはオブジェクト指向プログラムのソースコード中のメソッドから動詞と直接目的語と間接目的語を抽出し，フィルタリングを行うことで複数のプログラムに出現する動詞-目的語関係を収録した辞書を作成している [4]．本研究ではその辞書を拡張したものを使用してメソッド名の生成を行う．

Shepherd や Fry らはオブジェクト指向プログラムのソースコード中のメソッドやコメントから抽出した動詞と直接目的語の組を Feature Location 問題や Aspect Mining に適用している [3, 7].

また，Hill らはオブジェクト指向プログラムのソースコード中のメソッドから動詞と直接目的語と間接目的語の三つ組を抽出し，Feature Location 問題や Aspect Mining に適用する手法を提案した [5].

# 7　まとめと今後の課題

不適切なメソッド名がつけられることによって可読性が下がりプログラム理解に時間がかかるという問題において，命名に特有の難しさがあるメソッド名に着目した．そこで，本研究ではソースコード中に出現する動詞-目的語関係を収録した既存の辞書を用いて，開発者にメソッドの命名を支援する手法を提案し，その手法に基づいてツールを作成した．

作成したツールを使うことで開発者が適切な命名を行えるかどうか確認する実験を複数の被験者に対して実施したところ，ツールがあるときと無いときで有意な差があると言えなかった一方で，ツールの使用感に関するアンケートでは高い評価を得ることができた．そのため，開発者にメソッド名を提示するという手法は間違っておらず，改善を行うことで本手法はメソッドの命名支援になると考えた．

今後の課題としては，まず今回の実験で問題があった部分を改善し，より精緻な実験を行うことで正確な評価をする必要がある．その際，アンケートで指摘された点，考察で述べたツールの問題点については改善を行う．さらに，メソッド名候補の作成に辞書以外の情報，例えばユーザが記述したテキストファイルなどを使用する，既存の悪いメソッド名を本手法で改善する，メソッド名以外の識別子の命名支援を行うと言ったツールの機能追加も検討している．

## 謝辞

本研究において, 常に適切な御指導および御助言を賜りました大阪大学大学院情報科学研究科コンピュータサイエンス専攻井上 克郎教授に心より深く感謝いたします.

本研究において, 適切な御指導および御助言を賜りました大阪大学大学院情報科学研究科コンピュータサイエンス専攻松下 誠准教授に深く感謝いたします.

本研究において, 逐次適切な御指導および御助言を頂きました大阪大学大学院情報科学研究科コンピュータサイエンス専攻石尾 隆助教に深く感謝いたします.

本研究において, 常時適切な御指導および御助言を頂きました筑波大学システム情報系早瀬 康裕助教に深く感謝いたします.

本研究において, 数多くの御指導および御助言を頂きました大阪大学大学院情報科学研究科コンピュータサイエンス専攻眞鍋 雄貴特任助教に深く感謝いたします.

本研究において, 様々な御指導および御助言を頂きました大阪大学大学院情報科学研究科コンピュータサイエンス専攻鹿島 悠氏に深く感謝いたします.

最後に，その他様々な御指導，御助言等を頂いた大阪大学大学院情報科学研究科コンピュータサイエンス専攻井上研究室の皆様に深く感謝いたします.

## 参考文献


[1] Code conventions for the java programming language: 9. naming conventions. http://www.oracle.com/technetwork/java/javase/documentation/codeconventions-135099.html.

[2] Eclipse. http://www.eclipse.org/.

[3] Z.P. Fry, D. Shepherd, E. Hill, L. Pollock, and K. Vijay-Shanker. Analysing source code: looking for useful verb-direct object pairs in all the right places. *IET Software*, Vol. 2, No. 1, pp. 27–36, 2008.

[4] Yasuhiro Hayase, Yu Kashima, Yuki Manabe, and Katsuro Inoue. Building domain specific dictionaries of verb-object relation from source code. *in Proceedings of the 15th European Conference on Software Maintenance and Reengineering(CSMR '11)*, pp. 93–100, 2011.

[5] Emily Hill, Lori Pollock, and K. Vijay-Shanker. Automatically capturing source code context of nl-queries for software maintenance and reuse. *Proceedings of the 31st International Conference on Software Engineering(ICSE '09)*, pp. 232–242, 2009.

[6] Dawn Lawrie, Christopher Morrell, Henry Feild, and David Binkley. What 's in a name? a study of identifiers. *Proceedings of the 14th IEEE International Conference on Program Comprehension(ICPC '06)*, pp. 3–12, 2006.

