

第1回BlancoFW勉強会 作業手順 (Mac版)

文書の目的：作業を確実に再現させる

※別紙「第1回BlancoFW勉強会 作業手順 (Mac版) 作業・Log 画像.pdf」ファイルも併せて参照ください

1. ツールなどのインストールと設定

1. 「IntelliJ IDEA CE」をインストール (インストーラーは→[操作画像1](#))

どちらかの方法でインストールしてください

1. homebrew cask でインストール (homebrewのインストールが前提)

```
$ brew tap caskroom/cask  
$ brew cask install intellij-idea-ce
```

2. インストーラでインストール (→[操作画像1](#))

<https://www.jetbrains.com/idea/>

a. 折り返し (soft wrap) の設定 (→[操作画像2](#))

Preferences > Editor > General パネルで Soft Wraps の「Use soft wraps in editor」にチェックを入れる

2. JDK (Java Development Kit) のインストール

すでにインストールされているか、されているならバージョンを確認。

```
$ java -version
```

を実行

- a. インストールされていない場合 → インストールが必要です
- b. 以前のバージョンが表示される場合
(ver 1.6以上なら動作しますが、最新バージョンをインストールします)
 - 最新バージョンのインストールが必要です。
 - 環境変数の設定によるバージョンの切り替えが必要です。
- c. 最新バージョン (10.0.2) が表示される場合 → 環境変数の設定を確認してください。

参考

インストールされていない場合

```
1 $ java -version  
2 -bash: java: No such file or directory
```

以前のバージョンが表示される場合 (例はバージョン1.8)

```
1 $ java -version
2 java version "1.8.0_181"
3 Java(TM) SE Runtime Environment (build 1.8.0_181-b13)
4 Java HotSpot(TM) 64-Bit Server VM (build 25.181-b13, mixed mode)
```

最新バージョン (10.0.2) が表示される場合

```
1 $ java -version
2 java version "10.0.2" 2018-07-17
3 Java(TM) SE Runtime Environment 18.3 (build 10.0.2+13)
4 Java HotSpot(TM) 64-Bit Server VM 18.3 (build 10.0.2+13, mixed mode)
```

最新バージョンのインストール

どちらかの方法でインストールしてください

1. homebrew cask でインストール (homebrewのインストールが前提)

```
$ brew tap caskroom/cask
$ brew cask install java
```

2. インストーラでインストール

(参考: <https://eng-entrance.com/java-install-jdk-mac>)

- a. [公式ページ](#)で、「Accept License Agreement」にチェックを入れ、インストーラーをダウンロード
- b. ダウンロードファイルをダブルクリック、「JDK 10.0.2」ディスクがマウントされるので、「JDK 10.0.2.pkg」ファイルをダブルクリックしてインストール

インストールされているJAVAを確認する方法

コマンド `/usr/libexec/java_home -V` を使います。

```
1 $ /usr/libexec/java_home -V
2 Matching Java Virtual Machines (4):
3   10.0.2, x86_64: "Java SE 10.0.2" /Library/Java/JavaVirtualMachines/jdk
-10.0.2.jdk/Contents/Home
4   1.8.0_181, x86_64: "Java SE 8" /Library/Java/JavaVirtualMachines/jdk1.
8.0_181.jdk/Contents/Home
5   1.6.0_65-b14-468, x86_64: "Java SE 6" /Library/Java/JavaVirtualMachine
s/1.6.0.jdk/Contents/Home
6   1.6.0_65-b14-468, i386: "Java SE 6" /Library/Java/JavaVirtualMachines/
1.6.0.jdk/Contents/Home
7
8 /Library/Java/JavaVirtualMachines/jdk-10.0.2.jdk/Contents/Home
```

コマンド `/usr/libexec/java_home`

は `/System/Library/Frameworks/JavaVM.framework/Versions/Current/Commands/java_home` のシンボリックリンクで、javaインストール時に同時にインストールされます。

環境変数の設定では、この `/usr/libexec/java_home` を利用します。

環境変数の設定

以下、デフォルトのシェルである `BASH` の環境変数を設定します。

他のシェル (ZSH など) も同様に変更してください。

`~/.bash_profile` に以下を追加

```
1 export JAVA_HOME=$(/usr/libexec/java_home -v "10")
2 export PATH=$JAVA_HOME/bin:$PATH
```

