

第9章 アプレット

【学習内容とねらい】

本章では、Java 言語で作ったプログラムを Web ブラウザ上で動作させる方法を学習します。Java 言語には、これまで作成してきた Windows アプリケーションの他に、Web ブラウザ上で動作させる事のできる**アプレット**という形態があります。このアプレットを利用すれば、Web 上で Java プログラムを公開することもできます。

アプレットは Java 言語の普及当初は (Java 言語の機能の中で) 最も注目された機能で、当時は「アプレットによって Web ページは変わる!」と大いにもてはやされたものです。現在では、その”熱狂”はやや醒めたものの、今でも Java アプレットが幅広く用いられていることに変わりはありません。特に、Web 上の学習教材の開発などには、今でも着実に Java アプレットが活用されています。

さて、Web ブラウザ上で動作させるプログラムと言うと難しく聞こえるかも知れませんが、特殊な用途を除いては、これまでのアプリケーションの作成とほとんど変わることはありません。端的に言えば、これまでのフレームがアプレットに変わっただけです。ですから、本章を学習すれば、”簡単にアプレットを作成することができる”ということを体験できるはずです。上に述べたとおり、自作のアプレットを自分のホームページで公開する、ということも可能になります。興味のある人はチャレンジしてみてください。

<第9章の構成>

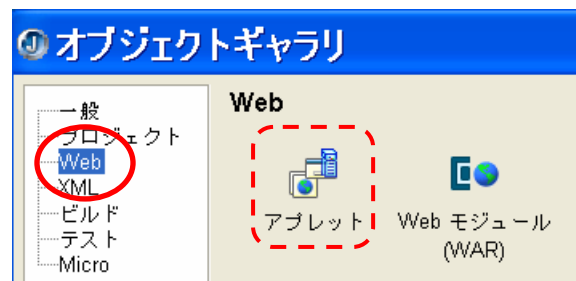
- | |
|-----------------|
| 9-1 アプレットの作成の仕方 |
| 9-2 アプレットの実行の仕方 |
| 9-3 アプレット作成の練習 |

9-1 アプレットの作成の仕方

【練習課題】

JBuilder を用いてアプレットを作成する場合、その工程はこれまで学習した通常のアプリケーションの作成とほとんど変わりません。まず、簡単なサンプル（アプレット）を作ってみましょう。

いつも通り [ファイル] → [新規] を選択します。その後「アプリケーション」ではなく、[Web]メニュー内にある「アプレット」を選択します。

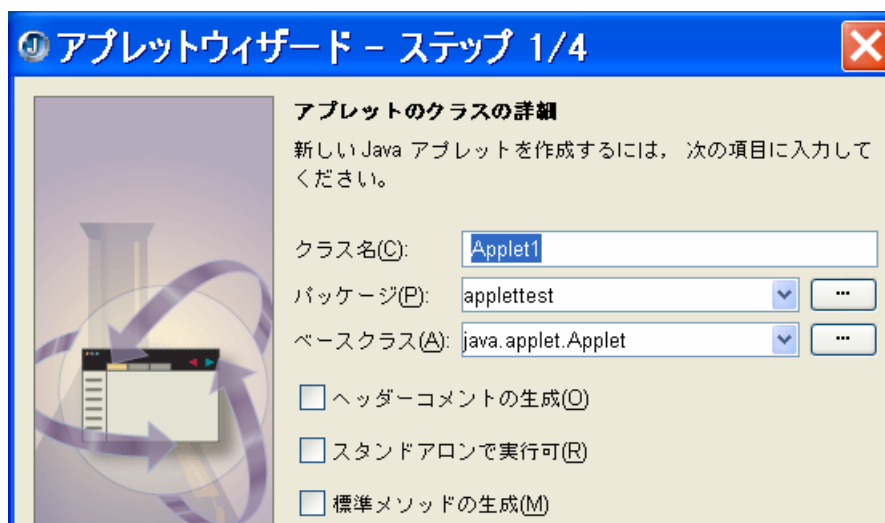


すると、これまでと同様、まずプロジェクトウィザードが現れます。ここでは、プロジェクト名を「AppletTest」にしました。なお、ディレクトリには、これまで通りプロジェクトフォルダの保管場所を指定します。

