

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Xcodeや、Metrowerks社のCodeWarriorや、あるいはBorland社のJBuilderなどがあります。

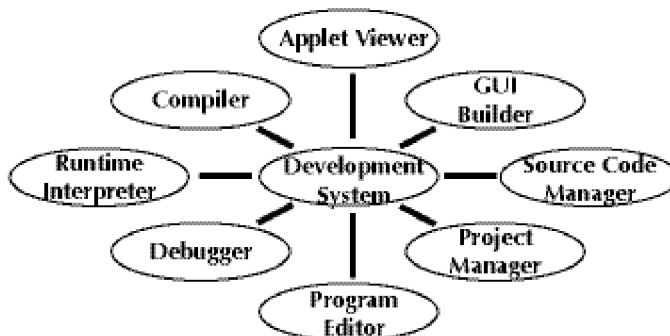


図2-2 統合型の開発環境

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。アッ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)

クラスファイルがいく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。



Sample.xcode

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Xcode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。

★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となります。ただし、Javaのエラーメッセージなどは、日本語で出ます。1つのプロジェクトのサイズが異常に大きくて、7MBぐらいになってしまい、やたらとプロジェクトを作るとディスクを圧迫するのがたまに傷ですが、快適に動いてくれます。



★Project Builder

Mac OS X 10.2までの、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。



★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、英語だけが辛いという人には、あまりお勧めできません。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。ダウンロードするファイルサイズも大きいし、Macintosh版には日本語版がないし、動きがトロいのですが、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のでしょうか。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。



★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、一度使っ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。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。

★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、Xcodeからも見ることができます。



★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.y 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、あるいは1.4.2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。

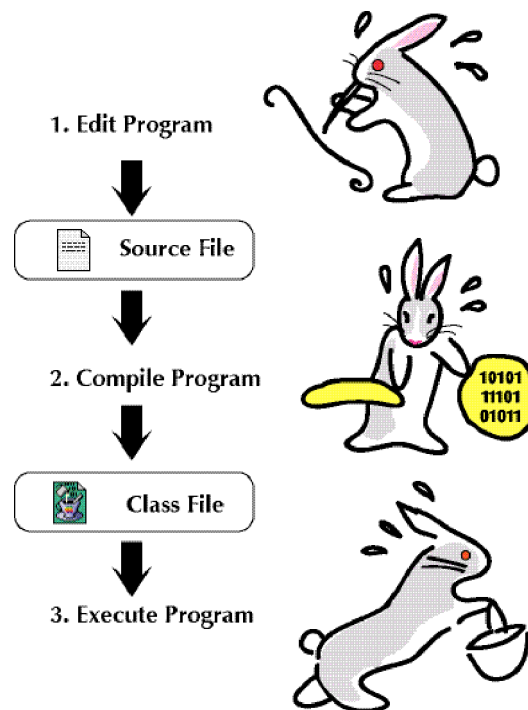


図2-3 プログラムを作って実行するまで

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同様で、次のような手順となります。なお、総合的な開発環境では、「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. 統合的な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けば良いのです。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。

- ★標準的な開発環境では、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。
- ★総合的な開発環境では、「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しれません。

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Xcodeの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Xcodeのセットアップ

Xcode 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はいるはずですが、

このDeveloperフォルダの下のApplicationsフォルダにXcode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。ビルドの設定などを聞いてきますが、OKボタンを押しておいてください。間違っても違う設定をしてしまった場合でも、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. Xcode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「New Project」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。少し下にスクロールさせてください。この中から、「Java AWT Applet」を選択します。