[7] David Shepherd, Lori Pollock, and K. Vijay-Shanker. Towards supporting on-demand virtual remodularization using program graphs. *in Proceedings of the 5th international conference on Aspect-oriented software development(AOSD '06)*, pp. 3–14, 2006.

[8] 伊藤斉志. C による探索プログラミング – 基礎から遺伝的アルゴリズムまで. オーム社, 2008.

# 付録

## A 実験使用の辞書を作成するために用いたソースコード集合一覧

表 16: apache.org

| プロジェクトディレクトリ名 | java ファイル数 | 総行数 |
|---|---|---|
| ant/ | 1238 | 258385 |
| beehive/ | 2933 | 369641 |
| cayenne/ | 2989 | 352995 |
| cocoon/ | 2844 | 494003 |
| commons-attributes/ | 56 | 6887 |
| commons-beanutils/ | 225 | 67779 |
| commons-betwixt/ | 393 | 47654 |
| commons-chain/ | 97 | 17702 |
| commons-cli/ | 103 | 18952 |
| commons-codec/ | 48 | 12784 |
| commons-collections/ | 469 | 110889 |
| commons-compress/ | 62 | 10457 |
| commons-configuration/ | 199 | 65918 |
| commons-csv/ | 20 | 3990 |
| commons-daemon/ | 12 | 1928 |
| commons-dbcp/ | 88 | 22454 |
| commons-dbutils/ | 41 | 6386 |
| commons-digester/ | 171 | 30527 |
| commons-discovery/ | 47 | 6127 |
| commons-el/ | 62 | 12321 |
| commons-email/ | 24 | 7952 |
| commons-fileupload/ | 41 | 8767 |
| commons-finder/ | 11 | 1285 |
| commons-flatfile/ | 61 | 5744 |
| commons-functor/ | 309 | 32326 |
| commons-i18n/ | 32 | 3494 |
| commons-id/ | 62 | 8578 |
| commons-io/ | 157 | 34210 |
| commons-javaflow/ | 93 | 9314 |
| commons-jci/ | 74 | 8274 |
| commons-jelly/ | 551 | 63241 |
| commons-jexl/ | 108 | 16432 |
| commons-jnet/ | 4 | 256 |
| commons-jxpath/ | 213 | 38801 |
| commons-lang/ | 247 | 100805 |
| commons-launcher/ | 17 | 4120 |
| commons-logging/ | 65 | 13288 |
| commons-math/ | 626 | 120572 |
| commons-modeler/ | 43 | 11070 |
| commons-monitoring/ | 99 | 9322 |
| commons-nabla/ | 126 | 10560 |
| commons-net/ | 204 | 43051 |
| commons-openpgp/ | 17 | 1785 |
| commons-performance/ | 18 | 3094 |
| commons-pipeline/ | 80 | 9254 |
| commons-pool/ | 50 | 17365 |
| commons-primitives/ | 457 | 54651 |

| | | |
|---|---|---|
| commons-scxml/ | 144 | 24751 |
| commons-transaction/ | 60 | 12340 |
| commons-validator/ | 128 | 26566 |
| commons-vfs/ | 310 | 47747 |
| db-derby-code/ | 2649 | 1003507 |
| db-torque-runtime/ | 107 | 30547 |
| excalibur/ | 941 | 155897 |
| felix/ | 2210 | 354850 |
| forrest/ | 232 | 33001 |
| hadoop-core/ | 1434 | 298206 |
| hivemind/ | 782 | 73101 |
| httpcomponents-httpclient/ | 312 | 51192 |
| httpcomponents-httpcore/ | 409 | 66452 |
| httpcomponents-oac.hc3x/ | 277 | 64597 |
| jackrabbit/ | 2471 | 451862 |
| jakarta-bcel/ | 434 | 63584 |
| jakarta-bsf/ | 149 | 20199 |
| jakarta-cactus/ | 296 | 47984 |
| jakarta-ecs/ | 391 | 70793 |
| jakarta-jcs/ | 476 | 87872 |
| jakarta-jmeter/ | 828 | 146384 |
| jakarta-oro/ | 98 | 17197 |
| jakarta-regexp/ | 14 | 5587 |
| james-server/ | 535 | 102620 |
| lenya/ | 3622 | 602967 |
| logging-chainsaw/ | 112 | 24626 |
| logging-log4j/ | 302 | 54620 |
| lucene/ | 1037 | 215926 |
| maven-components/ | 584 | 82245 |
| mina/ | 549 | 84625 |
| myfaces-tobago/ | 773 | 83636 |
| ode/ | 1249 | 142460 |
| poi/ | 2003 | 349983 |
| shale/ | 5197 | 891069 |
| struts1/ | 671 | 148326 |
| struts2/ | 1035 | 120877 |
| synapse-java/ | 654 | 85217 |
| tapestry/ | 1759 | 212616 |
| tomcat/ | 1204 | 340150 |
| turbine-core/ | 452 | 97846 |
| velocity-anakia/ | 15 | 2564 |
| velocity-dvsl/ | 20 | 3501 |
| velocity-engine/ | 381 | 73101 |
| velocity-texen/ | 20 | 3056 |
| velocity-tools/ | 147 | 40082 |
| webservices-scout/ | 119 | 17226 |
| xerces/ | 814 | 253364 |
| xmlbeans/ | 997 | 229874 |
| xml-commons/ | 523 | 62346 |
| xmlgraphics-commons/ | 304 | 53767 |
| xmlgraphics-fop/ | 1282 | 227278 |
| xml-security/ | 415 | 79035 |
| xml-xindice/ | 469 | 72310 |