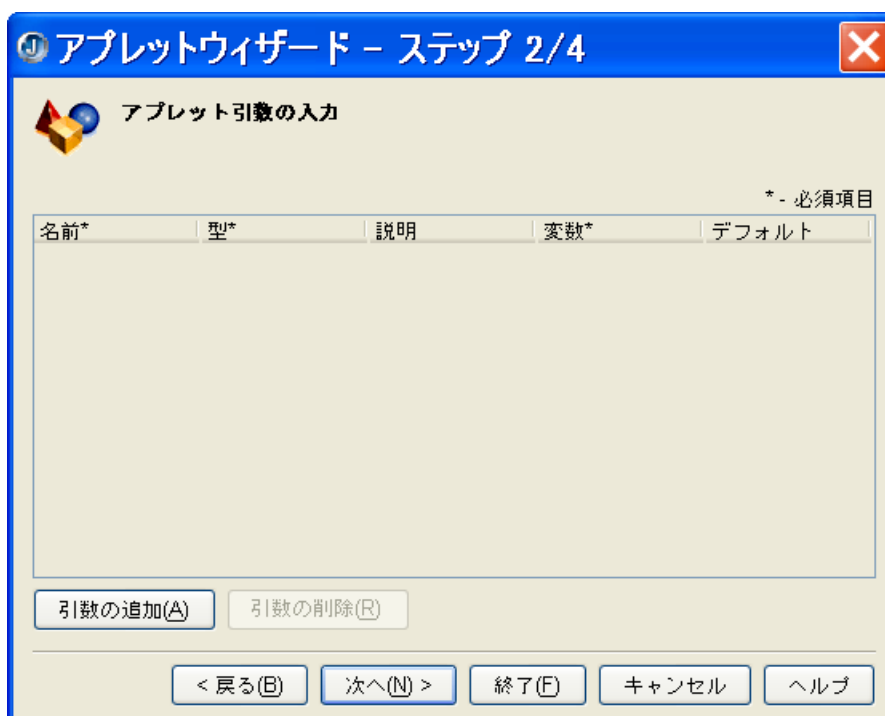


プロジェクトウィザードの 2/3 と 3/3 はデフォルト通りで結構です。

続いて「アプレットウィザード」が現れます。最初のページはデフォルトのままで結構です。ただ、ベースクラスが「java.applet.Applet」となっていることを確認して下さい。もし、ここが「java.applet.JApplet」になっていたら「java.applet.Applet」に戻して下さい。



2枚目のウィザードはそのままで結構です。



続いて3枚目はHTMLファイルの指定を行うウィザードです。ここでは、タイトルを「アプレットのテスト」に、名前は「TestApplet」のままにしました。また、高さを「200」に変更しています。

アプレットウィザード – ステップ 3/4

HTML の詳細
生成する HTML ページの詳細事項を 入力してください。

☒ HTML ページの生成(G)

タイトル(T):

名前(M):

コードベース(C):

幅(W): 高さ(H):

水平パディング(S): 垂直パディング(V):

位置合わせ(A):

< 戻る(B) 次へ(N) > 終了(F) キャンセル ヘルプ

ここに、「タイトル」は Web ブラウザのタイトルバーに表示される、HTML ファイルのタイトルです。「名前」は Web ブラウザのステータスバーに表示されるアプレットの名前です。幅と高さは、アプレットの幅と高さをピクセル単位で指定したものです。このようにアプレットが大きさを持つことから推測されるように、通常のアプリケーションのフレームが実はアプレットに対応しているのです。

最後のウィザードはデフォルトのままで結構です。

アプレットウィザード – ステップ 4/4

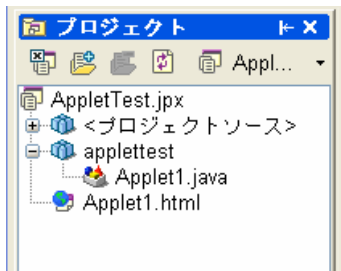
applet 設定の定義
以下のフィールドに入力することで新しい applet 設定を定義、作成できます。

☒ 実行時設定の作成(C)