ターミナルを再起動

バージョンの確認

```
1 $ java -version
2 java version "10.0.2" 2018-07-17
3 Java(TM) SE Runtime Environment 18.3 (build 10.0.2+13)
4 Java HotSpot(TM) 64-Bit Server VM 18.3 (build 10.0.2+13, mixed mode)
```

参考：環境変数のJAVA_HOMEによるJDKバージョンの切り替え

/usr/libexec/java_home -v "version no"で、JDKのディレクトリが返ってくる。

```
1 $ /usr/libexec/java_home -v "10"
2 /Library/Java/JavaVirtualMachines/jdk-10.0.2.jdk/Contents/Home
3
4 $ /usr/libexec/java_home -v "1.8"
5 /Library/Java/JavaVirtualMachines/jdk1.8.0_181.jdk/Contents/Home
6
7 $ /usr/libexec/java_home -v "1.6"
8 /Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home
```

これを利用して、環境変数PATHに\$JAVA_HOME/binを追加して、環境変数JAVA_HOMEのバージョンを切り替えることでJDKのバージョンを切り替えている。

参考：/usr/bin/javaを使用しない方法 (参考ページ)

JDKのインストールによって、/usr/binディレクトリにjavaがインストールされる。

```
1 $ ls -al /usr/bin/java
2 lrwxr-xr-x 1 root wheel 74 10 22 2017 /usr/bin/java -> /System/Library
  Frameworks/JavaVM.framework/Versions/Current/Commands/java
```

バージョン切り替えには/System/Library/Frameworks/JavaVM.framework/Versions/以下のディレクトリを変更・削除する必要があり、**非推奨**。

oracleが提供しているインストーラー（とhomebrew）でインストールされたjava

は/Library/Java/JavaVirtualMachines以下にあり、/usr/libexec/java_home -v "version no"で、JDKのディレクトリを取得できるので、ディレクトリの変更・削除が必要なく、こちらの利用を**推奨**。

したがって/usr/bin/javaシンボリックリンクは削除したいが、/usr/bin以下の領域はOSX El Capitan以降**システム整合性保護 (System Integrity Protection: SIP) のため変更不可**。

そのため、環境変数PATHの設定をexport PATH=\$JAVA_HOME/bin:\$PATHとし、**先に環境変数JAVA_HOMEで設定したoracleが提供しているjavaのディレクトリを読み込ませる**。

※意図していないバージョンが表示される場合

コマンド/usr/libexec/java_home -Vで表示されないバージョンが現れる場合。

(参考：古いJDKの削除方法 <https://memordm.com/highsierra-java8/>)

3.Antのインストール

すでにインストールされているか、されているならバージョンを確認。

```
$ ant -version
```

インストール

どちらかの方法でインストールしてください

- homebrewでインストール (homebrewのインストールが前提)

```
$ brew install ant
```

- 圧縮ファイルをダウンロードしてインストール

(参考: <https://knowledge.ron-tan.com/422/>)

- a. 公式ページで最新バージョンを確認 (ver 10.0.5)
- b. 任意のディレクトリにファイルをダウンロード

```
$ curl http://ftp.riken.jp/net/apache//ant/binaries/apache-ant-1.10.5-bin.tar.gz -o apache-ant-1.10.5.tar.gz
```

- c. ファイルを解凍

```
$ tar xvfz apache-ant-1.10.5.tar.gz
```

- d. 解凍したファイルを /usr/local/bin に移動

```
$ mv apache-ant-1.10.5 /usr/local/bin/
```

インストールの確認

```
1 $ ant -version
2 Apache Ant(TM) version 1.10.5 compiled on July 10 2018
```

環境変数の設定

以下、デフォルトのシェルであるBASHの環境変数を設定します。

他のシェル (ZSHなど) も同様に変更してください。

~/.bash_profile に以下を追加

```
export ANT_HOME=/usr/local/bin/apache-ant-1.10.5
```

```
export PATH=$ANT_PATH/bin:$PATH
```

ターミナルを再起動。

参考: ver1.8未満のJDKを利用する場合のAntのバージョン

java のver 1.8未満を使用する場合、Ant のver 1.9.13をインストール、環境変数を設定してバージョンを切り替える (今回は使用しない)

