

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>

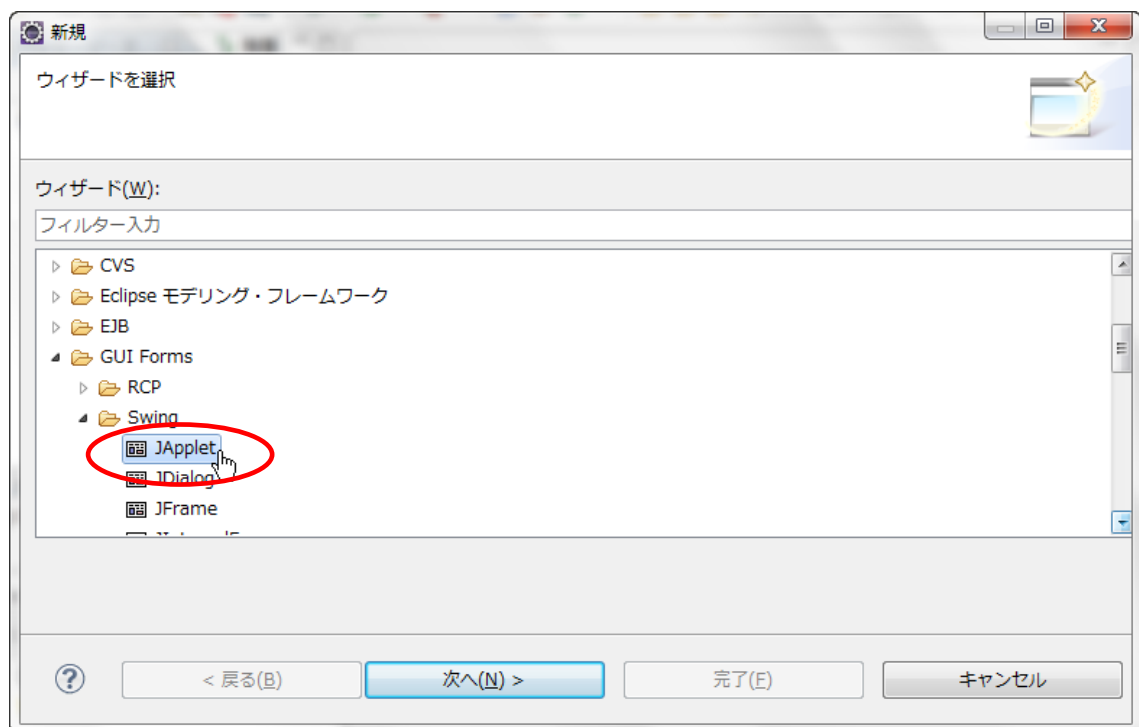
- | |
|---|
| <ul style="list-style-type: none">9-1 アプレットの作成の仕方9-2 Web ブラウザ上でのアプレットの実行の仕方9-3 アプレット作成の練習 |
|---|

