第 1 回 eclipse ゼミ

ゼミ担当者　：　芦辺　麻衣子，福田　正輝，鍵谷　武宏
指導院生　：　梶原　広輝，平岩　健一郎
開催日　：　2006 年 5 月 12 日

ゼミ内容: 本ゼミでは，プログラミング演習に向けて，Java の統合開発環境である eclipse について知っておくべき事柄を紹介する．

1　はじめに

近年，プログラミング言語として Java 言語が広まり，それに伴い，Java 用の統合開発環境（Integrated Development Environment：IDE）が作成されてきた．IDE の中でもオープンソースであるにも関わらず，製品版 IDE に引けを取らない充実した機能を有する点から，eclipse が注目されている．eclipse は，Java エディタやデバッガ，JSP エディタ，XML エディタ，バージョン管理（CVS）など，IDE に要求される機能を十分に備えている．また，Windows 版や Linux 版など多数の OS 向けバージョンが存在する．このような点から知的システムデザイン研究室では，eclipse を使って Java のコーディングを行う．本ゼミでは，eclipse の使用方法や便利なツール，プラグインのインストール方法について学習する．

2　インストール

はじめに，eclipse を使用する環境を整えるために，Java と eclipse のインストール方法について述べる．

2.1　Java のインストール

本資料では Windows 環境への Java(jdk) のインストールについて説明する．Java の環境には，J2EE，J2SE，J2ME があるが，本資料では最も一般的なデスクトップアプリケーション向けである J2SE を対象とする．また，マシン環境としては WindowsXP を対象としている．

2.1.1　インストール実行ファイルの取得

まず，下記のサイトより Java 開発環境のインストールプログラムをダウンロードする．

URL

http://java.sun.com/j2se/1.5.0/download.jsp

ダウンロードするファイルは上記 Web サイトの「Download JDK 5.0 Update 6」である．「Download JDK 5.0 Update 6」をクリックする．(Fig. 1 参照)

項目をクリックすると Fig. 2 のような画面へと移る．まず，Fig. 2 において丸で囲ってある場所の「Accept」を選択し，Fig. 3 の画面へ．

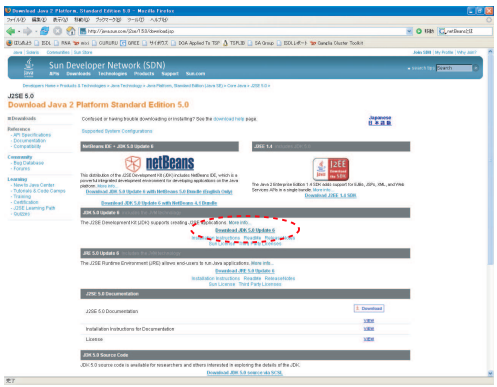

Fig. 1　J2SE 5.0 のダウンロード（出典：自作）

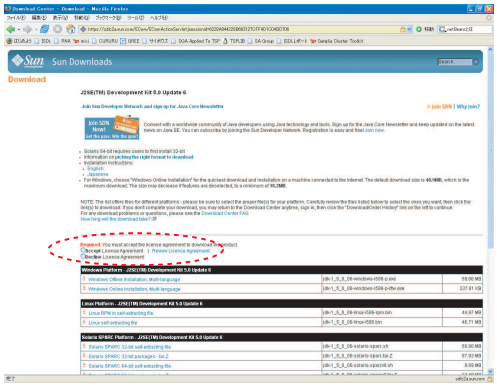

Fig. 2　使用条件（出典：自作）

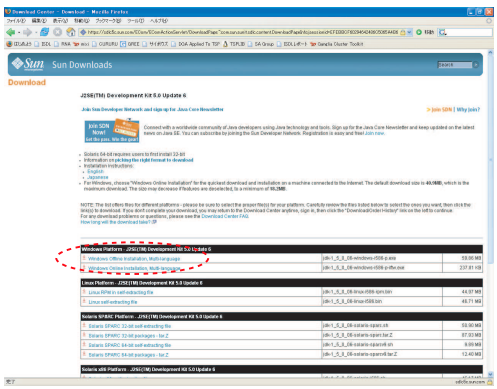

Fig. 3　OS 選択画面（出典：自作）

ここではインストール対象マシンの OS を選択する必要がある．本マニュアルでは Windows 環境を想定している．そのため，一番上にある「Windows Platform」の項目にある「Windows Offline Installation, Multi-language」をクリックする．するとファイルのダウンロード画面が表示されるので，デスクトップを指定してダウンロードする．(Fig. 4 参照) すぐ下の「Windows Online Installation, Multi-language」との違いはオフラインでのインストールが可能なことである．

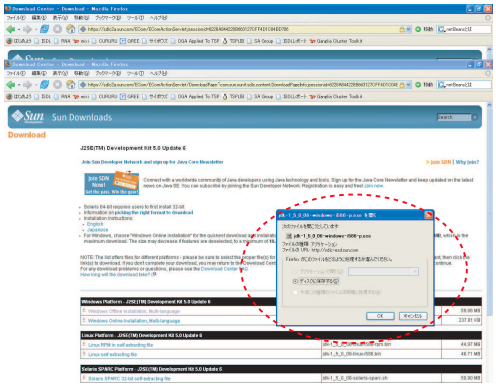

Fig. 4　ダウンロード（出典：自作）

2.1.2　Java のインストール

本項では 2.1.1 項で取得したプログラムより，実際にインストールするための方法について述べる．

まず，デスクトップ上にダウンロードしたファイル (Fig. 5 参照) をダブルクリックするとインストールウィザードが起動する．

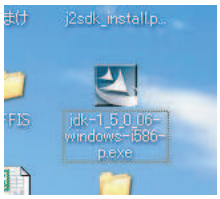

Fig. 5　ダウンロードしたファイル（出典：自作）

その後，Fig. 6 のライセンス契約画面に移るので「同意します」にチェックをつけ,「次へ」をクリックする．

次にカスタムセットアップの画面が出るので「次へ」をクリックする．(Fig. 7 参照)

すると JDK のインストールが始まる．(Fig. 8 参照) ちなみに JDK とは Java プログラミング言語を使ってアプレットやアプリケーションを記述する時に使用するソフトウェア開発キットである．

JDK のインストールが終わると，JRE のインストールに移る．(Fig. 9 参照)JRE とは Java プログラミング言語を使って記述されたアプレットやアプリケーションを実行するのに必要な環境である．ここも「次へ」をクリックする．

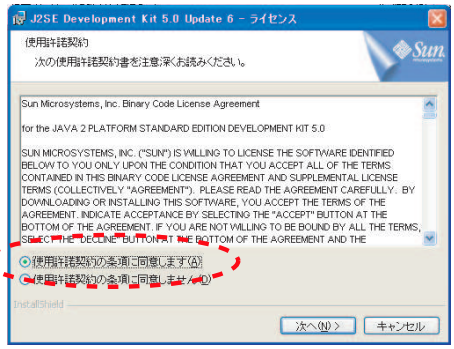

Fig. 6　ライセンス契約画面（出典：自作）

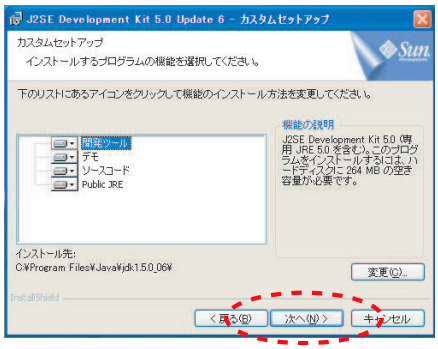

Fig. 7　カスタムセットアップ（出典：自作）

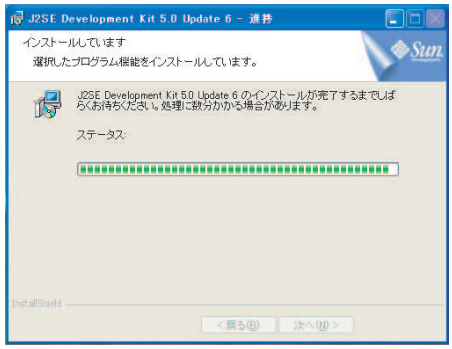

Fig. 8　JDK のインストール（出典：自作）

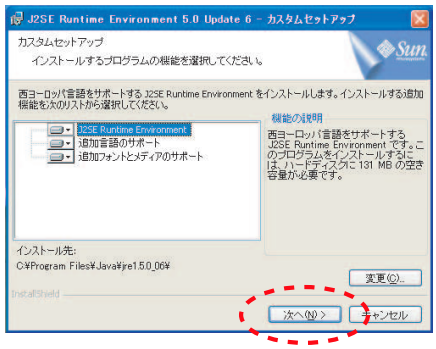

Fig. 9　JRE のインストール（出典：自作）

次にブラウザの登録に移り，これも「次へ」のボタンをクリックする．(Fig. 10 参照)

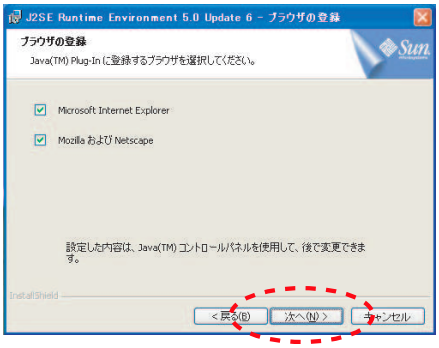

Fig. 10　ブラウザの登録画面（出典：自作）

その後，Fig. 11 に示すインストール完了メッセージが出たらインストールは完了である．

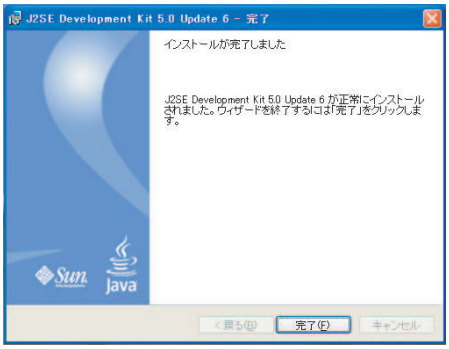

Fig. 11　インストールの完了（出典：自作）

2.1.3　環境設定

2.1.1 項，2.1.2 項で Java のインストールは完了したが，Java を動作させるために，環境変数の設定をする必要がある．

まず,「マイコンピュータ→右クリック→プロパティ」を選択し，システムのプロパティの画面を表示させる．(あるいは「コントロールパネル→システム」でも同様の画面に移動する．)

システムのプロパティの中から「詳細設定」のタブを選択し，環境変数をクリックする．(Fig. 12 参照)

Fig. 13 に示すような環境変数の設定が表示されるので，そのうちユーザー環境変数の新規をクリックする．すると，Fig. 14 の「変数の編集」画面となる．

ここでは，以下の値を入力する．

その結果 Fig. 15 のようになる．

PATH の設定

変数名：PATH

変数値：C:¥Program Files¥Java¥jdk1.5.0_06¥bin

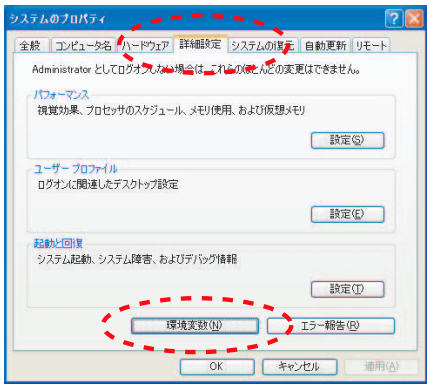

Fig. 12　システムのプロパティ（出典：自作）

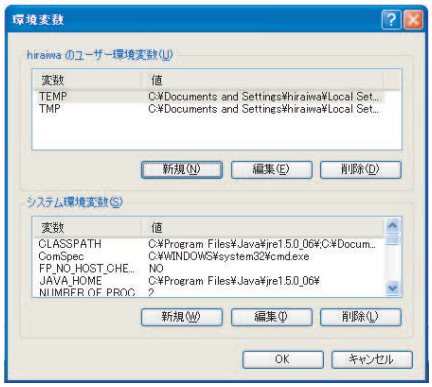

Fig. 13　ユーザー環境変数（出典：自作）

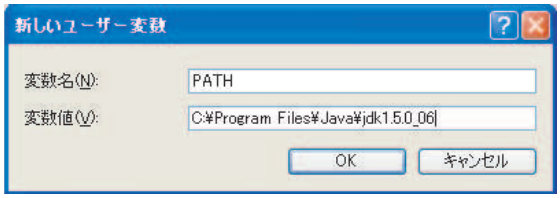

Fig. 14　変数の編集（出典：自作）

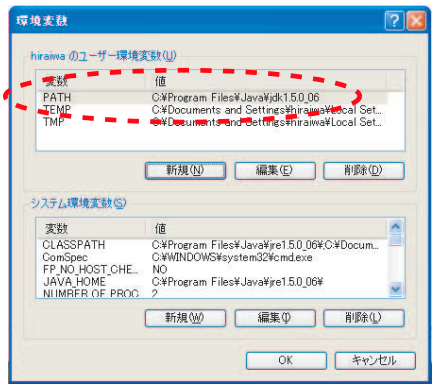

Fig. 15　環境変数確認（出典：自作）

ただし，すでに環境変数に PATH という変数が存在する場合は，PATH を選択し，編集ボタンを押すことによって編集することができるので，今ある値の最後に

PATH の追加

;C:¥Program Files¥Java¥jdk1.5.0_06¥bin

を追加すればよい．追加する時は，最初に「;(セミコロン)」を付けてから追加することに気をつける．追加後は以下のようになる．

PATH の追加後

C:¥~ 以前からある値 ~;C:¥Program Files¥Java¥jdk1.5.0_06¥bin

正しく追加されていることを確認したら「OK」ボタンをクリックし終了する．

2.1.4 インストールの確認

最後に，インストールの確認として，コマンドプロンプトより以下のコマンドを入力してみる．その結果 Fig. 16 のようにインストールした Java のバージョンが表示されたら，インストールは成功である．