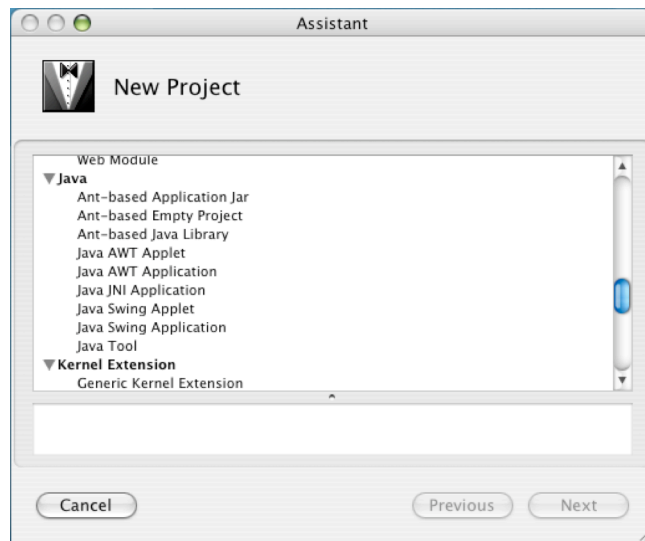


図2-4 Javaのアプレットのテンプレートを選択する

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Xcodeを使う場合に、いつでもそのフォルダが指定されることとなります。

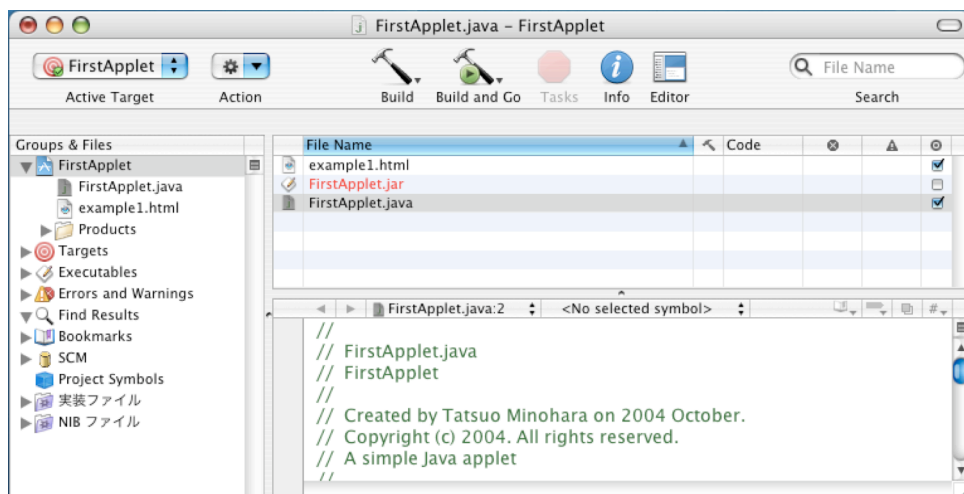


図2-5 ソースファイルを見してみる

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垂直線があります。ここに、今回必要になるファイルが並んでいます。FirstAppletというプロジェクトファイルが選択され

ている場合、右側の上のリストにつきのようなファイルが並んでいるはずですが。

FirstApplet.java	Javaのソースプログラムです
FirstApplet.jar	実行するクラスが入ったアーカイブファイルです。 まだ作られていませんので、赤く表示されています。
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。また、上の方にある「Editor」ボタンを押してみてください。次のようにこのウィンドウを一杯使って表示してくれます。

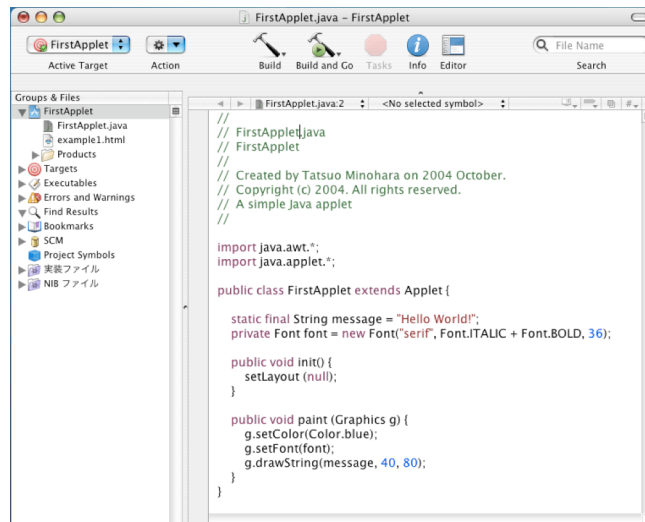


図2-6 Editorボタンでソースプログラムを中心に表示させる

もう一度「Editor」ボタンを押しますと、先程の表示に戻ってくれます。さあ、実行させてみましょう。メニューでも実行できますが、上部中央の左から2番目のボタン「BuildandGo」を押すのが簡単でしょう。メニューの場合は、「Build」メニューの「Build and Run」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。アプレットビューワが立ち上がり、次のような結果を見せてくれるはずですが。