表 17: eclipse.org

| プロジェクトディレクトリ名 | java ファイル数 | 総行数 |
|---|---|---|
| org.apache.lucene/ | 10 | 3799 |
| org.eclipse.ant.core/ | 36 | 6889 |
| org.eclipse.ant.tests.core/ | 20 | 2538 |
| org.eclipse.ant.tests.ui/ | 48 | 7844 |
| org.eclipse.ant.ui/ | 311 | 53760 |
| org.eclipse.compare.examples.xml/ | 28 | 5832 |
| org.eclipse.compare.examples/ | 3 | 267 |
| org.eclipse.compare.tests/ | 16 | 2828 |
| org.eclipse.compare/ | 177 | 41662 |
| org.eclipse.core.applicationrunner/ | 2 | 222 |
| org.eclipse.core.commands/ | 79 | 12814 |
| org.eclipse.core.contenttype/ | 34 | 5210 |
| org.eclipse.core.databinding.beans/ | 14 | 3610 |
| org.eclipse.core.databinding/ | 183 | 23878 |
| org.eclipse.core.expressions.tests/ | 18 | 1999 |
| org.eclipse.core.expressions/ | 45 | 5315 |
| org.eclipse.core.filebuffers.tests/ | 23 | 5034 |
| org.eclipse.core.filebuffers/ | 39 | 7444 |
| org.eclipse.core.filesystem/ | 22 | 3991 |
| org.eclipse.core.jobs/ | 31 | 7135 |
| org.eclipse.core.net/ | 29 | 4421 |
| org.eclipse.core.resources.compatibility/ | 54 | 7996 |
| org.eclipse.core.resources.spysupport/ | 2 | 98 |
| org.eclipse.core.resources/ | 254 | 60944 |
| org.eclipse.core.runtime.compatibility.auth/ | 8 | 1094 |
| org.eclipse.core.runtime.compatibility.registry/ | 6 | 542 |
| org.eclipse.core.runtime.compatibility/ | 36 | 7737 |
| org.eclipse.core.runtime.osgi/ | 100 | 18798 |
| org.eclipse.core.runtime/ | 23 | 8151 |
| org.eclipse.core.tests.harness/ | 35 | 4637 |
| org.eclipse.core.tests.net/ | 4 | 718 |
| org.eclipse.core.tests.resources.saveparticipant/ | 4 | 823 |
| org.eclipse.core.tests.resources.saveparticipant1/ | 1 | 196 |
| org.eclipse.core.tests.resources.saveparticipant2/ | 1 | 207 |
| org.eclipse.core.tests.resources.saveparticipant3/ | 2 | 256 |
| org.eclipse.core.tests.resources/ | 255 | 61797 |
| org.eclipse.core.tests.runtime/ | 108 | 21163 |
| org.eclipse.core.tools.resources/ | 29 | 5103 |
| org.eclipse.core.tools/ | 78 | 8485 |
| org.eclipse.core.variables/ | 16 | 1987 |
| org.eclipse.debug.core/ | 177 | 27370 |
| org.eclipse.debug.examples.core/ | 27 | 3760 |
| org.eclipse.debug.examples.ui/ | 52 | 4530 |
| org.eclipse.debug.ui/ | 735 | 127334 |
| org.eclipse.equinox.http.jetty/ | 5 | 851 |
| org.eclipse.help.appserver/ | 6 | 654 |
| org.eclipse.help.base/ | 128 | 20720 |
| org.eclipse.help.tests/ | 36 | 2550 |
| org.eclipse.help.ui/ | 100 | 19617 |
| org.eclipse.help.webapp/ | 52 | 8806 |
| org.eclipse.help/ | 82 | 8852 |
| org.eclipse.jdt.apt.core/ | 164 | 25650 |