インストールの確認

java -version

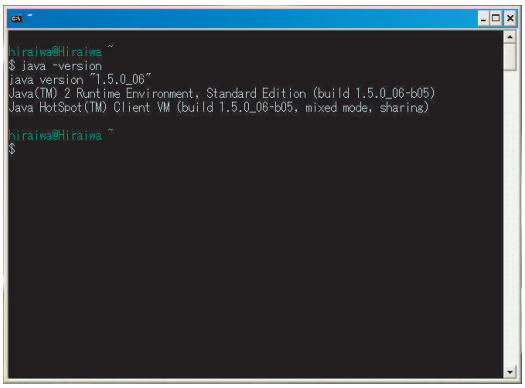

Fig. 16 実行チェック（出典：自作）

2.2 eclipse のインストール方法

eclipse とはオープンソースの統合ソフトウェア開発環境 (Integrated Development Environment:IDE) の一つである．eclipse を利用して Java プログラミングのコーディングを行っていく．

本マニュアルでは eclipse の本家である eclipse.org のサイトからダウンロードする方法について説明する．また，eclipse.org が提供している Language Pack を用いて日本語化も行う．

しかし，eclipse.org からダウンロードする際に，非常に時間を要したり，ダウンロードに失敗したりと容易でない．よって，他の以下の方法を用いてもよい．

- Java の書籍を購入し，付録の eclipse をインストールする．
- Vector(http://www.vector.co.jp/soft/dl/winnt/prog/se384127.html) のサイトから All-in eclipse をダウンロードする．

All-in eclipse とは eclipse バージョン 3.1.1 に Language Pack を適用し，インストーラ形式にしたものである．この方法は非常に簡単である．

2.2.1 eclipse の取得

まず，下記の eclipse.org のページから「3.1.1」をクリックし，バージョン 3.1.1 のダウンロードページへと進む．(Fig. 17 参照)

URL

http://download.eclipse.org/eclipse/downloads/index.php

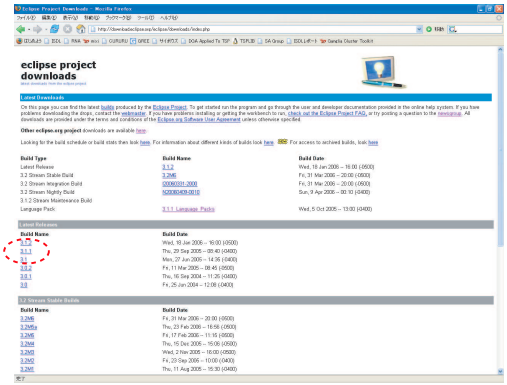

Fig. 17 バージョン選択画面（出典：自作）

次に OS 選択画面に移る．本マニュアルでは Windows 環境を想定している．よって Windows(Supported Versions) の右の「eclipse-SDK-3.1.1-win32.zip」をクリックし次に進む．(Fig. 18 参照)

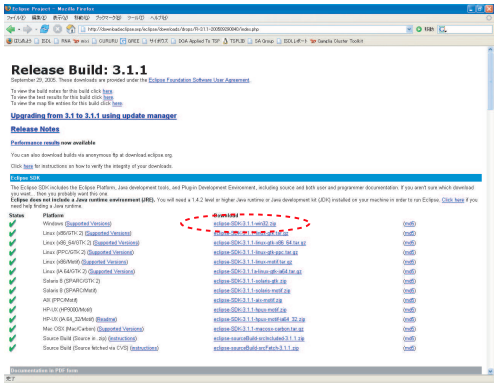

Fig. 18 OS 選択画面（出典：自作）

すると，Fig. 19 の画面に移るが，自動的に Fig. 20 の画面に移動する．もし移動しない場合は Fig. 19 の

「here」をクリックする．

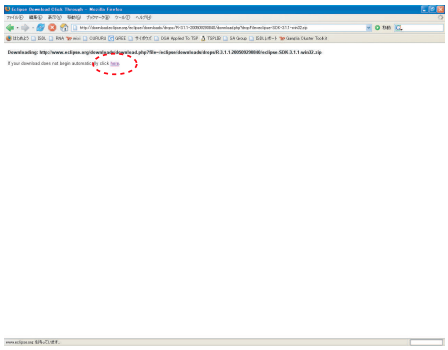

Fig. 19　eclipse のダウンロード（出典：自作）

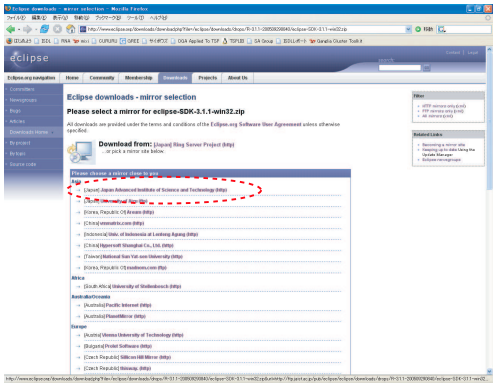

Fig. 20　ダウンロード（出典：自作）

次に Fig. 20 の画面でどのサイトからダウンロードするかを決定する．好みのサイトからダウンロードして構わないが，本マニュアルでは「Japan Advanced Institute of Science and Technology(http)」からダウンロードすることにする．すると「eclipse-SDK-3.1.1-win32.zip」のダウンロードが始まるのでデスクトップに保存する．(Fig. 21 参照)

何度ダウンロードしても失敗する場合は，書籍や Vector(URL:http://www.vector.co.jp/soft/dl/winnt/prog/se384127.html) からインストールする方法に変更する．

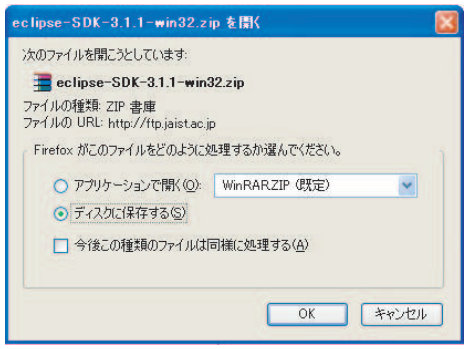

Fig. 21　ディスクへの保存（出典：自作）

2.2.2　eclipse の日本語化用ファイルの取得

日本語化を行うため以下のサイトの「3.1.1 Language Packs」をクリックする．(Fig. 22 参照)

URL

http://download.eclipse.org/eclipse/downloads/index.php

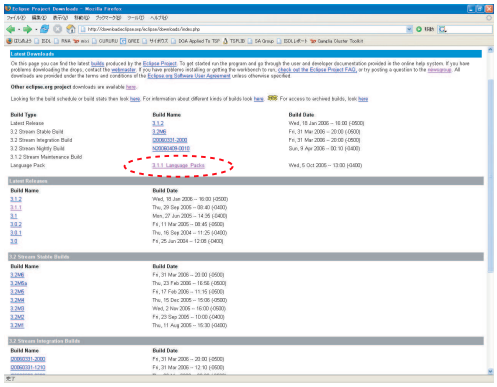

Fig. 22　Language Packs（出典：自作）

すると Fig. 23 のページに移動するので「NLpack1-eclipse-SDK-3.1.1a-win32.zip」をクリックする．

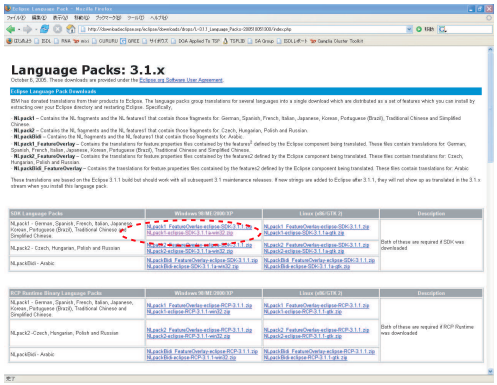

Fig. 23　Language Packs の選択（出典：自作）

次に Fig. 24 の画面に移るが，自動的にすぐに Fig. 25 の画面に切り替わる．もし切り替わらない場合は Fig. 24 の「here」をクリックする．

次に Fig. 25 の画面でどのサイトからダウンロードするのかを決定する．好みの場所からダウンロードして構わないが，本マニュアルでは「University of Aizu」をクリックし，ここからダウンロードすることにする．すると「NLpack1-eclipse-SDK-3.1.1a-win32.zip」のダウンロードが始まるのでデスクトップに保存する．(Fig. 26 参照)

2.2.3　eclipse の実行

本項では 2.1.1，2.1.2 項で取得したファイルを用いて，eclipse を日本語化しインストールする．

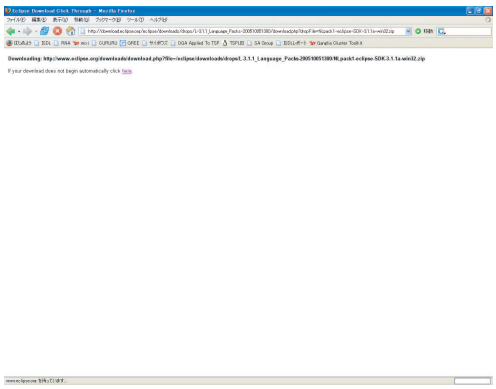

Fig. 24　Language Pack のダウンロード（出典：自作）

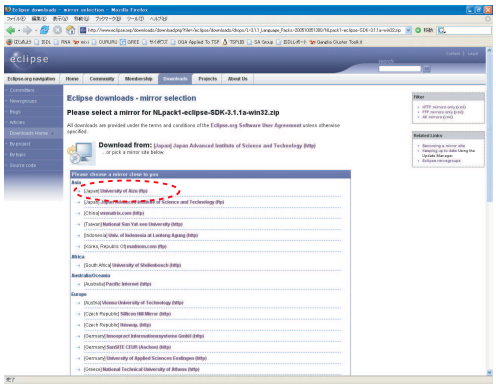

Fig. 25　ダウンロード（出典：自作）

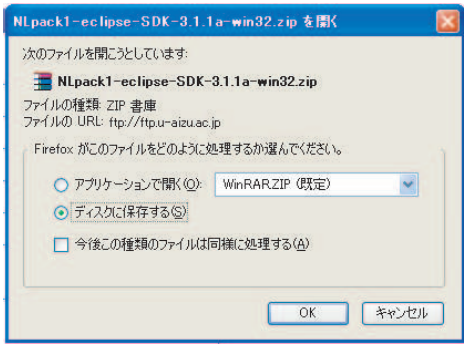

Fig. 26　ファイルの保存（出典：自作）

まず，デスクトップ上にダウンロードしたファイル「eclipse-SDK-3.1.1-win32.zip」と「NLpack1-eclipse-SDK-3.1.1a-win32.zip」を解凍する．(Fig. 27 参照)

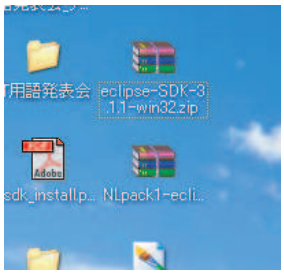

Fig. 27　ダウンロードしたファイル（出典：自作）

解凍によりできた「eclipse-SDK-3.1.1-win32」という名前のフォルダの中に「eclipse」というフォルダがあるので，このフォルダを自分の使いやすい場所に移動させる．Fig. 28 参照はマイ ドキュメントに移動させた例である．

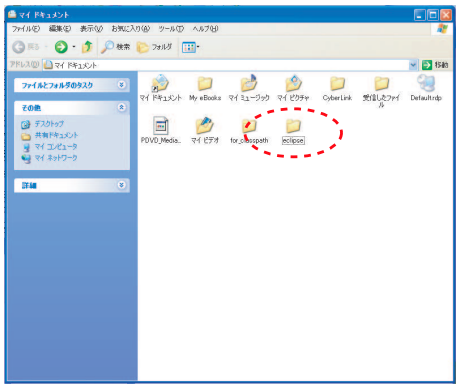

Fig. 28　ダウンロードしたファイル（出典：自作）

「eclipse」フォルダの中身は Fig. 29 のようになっている．

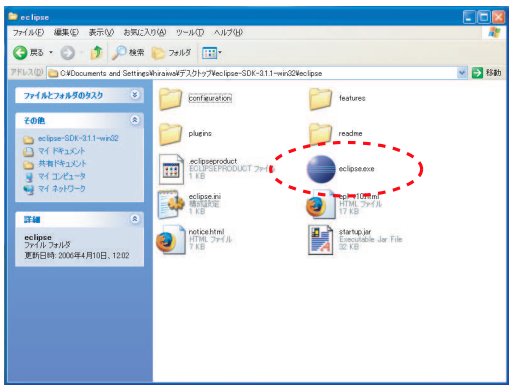

Fig. 29　eclipse-SDK-3.1.1-win32 の中身（出典：自作）

また，解凍したもう一つの「NLpack1-eclipse-SDK-3.1.1a-win32」というフォルダにも「eclipse」というフォ

ルダがあり，その中身は Fig. 30 のようになっている．

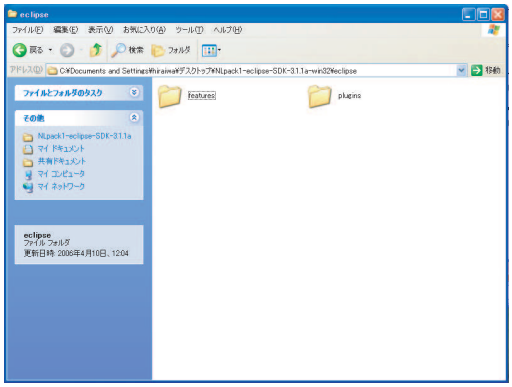

Fig. 30　NLpack1-eclipse-SDK-3.1.1a-win32.zip の中身（出典：自作）

この Fig. 30 の「plugins」と「features」のフォルダを Fig. 29 のフォルダ中にコピー (上書き) する．以上で eclipse の日本語化を含めたインストールは完了である．

eclipse の実行は Fig. 29 の「eclipse.exe」をダブルクリックすればよい．初めに Fig. 31 のようにワークスペースをどこに作成するか聞かれるので，マイ ドキュメントなど好きなフォルダを選択し「OK」ボタンを押せばよい．このワークスペースの中にソースコードのファイルが生成されることになる．