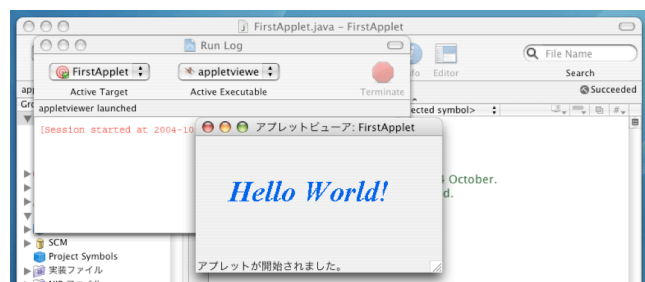


図2-7 AppletViewerが立ち上がり、実行結果を見る

日本語の入力についての補足をしておきます。日本語を使うときは、必ず設定しておいてください。

メインウィンドウの左の項目で、「Target」を選んでください。ソーステキストの中に日本語があった場合の解釈の仕方をここで設定できます。まず、「FirstApplet」というサブの項目を選びます。そうしますと下の図のように、右側のパネルの内側の左のアウトライン表示に「Settings」という項目がでてきますので、そのサブ項目の「Java Compiler Settings」を選びます。そうしますと、右側のメインのパネルの中に、「Source file encoding」というポップアップメニューが現れます。このポップアップメニューが「欧米 (Mac OS Roman)」になっている場合は、うまく日本語が実行時に表示されませんので、「日本語 (Mac OS)」に変更します。この段階で、Fileメニューから「Save」を選んで置くことよいでしょう。プロジェクト

ファイルに設定が保存されます。編集画面に戻るには、左側のアウトライン表示で、一番上に置かれている水色のアイコンの方の「FirstApplet」をもう一度クリックして選択し、右側のリストからファイルを選びます。

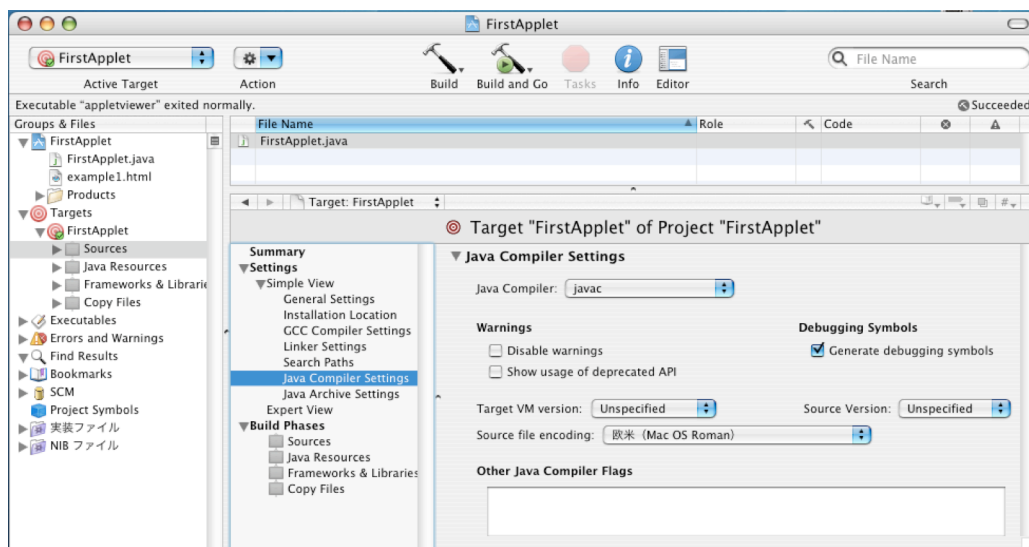


図2-8 日本語設定のためにエンコーディングを変更する

ソースプログラムの編集ウィンドウを別個に表示させると、FormatメニューのFileEncodingのサブメニューが選べるようになりますが、そこでファイルのエンコーディングの設定を変更した場合は、ここで説明した「Targets」の方のJava Compiler Settingsの方も、同じエンコーディングを設定しておく必要があります。そうしないと、コンパイラで日本語の書かれている部分などがエラーになってしまいます。

