

Java 6.0 と Eclipse3.2.1+UML の導入

高田司郎

2007 年 04 月 10 日

1 Java 6.0 の導入

2005 年度情報学科指定ノートパソコンに、Java の開発環境 Java 2 Platform Standard Edition 6.0 を導入する方法を説明する。

1.1 Java6.0 の入手

2005 年度学科指定ノートパソコンには、jdk-1.4.2_08 と jdk-1.5.0_02 が導入されている。ファイルは /usr/local/src/java/ 下に保管されている。同じ場所に、jdk-6 のファイルをダウンロードする。ただし src の所有者は info になっているため、他のユーザで導入作業をする場合は root になり chown など所有者を変更する。また Java と Eclipse の導入で /usr/local に空きが約 1.3G ほど必要なため df コマンドで空きを確認し、足らなければ /usr/local/src 以下の不要なファイルを消す。下記の例では java と eclipse を消去している。

```
[info@localhost info]$ df -k
ファイルシステム    1k-ブロック   使用中       空き  使用 %   マウント場所
/dev/hda10           3494492    2564004     752972   78% /
/dev/hda6              77749      10509      63226   15% /boot
/dev/hda8            4127076    429080    3488352   11% /home
none                 252684         0     252684    0% /dev/shm
/dev/hda7            5162796    3684816    1215724   76% /usr/local
/dev/hda5            29994720    593936    29400784    2% /win
[info@localhost info]$ su
Password:
[root@localhost info]# cd /usr/local
[root@localhost local]# chown -R info.users src <=== info は使用する id に
[root@localhost local]# cd src
[root@localhost src]# du -sk *
5304    SML
77608   SWI-Prolog
62136   TKGate
144540  acrobat
458580  eclipse
....
[root@localhost src]# rm -rf java eclipse; mkdir java eclipse
```

```
[root@localhost src]# df -k
```

<http://java.sun.com/javase/ja/6/download.html> にアクセスし、JDK 6 ダウンロードのリンクをクリックする。次に、Required の Accept ラジオボタンをクリックした後、Linux Platform - Java(TM) SE Development Kit 6 の Linux self-extracting file(ファイル名は jdk-6u1-linux-i586.bin) を /usr/local/src/java/ にダウンロードする。

1.2 インストール

インストールは、ダウンロードしたファイルを実行して展開した後、/usr/local/ に移動し、java のシンボリックリンクを変更するだけである。

```
[info@localhost info]$ cd /usr/local/src/java
[info@localhost java]$ su
Password:
[root@localhost java]# sh jdk-6u1-linux-i586.bin
```

使用許諾ライセンスが表示されるので、f を押して進む。終わりに yes/no が聞かれるので yes と入力するとファイルが展開される。次に /usr/local/ に展開されたディレクトリを移動し、java のシンボリックリンクを変更する。

```
[root@localhost java]# mv jdk1.6.0_01/ /usr/local/
[root@localhost java]# cd /usr/local
[root@localhost local]# rm -f java
[root@localhost local]# ln -s jdk1.6.0_01/ java
```

これで、Java は、6.0 に更新された。ただし、日本語の字化けをなくすために、以下の処理もしておく。

```
[root@localhost local]# cd /usr/local/java/jre/lib/fonts
[root@localhost fonts]# mkdir fallback; cd fallback
[root@localhost fallback]# ln -s /usr/X11R6/lib/X11/fonts/TrueType/kochi-gothic.ttf .
[root@localhost fallback]# ln -s /usr/X11R6/lib/X11/fonts/TrueType/fonts.dir .
```

1.3 確認テスト

```
[info@localhost info]$ javac -version
javac 1.6.0_01
[info@localhost info]$ java -version
java version "1.6.0_01"
Java(TM) SE Runtime Environment (build 1.6.0_01-b06)
Java HotSpot(TM) Client VM (build 1.6.0_01-b06, mixed mode, sharing)
```

1.4 ドキュメント

Java SE 6 ドキュメント を、<http://java.sun.com/javase/ja/6/download.html> からダウンロードする。ダウンロードのリンクをクリックした後、Required の Accept ラジオボタンをクリックし、jdk-6-doc-ja.zip を /usr/local/src/java/ にダウンロードする。

```
[info@localhost java]$ cd /usr/local/src/java
[info@localhost java]$ unzip jdk-6-doc-ja.zip
```

Java SE 6 API 仕様は, <file:///usr/local/src/java/docs/ja/index.html> にある . Firefox などファイルを開く .