9-1 アプレットの作成・実行

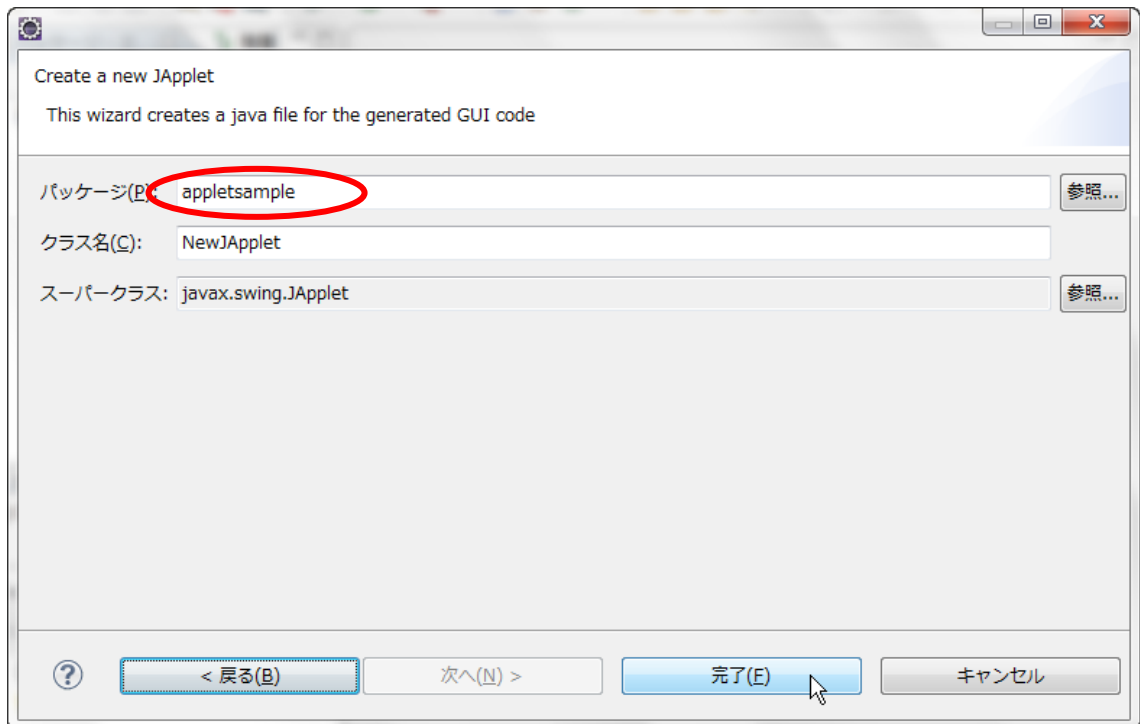
【練習課題】

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、Java プロジェクトを作成します。ここでは、プロジェクト名を「AppletSample」としました。そして、同プロジェクト内にアプレットを新規作成するために、[新規]→[その他]を選択し、下の様にテンプレート（ひな形）として「JApplet」を選択します。



[次へ] ボタンをクリックして現れる次の画面でパッケージ名を指定します。ここでは、これまで通り次の様にプロジェクト名を小文字で表し「appletsample」としました（Java プログラムでは、パッケージ名を小文字にするのが慣例です）。



パッケージ名指定後、[完了] ボタンをクリックすると設定が完了します。これで、アプレットのひな形（具体的には、NewJApplet.java というプログラム）が作成されました。

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

```
package appletsample;
```

```
import java.awt.Dimension;
```

```
import javax.swing.JApplet; ①
```

```
import javax.swing.JComponent;
```

```
import javax.swing.JFrame;
```

```
import javax.swing.SwingUtilities;
```

```
public class NewJApplet extends javax.swing.JApplet { ②  
    /**
```

```
     * Auto-generated main method to display this  
     * JApplet inside a new JFrame.
```

```
     */
```

```
    public static void main(String[] args) {
```

```
        . . .
```

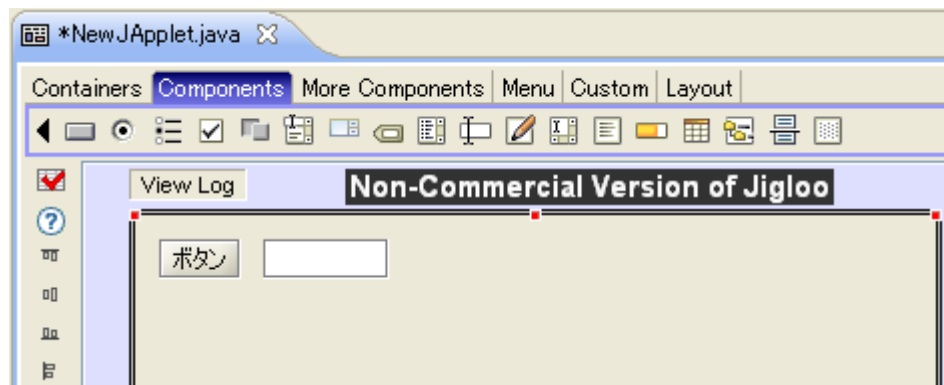
```
    }
```

```
    . . .
```

<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) `javax.swing.JApplet` パッケージをインポートしています。
- ② `JFrame` の代わりに「`JApplet`」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、`JApplet` 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。逆に言えば、これまで学習したプログラムとアプレットプログラムの違いは、フレームがアプレットに変わっただけ、という事になります。