以上で eclipse のインストールは完了である．

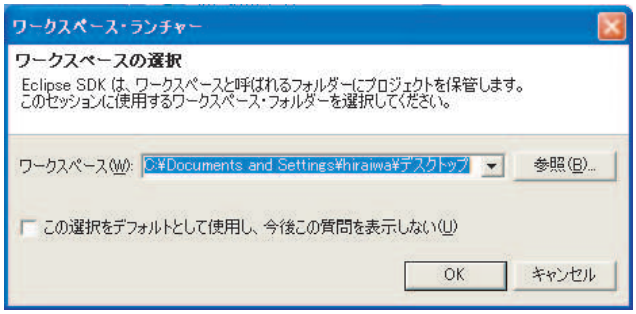

Fig. 31　ワークスペースの選択（出典：自作）

3　eclipse の基本的な使い方

3.1　プロジェクトの作成

eclipse を起動すると左側にパッケージエクスプローラが表示される．これはクラスやクラスの集合体であるパッケージの情報を表示し，管理するビューで，ここに自分の作成したプロジェクト名や，クラス名が表示される．プロジェクトを作るためには，まずメニューバーのファイルをクリックし Fig. 32 のように選択する．すると Fig. 33 の画面が表示されるので，Java プロジェクトを選択して，次へをクリックする．すると Fig. 34 の画面になるので，「HelloWorld」と入力し終了をクリック．これで「HelloWorld」というプロジェクトが作成され，Fig. 35 のようにパッケージエクスプローラに「HelloWorld」が追加される．

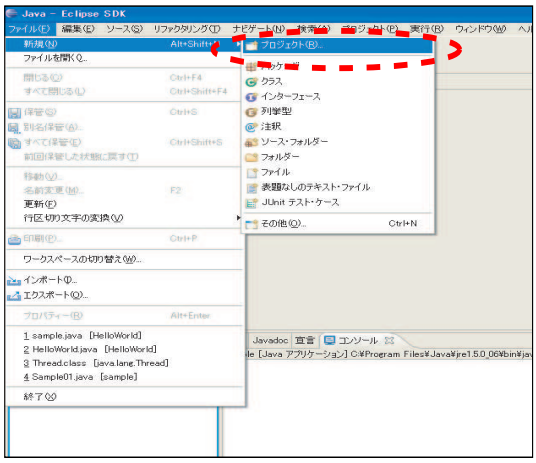

Fig. 32　新規プロジェクトの作成（出典：自作）

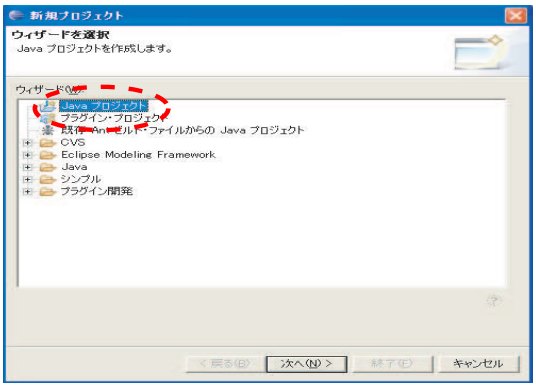

Fig. 33　Java プロジェクトの選択（出典：自作）

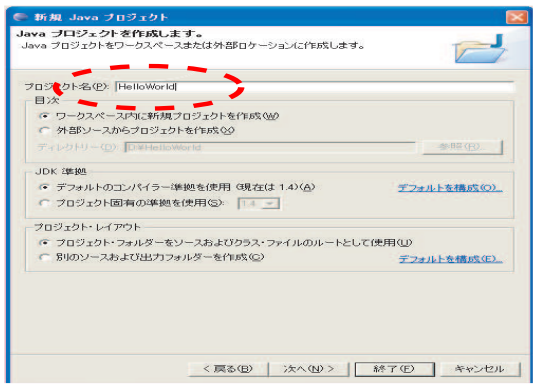

Fig. 34　プロジェクト名の入力（出典：自作）

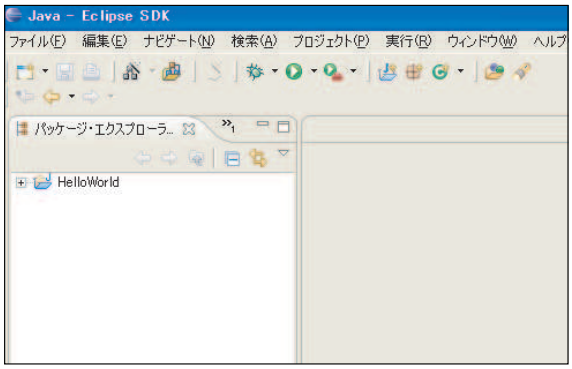

Fig. 35　HelloWorld（出典：自作）

3.2　クラスの作成

作成したプロジェクトである HelloWorld を選択し，右クリックし Fig. 36 のようにクラスを選択する．すると Fig. 37 の画面が表示されるので，「名前」のテキストフィールドに「Sample」と入力し，Fig. 37 のように選択し，「終了」をクリックする．これで Sample というクラスが作成される．

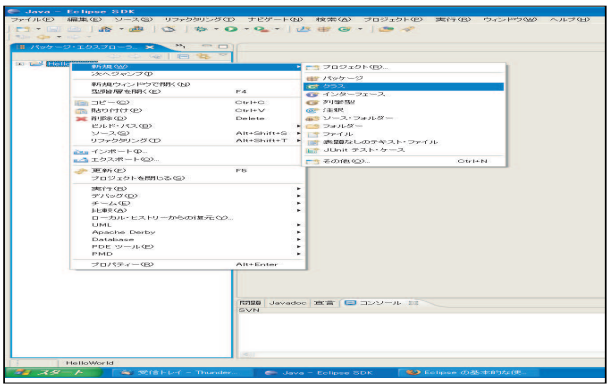

Fig. 36　新規クラスの作成（出典：自作）

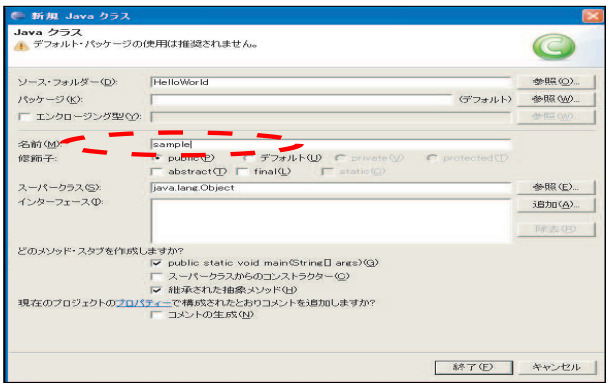

Fig. 37　クラス名の入力（出典：自作）

3.3　プログラムの実行

3.3.1　コマンドライン引数が無い場合

例として作成した HelloWorld というクラスで，Fig. 38 のように「System.out.println(”Hello”);」と入力する．この入力の際に「Ctrl + Space」でコードを補間することができる．例えば S で「Ctrl + Space」を実行すると，S の付くプログラム候補が表示される．そして入力の後，メニューバーから実行を選択し，Fig. 39 の画面のように選択する．すると，Fig. 40 の画面になるので，Java アプリケーションを選択して，新規をクリックする．今回は Java アプリケーションだが，Java アプレットを実行したい場合は，同じように Java アプレットを選択し，新規をクリックすることで実行できる．次に Fig. 41 で実行したいプロジェクト名とクラス名を入力し実行する．すると，プログラムが正確に入力されていれば，Fig. 42 のように，実行結果がコンソールに表示される．しかしプログラムを正確に入力できなかった場合は，コンパイルエラーが起こる．例えば，先ほどの入力で「;」を最後に付け忘れたとする．すると Fig. 43 のように，誤っているコードに印が付き，その印をマウスでクリックすると，Fig. 44 のようにエラーの詳細が表示される．

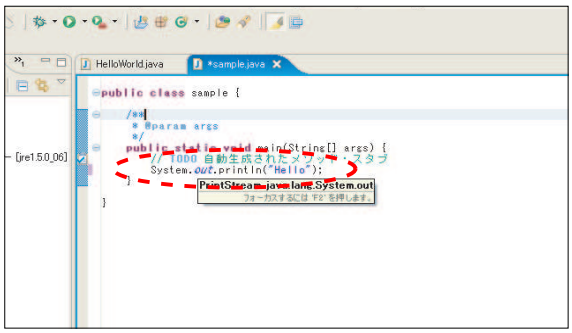

Fig. 38　プログラムの入力（出典：自作）

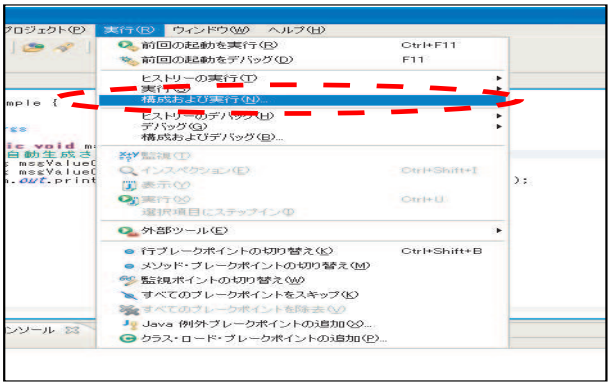

Fig. 39　構成および実行の選択（出典：自作）

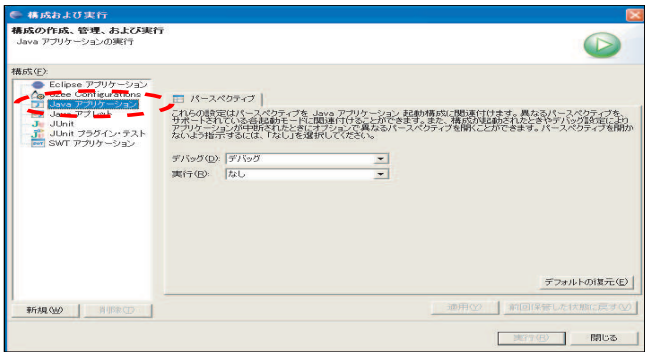

Fig. 40　プログラムの実行（出典：自作）

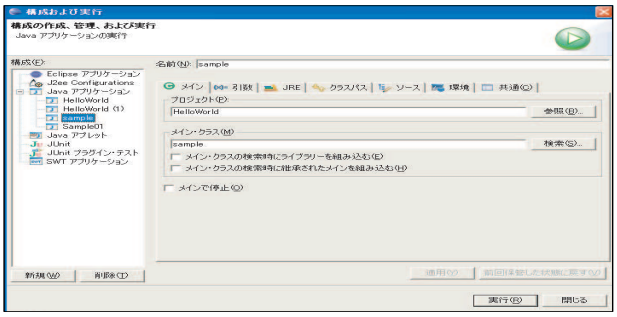

Fig. 41　プロジェクトとクラスの選択（出典：自作）

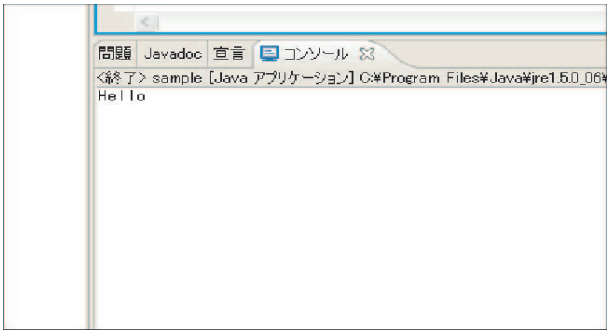

Fig. 42　実行画面（出典：自作）

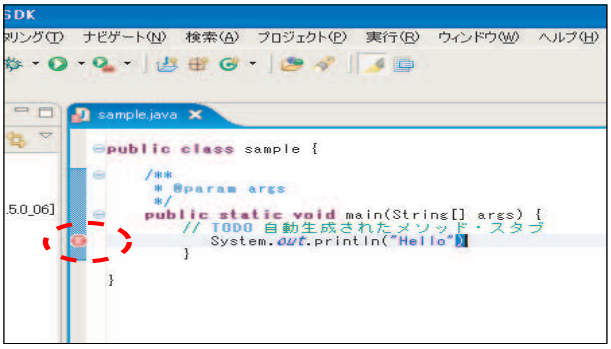

Fig. 43　エラーの部分の表示（出典：自作）

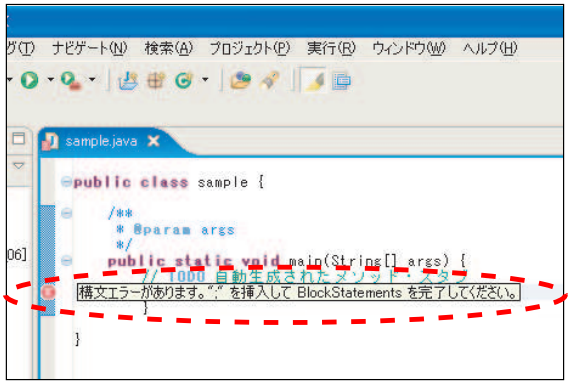

Fig. 44　エラー詳細（出典：自作）

3.3.2　コマンドライン引数がある場合

次に引数がある場合のプログラムの実行を考える．まず先ほど作成したプログラムを Fig. 45 のように変更する．そして，先ほどの実行と同じように，メニューバーから実行を選択し，Fig. 39 の画面のように選択する．すると画面が Fig. 41 のように切り替わるので，ここで引数を選択し，Fig. 46 のように引数の入力を行う．ここでは自分の名前と「！」を入力し，名前と「！」を半角スペースで空けておく．そして実行をクリックすれば，コンソールに実行結果である名前が表示される．この例では「私の名前は大阪です！」と表示される．