2 Eclipse の導入

次に, Eclipse 3.2.1 の導入方法を説明する .

2.1 Eclipse3.2.1 と EclipseUML の入手

2005 年度学科指定ノートパソコンには, Eclipse 2.1 と 3.0 が導入されている . ファイルは `/usr/local/src/eclipse/` 下に保管されている . まず, `/usr/local/src/eclipse/3.2` を作成後, そこに, Eclipse 3.2.1 とその日本語パックファイルをダウンロードする .

<http://www.eclipse.org/downloads/> にアクセスする . このマニュアル作成時は, 3.2.2 は日本語サポートがないため, 3.2.1 版をダウンロードするために, Download now: の All versions リンクをクリックする .

まず, Language Pack 3.2.1 Language Packs をクリック後, NLpack1-eclipse-SDK-3.2.1-gtk.zip をクリックすると, ダウンロードサイトを聞いてくるので, 適当に選択し, `/usr/local/src/eclipse/3.2/` にダウンロードする . 次に, <http://download.eclipse.org/eclipse/downloads/> に戻り, Latest Releases 3.2.1 をクリック後, eclipse-SDK-3.2.1-linux-gtk.tar.gz をクリックすると, ダウンロードサイトを聞いてくるので, 適当に選択して, `/usr/local/src/eclipse/3.2/` にダウンロードする .

次に, EclipseUML を入手する . <http://www.omondo.com/> にアクセスし, 左横の EclipseUML Free Edition for Eclipse 3.2 の下の download をクリックした後, EclipseUML Free installer for Windows and Linux: リンクをクリックし, eclipseUML_E320_FreeEdition_2.1.0.20061006.jar ファイルを `/usr/local/src/eclipse/3.2/` にダウンロードする .

2.2 インストール

インストールは, ダウンロードしたファイルを展開した後, `/usr/local/` に移動し, eclipse のシンボリックリンクを変更するだけである .

```
[info@localhost info]$ cd /usr/local/src/eclipse/3.2
[info@localhost 3.2]$ su
Password:
[root@localhost 3.2]# tar xzf eclipse-SDK-3.2.1-linux-gtk.tar.gz
[root@localhost 3.2]# unzip NLpack1-eclipse-SDK-3.2.1-gtk.zip
[root@localhost 3.2]# mv eclipse /usr/local/eclipse3.2.1
[root@localhost 3.2]# cd /usr/local
[root@localhost local]# rm -f eclipse
[root@localhost local]# ln -s eclipse3.2.1/ eclipse
```

これで, Eclipse は, 3.2.1 に更新された . 次に, EclipseUML をインストールする .

```
[root@localhost local]# cd /usr/local/src/eclipse/3.2
[root@localhost 3.2]# java -jar eclipseUML_E320_FreeEdition_2.1.0.20061006.jar
```

