

Chapter 2. 開発のしかた

2-1. Javaの開発環境

2-1-1. 開発環境の種類

使用しているオペレーティングシステムやコンピュータによって、同じJava言語でプログラムを開発と言ってもまったく異なると思います。ここでは、プログラムを開発するためのソフトウェア群を指して開発環境（Development Environment）と呼びます。この節では、個々の開発環境に依存しない最低限の共通事項をまとめてみました。開発環境は基本的には、次のような2つの分類ができます。

★Java開発キット（JDK）の開発環境

この開発環境では、基本的な開発に必要なソフトウェアが単体で用意されています。これを開発に必要な道具ということでツール（Tool）と呼びます。基本ツールとして、コンパイラとランタイム・インタプリタとアプレットの表示のために必要なアプレットビューワ（Applet Viewer）などが用意されています。プログラムを記述するためには、別途何らかのエディタを自分で用意する必要があります。また、この開発環境はネットワークからダウンロードしたりして無償で手に入れることができるものが多いのです。この開発環境の代表的なものとしては、標準となっているSun社のJDK（Java Development Kit）があります。

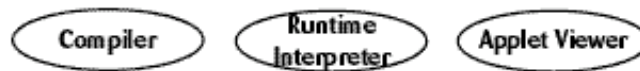


図2-1 開発キットの開発環境

★統合型の開発環境

上記の開発用ソフトウェアの他に、エディタや複数のプログラムを管理するソフトウェアが用意されています。これらの対話的なツールが揃っている環境をIDE（Integrated Development Environment：統合化開発環境）と呼ぶことがあります。プログラムの実行を途中で止めたり、その途中の状態を見ることができるデバッガ（Debugger）と呼ばれるツールも付属しています。また、グラフィックベースでソフトウェア開発を行なえるRAD（Rapid Application Development）ツールを付属していたり、むしろそちらを主体にしているものがあります。総合的な開発環境になっていますので、ボタンやメニューで簡単にコンパイルしたり、実行させたりすることができます。この開発環境は、有償の製品と供給されるものがほとんどです。代表的なものとしては、この本で標準として使うApple社のProject Builderや、Metrowerks社のCodeWarriorや、あるいはBorland社のJBuilderなどがあります。

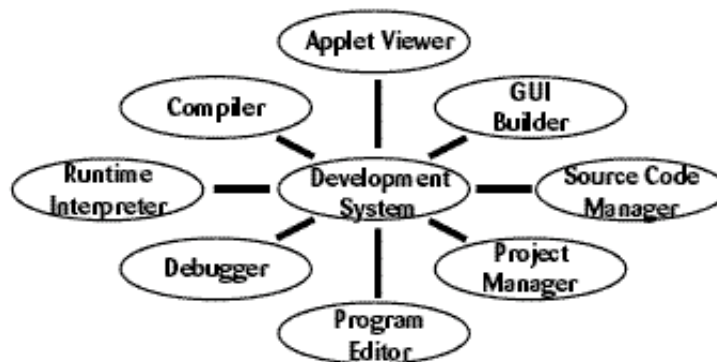


図2-2 総合型の開発環境

2-1-2. 使用するファイルの種類

★ソースファイル（ソーステキストファイル／ソースプログラムファイル）

Javaプログラミング言語で記述されたプログラムファイルです。これは自分で作成する必要があります。ファイル名に関しては、次のような約束事があります。

.javaというファイル名になっていなければいけません。の部分は、適当な名前を付けても構いませんが、ただし少し制限もあります（後で説明します）。また、オペレーティングシステムによっては、付けられる名前に制限があります。

例えば、Sample.javaという名前のファイルは、ソースプログラムファイルです。



Sample.java

★クラスファイル

中間コードに翻訳されたファイルです。これは、コンパイラによって生成されます。Javaでは、実行の単位がクラス（後の章で説明します）と呼ばれるものになっていますので、このような名前がついています。このファイル単体では実行することはできません。実行にはランタイム・インタプリタを必要とします。アプレットなどがネットワークを介して転送されるときは、このファイルが転送されます。この種類のファイルも名前に関して約束があります。