名前(M):

ベース設定(A):

< 戻る(B) 次へ(N) > 終了(F) キャンセル ヘルプ



これで、アプレットのひな形が作成されました。作成されたのは、(プロジェクトペインを見れば分かるように) プロジェクトファイルの他に「Applet1.java」と「Applet1.html」の二つのファイルです。

ここで「Applet1.java」のソースを見ると、次のようになっているはずです。

```
package applettest;
```

```
import java.awt.*;
```

```
import java.awt.event.*;
```

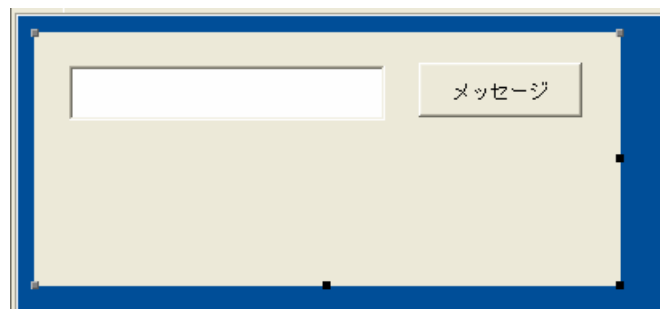
```
import java.applet.*; ①
```

```
public class Applet1 extends Applet { ②
    private boolean isStandalone = false;
    //引数値の取得
    public String getParameter(String key, String def) {
        return isStandalone ? System.getProperty(key, def) :
            (getParameter(key) != null ? getParameter(key) : def);
    }
    . . .
```

<解説>

- ① 今の場合、アプレットの作成なので、(アプレット作成に必要なクラスが用意されている) java.applet パッケージをインポートしています。
- ② JFrame の代わりに「Applet」が入っています。これは、アプレット作成の場合は、JFrame クラスではなく、Applet クラス (のオブジェクト) の上に色々なコンポーネントを貼り付けてプログラムを作成する事を意味しています。

それでは、「設計」ビューに切り替えて次のようなアプレットを作りましょう。これは、テキストフィールドとボタンを一つずつ貼り付けたアプレットで、作り方はこれまでのアプリケーションの場合と全く同様です。

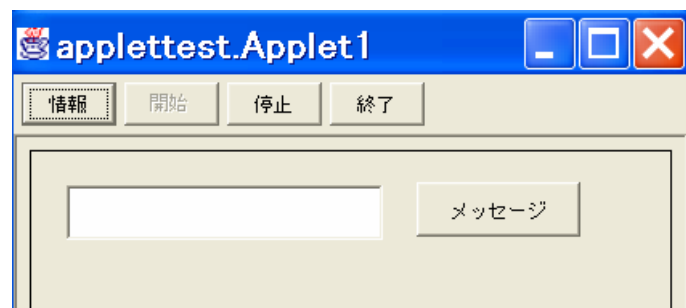


そして、「メッセージ」ボタンをクリックしたときのイベントハンドラを次のようにしましょう。

```
void jButton1_actionPerformed(ActionEvent e) {  
    jTextField1.setText("アプレット作成、成功!");  
}
```

見て分かる通り、これはボタンをクリックするとテキストフィールド欄に「アプレット作成、成功!」と表示させるプログラムです。

作成したら、アプリケーションの場合と同じように実行してみましょう。実行すると、次のような画面が現れるはずです。



ここで、「メッセージ」ボタンをクリックすると、次のように所定のメッセージが表示されます。



以上は、作成したプログラムの動作確認なのですが、これでは（通常の）アプリケーションの場合と変わらず、まだ Web ブラウザ上で動作させてはいません。そこで、次節でアプレットを Web ブラウザから呼び出す方法を確認しておきましょう。

9-2 アプレットの実行の仕方

Web ブラウザからアプレットを呼び出すには、HTML ファイルが必要です。そのための HTML ファイルが、(JBuilder によって自動生成された)「Applet1.html」です。「Applet1.html」をソース表示で見ると、次のようになっています。