| | | |
|---|---|---|
| org.eclipse.jdt.apt.pluggable.core/ | 14 | 1395 |
| org.eclipse.jdt.apt.pluggable.tests/ | 24 | 2215 |
| org.eclipse.jdt.apt.tests/ | 133 | 12482 |
| org.eclipse.jdt.apt.ui/ | 14 | 3246 |
| org.eclipse.jdt.compiler.apt.tests/ | 73 | 8523 |
| org.eclipse.jdt.compiler.apt/ | 44 | 9576 |
| org.eclipse.jdt.compiler.tool.tests/ | 5 | 1795 |
| org.eclipse.jdt.compiler.tool/ | 8 | 2741 |
| org.eclipse.jdt.core.manipulation/ | 51 | 7474 |
| org.eclipse.jdt.core.tests.builder/ | 24 | 12416 |
| org.eclipse.jdt.core.tests.compiler/ | 175 | 342261 |
| org.eclipse.jdt.core.tests.model/ | 4462 | 437806 |
| org.eclipse.jdt.core.tests.performance/ | 18 | 44737 |
| org.eclipse.jdt.core/ | 1183 | 428132 |
| org.eclipse.jdt.debug.jdi.tests/ | 95 | 11777 |
| org.eclipse.jdt.debug.tests/ | 365 | 117814 |
| org.eclipse.jdt.debug.ui/ | 394 | 60556 |
| org.eclipse.jdt.debug/ | 465 | 72365 |
| org.eclipse.jdt.junit.runtime/ | 29 | 2608 |
| org.eclipse.jdt.junit/ | 112 | 21974 |
| org.eclipse.jdt.junit4.runtime/ | 6 | 364 |
| org.eclipse.jdt.launching.j9/ | 25 | 5413 |
| org.eclipse.jdt.launching.macosx/ | 15 | 1917 |
| org.eclipse.jdt.launching/ | 102 | 22797 |
| org.eclipse.jdt.text.tests/ | 161 | 23925 |
| org.eclipse.jdt.ui.examples.javafamily/ | 42 | 4508 |
| org.eclipse.jdt.ui.examples.projects/ | 5 | 684 |
| org.eclipse.jdt.ui.tests.refactoring/ | 6851 | 120466 |
| org.eclipse.jdt.ui.tests/ | 220 | 99518 |
| org.eclipse.jdt.ui/ | 2031 | 495198 |
| org.eclipse.jface.databinding/ | 79 | 11290 |
| org.eclipse.jface.examples.databinding/ | 65 | 10039 |
| org.eclipse.jface.snippets/ | 72 | 14567 |
| org.eclipse.jface.tests.databinding.conformance/ | 29 | 3748 |
| org.eclipse.jface.tests.databinding/ | 198 | 26982 |
| org.eclipse.jface.text.tests/ | 19 | 3680 |
| org.eclipse.jface.text/ | 321 | 74182 |
| org.eclipse.jface/ | 377 | 100673 |
| org.eclipse.jsch.core/ | 17 | 1797 |
| org.eclipse.jsch.tests/ | 1 | 60 |
| org.eclipse.jsch.ui/ | 16 | 3583 |
| org.eclipse.ltk.core.refactoring.tests/ | 27 | 3296 |
| org.eclipse.ltk.core.refactoring/ | 138 | 23238 |
| org.eclipse.ltk.ui.refactoring.tests/ | 10 | 776 |
| org.eclipse.ltk.ui.refactoring/ | 127 | 20658 |
| org.eclipse.pde.p2.ui/ | 7 | 882 |
| org.eclipse.platform/ | 4 | 598 |
| org.eclipse.releng.basebuilder/ | 43 | 11343 |
| org.eclipse.releng.eclipsebuilder/ | 1 | 524 |
| org.eclipse.releng.tests/ | 2 | 1544 |
| org.eclipse.releng.tools/ | 47 | 7807 |
| org.eclipse.scripting.adapters.javascript/ | 1 | 129 |
| org.eclipse.scripting.examples/ | 4 | 1389 |
| org.eclipse.scripting.tests/ | 2 | 155 |
| org.eclipse.scripting/ | 216 | 38384 |