***.classというファイル名という名前になっていなければいけません。

例えば、NSArray.classという名前のファイルは、クラスファイルです。



NSArray.class

★パッケージファイル（クラスライブラリ）

クラスファイルが幾つか合わさったものになっています。複数のファイルが一つにまとめられたファイルをアーカイブ（Archive）ファイルと呼ぶことがあります。Jar形式のものとZip形式のものがあります。



CoreAudio.jar



QTJava.zip

★プロジェクトファイル

総合型の開発環境では用意されているもので、複数のソーステキストからプログラムが構成されていたり、実行に必要とされている画像・音声ファイルまでまとめて管理してくれるものです。



BlastApp.pbproj

2-2. Mac OS X上での開発環境

2-2-1. 代表的な開発環境

代表的な開発環境について簡単に紹介します。本書ではProject Builderを中心に取り上げていきますが、付録の部分にJDKなどの環境での開発の仕方を記しておきますので、そちらを参照してください。

★Project Builder (Xcode)

Mac OS Xでの、中心的な開発環境です。統合的な開発環境で、Cocoaなどのライブラリを利用する場合は、Interface Builderとも連携をします。Java以外にも、C/C++、Objective-C、AppleScriptなどの言語で開発することもできます。Mac OS X 10.3 (コード名: Panther) からは、Project Builderの後継のXcodeが使われることになりました。



★CodeWarrior

元はMacOS用の中心的な開発環境だったのですが、Apple社が自社でProject Builderなどの開発環境を用意したことから、Windowsでの開発環境に主力を注いでいます。簡単で直感的な統合開発環境で、Project Builderも、CodeWarriorを手本に作られているところがあります。Java以外にC/C++言語で開発が出来ます。特に、現在は組み込み機器やゲームマシンのPlayStationなどの、実行するコンピュータが異なるクロス開発環境として良く使われています。

★JBuilder

フリー版がある統合的な開発環境です。ただし、英語版なので、英語だけが辛いという人には、あまりお勧めできません。

★JEdit

アプリケーションだけを動かすことに限定した、シェアウェア版の開発環境です。AppleScriptのように、プログラムをエディタで書いて、保存して、実行(再生)ボタンを押すと、自動的にコンパイルされ、実行されますので、簡単なアプリケーションを記述するときには、一度試してみることをお勧めします。

★JDK

Sun MicrosystemsのJDK (Java Developer Kit) に準拠した、Apple社が開発している基本的な開発環境です。上記のすべての開発環境は、このJDKのコンパイラや、アプレットビューワ、JavaVMなどを用いています。あらかじめ、一般のエディタでJavaのプログラムを記述しておけば、ターミナルのアプリケーションから、JDKで用意されているコマンドを直接実行して、コンパイラやアプレットビューワを用いることもできます。

2-2-2. MacOS Xに含まれる補助ツール

Developer ToolのCDをインストールすると、補助的なツールが付いてきます。Cocoaで利用するInterface Builderを除き、Javaに関連するいくつかの補助ツールを紹介します。

以下のものは、/Developer/Applicationsの下にあります。

★JavaBrowser

Javaのクラスライブラリを指定して、どのようなクラスが用意されているのか、利用できる内容を表示させることができます。往年のオブジェクト指向開発言語、Smalltalkのクラスブラウザをまねて作ってあります。ただし、プログラムで利用する基本的なクラスライブラリの内容は、Project Builderからも見ることができます。また、このJavaBrowserは、Finderのサポートサブメニューに登録されており、Finderに切り換えて、Shift + Command + Jのショートカットキーで立ち上げることができます。

★MRJAppBuilder

Javaのアプリケーションを、Mac OS Xの環境で単体で稼働できるようにするためのツールです。

★JarBundler

JavaのJarファイルをアプリケーションとして稼働させるためのツールです。

以下のものは、Developer ToolのCDをインストールしなくても、標準で添付されています。アプリケーション→ユーティリティ→Javaのフォルダ、すなわち/Applications/Utility/Javaのフォルダにあります。

★Java Web Start

Sun Microsystems社が提唱するWeb Startアプリケーションをサポートするためのアプリケーションです。

★Applet Launcher

JDKの標準では、アプレットを実行させた場合、Sun Microsystems社のJDKに準拠したアプレットビューワが立ち上がります。それに替わるApple社独自のアプレットビューワになっています。