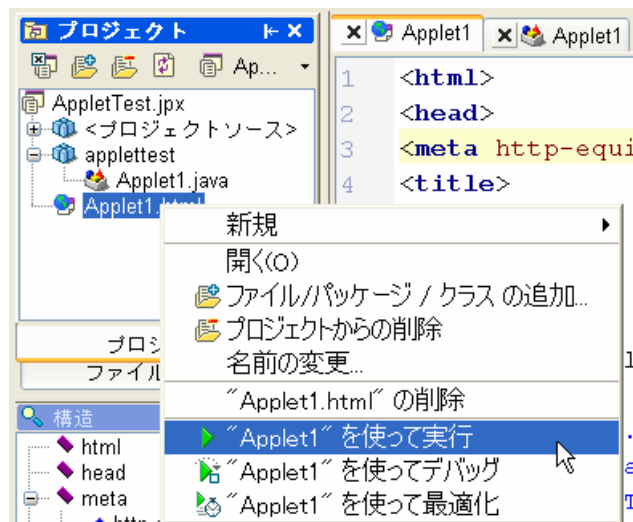
```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=Shift_JIS">
<title>
アプレットのテスト
</title>
</head>
<body>
applettest.Applet1 は Java が使用できるブラウザで下に表示されます<br>
<applet
  codebase = "."
  code     = "applettest.Applet1.class"
  name     = "TestApplet"
  width    = "400"
  height   = "200"
  hspace   = "0"
  vspace   = "0"
  align    = "middle"
>
</applet>
</body>
</html>
```

中ほどに、<applet …/applet>タグがありますが、これが、アプレットを呼び出すタグです。その中の

code = applettest.Applet1.class

という部分で、「applettest」フォルダ内にある「Applet1.class」というクラスファイルを呼び出しています。つまり、これにより、アプレットが実行される訳です。このファイル名を変えれば、該当する名称のアプレットが実行されることになります。

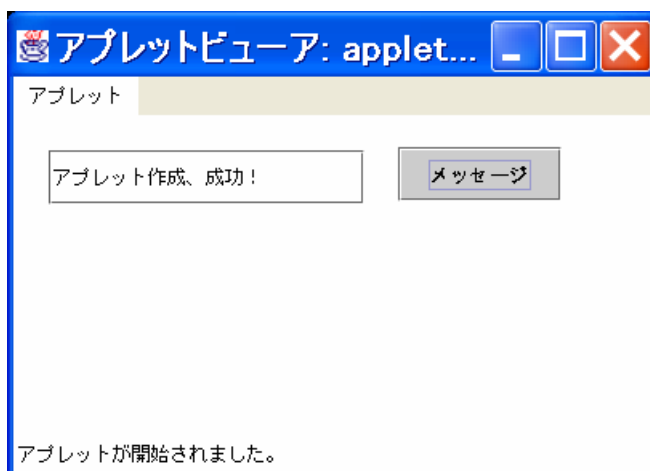
それでは、この「Applet1.html」からアプレットを呼び出してみましょう。それには、次ページのようにプロジェクトペインの「applet1.html」を右ボタンクリックして現れるプルダウンメニューから「Applet1 を使用して実行」を選択します。