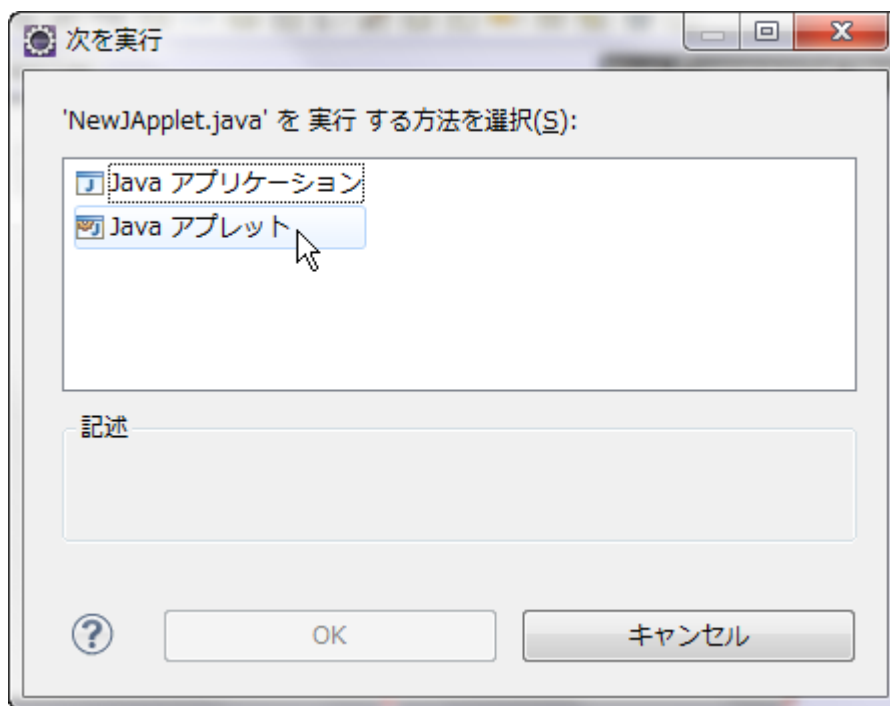
それでは、ボタンをクリックするとメッセージが表示される、という簡単なアプレットを作ってみましょう。アプレットへのコンポーネントの配置の仕方は、フレームの場合と全く同様です。次の様に [ボタン] コンポーネントとテキストフィールドコンポーネントを配置して下さい。



さらに、[ボタン] をクリックしたときに、テキストフィールドに「成功!」と表示されるようにしましょう。そこで、[ボタン] クリック時のイベントハンドラを次の様に記述します。

```
private void jButton1ActionPerformed(ActionEvent evt) {  
    jTextField1.setText("成功!");  
}
```

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、「Java アプレット」を選択し、[OK] ボタンをクリックすると、下の様なアプレットビューアが起動し、実行結果を確認できます。