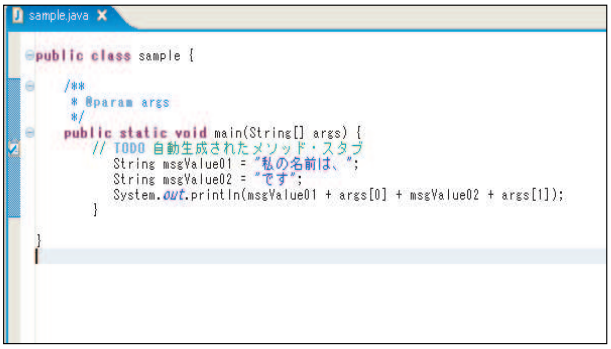

Fig. 45　引数のあるプログラムの入力（出典：自作）

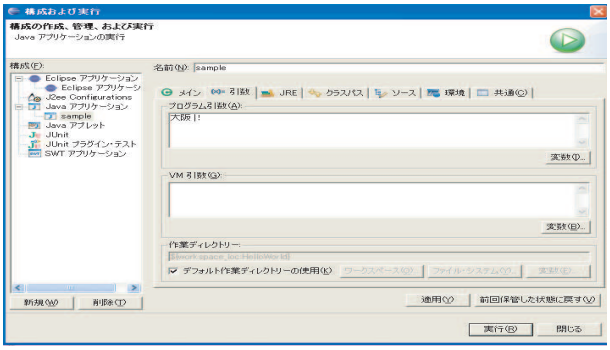

Fig. 46　引数の入力（出典：自作）

3.3.3　アイコンを使った実行方法

今回は実行方法として，メニューバーから実行を選択した．しかし他の実行方法として，Fig. 47 のように，実行のアイコンが用意されているので，main 文のあるファイルをパッケージエクスプローラで選択している場合は，Fig. 47 に示すアイコンをクリックし，Fig. 48 のように選択することで実行することができる．また，引数のあるプログラムや，Java アプレットを実行したい場合は，Fig. 47 に示すアイコンをクリックし，今度は構成および実行を選択すれば Fig. 41 の画面に切り替わるので，後は同じように設定すれば，実行できる．

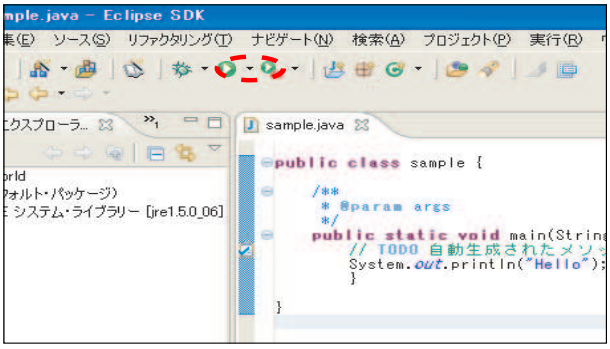

Fig. 47　実行のアイコン（出典：自作）

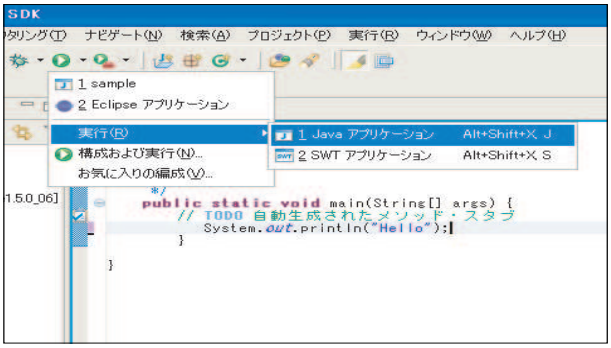

Fig. 48　違う実行方法（出典：自作）

3.4　既存プロジェクトのインポート

eclipse のプロジェクトを移動する際，まず eclipse のワークスペースの場所から，移動したいプロジェクトをコピーし，適当な場所に保存しておく．次に Fig. 49 のように，eclipse のパッケージエクスプローラー上で右クリックし，インポートを選択．その後 Fig. 50 のように表示されるので，既存プロジェクトをワークスペースへを選択．すると Fig. 51 のように表示されるので，ルートディレクトリの参照から，コピーしてきたプロジェクトを選択し終了をクリックすれば，既存プロジェクトのインポートが完了する．なお同様の操作はファイルメニューのインポートからも可能である．

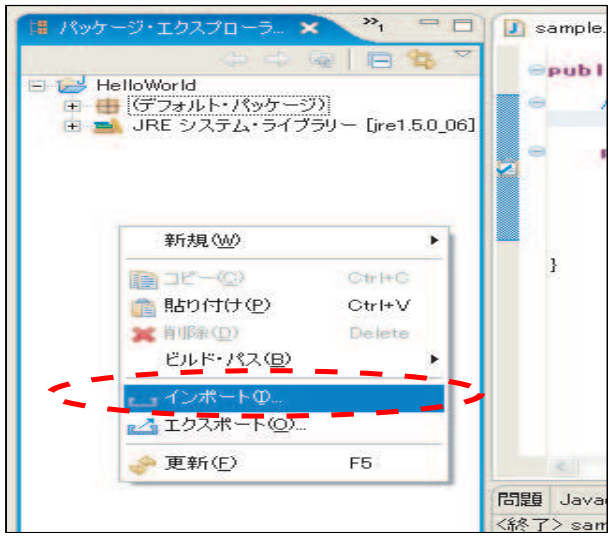

Fig. 49　インポートの選択（出典：自作）

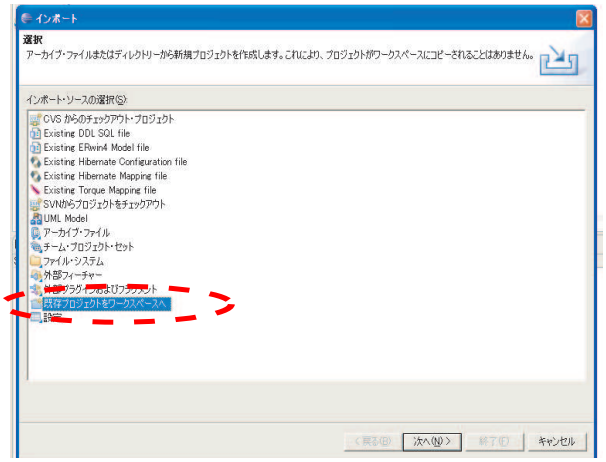

Fig. 50　既存プロジェクトのワークスペースへのインポート（出典：自作）

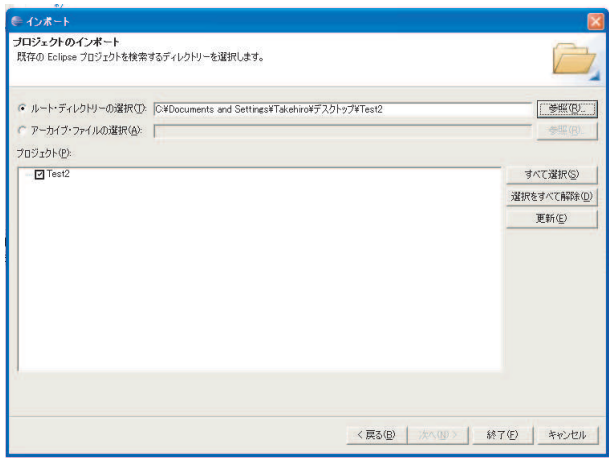

Fig. 51　インポートするプロジェクトの選択（出典：自作）

3.5　デバッグ

デバッグとはプログラムの誤りを探し，取り除くことであり，eclipse には，その発見や修正を支援するデバッガが備わっている．その使用法を次の 1 から 6 に示す．

1. ブレークポイントの設定
   ブレークポイントとは，実行中のプログラムを一時停止させる場所を指定するための「目印」のことで，Fig. 97 のように停止させたいプログラムが表示されたウィンドウの左側をダブルクリックすると水色の丸いマークで表示される．デバッグでは，ブレークポイントがついたコードを実行する直前で停止する．

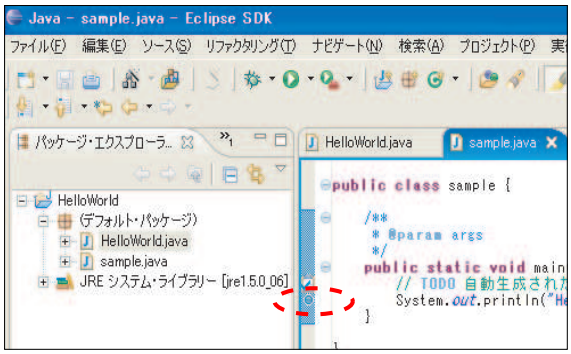

Fig. 52　ブレークポイントの選択（出典：自作）

2. デバッグの実行
   Fig. 53 のようにツールバーから実行を選択し，デバッグ，Java アプリケーションを選択．するとパースペクティブを開きますかと聞かれるので，「はい」を選択する．すると画面が Fig. 54 に切り替わる．Fig. 54 の右上には，プログラムを停止した時点での変数の値が表示される．表示されていない場合は，Fig. 55 のようにウィンドウ，ビューの表示，変数で，表示させることができる．

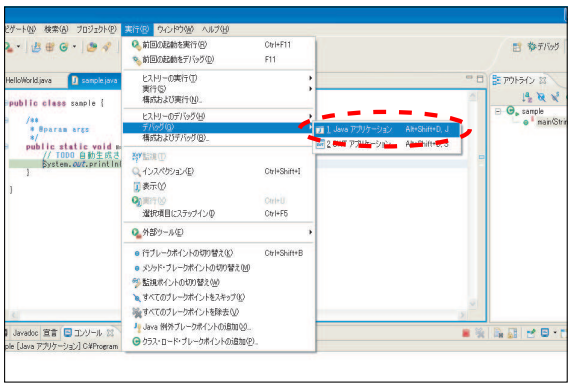

Fig. 53　デバッグの実行（出典：自作）

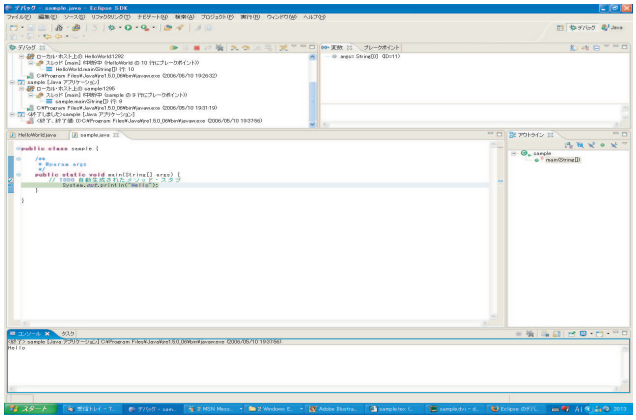

Fig. 54　デバッグの実行画面（出典：自作）

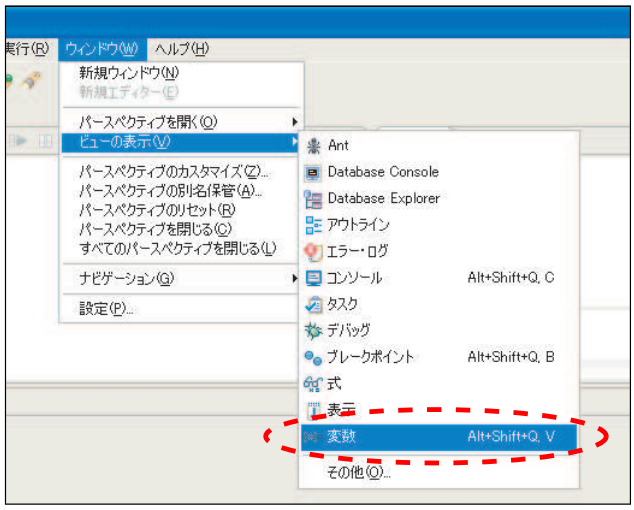

Fig. 55　変数のビューの表示のさせ方（出典：自作）

3. プログラムの再開
   停止したプログラムを実行させるには，Fig. 56 の示すアイコンをクリックすることで実行することができる．

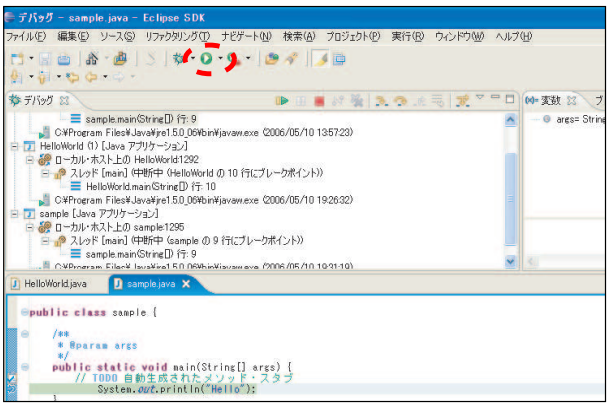

Fig. 56　プログラムの再開（出典：自作）

4. ステップオーバー
   ステップオーバーとは，プログラムが停止した状態からコードを 1 つ実行する操作のことで，そのメソッドのコードを全て実行した直後で停止する．また Fig. 57 の示すアイコンをクリックすることで実行できる．