すると、次のように**アプレットビューア**が起動し、アプレットを動作させることができます。



アプレットビューアが起動したところ



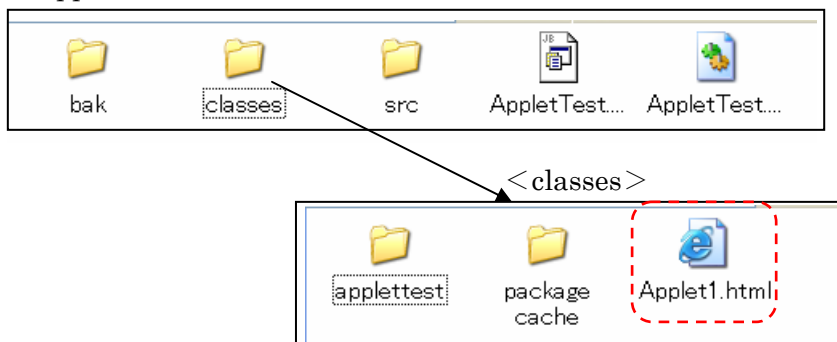
ボタンをクリックすると、メッセージが表示されます。

このようにアプレットビューアは、Web ブラウザ上での動作をシミュレートしてくれるので、画面の体裁を含めて動作確認をすることができます。しかし、やはり最終的には

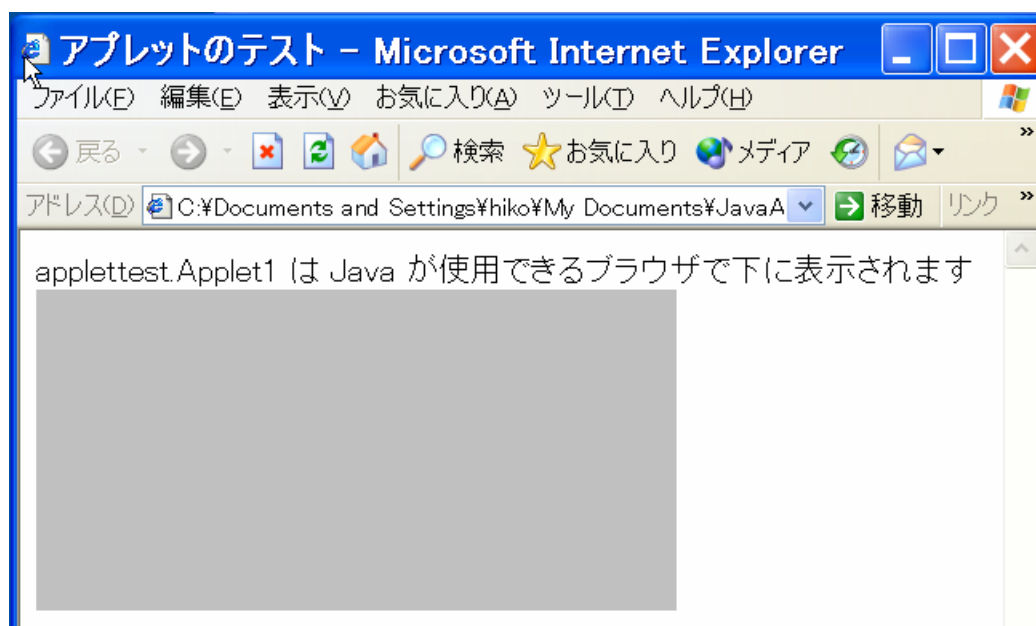
Internet Explorer(IE)などの一般のブラウザで動作させたいものです。

そこで、ひとまず JBuilder から離れて、Applet1.html を IE で直接開いてみましょう。Applet1.html ファイルは、アプレットを作成した「AppletTest」フォルダ内の「classes」フォルダ内にあります。

<AppletTest フォルダの中身>



ところが、この HTML ファイルを IE で開いても、(一般には) 下のようにアプレットはうまく動作しません。



実は、(色々な事情により) IE を始め大半のブラウザは、JDK1.2 以降、つまり Java2 以降のバージョンをサポートしていないのです。それを解決するため、サン・マイクロシステムズ社は、**Java プラグイン**を配布しています。これは、JRE (Java Runtime Environment) に付属しているもので、この JRE をインストールすると、Web ブラウザ上で問題なく Java アプレットを動作させることができます。JRE のインストールについては次ページを参照して下さい。

JRE のインストール

サン・マイクロシステムズ社は、次のページで、Java プラグインが付属している JRE を提供しています。

<http://java.sun.com/j2se/1.4.2/ja/download.html>

このページから、JRE の最新バージョン (v.1.4.2_07 あるいはそれ以降) をダウンロードできます。画面に指示に従えばダウンロードできますが、途中で英語のページになるので一応以下に要領を示しておきます。