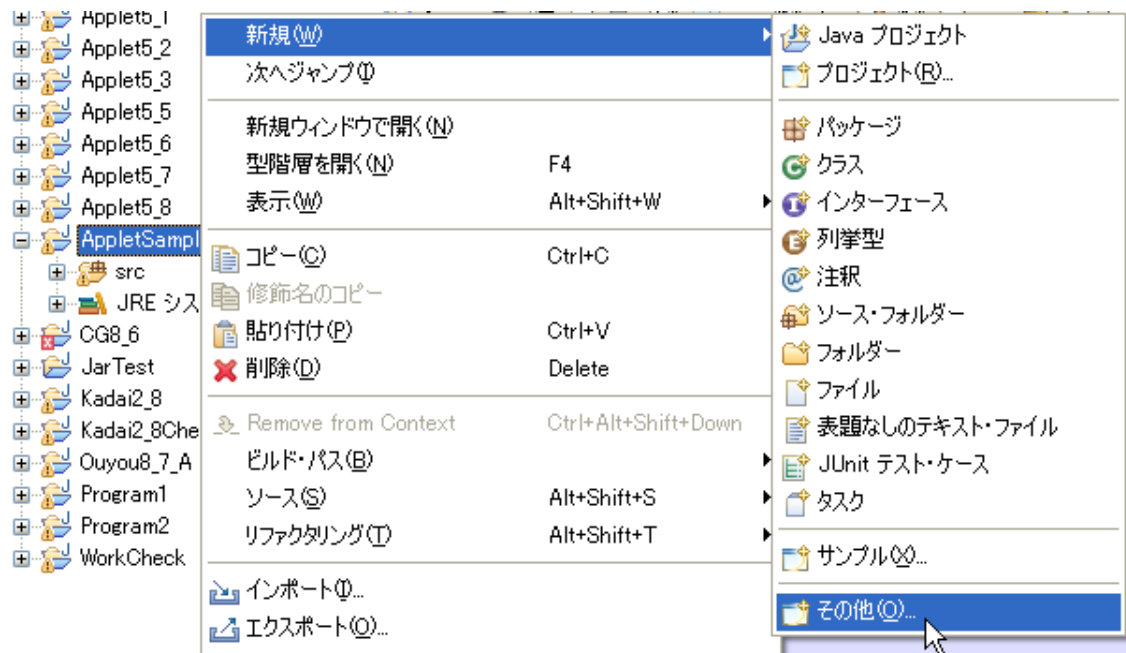


以上は、作成したプログラムの動作確認なのですが、これでは（通常の）アプリケーションの場合と変わらず、まだ Web ブラウザ上で動作させてはいません。そこで、次節でアプレットを Web ブラウザから呼び出す方法を確認しておきましょう。

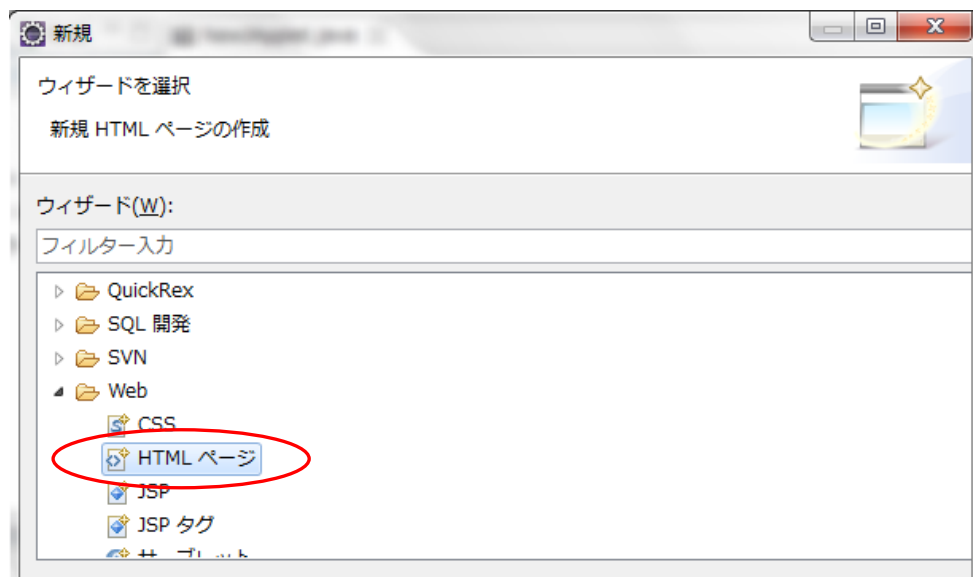
9-2 Web ブラウザ上でのアプレットの実行の仕方

アプレットは Web ブラウザ上で動作させる事ができます。というよりも、それが本来の使い方です。そのためには、アプレットを呼び出す HTML ファイルを作成する必要があります。次の要領で作成して下さい。