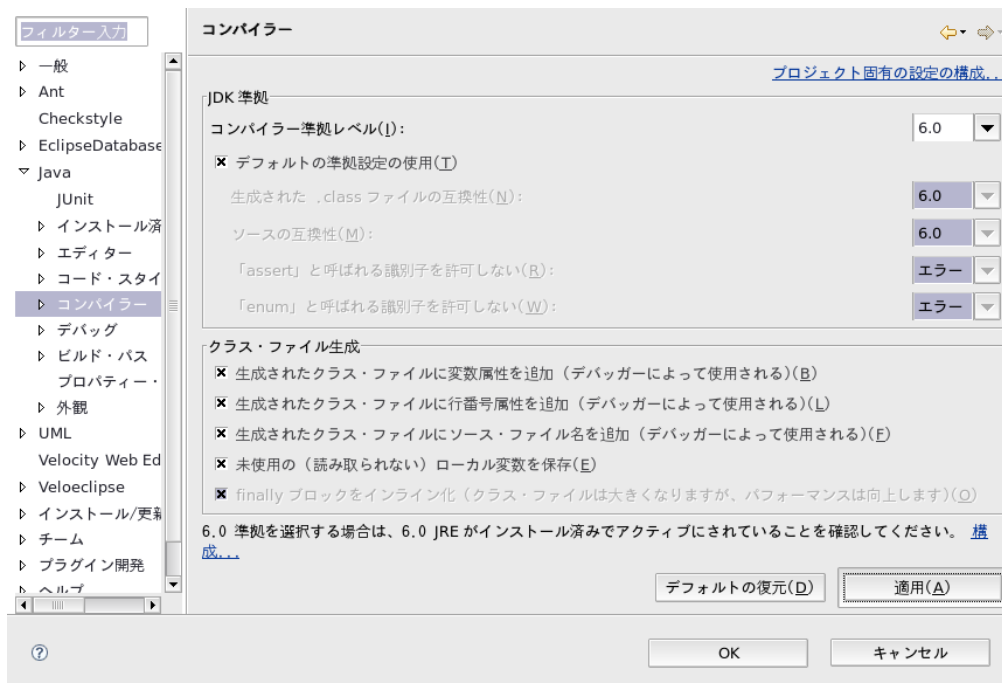


図 1 コンパイラー設定ウィンドウ

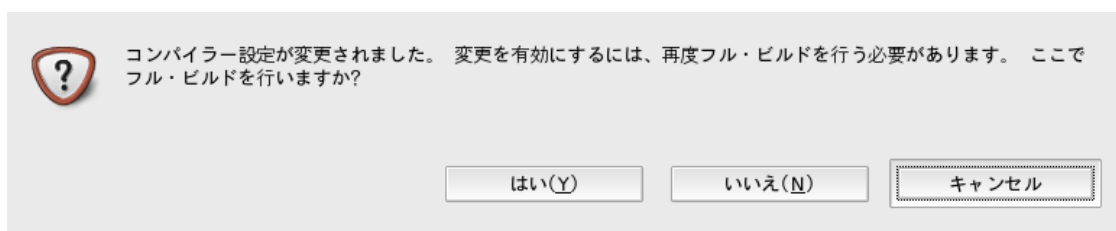


図 2 フルビルド確認ウィンドウ

すると、EclipseUML の導入が開始する。「OK」や「次に」をクリックし、「ライセンス利用許諾に同意します」を選択して「次へ」。インストールするパスを聞いてくるので、`/usr/local/eclipse` になっていることを確認し、「次へ」をクリックして行けばインストールが終了する。

2.3 確認テスト

その後、`root` を `exit` して `info` に戻り、以下の確認テストを行う。Eclipse はバージョンアップ時は念のために `workspace` のバックアップを取っておくこと。下記の例では、`workspace.20070411` をバックアップとして作成している。

```
[info@localhost info]$ cp -R workspace workspace.20070411
[info@localhost info]$ eclipse &
```

Eclipse3.2 が起動すれば導入は成功である。なお、実行開始時に、ワークスペースをどこにするか聞いてくる。今まで同様に、`/home/info/workspace` でよければ、デフォルトとして選択し、「OK」を押す。