上のアドレスに接続すると以下のページが現れるので、ここで、「J2SE JRE のダウンロード」をクリックしてください。

**netBeans**
NetBeans は無料で強力なオープンソースの Java IDE ツールです。[詳細はこちら](#)
[NetBeans のダウンロード](#)

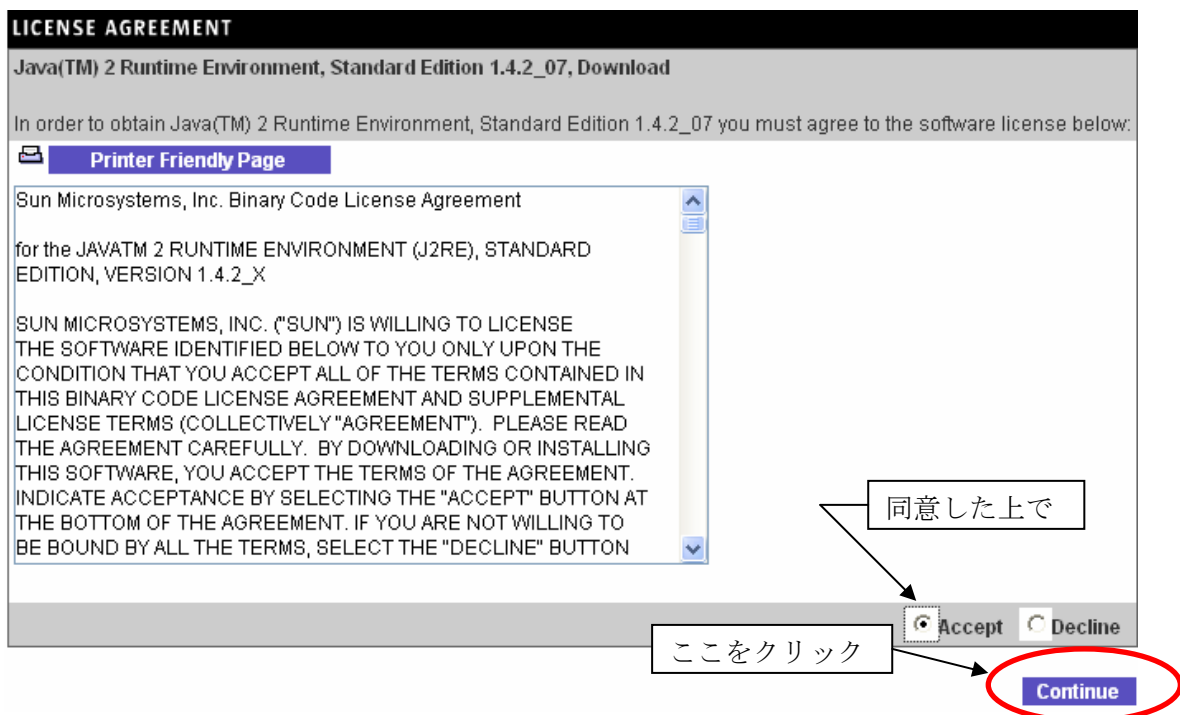

Java 2 Enterprise Edition 1.4 SDK は EJB、JSP、XML、Web サービス API をひとつのバンドルでサポートします。[詳細はこちら](#)
[J2EE 1.4 SDK のダウンロード](#)

J2SE v 1.4.2_07 SDK JVM テクノロジーを含む
J2SE Software Development Kit (SDK) は J2SE アプリケーションの作成をサポートします。[詳細はこちら](#)
[J2SE SDK のダウンロード](#)
[インストール手順](#) [ReadMe](#) [リリースノート](#)
[サンの使用許諾契約](#) [サン以外の使用許諾契約](#)

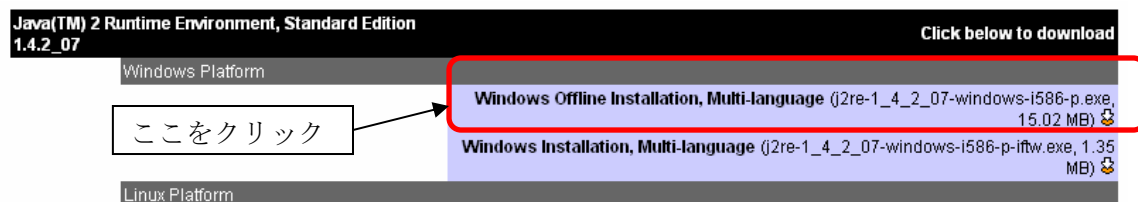
J2SE v 1.4.2_07 JRE JVM テクノロジーを含む
J2SE Java Runtime Environment (JRE) を使うと Java アプリケーションを実行できます。[詳細はこちら](#)
[J2SE JRE のダウンロード](#)
[インストール手順](#) [ReadMe](#) [リリースノート](#)
[サンの使用許諾契約](#) [サン以外の使用許諾契約](#)