- ① パッケージエクスプローラから、作成したプロジェクト「AppletSample」を選択し右ボタンクリックします。そして、「新規」→「その他」と選択します。

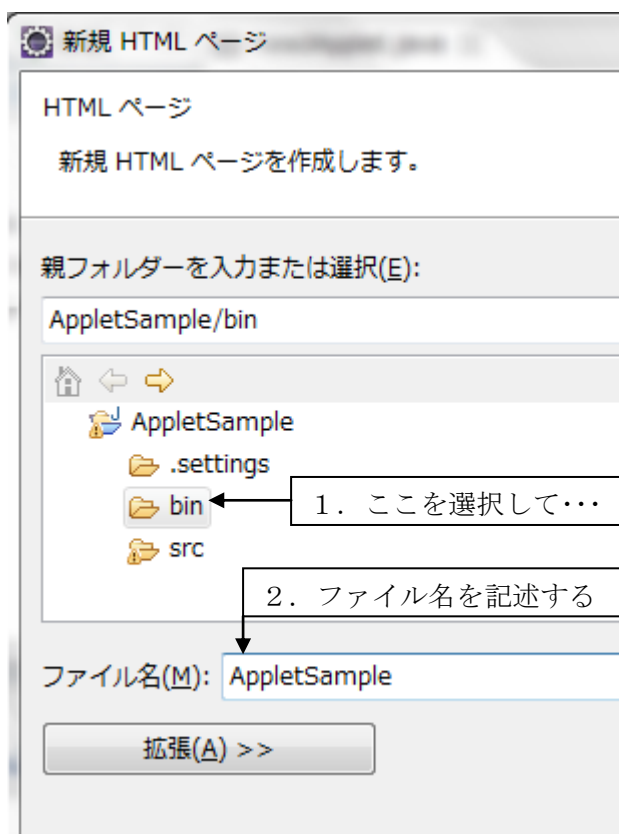


- ② そして「Web」→「HTML ページ」と選択し「次へ」ボタンをクリックします。



- ③ 次に、親フォルダ（ファイルを新規作成する場所）として、プロジェクト内の「bin」フォルダを選択し、続いて（作成する HTML）ファイル名を指定します。ここでは、プロジェクトと同名としました。

