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する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・インター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（RapidApplicationDevelopment）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ProjectBuilder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・インター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（クラスライブラリィ）
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ProjectBuilder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
Mac OS X 10.2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、InterfaceBuilder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。

★Xcode
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。Project　Builderと異なり、まだ日本語化されていません。英語のメニューを使うことになります。ただし、Javaのエラーメッセージなどは、日本語で出ます。

★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、英語だらけが辛いという人には、あまりお薦めできません。実行方法も選ぶ形になります。

★Eclipse
フリー版です。特に、UMLと共に開発を行なうときに使われる統合開発環境です。いろいろ面倒なのですが、使いたい人はトライしてみてください。

★Netbeans
すべてJava自体で記述された統合開発環境です。フリー版です。少し遅いですが、使えることができるのではないでしょうか。ただし、Mac OS X用のインターフェースではありませんので、ちょっと洗練されていません。

★JJEdit
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、一度使ってみることをお薦めします。

★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がもできます。ただし、JDKの中で、Javaの実行環境やライブラリなどは、Mac OS Xに組み込まれて入ってきますの

で、特にインストールする必要はありません。

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。Mac OS X 10.3からは、更に多くのツールが含まれるようになりました。

★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も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ProjectBuilder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。

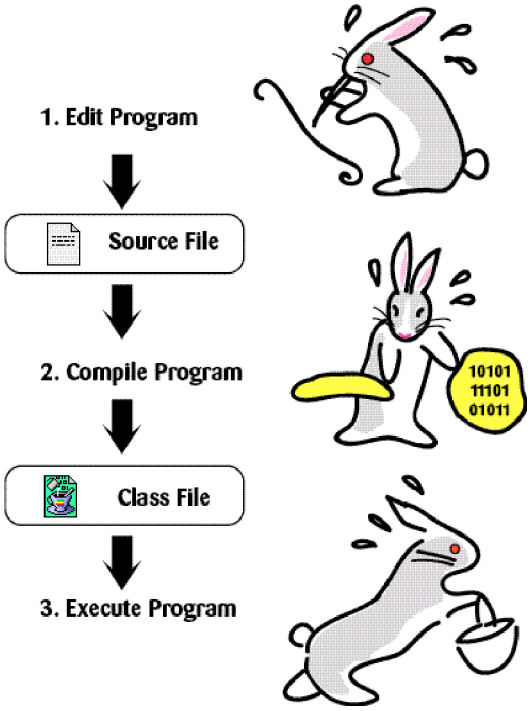

図2-3　プログラムを作って実行するまで

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、既に書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MacOS９で動いていたMRJSDK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、コンパイルを行な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・インター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・インタープリタを起動して、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

本書では、以下の章において、主にアプレットを開発することを中心に説明します。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。

２－４－１．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下さい。

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るはずです。

このDeveloperフォルダの下のApplicationsフォルダにProjectBuilder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。間違っ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？

２－４－２．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ProjectBuilder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」、「ブレークポイント」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。なお、ダブルクリックしますと、別ウィンドウにその内容を表示してくれます。

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。

２－４－３．Project Builderでのアプレット作成時におけるファイル構成

今回プロジェクタ名として、「FirstApplet」を選びましたので、そのようなフォルダが指定されたフォルダに、サブフォルダとして作られているはずです。そのフォルダを開いてみてください。簡単なプロジェクトでも、フォルダ全体として1MBぐらいの容量があります。Project Builder上で見える２つのファイル（FirstApplet.javaとexample1.html）以外に、次のようなファイルとフォルダがあります。

FirstApplet.pbproj	Project Builder用のプロジェクトファイル
build	ビルド時に作成されたフォルダ

このFirstApplet.pbprojは、ダブルクリックすると、Project Builderで開かれるプロジェクトファイルです。buildフォルダの下には、更に次のような構成になっています。

example1.html	実行用にコピーされたHTMLのWebページ
FirstApplet.jar	実行用のJar形式のアーカイブファイル
FirstApplet	クラスファイルへのエイリアス
FirstApplet.build	途中で作られたファイルが入っているフォルダ