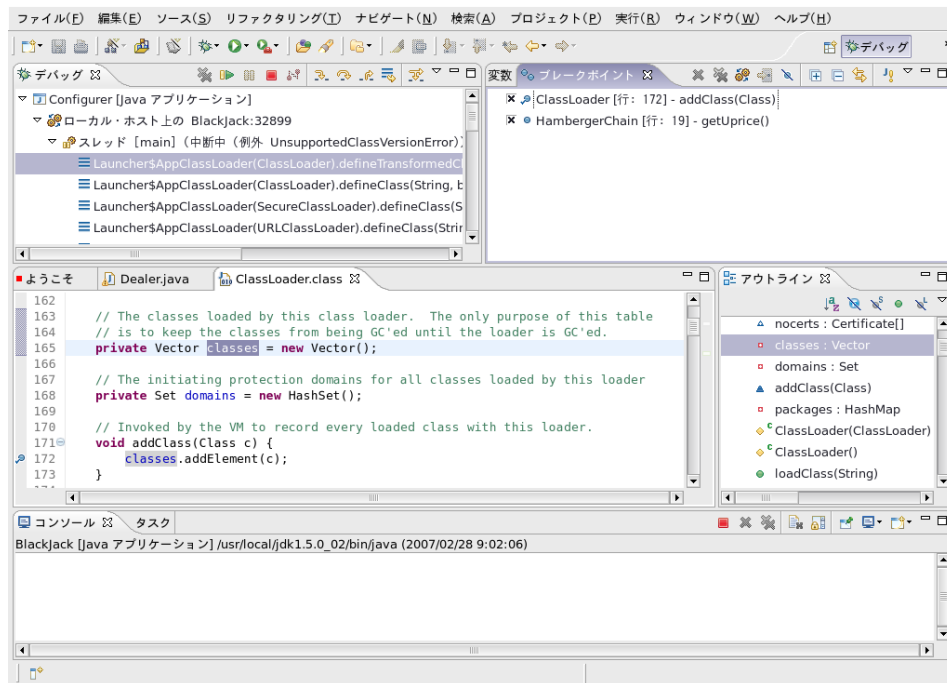


図 3 デバッガーの実行ウィンドウ

2.4 java 6.0 の設定

Java を、6.0 にバージョンアップしたため、Eclipse で使用するデフォルトの Java バージョンを設定する。Eclipse を起動し、プルダウンメニューの「ウィンドウ」→「設定」を選択する。図 1 のように、「Java」→「コンパイラ」を選択して、コンパイラ準拠レベルを 6.0 にし、「適用」を押す。すると、図 2 で確認されるので「OK」を押す。

2.5 UML の起動

処理対象のパッケージをマウスで右クリックし、Open UML で行いたい機能を指定すると、UML は起動する。

3 Eclipse のデバッガー

記述したプログラムが予想外の動きをする。プログラムを眺めてもバグが見付かりそうにない。このような場合は、Eclipse の強力なデバッガーを使用することを勧める。デバッガーの起動は、右上の Java マークの左 (パースペクティブを開く) をクリックして、デバッグを選択すれば起動される。主な機能としては、ステップ実行、ブレークポイント設定、変数などオブジェクトの参照などが挙げられる。図 3 は、AddClass の 172 行目でブレークした図である。右上のウィンドウにおいて「変数」にすれば各種、変数の内容が参照できる。また、デバッグウィンドウのメニューを操作すればステップ実行ができる。

以下に、主な機能を列挙する。

- ブレークポイントの設定：ブレークポイントを設定して、デバッグ作業を開始する。プログラムはブレークポイントに達するまで実行される。この時点での状態を調べることが出来る。
- コードのステップ実行：プログラムがブレークポイントで停止している状態で、1 行ずつ実行することが出来る。

- 指定した行まで実行：ブレークポイントが設定されていない行で、プログラムを停止することが出来る。
- 式や変数の監視：プログラム実行時に、変数や式の値を監視することが出来る。
- ブレークポイントのヒットカウントの設定：ブレークポイントの指定回数に到達したときだけ、そのブレークポイントで停止することが出来る。
- ブレークポイント条件の設定：変数に特定の値が含まれるなど、プログラムがある状態を満足するときだけ、その実行を停めることが出来る。
- 式の評価：デバッグ時に式の評価をすることが出来る。
- デバッグ時の変数への値設定：デバッグ時に変数の値を変更して、実行を続けることが出来る。
- その場でプログラムの変更：プログラムのエラーを発見した時は、デバッグ作業中に修正してテストを続けることが出来る。

これらの使用方法については、Eclipse のヘルプに概要が説明されている。