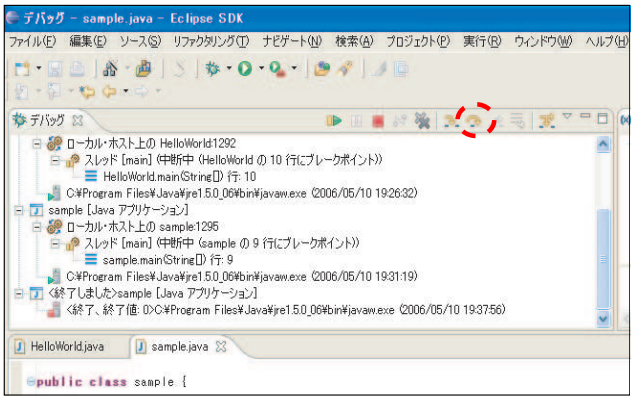

Fig. 57　ステップオーバー（出典：自作）

5. ステップイン
   ステップインはステップオーバーと同じように，プログラムが停止した状態からコードを 1 つ実行する操作のことで，そのメソッドの先頭のコードを実行する直前で停止する．また Fig. 58 の示すアイコンを選択することで行うことができる．

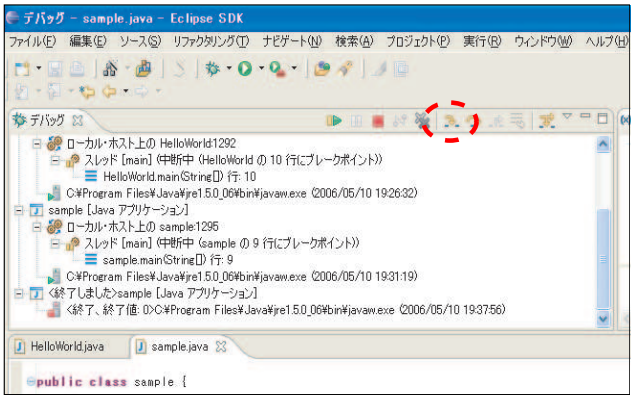

Fig. 58　ステップイン（出典：自作）

6. 画面の切り替え
   デバッグの画面から，元の画面に戻すには，Fig. 59 の示すアイコンを選択することで行うことができる．また同様に Fig. 60 のようにメニューバーのウィンドウからパースペクティブを開くを選択し，Java を選択すれば同じように画面を切り替えることができる．

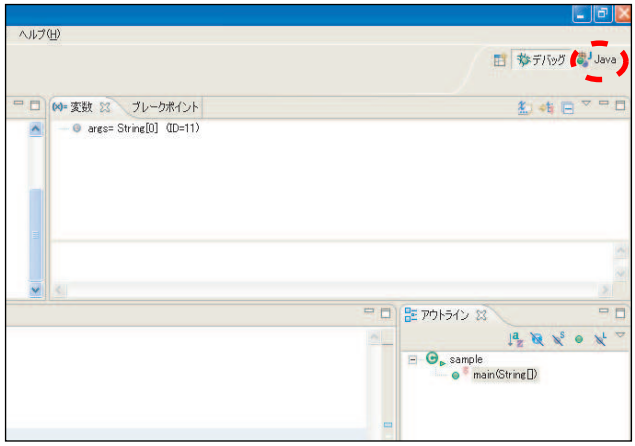

Fig. 59　画面の切り替え 1 （出典：自作）

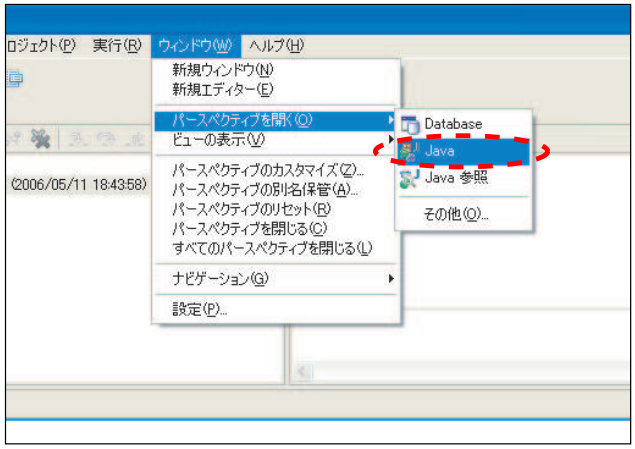

Fig. 60　画面の切り替え 2 （出典：自作）

3.6　Javadoc

3.6.1　Javadoc とは

Javadoc とは，Java のコードの中にコメントを埋め込むことによって，クラス，変数，関数の説明を HTML 形式で見られるようにしたものである．Java クラスの仕様書の標準の書式である．Table 1 に，主な Javadoc のタグを示す．

Table 1　Javadoc の主なタグ

タグ	概要
@author	作成者
@exception	例外
@throws	例外
@param	引数
@return	戻り値
@see	関連項目
@since	導入されたバージョン
@version	バージョン情報

3.6.2　リファレンスの生成

はじめに，コードの中に Javadoc コメントを追加していく必要がある．追加は，新しいクラスを作成する際に，Fig. 61 の画面で「コメントの生成」にチェックを付ければ良い．もし，自分で生成したい場合は，/**～*/で囲み@を打つと Fig. 62 のように Javadoc タグの補完が出てくるので，容易にタグを入力することができる．あとは，出てきたタグに従って，説明を加えることにより，コードの中に Javadoc タグを生成することができる．今回は，Fig. 63 のようなコード Test.java を作成した．

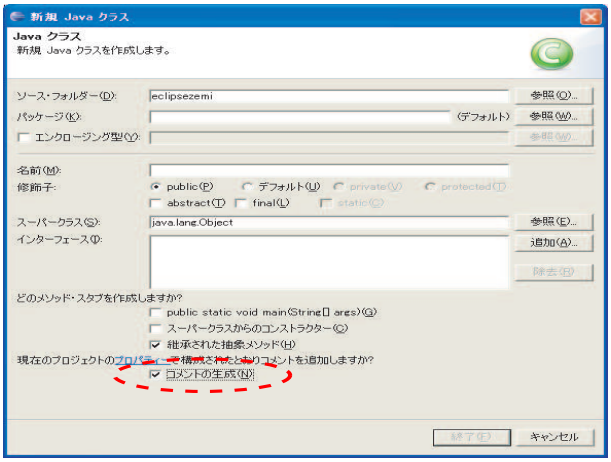

Fig. 61　コメントの生成（出典：自作）

Fig. 62　Javadoc タグの生成（出典：自作）

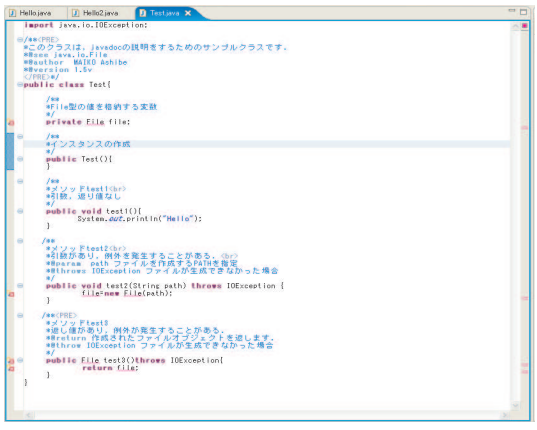

Fig. 63　Test.java（出典：自作）

実際に，リファレンスを作成する手順を以下に示す．メニューバーの「プロジェクト」にある「Javadoc の生成」を選択する．(Fig. 64 参照）すると，Fig. 65 のような Javadoc 生成ウィザードが起動する．

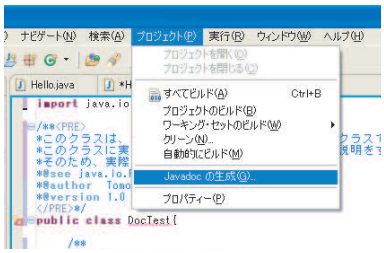

Fig. 64　Javadoc の生成選択（出典：自作）

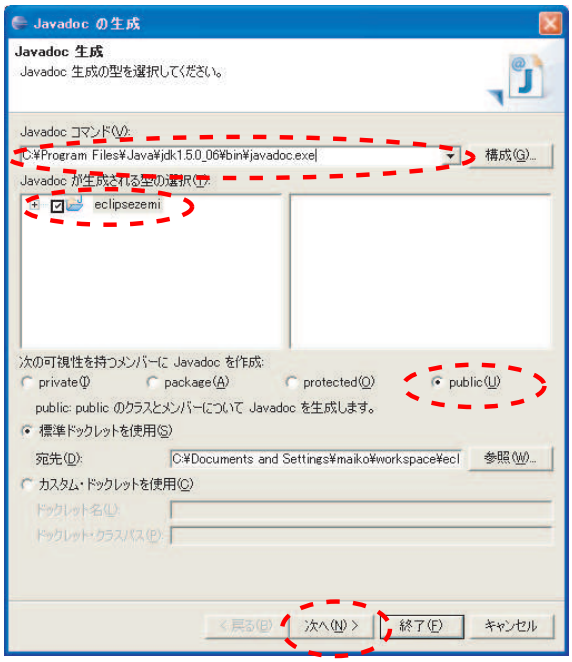

Fig. 65　Javadoc 生成ウィザード（1）（出典：自作）

この画面では，Javadoc コマンドにインストールした Java のフォルダの中にある javadoc.exe のパスを入力する．ここでは，「`C:\Program Files\Java\jdk1.5.0_06\bin \javadoc.exe`」とした．Javadoc メンバの範囲を選択し，「次へ」を押す．今回は，「public」を選択した．すると，Fig. 66 のような画面になるので，ここでは「次へ」を押す．また，次の Fig. 67 では「終了」を選択する．これで，リファレンスが生成された．作成した，HTML は Fig. 65 のドックレットの宛先として指定した箇所に出来ている．eclipse からは，Fig. 68 のように，doc の中に出来ている index.html を右クリックし，「アプリケーションから開く」にある「Web ブラウザー」を選択する．これにより，リファレンスを閲覧することができる．Fig. 69 が作成した HTML である．

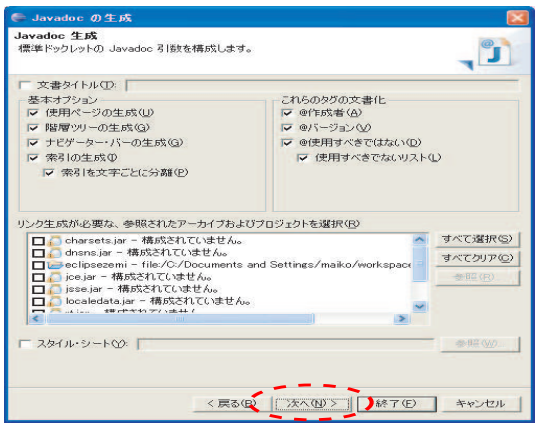

Fig. 66　Javadoc 生成ウィザード（2）（出典：自作）

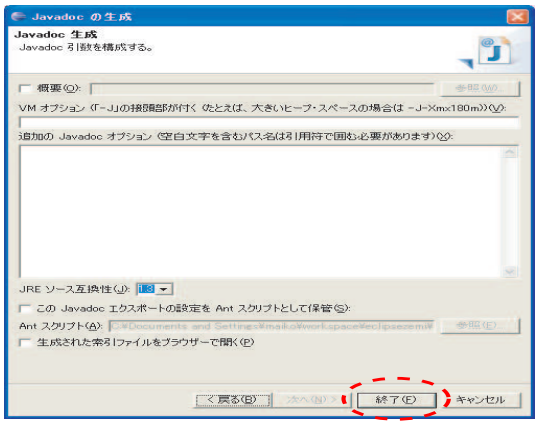

Fig. 67　Javadoc 生成ウィザード（3）（出典：自作）

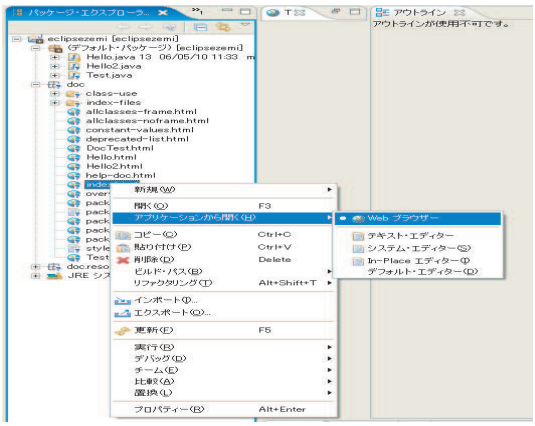

Fig. 68　Web ブラウザー選択（出典：自作）

Fig. 69　Test.java の Javadoc リファレンス（出典:自作）

4　プラグイン

4.1　subclipse

subclipse は，バージョン管理ツール Subversion の eclipse 用フロントエンドプラグインである．Subversion の開発元である tigris.org で開発されている．使用感は，標準で提供されている CVS フロントエンドと変わらない．