また、XcodeメニューのPreferenceダイアログの「Text Editing」のタグには、「Default Encoding」の設定ポップアップメニューがあります。通常は、「Systemdefault（日本語（MacOS）」という項目になっているのですが、特に日本語を使うことを前提とするときは、これを「日本語（Mac OS）」に変更しておくといいでしょう。これ以降のプロジェクト作成のときに、参照されます。

2-4-3. Xcodeでのアプレット作成時におけるファイル構成

今回プロジェクト名として、「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が、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6.8MBぐらいの容量があります。Xcode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.xcode	Xcode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.xcodeは、ダブルクリックすると、Xcode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

Xcode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、アプリケーションの場合に、ここで最初に実行される特定のクラスを指定するために、Jar Bundlerなどを利用します。

さて、プロジェクトのフォルダ自体は6.8MBもあります。これは、インデックスファイルのサイズが大きいのです。インデックスのあるフォルダは、6.5MBもあります。手持ちのディスクの容量が少なく、そのプロジェクトをそれ以降あまり用いないときは、このインデックスファイルを消しても良いのではないかと思います。インデックスファイルが置かれているフォルダは以下の通りです。

FirstApplet/build/FirstApplet.build/FirstApplet.pbxindex

2-4-4. Xcode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。

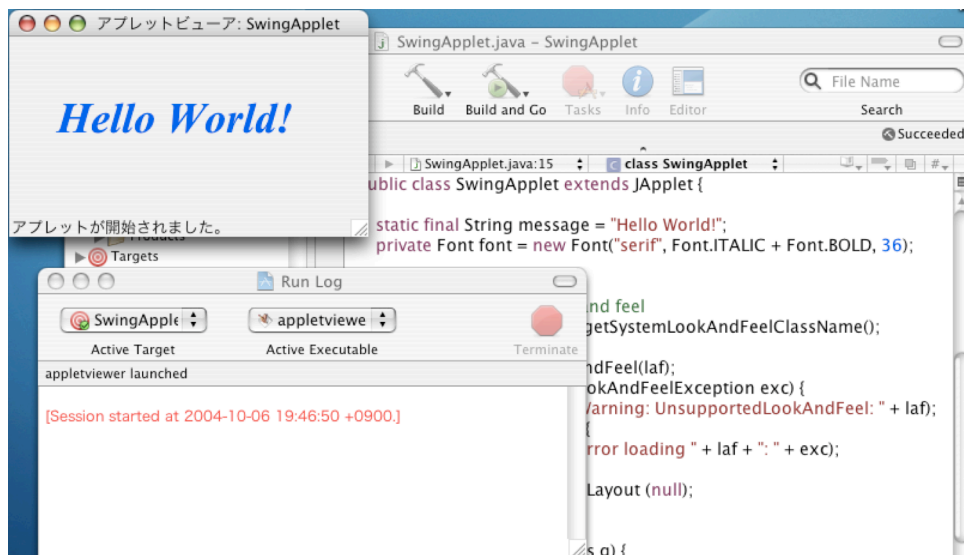


図2-9 Swingのアプレットの実行結果

2-4-5. Xcodeのエラーの表示

プログラムを自分で書くようになりますと、プログラムが規則に添った形でない場合にエラーが表示されます。この場合、Xcodeがどのようにエラーを表示するか見てみましょう。たとえば、FirstApplet.javaのテキストで、setFontとなっている部分をsetfontにわざと間違えて編集し、保存します。コンパイラが、大文字と

小文字の違いを見分けて、エラーを表示してきます。エラーを見るには、左側のアウトライン欄の「Errors and Warnings」を項目を選び、右向きの三角形をクリックして、下向き▼にします。FirstApplet.javaのサブ項目をクリックすると、右側の上部にエラーの一覧リストを表示してくれます。そして、そのリストの中で選択されているエラーの箇所を下のプログラムの画面の該当する行に赤い○に×が入ったマークを表示してくれます。それが、エラーのある行です。