| | | |
|---|---|---|
| org.eclipse.sdk.tests-feature/ | 117 | 20615 |
| org.eclipse.search.tests/ | 22 | 2728 |
| org.eclipse.search/ | 152 | 24096 |
| org.eclipse.swt.examples.browser.demos/ | 5 | 751 |
| org.eclipse.swt.examples.browser/ | 2 | 79 |
| org.eclipse.swt.examples.controls/ | 3 | 116 |
| org.eclipse.swt.examples.launcher/ | 5 | 1012 |
| org.eclipse.swt.examples.layouts/ | 2 | 81 |
| org.eclipse.swt.examples.ole.win32/ | 4 | 1110 |
| org.eclipse.swt.examples.paint/ | 2 | 157 |
| org.eclipse.swt.examples/ | 139 | 36312 |
| org.eclipse.swt.graphics.text/ | 46 | 35701 |
| org.eclipse.swt.opengl.examples/ | 17 | 4481 |
| org.eclipse.swt.opengl/ | 17 | 4200 |
| org.eclipse.swt.snippets/ | 308 | 23486 |
| org.eclipse.swt.tests/ | 241 | 55513 |
| org.eclipse.swt.tools/ | 48 | 13905 |
| org.eclipse.swt/ | 1679 | 645217 |
| org.eclipse.team.core/ | 172 | 27526 |
| org.eclipse.team.cvs.core/ | 187 | 34854 |
| org.eclipse.team.cvs.ssh/ | 13 | 3057 |
| org.eclipse.team.cvs.ssh2/ | 7 | 726 |
| org.eclipse.team.cvs.ui/ | 322 | 64171 |
| org.eclipse.team.examples.filesystem/ | 101 | 12030 |
| org.eclipse.team.ftp/ | 18 | 1985 |
| org.eclipse.team.tests.core/ | 20 | 1765 |
| org.eclipse.team.tests.cvs.core/ | 83 | 18025 |
| org.eclipse.team.ui/ | 310 | 61080 |
| org.eclipse.team.webdav/ | 14 | 1477 |
| org.eclipse.test.performance.data/ | 1 | 73 |
| org.eclipse.test.performance/ | 56 | 9883 |
| org.eclipse.test/ | 4 | 1009 |
| org.eclipse.text.tests/ | 32 | 12894 |
| org.eclipse.text/ | 127 | 25160 |
| org.eclipse.tomcat/ | 8 | 1479 |
| org.eclipse.ua.tests/ | 118 | 12560 |
| org.eclipse.ui.browser/ | 51 | 7287 |
| org.eclipse.ui.carbon/ | 1 | 454 |
| org.eclipse.ui.cheatsheets/ | 124 | 17990 |
| org.eclipse.ui.cocoa/ | 2 | 573 |
| org.eclipse.ui.console/ | 56 | 9397 |
| org.eclipse.ui.editors.tests/ | 5 | 595 |
| org.eclipse.ui.editors/ | 124 | 29933 |
| org.eclipse.ui.examples.components/ | 25 | 2133 |
| org.eclipse.ui.examples.contributions/ | 31 | 2537 |
| org.eclipse.ui.examples.fieldassist/ | 8 | 1440 |
| org.eclipse.ui.examples.javaeditor/ | 39 | 3315 |
| org.eclipse.ui.examples.job/ | 16 | 1656 |
| org.eclipse.ui.examples.multipageeditor/ | 3 | 378 |
| org.eclipse.ui.examples.navigator/ | 7 | 777 |
| org.eclipse.ui.examples.presentation/ | 22 | 3560 |
| org.eclipse.ui.examples.propertysheet/ | 16 | 2864 |
| org.eclipse.ui.examples.rcp.browser/ | 11 | 1196 |
| org.eclipse.ui.examples.readmetool/ | 31 | 3577 |
| org.eclipse.ui.examples.undo/ | 14 | 1711 |