ここをクリック

続いて次ページのようにライセンスの同意確認画面が現れるので、内容を確認した上で (同意できる場合は) **Accept** 欄をチェックして **[Continue]** ボタンをクリックします。



最後に以下のダウンロードファイル選択画面が出てくるので、「Windows Platform」欄の「Windows offline ...」部分をクリックして下さい。



これにより、ファイルのダウンロードを確認するポップアップ画面が現れるので [OK] ボタンをクリックするとダウンロードが開始されます。ファイルはデスクトップ上にダウンロードすると良いでしょう。

ダウンロードが終了したら、当該実行ファイルをクリックするとインストールが開始されます。

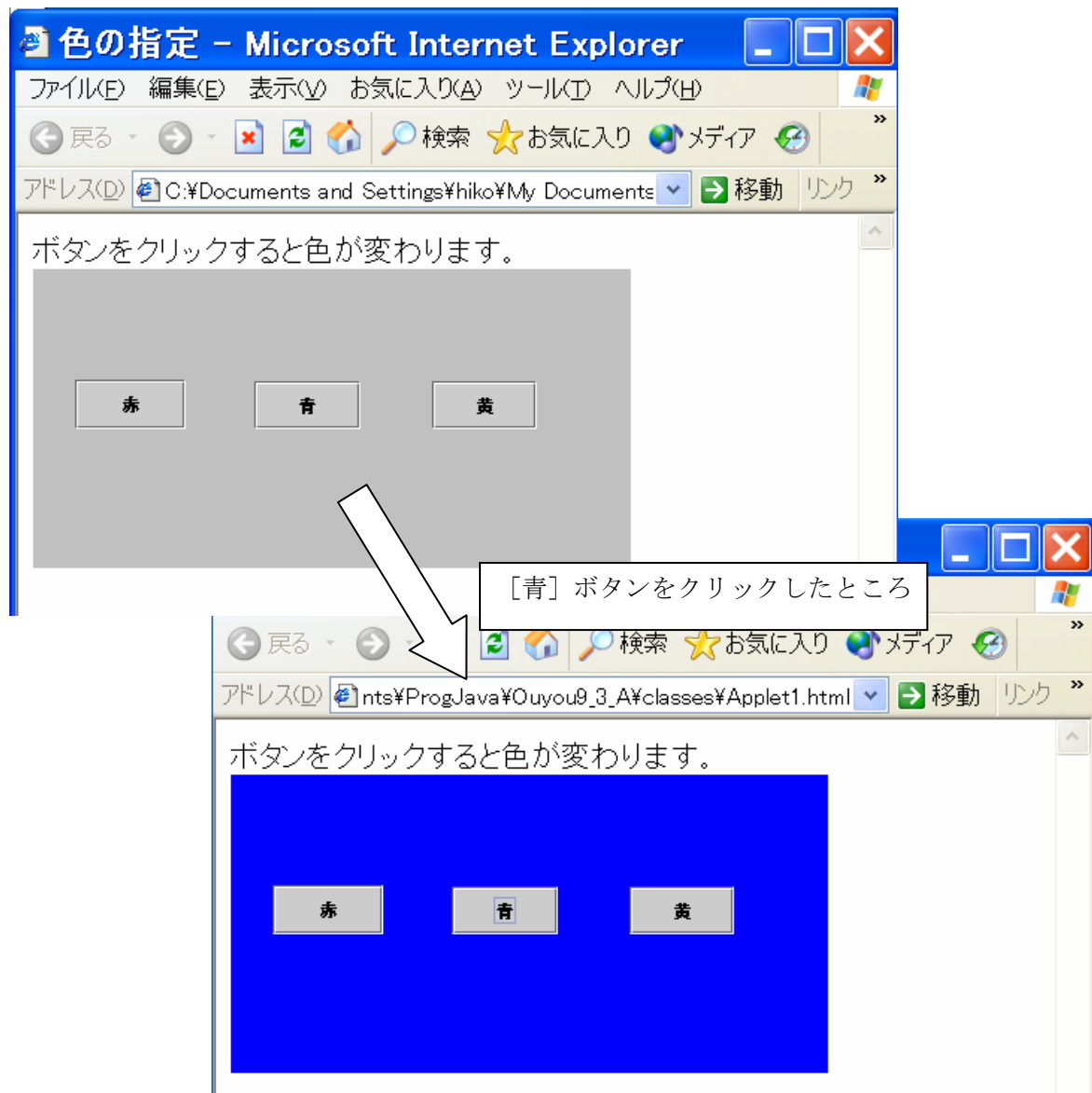
インストールが完了したら、ブラウザ上で、アプレットが動作する事を確認して下さい。