入力後「次へ」ボタンをクリックして次に進みます。その次の画面では、特に変更する必要はないので、「終了」ボタンをクリックします。これで作成準備完了です。



- ④ すると、次の HTML 文のひな形が現れます。ここでは、タイトルを変更して、アプレット呼び出しの HTML 文を記入します。



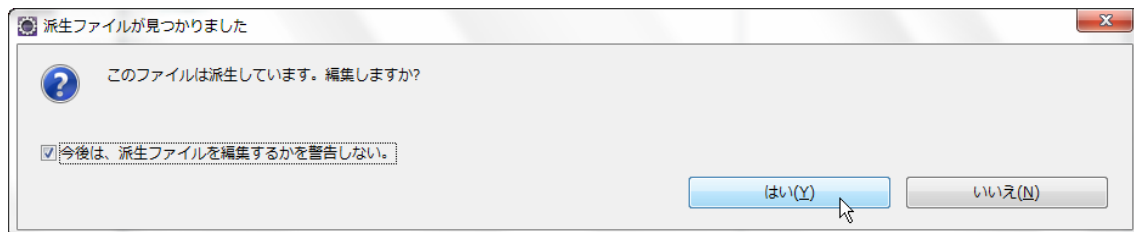
- ⑤ ここでは、次ページのようにタイトルを「アプレットサンプル」とし、アプレット呼び出しタグを点線枠内のように記述します。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=windows-31j">
<title> アプレットサンプル </title>
</head>
<body>
<applet
  codebase="."
  code="appletsample.NewJApplet.class"
  width="400"
  height="250"
>
</applet>
</body>
</html>
```

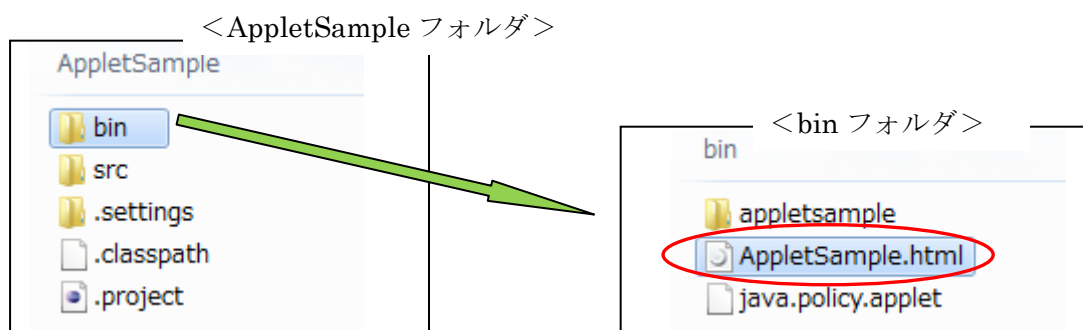
Web ページのタイトル

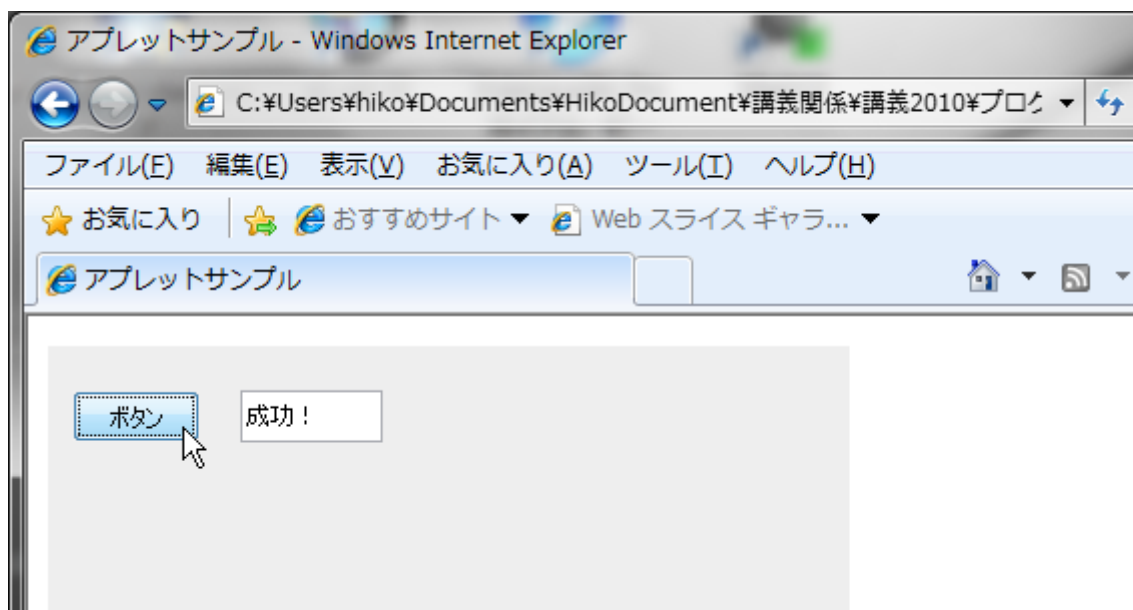
アプレット呼び出しタグ

なお、記入を始めると次のような確認画面が現れますが、ここでは、「今後は、派生ファイルを編集するか警告しない」欄にチェックを入れ、[はい] を選んで下さい。



- ⑥ 記述が完了したら、作成した html ファイル「AppletSample.html」を保管します。
その後、ワークスペース内の「AppletSample」→「bin」フォルダ内にある「AppletSample.html」をダブルクリックすると、ブラウザが起動し、次ページのようにアプレットが Web ブラウザに表示されます。





もし、上のようにボタンやテキストフィールドが表示されない場合は、サン・マイクロシステムズ社が用意している **Java プラグイン** をインストールする必要があります。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以下を参照して下さい。

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/javase/ja/6/download.html>

このページから、JRE の最新バージョンをダウンロードできます。画面の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様なページ (2010 年 9 月 17 日現在) が現れるので、ここで、次のように [JRE ダウンロード] ボタンをクリックしてください。

Java SE ダウンロードの詳細は、以下のとおりです。

Java Platform, Standard Edition	