| | | |
|---|---|---|
| org.eclipse.ui.examples.views.properties.tabbed/ | 94 | 20005 |
| org.eclipse.ui.externaltools/ | 38 | 6962 |
| org.eclipse.ui.forms.examples/ | 28 | 3158 |
| org.eclipse.ui.forms/ | 80 | 21332 |
| org.eclipse.ui.ide.application/ | 6 | 2423 |
| org.eclipse.ui.ide/ | 531 | 120756 |
| org.eclipse.ui.internal.r21presentation/ | 14 | 6420 |
| org.eclipse.ui.intro.universal/ | 23 | 4444 |
| org.eclipse.ui.intro/ | 86 | 19597 |
| org.eclipse.ui.navigator.resources/ | 34 | 5471 |
| org.eclipse.ui.navigator/ | 137 | 23976 |
| org.eclipse.ui.net/ | 14 | 1676 |
| org.eclipse.ui.presentations.r21/ | 14 | 7335 |
| org.eclipse.ui.tests.browser/ | 11 | 870 |
| org.eclipse.ui.tests.forms/ | 10 | 1091 |
| org.eclipse.ui.tests.harness/ | 22 | 2930 |
| org.eclipse.ui.tests.navigator/ | 40 | 3818 |
| org.eclipse.ui.tests.performance/ | 74 | 7844 |
| org.eclipse.ui.tests.rcp/ | 22 | 2907 |
| org.eclipse.ui.tests.views.properties.tabbed/ | 105 | 7136 |
| org.eclipse.ui.tests/ | 696 | 91142 |
| org.eclipse.ui.tutorials.rcp/ | 23 | 853 |
| org.eclipse.ui.versioncheck/ | 2 | 112 |
| org.eclipse.ui.views.properties.tabbed/ | 36 | 6235 |
| org.eclipse.ui.views/ | 33 | 5832 |
| org.eclipse.ui.win32/ | 2 | 814 |
| org.eclipse.ui.workbench.compatibility/ | 2 | 392 |
| org.eclipse.ui.workbench.texteditor.tests/ | 13 | 2421 |
| org.eclipse.ui.workbench.texteditor/ | 160 | 39850 |
| org.eclipse.ui.workbench/ | 1309 | 287745 |
| org.eclipse.ui/ | 2 | 353 |
| org.eclipse.update.configurator/ | 29 | 6868 |
| org.eclipse.update.core/ | 276 | 49144 |
| org.eclipse.update.examples/ | 11 | 1262 |
| org.eclipse.update.scheduler/ | 8 | 1006 |
| org.eclipse.update.tests.core/ | 77 | 9235 |
| org.eclipse.update.ui/ | 100 | 17321 |
| org.eclipse.vcm.core.cvs.ext/ | 3 | 135 |
| org.eclipse.vcm.core.cvs.ssh/ | 10 | 2654 |
| org.eclipse.vcm.core.cvs/ | 103 | 9300 |
| org.eclipse.vcm.core/ | 63 | 8324 |
| org.eclipse.vcm.tests.core.cvs.ssh/ | 6 | 316 |
| org.eclipse.vcm.tests.core.cvs/ | 7 | 932 |
| org.eclipse.vcm.tests.core/ | 20 | 2802 |
| org.eclipse.vcm.ui.cvs/ | 33 | 4427 |
| org.eclipse.vcm.ui/ | 102 | 13966 |
| org.eclipse.webdav/ | 120 | 22315 |

表 18: sourceforge.org

| プロジェクトディレクトリ名 | java ファイル数 | 総行数 |
|---|---|---|
| acqlite/ | 2737 | 458733 |
| adempiere/ | 4143 | 1155868 |
| adito/ | 1982 | 306562 |

| | | |
|---|---|---|
| advanced-gwt/ | 111 | 19334 |
| agilewiki/ | 726 | 89373 |
| antlrv3ide/ | 197 | 20842 |
| aria/ | 1272 | 266199 |
| armedbear-j/ | 797 | 199355 |
| artifactory/ | 272 | 30348 |
| atunes/ | 520 | 94001 |
| avr-eclipse/ | 210 | 44985 |
| bt747/ | 324 | 87566 |
| cdk/ | 210 | 47586 |
| coopnet/ | 122 | 25729 |
| cruisecontrol/ | 912 | 137566 |
| daimonin/ | 20 | 4864 |
| dapper/ | 271 | 39001 |
| datacrow/ | 1764 | 348198 |
| davmail/ | 42 | 8304 |
| dbunit/ | 486 | 67912 |
| dimdim/ | 1219 | 174027 |
| dita-op/ | 69 | 8320 |
| dita-ot/ | 125 | 22789 |
| donnerlaparole/ | 170 | 27928 |
| dooble/ | 469 | 124880 |
| dozer/ | 568 | 41919 |
| eclim/ | 308 | 35199 |
| eclipsesql/ | 432 | 54875 |
| ejbca/ | 1255 | 221922 |
| erlide/ | 677 | 122251 |
| findbugs/ | 1828 | 220400 |
| firebird/ | 711 | 227708 |
| fireflyclient/ | 207 | 31052 |
| follow/ | 70 | 8333 |
| freecol/ | 485 | 144748 |
| freelims/ | 31 | 18289 |
| freemarker/ | 377 | 60990 |
| frostwire/ | 2626 | 547046 |
| fuber/ | 238 | 30440 |
| ganttproject/ | 1112 | 187868 |
| gdcm/ | 5 | 650 |
| geneontology/ | 1436 | 226718 |
| gep/ | 123 | 13936 |
| gham/ | 798 | 88893 |
| gpsmid/ | 202 | 36448 |
| hibernate/ | 3896 | 445894 |
| hibernate4gwt/ | 270 | 25333 |
| hipergate/ | 553 | 178173 |
| hyperic-hq/ | 3097 | 521323 |
| iplist/ | 7 | 2770 |
| j2s/ | 1898 | 463020 |
| jabref/ | 601 | 122809 |
| jailer/ | 77 | 27115 |
| java-ml/ | 258 | 34839 |
| jaxodraw/ | 307 | 70495 |
| jfreechart/ | 1130 | 288757 |
| jgnucashlib/ | 204 | 93576 |
| j-interop/ | 205 | 41578 |