ProjectBuilderでは、コンパイルされると生成されるのは、classファイルではなく、jarファイルであるというのがわかります。もちろん、このjar形式でアーカイブされているファイルの中に、FirstApplet.classも含まれています。途中で生成される単体のclassファイルの方は、かなり奥深くはいっており、以下のフォルダの中にあります。パスの区切りにスラッシュ（/）を使いました。

FirstApplet/build/FirstApplet.build/FirstApplet.build/JavaClasses/FirstApplet.class

ちなみに、jarファイルの方をダブルクリックすると、自動的にJar Launcher（このアプリケーションは、次のフォルダにあります：/System/Library/CoreServices）が起動されます。現在何も指定されていないので、何も起きませんが、ここで特定のクラスを指定するために、Jar Bundlerなどを利用します。

２－４－４．Project BuilderでSwingのアプレットを作成し、実行する

基本的には、AWTのアプレットのやり方とほとんど同じです。たとえば、プロジェクト名を、SwingAppletと致しましょう。次のようなやり方で、アプレットを実行させることができます。

1．新規プロジェクトで、Java Swing Appletを選ぶ
2．フォルダを選択し、プロジェクト名をSwingAppletと入力する。
3．ビルド＆実行のボタンで、コンパイルと実行を行なう。

Javaのソースプログラムを見ますと、AWTのアプレットと少し異なるのがわかります。その内容はともかく、Swingライブラリを使ったアプレットも同じように動くことが確認できればよしと致します。

２－４－５．Project Builderのその他の機能

いろいろな機能が用意されていますので、いくつかのJavaに関係する代表的な機能だけ、説明致します。

★タブの収納
タブをダブルクリックするとそのタブで表示されていた部分を収納します。左右や上下に分かれていたウィンドウを見やすく整えることができます。もう一度、そのタブをダブルクリックすると、表示されます。

★ターゲットのタブ
垂直タブの中から、「ターゲット」という名前のついたタブをクリックしてみてください。AWTのアプレットの例を利用して説明しますと、「FirstApplet」という名前の項目があります。それをクリックすると、Javaのコンパイルなどの設定を変更することができます。通常は、変更する項目はほとんどありませんが、日本語を表示させたりする場合は、「ソースファイルのエンコーディング」の項目を日本語の文字コードに設定したりすることがあります。標準設定の状態では、ソースファイルの文字コードがUTF-8に設定されていることが多いので、アプレットの画面上で日本語を表示させるときは、ここで「UTF-8」を選んでおきます。また、「実行可能ファイル」の項目の下に「appletviewer」がありますが、これも、Apple純正のAppletLauncherに変更することも可能です。

★クラスのタブ
これは、JavaBrowserの機能と似ています。Object（java.lang）の項目の下には、標準のクラスライブラリに入っているすべてのクラスが、ABC順で並んでいます。知りたいクラスを選択し、そのメンバーの欄をクリックすると、ウィンドウの右側に、その使い方を表示してくれます。ただし、説明が英語なのですが、気にせず、どんどん調べていきましょう。

★ファイルの追加
単純なプログラムの場合は、１つのJavaのソースファイルでプログラムを記述していきますが、複雑なものになると、複数のソースファイルに分割して、プログラムを記述することになります。既存のファイルがある場合は、「プロジェクト」メニューの「ファイルの追加」で、そのプロジェクトにファイルを追加することができます。新規にソースファイルを作って追加したい場合は、「ファイル」メニューの「新規ファイル」を選び、出てきたダイアログで（スクロールした一番下にあるのですが）、「Pure Java」という項目の中にある「Java　class」を選びます。その次のダイアログなどで、プロジェクトのフォルダに自動的に置いてくれたり、ソースファイルの名前を入力したり、プロジェクトに自動追加したりする設定ができます。