4.1.1　subclipse のインストール

はじめに subclipse をインストールする際に，必要となってくる eclipse のプロキシサーバの設定を行う．eclipse を起動し，メニューバーのウィンドウにある設定を選択する．(Fig. 70 参照）すると，Fig. 71 のようなウィンドウが出てくるので,「プロキシー設定の HTTP プロキシー接続を使用可能にする」にチェックを入れ，プロキシの設定を行う．HTTP プロキシ・ホスト・アドレスは,「proxyt.doshisha.ac.jp」で，HTTP プロキシ・ホスト・ポートは,「8080」で設定する．(Fig. 71 参照）これで，eclipse におけるプロキシ設定が完了した．ここから，実際に subclipse のインストールを行う．メニューバーのヘルプにあるソフトウエア更新から検索とインストールを選択する．(Fig. 72 参照）すると，Fig. 73 のようなウィンドウが出てくる．ここでは,「インストールする新規フィーチャーを検索」にラジオボタンを選択し,「次へ」をクリックする．(Fig. 73 参照）

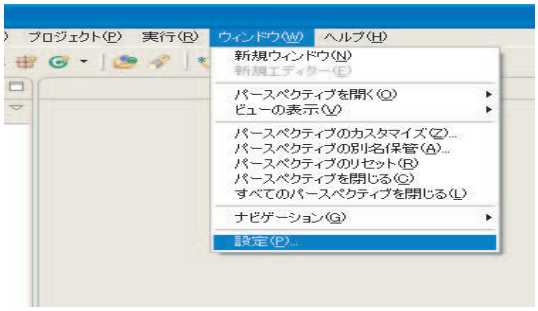

Fig. 70　プロキシサーバの設定（出典：自作）

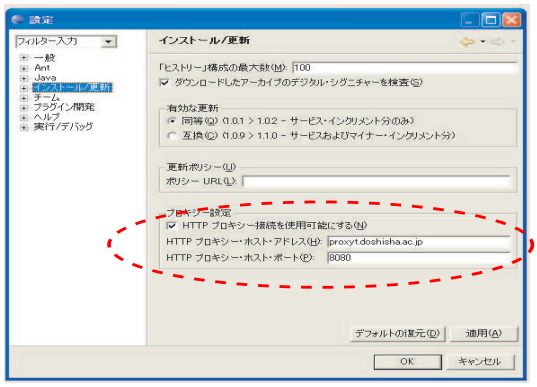

Fig. 71　プロキシサーバの設定（出典：自作）

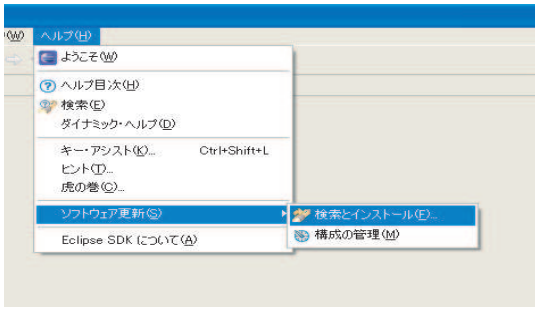

Fig. 72　検索とインストールの選択（出典：自作）

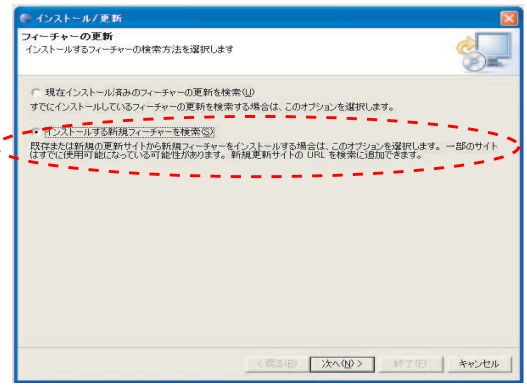

Fig. 73　新しいプラグイン定義の追加（出典：自作）

次に，Fig. 74 の画面に移るので，「新規リモート・サイト」ボタンをクリックし，subclipse を定義する．この時，名前は「subclipse」で，URL は「http://subclipse.tigris.org/update」とする．名前と URL を入力して「OK」ボタンを押す．(Fig. 74 参照) Fig. 75 の画面になるので，「subclipse」にチェックが入っていることを確認し，「終了」ボタンを押すと，Fig. 76 の画面に変わる．ここでは，subclipse にチェックを付け，インストールするプラグインを選択する．選択した後，「次へ」をクリックする．

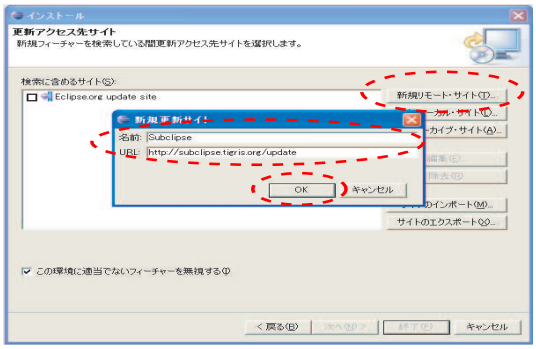

Fig. 74　subclipse の定義（出典：自作）

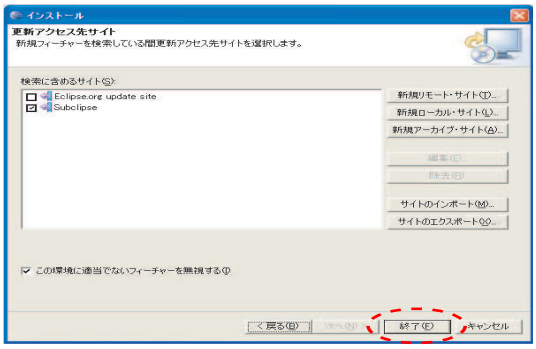

Fig. 75　定義終了（出典：自作）

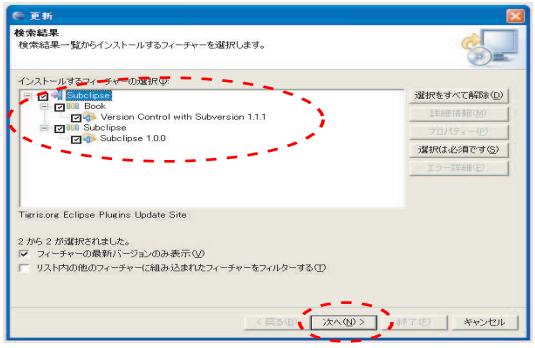

Fig. 76　プラグインの選択（出典：自作）

次に，Fig. 77 の画面で,「使用条件の条項に同意します」にチェックを付け,「次へ」をクリックする．すると，Fig. 78 の画面になるので，プラグインのインストール先を確認し,「終了」ボタンを押す．subclipse のインストールが開始されるが，プラグインにデジタル署名がされていない場合，インストールを続行するか確認されるので，ここでは「全てインストール」をクリックする．(Fig. 79 参照)

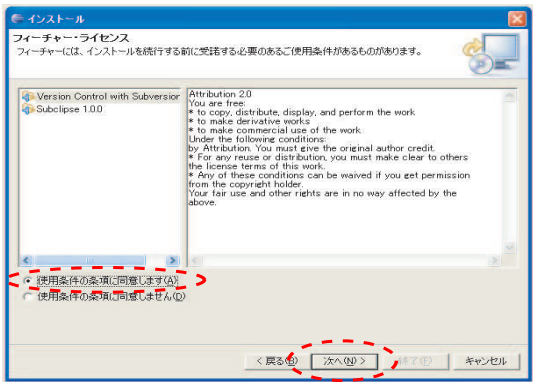

Fig. 77　ライセンスの同意確認（出典：自作）

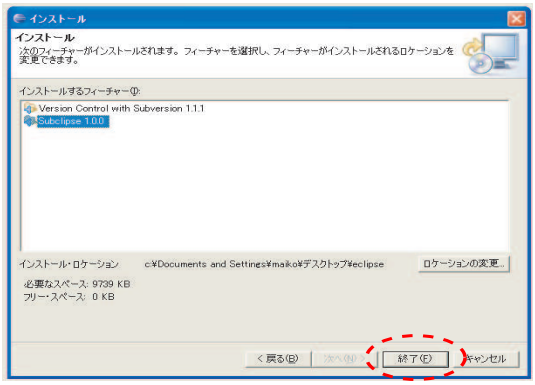

Fig. 78　インストール先の確認（出典：自作）

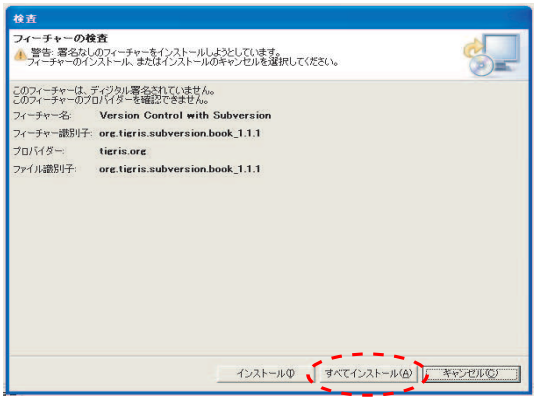

Fig. 79　署名の確認（出典：自作）

インストールが全て終わると，Fig.80 のような再始動を要求されるので,「はい」をクリックする．これで，subclipse のインストールは完了する．メニューバーのヘルプの「ソフトウエア更新」にある「構成の管理」を選択し, (Fig. 81 参照) Fig. 82 のように,「subclipse 1.0.0」と「Version Control with Subversion 1.1.1」が表示されていれば subclipse のインストールはできている．

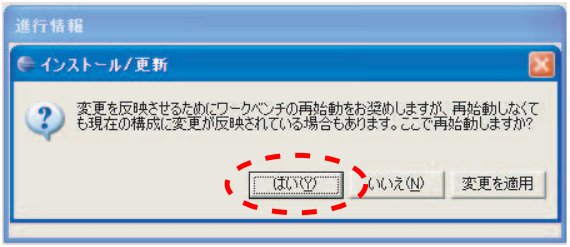

Fig. 80　再始動確認画面（出典：自作）

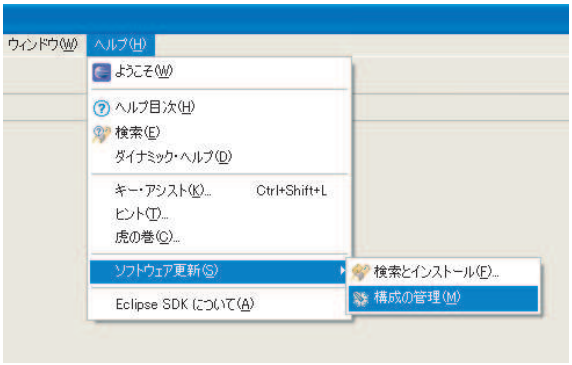

Fig. 81　構成の管理選択画面（出典：自作）

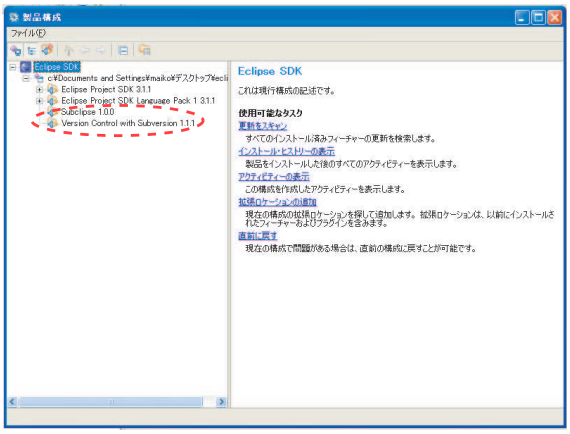

Fig. 82　プラグインの確認画面（出典：自作）

4.1.2　subclipse の使用方法

subclipse を使用するためには，ファイル群をまとめて管理しておくレポジトリを duke 内に作成する必要があるので，先にその手順について述べる．まず，cygwin を起動させ duke 内に以下のコマンドで入る．

```
$ ssh mashibe@duke.doshisha.ac.jp
```

すると，パスフレーズが聞かれるので入力すると duke 内に入ることができる．次に，自分のディレクトリ内にレポジトリを以下のコマンドで作成する．ここでは，repos というレポジトリを作成する．

```
$ cd /home/svn/personal/mashibe
$ svnadmin create repos
```

これでレポジトリは完成したので，次に subclipse を使った eclipse でのコミットの方法に移る．今回は，eclipsezemi というプロジェクトを作り，その中に Hello.java と Hello2.java を作成し，Hello.java のみをコミットすることにする．まず，Subversion のインターフェースを選択する．メニューバーのウィンドウにある設定を選択する．(Fig. 83 参照) チームの中にある SVN を選び，SVN インターフェースの「JavaSVN」にチェックを入れ，「OK」ボタンを押す．(Fig. 84 参照)

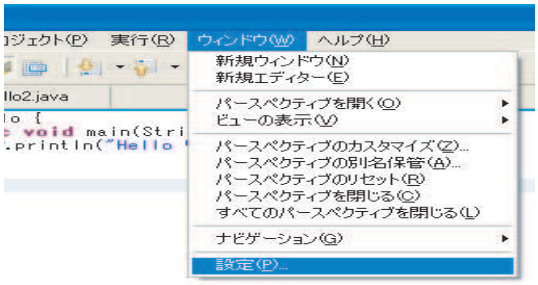

Fig. 83　設定選択画面（出典：自作）

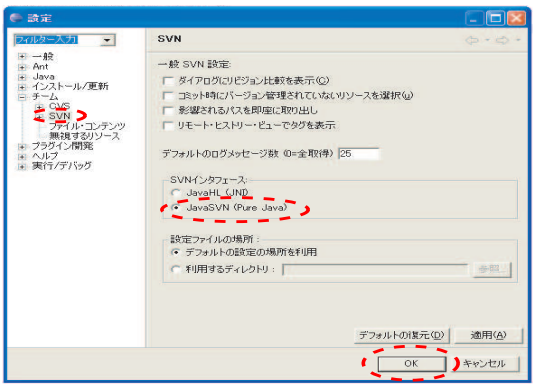

Fig. 84　インターフェース選択画面（出典：自作）

次に，repos にインポートする手順を述べる．プロジェクト eclipsezemi を右クリックし，「チーム」の「プロジェクトの共用」を選択する．(Fig. 85 参照) すると，Fig. 86 のようなウィンドウが出てくるので，「SVN」をクリックして「次へ」を押す．続いて，Fig. 87 ではロケーションに「svn+ssh://mashibe@duke.doshisha.ac.jp/home/svn/personal/mashibe/repos」と打ち込み，「次へ」を押す．