| | | |
|---|---|---|
| jitterbit/ | 485 | 37527 |
| jmol/ | 439 | 158875 |
| jmri/ | 2249 | 352202 |
| jnetpcap/ | 174 | 39751 |
| josso/ | 472 | 60650 |
| jpen/ | 57 | 5822 |
| jpwsafe/ | 119 | 17327 |
| jspx-bay/ | 115 | 15973 |
| jupload/ | 80 | 20688 |
| jvlt/ | 185 | 21349 |
| jython/ | 837 | 276114 |
| keros/ | 1 | 11 |
| kmlcsv/ | 91 | 7713 |
| kolmafia/ | 206 | 83327 |
| liquibase/ | 643 | 53680 |
| logicaldoc/ | 298 | 44638 |
| makagiga/ | 835 | 167531 |
| makumba/ | 388 | 67444 |
| matrex/ | 1082 | 92132 |
| megamek/ | 1328 | 322713 |
| megameklab/ | 33 | 14266 |
| microemulator/ | 310 | 44598 |
| micrologˆ/ | 105 | 14285 |
| nekohtml/ | 38 | 12683 |
| obpm/ | 1100 | 169144 |
| octave/ | 79 | 18853 |
| ogre4j/ | 1241 | 272848 |
| omegat/ | 244 | 49305 |
| openbravopos/ | 483 | 65343 |
| opencsv/ | 17 | 2714 |
| openmodeller/ | 10 | 716 |
| opennms/ | 2600 | 480144 |
| openqrm/ | 1564 | 240107 |
| openxava/ | 2483 | 285838 |
| opproject/ | 578 | 141513 |
| osmius/ | 640 | 76786 |
| oswing/ | 960 | 191123 |
| pcgen/ | 2976 | 587511 |
| pdfsam/ | 490 | 82032 |
| phex/ | 809 | 167087 |
| pmd/ | 1501 | 179756 |
| posterita/ | 785 | 166038 |
| redmin-mylyncon/ | 57 | 8259 |
| regexpeditor/ | 30 | 2910 |
| reprap/ | 204 | 64229 |
| rodin-b-sharp/ | 4645 | 758507 |
| romaframework/ | 1155 | 90594 |
| saxon/ | 580 | 118117 |
| sblim/ | 4923 | 1612974 |
| schemaspy/ | 61 | 11605 |
| smartfrog/ | 2954 | 433726 |
| soapui/ | 902 | 127764 |
| staf/ | 382 | 124210 |
| storybook2/ | 220 | 40331 |
| supercsv/ | 135 | 8894 |

| | | |
|---|---|---|
| sweethome3d/ | 161 | 59663 |
| symmetricds/ | 271 | 31461 |
| taverna/ | 3532 | 488151 |
| tvbrowser/ | 961 | 198106 |
| ucdetector/ | 148 | 10453 |
| unicore/ | 5196 | 927124 |
| unitils/ | 386 | 57919 |
| veryquickwiki/ | 227 | 32662 |
| villonanny/ | 53 | 7281 |
| vnc-tight/ | 42 | 11554 |
| webadmin/ | 26 | 9401 |
| webcamstudio/ | 420 | 59181 |
| wiki2xhtml/ | 92 | 18841 |
| wintvcap-gui/ | 94 | 33390 |
| wsmostudio/ | 114 | 27796 |
| xbrlapi/ | 408 | 45933 |
| xmm/ | 421 | 144993 |
| xpstudio/ | 71 | 3907 |
| yale/ | 2610 | 381842 |
| zk1/ | 1266 | 177381 |
| zkforge/ | 669 | 147504 |

## B アンケート

(ア) ツールで表示されたメソッド名は、実際にそのクラスで使用されそうな名前だと思いましたか？
① とてもそう思う
② そう思う
③ どちらとも言えない
④ あまりそう思わない
⑤ 全くそう思わない