★表示ファイルの選択、削除
あるプロジェクトのソースファイルを編集しているときに、一時的に、別のプロジェクトのソースファイルを見たり、その一部をコピーしたりしたい場合があります。このときは、「ファイル」メニューの「開く」で、別のプロジェクトフォルダにあるJavaのソースファイルを開きます。このとき、ウィンドウの上半部と下半部の境目のところに表示されているファイル名が、ポップアップメニューのような形になります。これで、切り替えすることもできます。中心となって編集したいソースファイルは、別ウィンドウに表示に表示させることもできます（「ファイル」タブで、ファイル名をダブルクリックします）から、同時に２つ見ることもできます。

２－４－６．アプレットとWebページ

アプレットは、SafariやInternetExplorerでも見ることができます。ホームページ用のフォルダにHTML形式のファイルとJarファイルをコピーして実際に動くかどうか確かめてみてください。MacintoshでローカルにWebサーバを立ち上げるには、以下のように致します。Mac OS X上では、自分でWebサーバとしてApacheというプログラムを起ち上げることができます。

システム環境設定（アプリケーション）→共有→サービス
→パーソナルWeb共有の項目にチェックマークを入れる

この、ApacheのWebサーバは、以下のフォルダにあるindex.htmlというファイルを表示しています。

ユーザのフォルダ/Sites

このフォルダに、先ほどのbuildのフォルダから、example1.htmlと、FirstApplet.jarの２つファイルをコピーしてみてください。そして、次のようなURLを、Safari上で入力します。

http://localhost/example1.html

実際に見えましたでしょうか？個人のMacinotshで、一時的にWebサーバを立ち上げる場合は、これで構いません。しかし、インターネットに公開されているWebサーバで管理されているサイトで、自分のホームページ上でアプレットを動かすことも可能です。その他の場合は、各プロバイダや学校組織あるいは企業で方法は異なると思いますので、各自で調べてみて下さい。だいたいは、ファイルをWebサーバーが稼働しているマシンに転送するような形になります。

他の大学などでの環境では、例えばApacheなどでこのような設定がなされているのですが、ユーザのホームディレクトリの下にwwwや、public_htmlという名前（この名前は異なるかも知れません）のディレクトリが用意されていて、そこにHTMLのファイルと、クラスファイルを転送すれば良いという場合が多いと思います。これらのフォルダが、直接Macintoshにマウントできる場合は、そこにコピーするだけです。そうでない場合は、ターミナルなどの端末ソフトを利用して、UNIXのコマンドを使って転送する必要があります。

例として、macosxserver01というコンピュータに設けられているユーザ用のホームディレクトリの中に、public_htmlという外部公開用のフォルダがあり、そこに、example1.htmlというファイルと、FirstApplet.jarというファイルを転送しています。

```
% scp example1.html macosxserver01:~/public_html
% scp FirstApplet.jar macosxserver01:~/public_html
```

scpは、コピーをするコマンドです。%は、オンライン端末上でシステムが表示してくれているものです。この操作をした後に、Safariを立ち挙げて、つぎのようにURLを打ち込んでみて下さい。

http://サーバマシン名/~自分のログイン名/example1.html

自分のログイン名の部分は、各組織で用意されているユーザ名です。ところで、なぜソーステキスト（.javaで終わるファイル）は、転送しなくていいのでしょうか？わからない方は、第１章をもう一度読み直してみて下さい。

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、簡単で短いアプリケーションをJJEditを使って作成することと致します。

２－５－１．JJEditでアプリケーションを作成し、実行する

JJEdit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、JJEdit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。また、クラス名が以下のプログラムでは、FirstApplicationになっていますが、間に空白を入れないでください。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、気にせずに打ってください。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、「Compile」と書かれたボタンを押してみてください。エラーがある場合は、エラーメッセージが表示されますので注意して直してみて下さい。直したら、もう一度保存し、コンパイルボタンを押します。コンパイルが終了すると、「Run」ボタンがアクティブになります。このボタンを押せば実行が行なわれます。

２－６．課題

課題　2-1.
もし、自分用のWebページを持っている場合は、この章で作成したアプレットのクラスファイル（HelloWeb.class）とHTMLファイル（Tester.html）を、そのWebページのフォルダ（ディレクトリ）にコピーして、外部からSafariやInternet Explorerでアプレットの実行ができるかどうか確かめてみなさい。