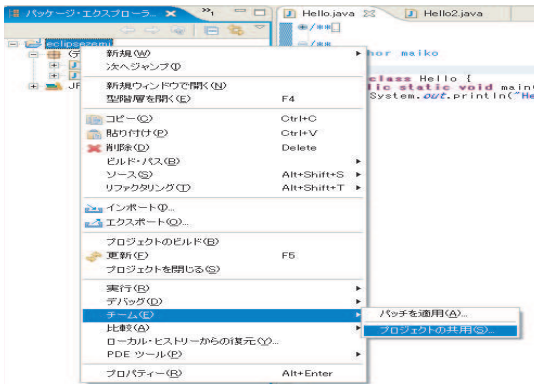

Fig. 85　プロジェクトの共用選択（出典：自作）

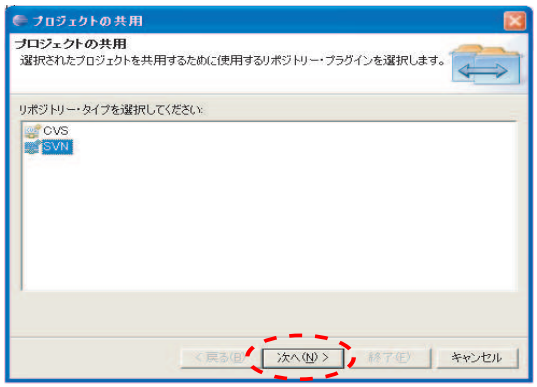

Fig. 86　リポジトリータイプの選択（出典：自作）

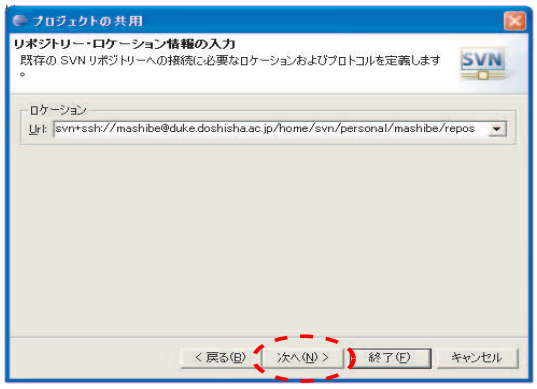

Fig. 87　ロケーション情報の入力（出典：自作）

Fig. 88 のような画面が出てくるので，ユーザ名を確認し，「プライベート・キー認証」にチェックを入れ，キーファイルに cygwin の秘密鍵を参照し，その下にパスフレーズを入れる．最後に，「情報を保存」にチェックを入れてから「OK」を押す．次に，コミットするファイルを選ぶ画面が出てくるので，今回は Hello.java のみを選択する．(Fig. 89 参照) これで，最初に作成したレポジトリ repos に，Hello.java がコミットできた．また，パッケージエクスプローラ上で，レポジトリ管理されているファイルには黄色いタンクが，そうでないファイルには「？」マークが付くので見分けられる．(Fig. 90 参照)

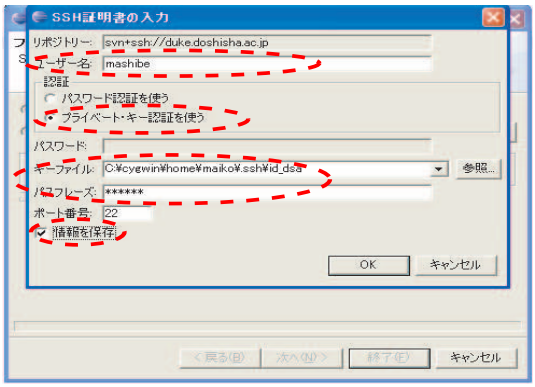

Fig. 88　SSH 証明書の入力（出典：自作）

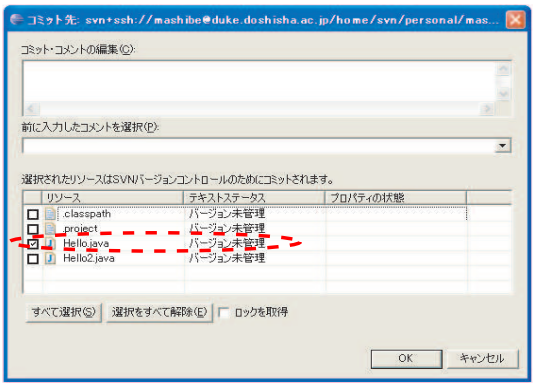

Fig. 89　ファイルの選択（出典：自作）

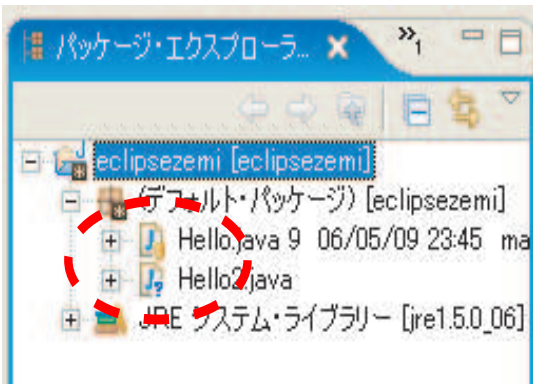

Fig. 90　コミットの確認（出典：自作）

一度，コミットしたファイルを編集して，再度コミットし直したい場合は，コミットしたいファイルを右クリックし，コミットを選択する．(Fig. 91 参照) ただし，

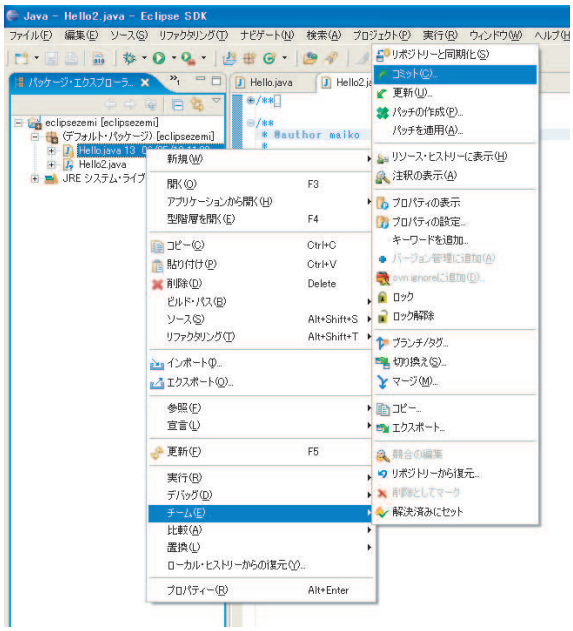

Fig. 91　二回目以降のコミット（出典：自作）

subclipse でレポジトリにアップするのは，ソースのみとする.class ファイルや他のデータをアップするのは望ましくない．

4.2　OMONDO Eclipse UML

OMONDO Eclipse UML とは，Eclipse 上で UML 図を作成するためのプラグインである．

4.2.1　OMONDO Eclipse UML について

Unified Modeling Language(UML) は，システムを視覚化したり，仕様や設計を文書化したりするための表現方法である．

本項では主に OMONDO Eclipse UML のインストール方法について説明する．

4.2.2　OMONDO Eclipse の UML のインストール

はじめに，「http://www.eclipseuml.com/」のページに移る (Fig. 92)．

Fig. 92 では，画面左の「download...」というところをクリックする．すると Fig. 93 へ画面が移動する．Fig. 93 へ移動したら Fig. 93 に示した○部分をクリックする．

Fig. 93 の上段には「Auto-Installer」という項目があり，Fig. 93 はその画面を表示している．ここで，VERSION が 2.1.0.20050927 と書いてある部分があるので，そこをクリックする．

ここで，Fig. 94 のように「ファイルを保存する」を選択する．ここでは保存先をわかりやすくするためにデスクトップに保存する．

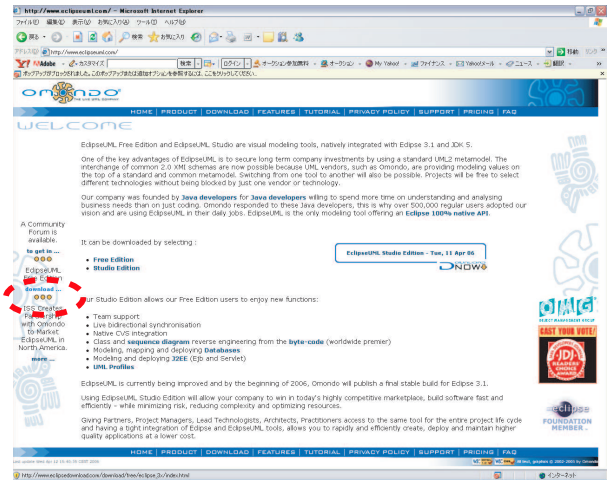

Fig. 92　OMONDO UML のインストール (1)(出展：自作)

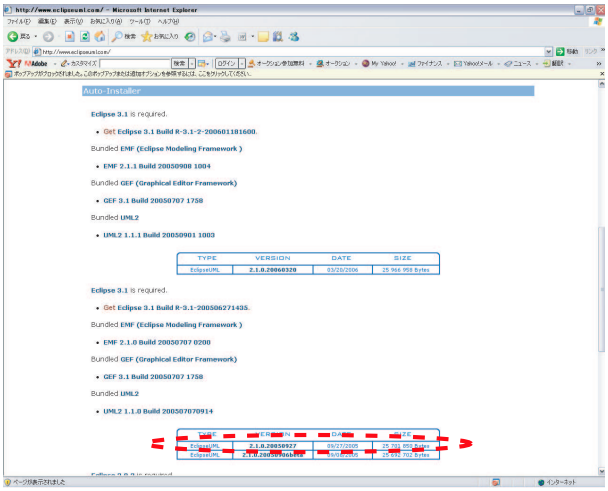

Fig. 93　OMONDO UML のインストール (2)(出展：自作)

さきほどデスクトップに保存したファイル (eclipseUML_E310_freeEdition_2.1.0.20050927.jar) を実行する．

ファイルを開くとインストールに関する文章が出力されるので，ここではデフォルトで「次へ」を選択していく．ライセンス画面 (Fig. 95) では内容に同意するを選択し，「次へ」を選択．

パスの選択 (Fig. 96) については，「feature」および「plugin」がおいてあるフォルダのパスを選択をする必要があるので注意する．

ここで「Eclipse」を実行し，上のツールバーから [ヘルプ] → [ソフトウエア更新] → [構成の管理] を選択する (Fig. 97)．そこで，UML の名の入ったフォルダ (EclipseUML Free Edition 2.1.0 および UML2 1.1.1) があるので、確認できる．

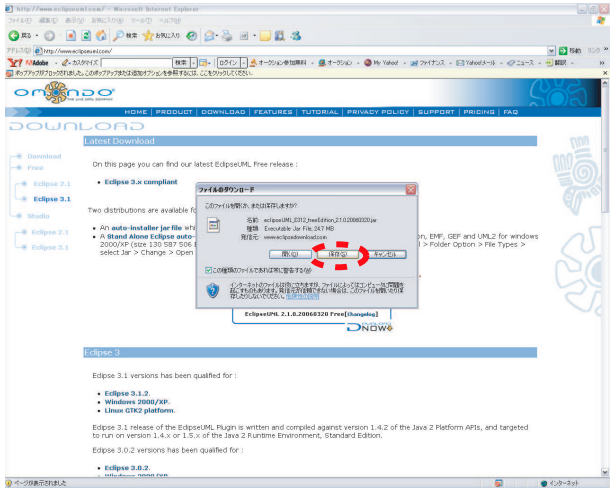

Fig. 94　OMONDO UML のインストール (3)(出展：自作)

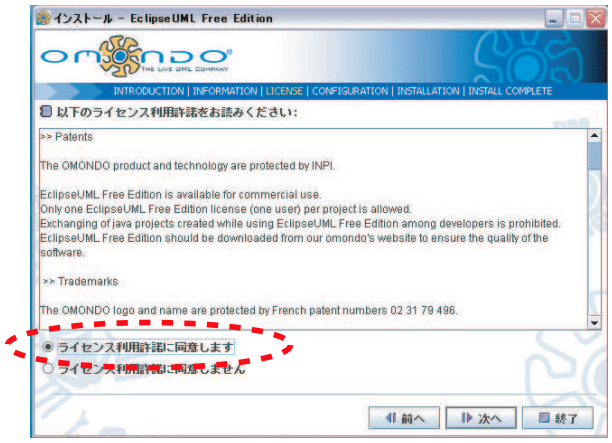

Fig. 95　OMONDO UML のインストール (4)(出展：自作)

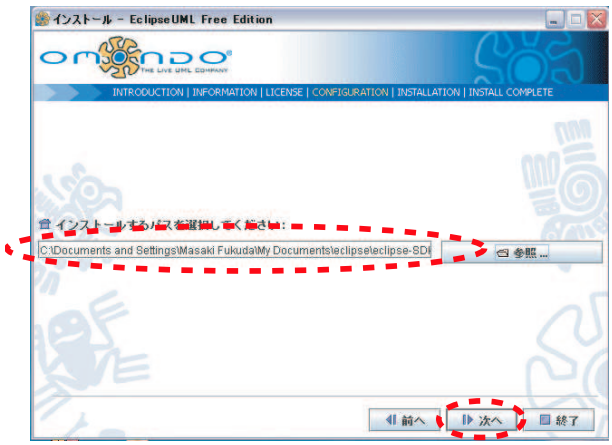

Fig. 96　OMONDO UML のインストール (5)(出展：自作)