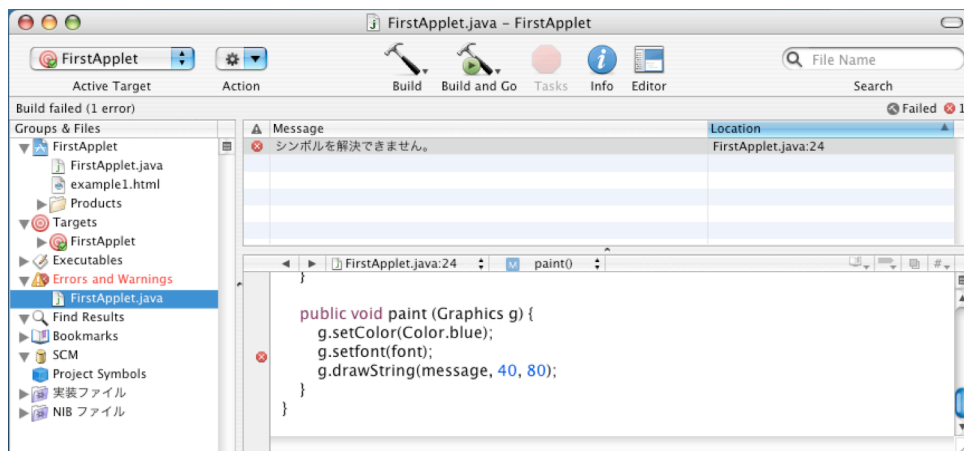


図2-10 エラーの箇所を表示する

エラーが表示されたら、その部分を直して、もう一度、保存してから再コンパイル（BuildandGoのボタンを押すので構わないでしょう）してみてください。エラーがあるうちは、プログラムはコンパイルしている段階で処理が中断され、実行はされません。

2-4-6. Xcode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。

★Targets

日本語の設定のときに説明しましたが、「Targets」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、中国語や日本語を表示させたりする場合は、「source file encoding」の項目をその国の語の文字コードに設定したりすることがあります。この文字コードがUTF-8に設定されている場合は、Unicodeになりますので、さまざまな国の文字を同時に表示させることが可能となります。また、「Executables」の項目の下に「appletviewer」がありますが、これをダブルクリックすると実行時の設定を変更できるダイアログが現れます。ここで、Apple純正のApplet Launcherに変更することも可能です。

★クラスを見る

これは、Java Browserの機能と似ています。Projectメニューの「ClassBrowser」という項目から起動します。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。上方についているポップアップメニューを「Flat,allclasses」にすると、すべてのクラスがでてきますので、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、ウィンドウの右の下側に、その定義を表示してくれます。

Java Browserも、このClass Browserもそれほどきちんとした説明を出してくれませんが、英語でのクラスの説明をみる場合は、Safariで以下のフォルダにあるファイルを開いてみてください。

</Developer/ADC Reference Library/documentation/index.html>

ここで、「Java」というリンクを辿り、そのページから「API Reference」というリンクを辿ります。

ただし、Xcode Developer Toolをインストールした段階では、このクラスの説明は展開されておらず、文字端末（ターミナル）で以下のようにコマンドを入力する必要があります。ここでは、2行に分かれて書かれています。1行で続けて入力してください。管理者のパスワードを聞いてきますので、自分用のMacintoshの場合は、自分のパスワードを入力します。最初の%は入力する必要はありません。

```
% sudo /Developer/ADC/Reference/Library/  
documentation/Java/scripts/unjarJavaDocumentation.sh
```

日本語での説明は、インターネット上で以下のURLから迎えることができます。URLが替わるかも知れないのですが、日本のSUNのJava Developer Connectionのページで、「J2SEドキュメントリスト」から見るることができます。

<http://sdc.sun.co.jp/java/index.html>

★ファイルの追加

単純なプログラムの場合、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、「Project」メニューの「Add to Project」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、「File」メニューの「New File」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、「File」メニューの「Open」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界線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になっています。これで、切り替えることもできます（境界線上の左右の三角ボタンでも切り替えられます）。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立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Macintosh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サーバが稼働しているマシンに転送するような形になります。

たとえば、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、unixserver01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 unixserver01:~/public_html
% scp FirstApplet.jar unixserver01:~/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を立ち上げて、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見て下さい。

2-5. アプリケーションの開発の例

Xcodeのテンプレートで「Java Tool」というのがあります。これが、アプリケーションを作る役割を担っています。同じアプリケーションを作るのに、使い勝手のよいJEdit も利用してみることになります。