(イ) メソッド名リストには解答に用いた名前が表示されましたか？ 以下の４つから選択してください。
① 解答に用いた完全なメソッド名が表示された
② 解答に用いた動詞を使用したメソッド名が表示された
③ 動詞も解答に用いたものと異なるが、解答のヒントとなるメソッド名は表示された
④ 表示されなかった （④を選択した方は次の問に進んでください）

(ウ) 解答に用いた(ヒントとなった)メソッド名を記述してください
（ ）

(エ) 解答に用いた(ヒントとなった)上のメソッド名を、リストからすぐに見つけることはできましたか？
① すぐに見つけた
② 少し探して見つけた
③ 少し苦労して見つけた
④ かなり苦労して見つけた

図 10: 表示されたリストに関するアンケート

事前アンケート

名前　（　　）

1. データベース(以下 DB)について

(ア) DB を用いた Java プログラムを書いたことはありますか？

①．ある　　②　ない　　（「ない」と答えた方は (エ) に進んでください ）

(イ)「ある」と答えた方にお聞きします。最近ではいつぐらいに書きましたか？

（　　）年（　　）ヶ月 前

(ウ)「ある」と答えた方にお聞きします。DB を用いて作成した最も大きなプログラムは何行でしたか？

約（　　）行

(エ) DB についての以下の記述に当てはまるものを全て選択してください。(複数可)

①．別の言語で DB プログラムを記述したことがある

②．DB プログラミングに関する本を読んだことがある

③．SQL 文を記述したことがある

2. GUI について

(ア) GUI を用いた Java プログラムを書いたことはありますか？

①．ある　　②　ない　　（「ない」と答えた方は (エ) に進んでください ）

(イ)「ある」と答えた方にお聞きします。最近ではいつぐらいに書きましたか？

（　　）年（　　）ヶ月 前

(ウ)「ある」と答えた方にお聞きします。GUI を用いて作成した最も大きなプログラムは何行でしたか？

約（　　）行

(エ) GUI についての以下の記述に当てはまるものを全て選択してください。(複数可)

①．別の言語で GUI プログラムを記述したことがある

②．GUI プログラミングに関する本を読んだことがある

3. Web アプリについて

(ア) Java で Web アプリを開発したことはありますか?

①．ある　　②　ない　　（「ない」と答えた方は (エ) に進んでください ）

(イ)「ある」と答えた方にお聞きします。最近ではいつぐらいに書きましたか？

（　　）年（　　）ヶ月 前

(ウ)「ある」と答えた方にお聞きします。作成した最も大きな Web アプリプログラムは何行でしたか？

約（　　）行

(エ) Web アプリについての以下の記述に当てはまるものを全て選択してください。(複数可)

①．別の言語で Web アプリを記述したことがある

②．Web アプリ作成に関する本を読んだことがある

4. XML について

(ア) XML を用いた Java プログラムを作成したことはありますか？

①．ある　　②　ない　　（「ない」と答えた方は (エ) に進んでください ）

(イ)「ある」と答えた方にお聞きします。最近ではいつぐらいに書きましたか？

（　　）年（　　）ヶ月 前

(ウ)「ある」と答えた方にお聞きします。XML を用いて作成した最も大きなプログラムは何行でしたか？

約（　　）行

(エ) XML についての以下の記述に当てはまるものを全て選択してください。(複数可)

①．XML を記述したことがある　　②　別の言語で XML プログラムを記述したことがある

③　XML に関する本を読んだことがある

図 11: 事前アンケート

ツール使用後アンケート

1. ツールで表示されるメソッド名リストは見やすかったですか？
   (ア) とても見やすかった
   (イ) 見やすかった
   (ウ) どちらとも言えない
   (エ) やや見にくかった
   (オ) 非常に見にくかった

2. メソッド名リスト表示までの待ち時間はどう感じましたか？
   (ア) 十分に早かった
   (イ) 実用に支障がない程度には早かった
   (ウ) 実用可能ではあるが少し遅かった
   (エ) 実用に支障があるぐらい遅かった

3. あなたがプログラムを書いているときに、このツールを使用したいと思いますか？
   (ア) とてもそう思う
   (イ) ややそう思う
   (ウ) どちらとも言えない
   (エ) あまりそう思わない
   (オ) 全くそう思わない

4. ツールのよかったところはどこですか？

5. ツールの悪かったところはどこですか？

6. その他自由な意見、感想を書いてください。

図 12: ツールの使用感に関するアンケート