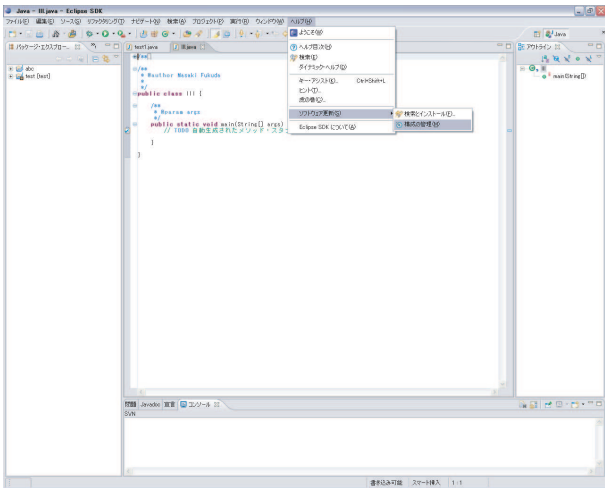

Fig. 97　OMONDO UML のインストール (6)(出展：自作)

以上の手順で UML の使用が可能となった．

4.3　CrossJPropEditor

CrossJPropEditor とは，プロパティファイルを表形式で編集するエディタである．

4.3.1　CrossJPropEditor のインストール

CrossJPropEditor インストールの手順を以下に示す．

はじめに，「http://www.javable.jp/tools/propeditor/」のページに移る．Fig. 98 において○の部分 (2 行あるうちの上段) をクリックすると Fig. 99 へ移る．

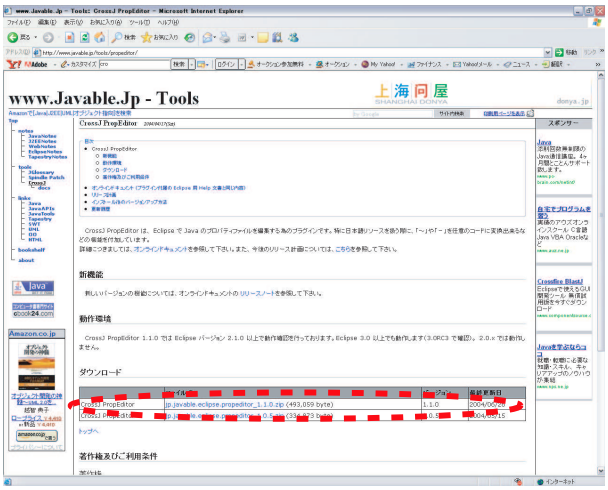

Fig. 98　CrossJPropEditor のインストール (2)(出展：自作)

ここでファイルの保存を選択する．

ファイルの保存に関しては前項目同様，一旦デスクトップに保存し，そのフォルダに含まれる内容 (「feature」および「plugin」) を「eclipse.exe」のあるフォルダの「feature」および「plugin」フォルダにすべて上書きする (Fig. 100)．

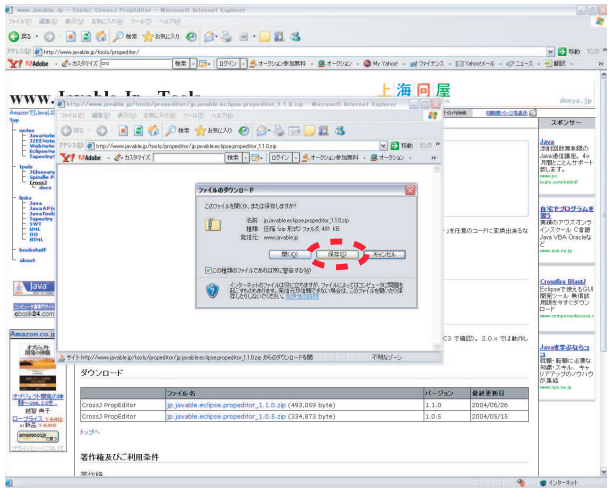

Fig. 99　CrossJPropEditor のインストール (3)(出展：自作)

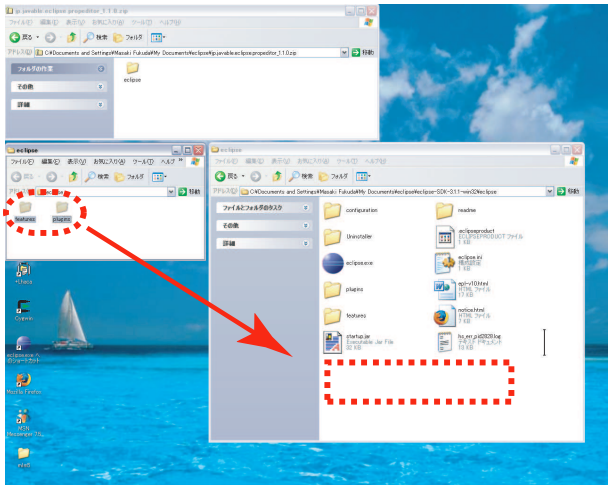

Fig. 100　CrossJPropEditor のインストール (5)(出展：自作)

ここまでの作業で CrossJPropEditor の使用環境は整った．

4.3.2　CrossJPropEditor の使い方

画面左側に位置しているパッケージ・エクスプローラの枠内にカーソルを合わせ (Fig. 101)，右クリックをし，[新規] → [ファイル] を選択し，ファイル名を入力する．このときのファイル名についてであるが，拡張子は必ず「.properties」とする (Fig. 102)．今回は「input」というクラス名に設定したので，「input.properties」というプロパティファイルを新規作成した．

先ほど作成したファイルを右クリックし、[アプリケーションから開く] → [CrossJPropEditor] を選択する (Fig. 103)．そうすると Fig. 104 のような画面になるので Fig. 104 上にあるカーソルの位置で右クリックをし，新規追加を選択する．

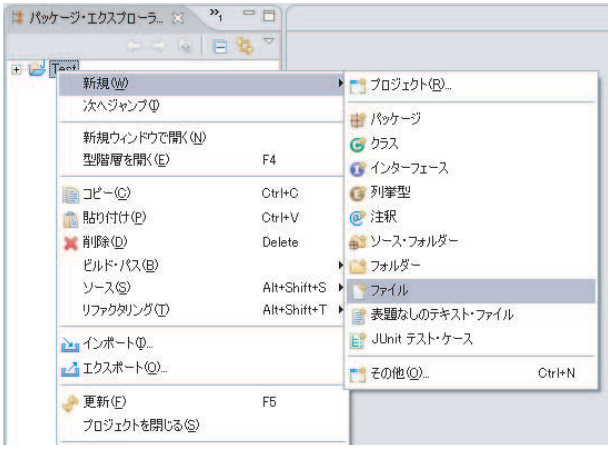

Fig. 101　CrossJPropEditor の利用 (1)(出展：自作)

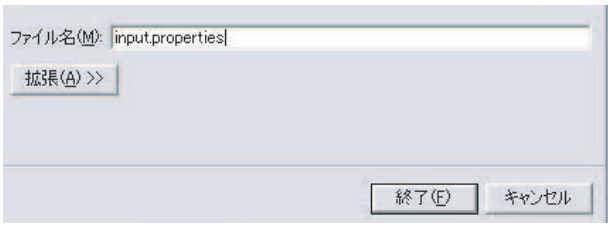

Fig. 102　CrossJPropEditor の利用 (2)(出展：自作)

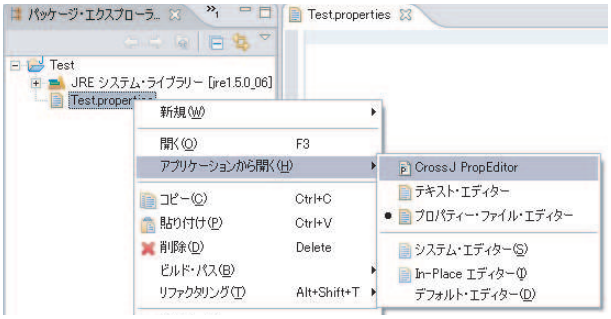

Fig. 103　CrossJPropEditor の利用 (3)(出展：自作)

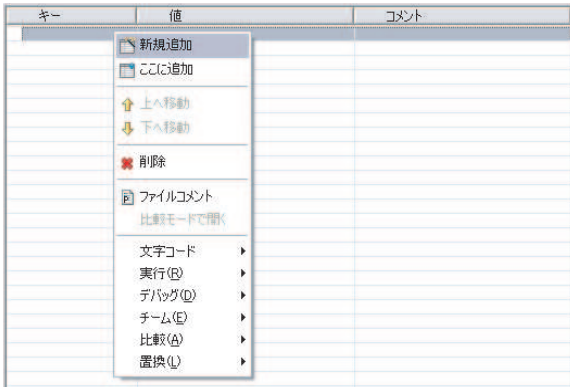

Fig. 104　CrossJPropEditor の利用 (4)(出展：自作)

次にキーとその値を決定する (Fig. 105).

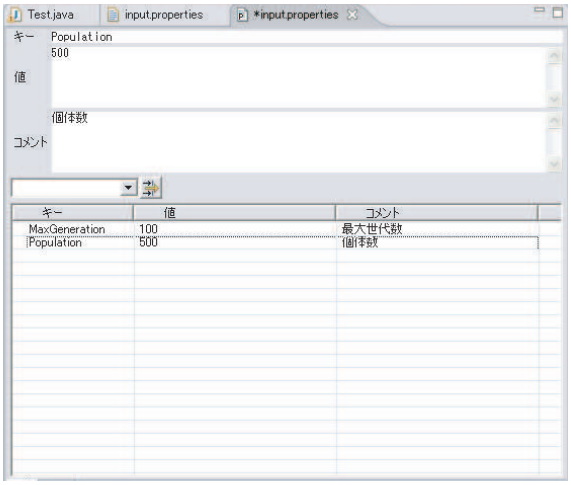

Fig. 105　CrossJPropEditor の利用 (5)(出展：自作)

値を入力してデータの保存後,「workspace」フォルダに入り，先ほど作成したクラスのフォルダに入る．今回は「Test」というクラスを作成したので,「Test」フォルダに入る．

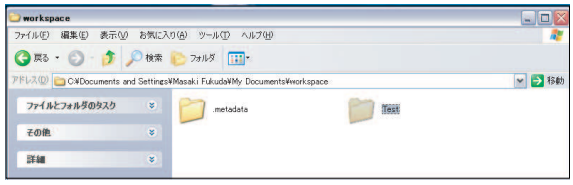

Fig. 106　CrossJPropEditor の利用 (6)(出展：自作)

「Test」ファイルの中に先ほど作成した「input.properties」というファイルがあるので，その中身は Fig. 107 のようになっている．

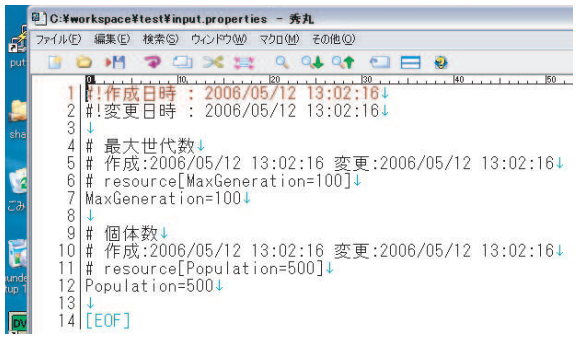

Fig. 107　CrossJPropEditor の利用 (7)(出展：自作)

実際には Fig. 108 のように Properties クラスを利用したファイル入力を用いて，各変数にプロパティファイルの内容を代入していく．

このように CrossJPropEditor はシミュレーデッドア

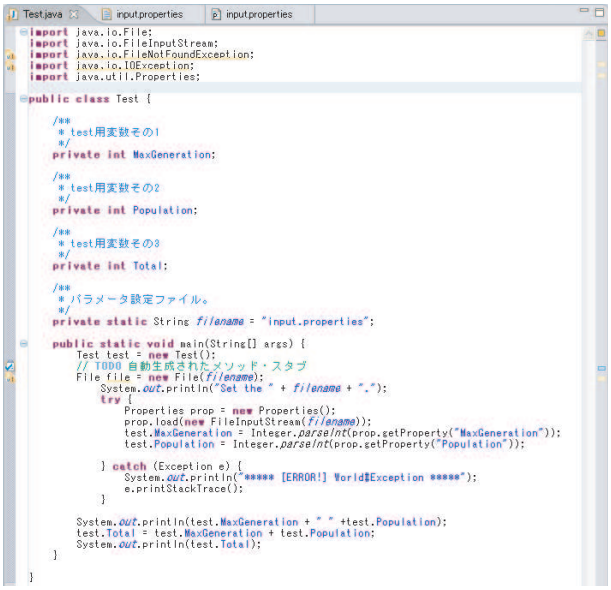

Fig. 108　CrossJPropEditor の利用 (8)(出展：自作)

ニーリングや遺伝的アルゴリズム等のパラメータを管理するのに非常に便利なプラグインである.

参考文献

1) Javadoc での API ドキュメント作成

`http://park15.wakwak.com/~unixlife/java/dev-javadoc.html`

2) subclipse のインストール方法

`http://server.seasar.org/manual/install_subclipse.html`

3) Javadoc

`http://programnet.hp.infoseek.co.jp/omake/javadoc.html`

4) OMONDO eclipse UML

`http://www.eclipseuml.com/`

5) CrossJPropEditor

`http://www.javable.jp/tools/propeditor/`