JDK 6 Update 21 (JDK または JRE) このリリースには、Java ランタイムにおける重要なセキュリティの更新が含まれています。今すぐこの更新をご利用ください。 » 詳細はこちら どの Java が必要? Java アプリケーションおよびアプレットを実行するには、システム上に JRE (Java Runtime Environment) がコピーされている必要があります。Java アプリケーションおよびアプレットを開発するには、JRE を含む JDK (Java Development Kit) が必要です。 注意: Firefox 3.6 ブラウザは Java SE 6 Update 10 以降が必要になります。Java ベースの Web アプリケーションが動作しないためです。詳細はこちら。	JDK ダウンロード ドキュメント ▾ JRE ダウンロード ドキュメント ▾

続いて現れるページの「Platform」欄から「Windows」を選択し、ライセンス契約の同意欄にチェックをつけます。そして [continue] ボタンをクリックします。

Java SE Runtime Environment 6u21

Provide Information, then Continue to Download

There is more information on the available files for download on the [Supported System Configurations](#) page.

Select Platform and Language for your download:

Platform: ① Windows を選択する。

Language: Multi-language

☒ I agree to the [Java SE Runtime Environment 6u21 with JavaFX License Agreement](#). ② 同意欄にチェックする。

Optional: Please Log In or Register for additional functionality and [benefits](#).
Or, click "Continue" now to proceed without Log In or Registration.

User Name:
Example: jim23 or jim@company.com

Password:

» [Register Now](#)
» [Why Register?](#)
» [Forgot User Name or Password ?](#)

Continue » ③ [Continue] ボタンをクリックする。

すると、次のダウンロード画面が現れるので、ここで、「Windows Offline Installation」欄の実行ファイルをクリックします。

Instructions: Click the file name to start the download.