★Input Method HotKey

Javaのアプリケーションを起動しているときに、使うことができるホットキー（ショートカットキー）を定義するための環境設定ツールです。

★Java 1.x.1 Plugin Settings

Safariや、Internet ExplorerなどのWebブラウザなどで、Javaのアプレットを表示する際の環境設定を行なうことができます。このアプリケーションを使って、通常は表示されない、コンソール画面などを表示させる設定などもできます。Mac OS Xで用意されている、JDKの2つの版用に、1.3.1用のものと、1.4.1用のものがあります。

★Developer Toolに含まれるJavaのサンプル

標準のクラスライブラリであるJFCや、AWT、Swing以外を用いたサンプルプログラム以外に、QuickTime for JavaやCocoaなどのApple独自のクラスライブラリを用いたサンプルが用意されています。これらは、以下のフォルダにあります。実際に動かすには、アプレット以外のものは、Project Builder用のプロジェクトファイルを開き、そこで実行させる必要があります。Appkitフォルダには、BlastAppなどのゲームも入っていますので、一度実行させてみてください。

/Developer/Examples/Java

2 - 3. Javaプログラムの開発の仕方

プログラムを作って実行するまでは、次の3つの過程が必要であることは、これまでの説明からおわかりのことでしょう。

1. プログラムを編集（作成）する
2. プログラムをコンパイルする
3. 実行する

編集・作成の結果として、ソースファイルが作成されますし、コンパイルの結果としてクラスファイルが自動的に作成されます。各過程について以下にもう少し詳しく説明しましょう。ただし、この節では一般的な開発環境でどのように行なうかということを説明します。特定の開発環境に依存する部分については、本書ではProject Builderを扱いますが、それ以外の開発環境を利用している場合は、付録を参照するなり、その開発環境に添付されているマニュアルを読むなりして下さい。また、アプリケーションとアプレットの場合では、少しやり方が異なりますので、アプレットは次節で別に説明します。

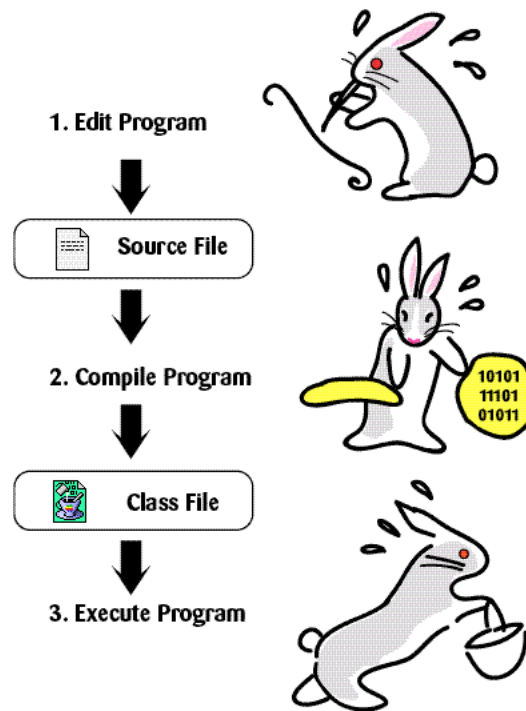


図2-3 プログラムを作って実行するまで

なお、総合的な開発環境の場合は、プログラムを作成する前に、プロジェクトを新規に作成する必要があるものが多いようです。この場合は、マニュアル等に従って新規のプロジェクトを一つ作成してください。

2-3-1. Java言語プログラムの作成の仕方

ワープロを使ったことがある人であれば、通常の文書を書くのとはほとんど同じで、次のような手順となります。なお、総合型の開発環境では、「新規プロジェクト」等のメニューを選択すれば、プロジェクトを新規に作成され、その中のプロジェクトの中のファイルを選択すれば、自動的にエディタが起動されます。また、プロジェクトで新規にファイルを作成したい場合は、「新規ファイル」等のメニューを選択します。