[AntのJava対応バージョン](#)

Ant Version	Minimum Java Version
Any 1.9.x release and the git branch 1.9.x	1.5
Any 1.10.x release and the current git master branch	1.8

4.SQLite3のインストール ※Macにはデフォルトでインストールされています

バージョンの確認

```
$ sqlite3 -version
```

5. (gitのインストール.....は、本来は必要ですが割愛)

2. antでBlancoDbEEをビルドし、JDBCコードを生成する

1.以下のgit branchを同じディレクトリに配置 (git clone または zipファイルをダウンロードして解凍)

- <https://github.com/igapyon/blancoDbEE>
- <https://github.com/igapyon/blanco2gDistribution>

配置のイメージ

```

1  .
2  |— blanco2gDistribution
3  |— blancoDbEE

```

→ `./blancoDbEE` フォルダに移動

2.IntelliJ IDEA CE で `./blancoDbEE` ディレクトリを読み込み (→操作画像3)

3./blancoDbEE/build.xmlを変更 (4箇所)

```

1      12
2      -      <property name="encoding" value="Windows-31J" />
3      +      <property name="encoding" value="UTF-8" />
4
5      151
6      -      <pathelement path="${eclipse.dir}/plugins/
org.junit_3.8.2.v20080602-1318/junit.jar" />
7      +      <!-- <pathelement path="${eclipse.dir}/plu
gins/org.junit_3.8.2.v20080602-1318/junit.jar" /> -->
8
9      171
10     -      <blancodb basepackage="my.db" jdbcdriver="org.sqlite.JDBC"
jdbcurl="jdbc:sqlite:.\test\data\sqlite\sqlite.db" jdbcuser="sqlite" j
dbcpassword="password" metadir="test/data/sqlite/" targetdir="test/result/
blanco" sql="true" table="true" encoding="Shift-JIS" />
11    +      <blancodb basepackage="my.db" jdbcdriver="org.sqlite.JDBC"
jdbcurl="jdbc:sqlite:./test/data/sqlite/sqlite.db" jdbcuser="sqlite" jdbcp
assword="password" metadir="test/data/sqlite/" targetdir="test/result/blan
co" sql="true" table="true" encoding="UTF-8" />

```

4. `./blancoDbEE` ディレクトリで `$ ant meta` 実行 (→参考log.1)

このとき、

- `./blancoDbEE/tmp`

ディレクトリが作成されています。

再実行する場合にはこのディレクトリを削除してから行ってください。

5.成果物の確認

- `./blancoDbEE/test/result/blanco/main/my/db/`以下のファイル

3. 生成したJDBCをテストする

1. `./blancoDbEE/test/src/sqlite/SimpleSqliteTester.java`の変更

```
1 9
2 -      final Connection conn = DriverManager.getConnection("jdbc:sqlite:
  e:./test\\data\\sqlite\\sqlite.db");
3 +      final Connection conn = DriverManager.getConnection("jdbc:sqlite:
  e:./test/data/sqlite/sqlite.db");
```

2. (IntelliJ IDEA CEの設定1) 使用するJDKを設定する (→[操作画像4](#))

3. (IntelliJ IDEA CEの設定2) `./blancoDbEE/test/src/sqlite`ディレクトリを「open module setting」から**Sources**にマークする (→[操作画像5](#))

4. (IntelliJ IDEA CEの設定3) 以下のディレクトリをライブラリに登録する (→[操作画像6](#))

- `./blancoDbEE/lib`
- `./blancoDbEE/lib.ant`
- `./blancoDbEE/lib.test`

5. IntelliJ IDEA CE で

`./blancoDbEE/test/src/sqlite/SimpleSqliteTester.java` を実行する (→[操作画像7](#))

ツールウィンドウで「`./blancoDbEE/test/src/sqlite/SimpleSqliteTester.java`」を右クリック、Run

エラーが表示されるが、実行ファイルがビルドされていないため。ビルドは自動で行われ、`./blancoDbEE/bin`フォルダが作成され、再実行の際にはエラーは出ない。

実行結果

```
1 /Library/Java/JavaVirtualMachines/jdk-10.0.2.jdk/Contents/Home/bin/java
  ...
2 Search result: [Field 1 value]
3
4 Process finished with exit code 0
```

以上