9-3 アプレット作成の練習

前節までに学習した通り、アプレットを利用したプログラムは、フレームがアプレットに変わった事をのぞけば、その作成方法はこれまでのアプリケーションとほとんど同じです。ですから、これまで作ったプログラムをアプレットに変更することは難なくできるはずです。実際に、2題ほど課題プログラムでアプレットを作ってみましょう。

【応用課題 9-3-A】

【基礎課題 3-3-3】で作った、「ボタンをクリックすればフレームの色が変わる」プログラムをアプレットとして作り、下のようにブラウザ上で動作させて下さい。



ヒント

フレームの色を指定する場合には、次のように `contentPane` コンポーネントの `background` プロパティを変えました（下は赤色に指定した場合）。→ p.59 参照

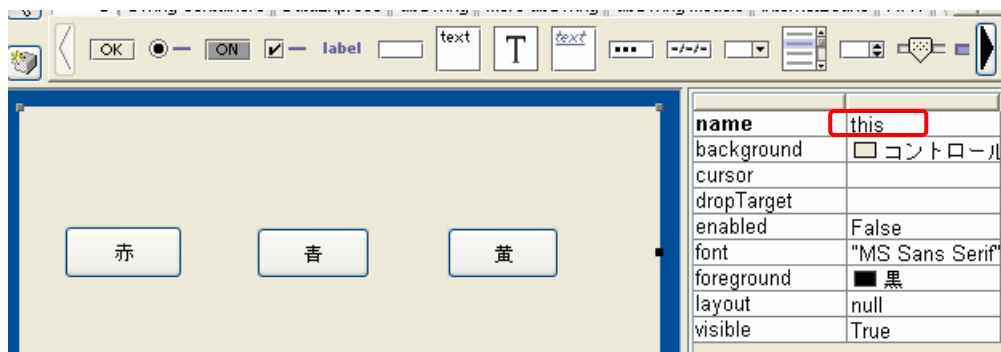
```
contentPane.setBackground(Color.red);
```

しかし、アプレットの場合はアプレットの `background` プロパティを直接指定します。しかし、（今記述しようとしている）ボタンクリックのイベントハンドラは、アプレットクラス `Applet1` の中のメソッドです。ところが、メソッドの中から、

```
Applet1.setBackground(Color.red);
```

等とクラス名自身、つまり自分自身を指定することはできません。では、どうすればよいのでしょうか？

ここで、アプレット作成画面を設計ビューに切り替え、アプレット自身が選択された状態にしてみてください。すると、下のようにその名前（name）は **this** となっていることが分かるでしょう。



このように **this** 変数は、今定義しているクラス内で自分自身を指す場合に用いられます。（より正確に言うと当該クラスのオブジェクトを指します）。ですから、例えば「赤」ボタンのイベントハンドラは

```
void jButtonRed_actionPerformed(ActionEvent e) {  
    this.setBackground(Color.red);  
}
```

と記述すれば良いのです。その他のボタンのイベントハンドラについても同様です。なお、この **this** に対して、継承元のクラス（つまりスーパークラス）を指す変数が 7-4 節（p.186～187）で学習した **super** です。対にして覚えておくと良いでしょう。

【応用課題 9-3-B】

今度は、8-3 節の【練習課題】で作成したプログラムを、下のようにアプレットとして作成して下さい。