Available Files

File Description and Name	Size
Windows Offline Installation	15.32 MB
jre-6u21-windows-i586.exe	

インストール実行ファイルをクリックする。

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 [保存] ボタンをクリックしてダウンロードを開始して下さい。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ダブルクリックするとインストールが開始されます。



インストールが開始すると次の開始画面が現れるので、ここで [インストール] ボタンをクリックして下さい。



次の画面が現れたらインストール完了です。[閉じる] ボタンをクリックして作業を終了して下さい。そして、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。

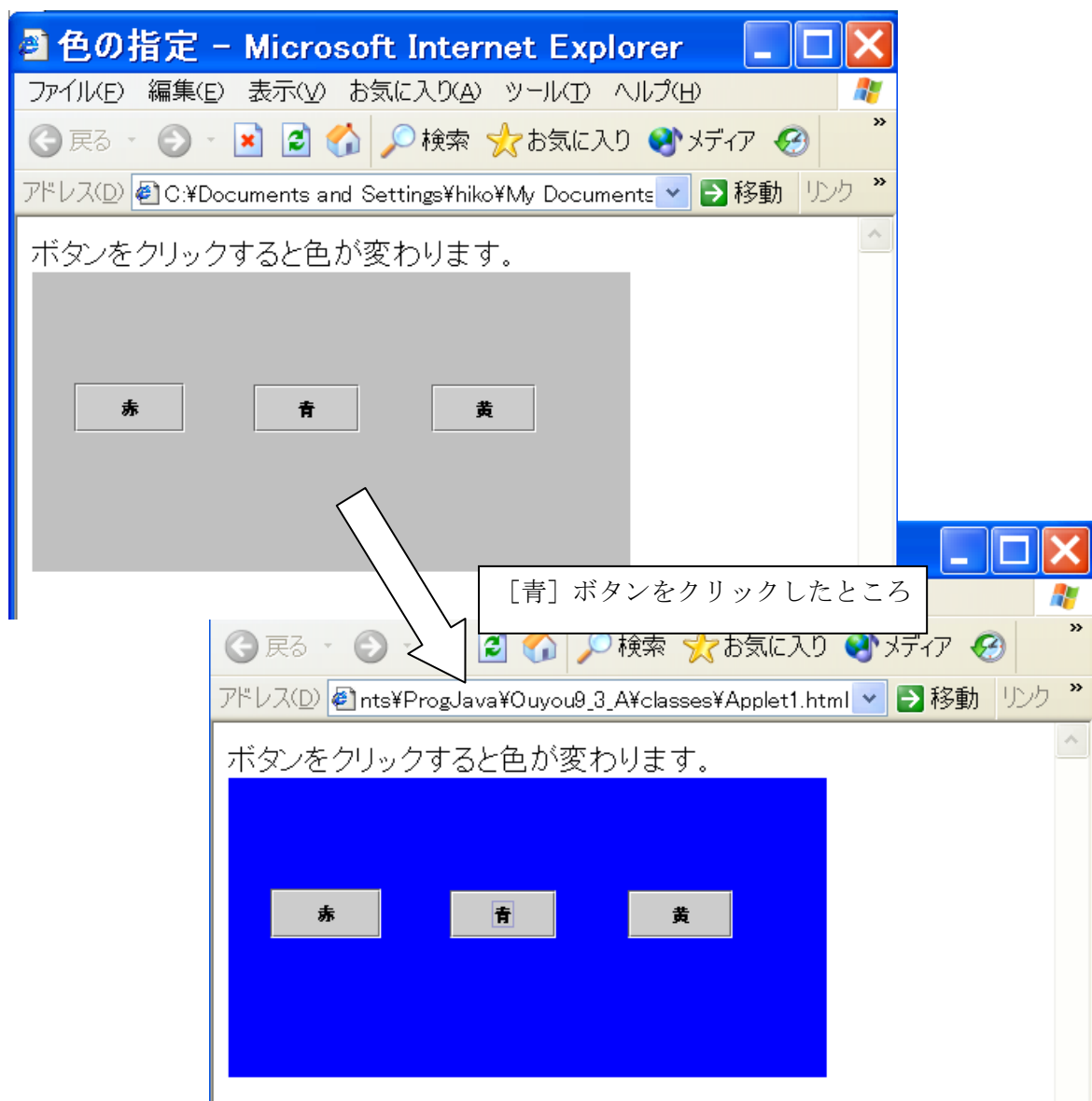


9-3 アプレット作成の練習

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

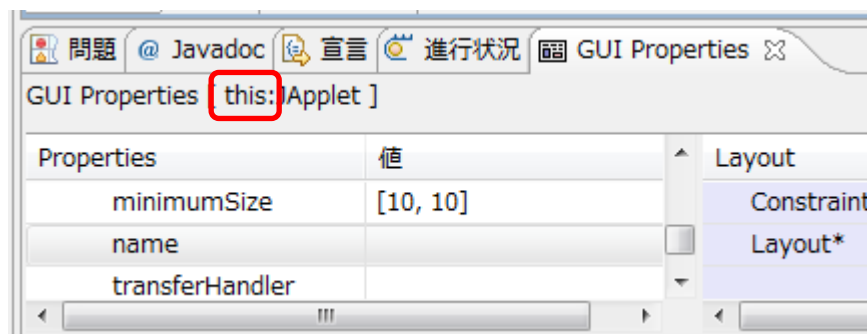
フレームの色を指定する場合には、次のように `contentPane` コンポーネントの `background` プロパティを変えました（下は赤色に指定した場合）。→ p.62 参照

```
getContentPane().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 `NewJApplet` 中のメソッドです。ところが、メソッドの中から、

```
(NewJApplet クラスのオブジェクト).setBackground(Color.red);
```

と `NewJApplet` クラスから生成されたオブジェクト名を指定することはできません。クラス定義時点ではオブジェクト名は未定だからです。では、どうすればよいのでしょうか？ここで、エディタ画面を `GUI Editor` 画面に切り替え、アプレット自身が選択された状態にしてみてください。そして画面下方の「`GUI Properties`」欄を見ると、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 jButtonRedActionPerformed(ActionEvent evt) {  
    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.192～193）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。