1. 統合型の開発環境では新規のプロジェクトを作成する
2. エディタを起動し、新規に文書を作成するか、既にかかれている文書を編集する
3. エディタの中でJava言語のプログラムを入力・編集していく
4. プログラムを記述終了後、ファイルに保存する

プログラムを保存した後、必ずしもエディタを終了する必要はありません。変更内容がファイルに保存されていれば良いのです。前に述べたように、ファイルを終了する際に、Java言語のプログラムでは、ファイル名が必ず「.java」で終わっていなければならないことに注意してください。ファイル名の終わりの接辞詞（ピリオド以降）をサフィックス（Suffix）と呼ぶことがあります。なお、ファイル名はサフィックスの前の部分はクラス名（後で述べます）と同じにしなければならないという制約があります。

2-3-2. Java言語プログラムのコンパイルの仕方

プログラムを中間コード（JavaVM用のコード）に変換するために、コンパイルします。これも開発環境によって様々ですが、次のいずれかに大別されます。

- ★標準型の開発環境では、コンパイラを起動して、コンパイルするファイルを指定します。
- ★総合型の開発環境では、「コンパイル」や「ビルド」、あるいは「make」といったメニューを選択するか、ウィンドウ内にある対応するボタンを押します。

標準型の開発環境でも、例えばMacOS 9で動いていたMRJ SDKなどの場合にはコンパイラのアイコン上にプログラムのファイルのアイコンをドラッグして離してやる（ドラッグ&ドロップ）すると、それだけでコンパイラが起動し、コンパイルを行ってくれるというわかりやすいやり方もあります。

プログラムの記述の仕方に問題がなかった場合は、接尾詞のjavaが、classに変わった形の名前のファイルが自動的に作成されます。成功した場合は、往々にして何も表示されないことが多いため、本当に成功したかどうか疑問に思うことがあります。逆に、プログラムの記述に間違いがあったときは、間違った場所がエラー（Error）として表示されますので、失敗したことがわかると思います。エラーについては別節で説明します。

2-3-3. Java言語アプリケーションプログラムの実行の仕方

めでたくもコンパイルに成功した場合について説明しましょう。アプリケーションとして記述されたプログラムをコンパイルした結果として生成されたクラスファイルは、そのままランタイム・インタプリタを使って実行することができます。

★標準型の開発環境では、ランタイム・インタプリタを起動して、クラスファイルを指定して実行します。

★総合型の開発環境では、「実行」あるいは「Run」といったメニューを選択するか、対応するボタンを押します。

総合型の開発環境では、標準でデバッガが起動される設定になっているものもあります。そのような環境では、メニューに「デバッグ」あるいは「Debug」といった項目が表示されています。なお、ほとんどの統合的な開発環境では、コンパイルと実行まで、同時に行なってくれるメニューやボタンが用意されています。ですから、コンパイルされて、実行というのが、別々に行なわれているというのがわかりにくいかも知れません。

2-4. アプレットの開発の例

アプレットの開発例を説明する前に、最初に、新たにDevelopment ToolのCDをインストールするところから、説明します。

★Developer ToolとProject Builderのセットアップ

Developer ToolのCDが、購入したMac OS Xについていない場合は、以下のWebページから申し込みます。ADC（Apple Developer Connection）会員なら無料でダウンロードできるとのことですが、まずメンバーシップを得ることから始まります。オンラインのメンバーシップは無料ですので、それに登録してから、ダウンロードをしてみてください。

http://developer.apple.com/ja/tools/	開発ツールのページ
http://developer.apple.com/ja/membership/	メンバーシップのページ

CD-ROMあるいは、ダウンロードしたディスクイメージをマウントすると、インストーラが付いていますので、それでインストールしてみてください。インストールには時間が結構かかります。システムのあるディスクパーティションに、Developerというフォルダができています。

このDeveloperフォルダの下にApplicationsフォルダにProject Builderが用意されています。これをダブルクリックで立ち上げてみてください。最初に、どのようなウィンドウの設定にするか尋ねてくるダイアログが現われます。3つの設定が行なわれます。ビルドの設定→ウィンドウの設定→ファイル管理と進んでいきますが、この2番目のダイアログの「ウィンドウの設定」において、なるべく混乱を避けるために、「1つのウィンドウで表示する」を選んでください。間違えて別のものを選んだ場合でも、後でアプリケーションメニューの「環境設定」のダイアログから変更することができます。うまく、立ち上がりましたでしょうか？