2-5-1. Xcodeでアプリケーションを作成し、実行する

Xcodeで、新規プロジェクトを作成します。このとき、テンプレートとしては、「Java Tool」を選びます。注意して欲しいのは、XcodeのJava AWT Applicationというテンプレートです。これは、ウィンドウを伴うアプリケーション用なので選ばないで下さい。説明できないような難しいプログラムをいきなり作ってくれます。テンプレートを選んだ後は、プロジェクトの名前を入力してみてください。ここでは、「FirstApp」という名前にします。これ以降は、アプレットを作成するときと手順が同じです。実行についても同じように行なえます。

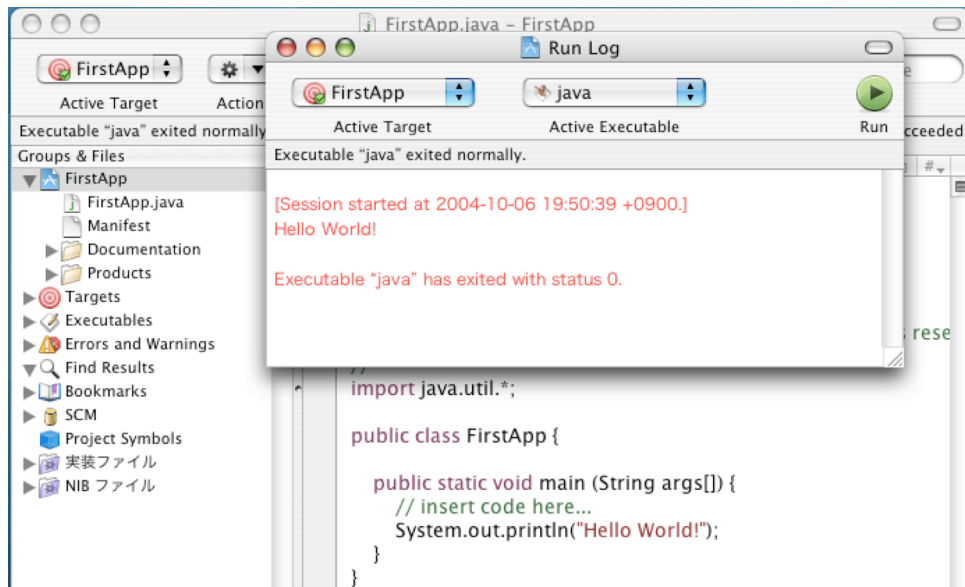


図2-11 XcodeによるJavaアプリケーションの実行

実行されると、白い画面にメッセージが表示されます。この画面は、実行時に起動される標準端末を疑似的に再現するもので、RunLogという名前がついています。RunLog（あるいはデバッグするときに使われるのはDebug Log）は、pseudo terminalというソフトウェアで実現されているのですが、このソフトウェアが日本語をうまく通さないので、日本語のメッセージをRunLogに表示をさせることができません。日本語で表示をみたい場合は、次に説明するJJEditを利用した方が良いでしょう。

2-5-2. 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、間に空白を入れしないでください。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!");
    }
}
```

インデントされた行を左に戻す場合は、Deleteキーを使います。入力できたら、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ください。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。

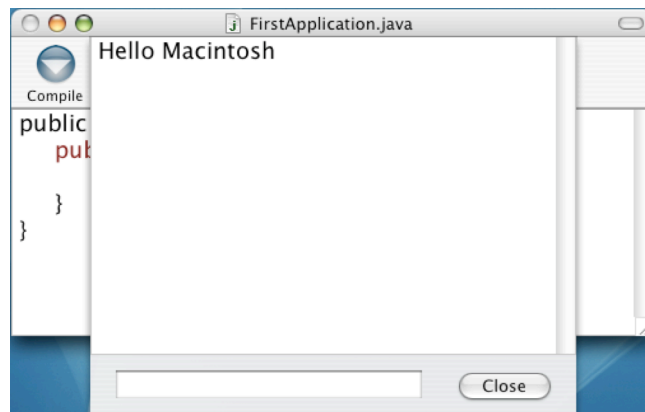


図2-12 JJEditによるアプリケーションの実行

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ください。