★Project Builderでアプレットを作成し、実行する

まず、ファイルメニューの「新規プロジェクト」を選択します。そうすると、次のようなダイアログが出てくるはずです。この中から、「Java AWT Applet」を選択します。

うまく「Java AWT Applet」を選択できましたでしょうか？もし、違うのを選んだ場合は、「キャンセル」ボタンを押して、もう一度、新規プロジェクトを作るところからやり直して下さい。

次のダイアログでは、プロジェクトの名前を入力します。これが、クラス名などにもなりますので、注意してください。大文字で始め、それ以降はなるべく小文字で記述します。なお、空白は入れないで下さい。ここでは、例として「FirstApplet」と入力してみましょう。それから、プロジェクトのディレクトリという項目で、このプロジェクトを保管するフォルダを指定してください。右側のボタンでファイルブラウザを表示させることができます。授業などで行なう場合は、授業で指定されたフォルダに保管するようにして下さい。一旦、そのフォルダを指定しておきますと、それ以降Project Builderを使う場合に、いつでもそのフォルダが指定されることになります。

さて、画面の説明をしましょう。やや左側に、ウィンドウを左右に2つに区切るいくつかのタブがあります。「ファイル」、「クラス」、「ブックマーク」、「ターゲット」、「ブレイクポイント」などがあります。主に使うの「ファイル」のタブです。ファイルのタブを選択しておいて下さい。下の方では、索引を作成している旨が表示されています。この索引作成には、時間が掛かりますので、気にしないで置いておき、別の作業に取りかかりましょう。

ファイルに2つあることに注意して下さい。

FirstApplet.java	Javaのソースプログラムです
example1.html	Javaを実行させるためのHTMLで書かれたWebページです

それぞれのファイルをクリックして中を見てみてください。

それから、実行させます。メニューでも実行できますが、左から3番目のボタン「ビルド&実行」を押すのが簡単でしょう。メニューの場合は、「ビルド」メニューの「ビルドと実行」を選びます。ショートカットキーのCommand+Rで、同じことができます。

★Project BuilderでSwingのアプレットを作成し、実行する

2-5. アプリケーションの開発の例

Project BuilderのテンプレートでJava AWT Applicationというのがあるのですが、これは説明できないようないきなり難しいプログラムを作ってくれますので、最後の方の章で説明することにして、ここでは、簡単に短いアプリケーションをJEditを使って作成することと致します。

★JEditでアプリケーションを作成し、実行する

JEditを英語のアップル社のホームページからダウンロードします。www.apple.comです。ここで、「Mac OS X」のタブをクリックし、「downloads」のタブをクリックしてください。そして、右側に配置されている分野ごとのリンクから、「Development Tools」のリンクをクリックして下さい。このページにあります。何ページかありますので、次のページなどをみてみてください。ディスクイメージであります。

さて、JEditを立ち上げます。次のようなプログラムを自分で入力してみてください。大文字小文字の区別をしてください。【と】の波括弧（英語ではbraceと呼ばれています）も正確に打ち込んでください。本書では、わかりやすくするために文字が太字になっていたり斜体になっていますが、大抵のエディタではそんな機能はありませんので、気にせず打ってください。空白があるところとないところの区別もしてください。また、2行目から4行目に掛けて行なわれている左側にテキストをずらす記述の仕方は字下げ（Indent）と呼ばれ

ていますが、これもしてみてください。字下げにはTABキーを用います。

```
public class FirstApplication {  
    public static void main( String [] args ) {  
        System.out.println("Hello, World!");  
    }  
}
```

入力できましたら、ファイルメニューの「Save」で保存します。このときのファイル名は、上記のプログラムで書かれているクラス名（public classの後の名前）を用いて、「FirstApplication.java」というファイル名にします。この名前にしないとうまくコンパイル実行できないで注意して下さい。

保存ができましたら、実行（再生）ボタンを押します。これでコンパイルと実行が行なわれます。エラーがある場合は、実行されないで、エラーメッセージが表示されますので注意して直して下さい。