

付録 A 既存のクラスの利用の仕方

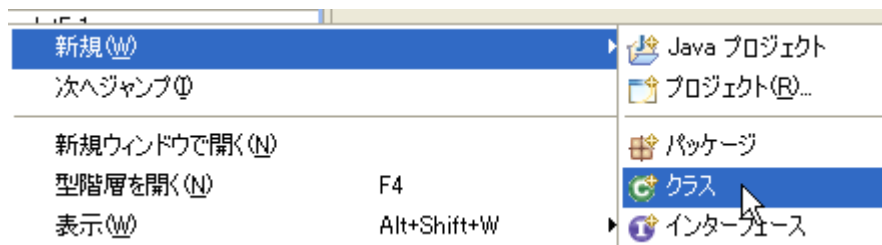
第7章では、フレームクラス（`NewJFrame.java`）とそこから呼び出されるクラス（`Meibo.java` など）を同じプロジェクト内、つまり同じパッケージ内に定義しました。しかし、一般には、別のパッケージ（フォルダ）に保管されているクラスを利用する場合があります。ここでは、その方法を説明します。

なお、**フォルダ**はJavaの用語では**パッケージ**に対応するので、以下パッケージという用語を用いることにします。それでは、以下に、具体的な例題を用いて、ある任意のパッケージにあるクラスを利用する方法を学習しましょう。

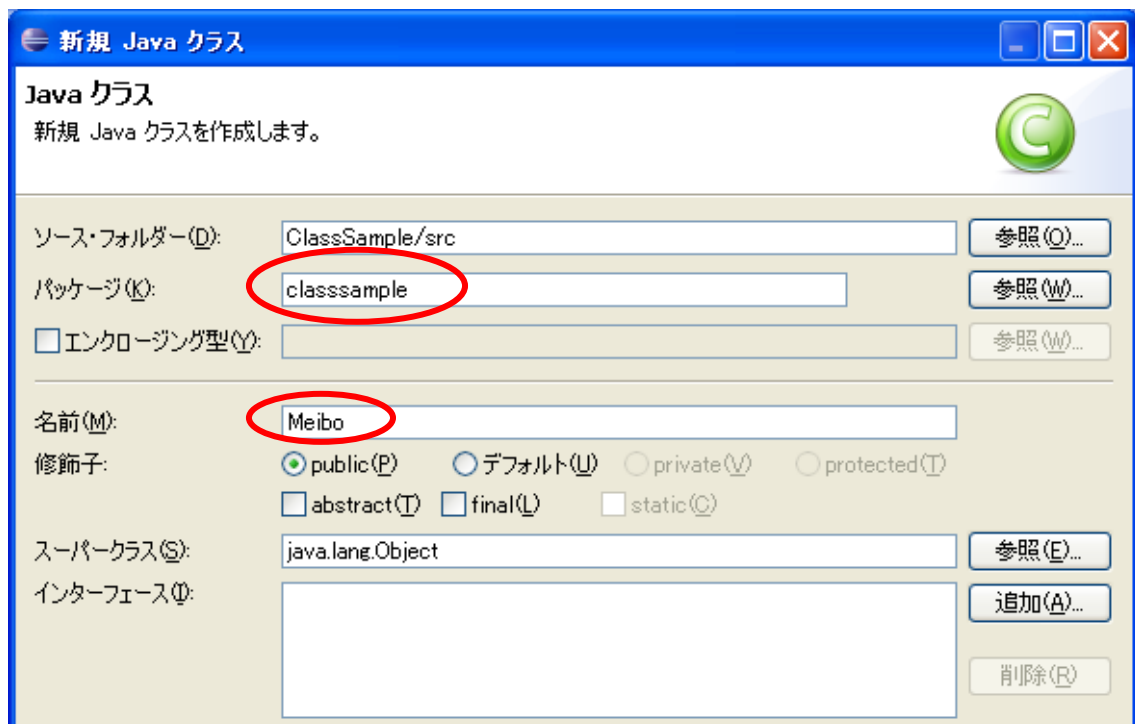
【例題 A-1 クラスの作成】

まず（利用される）クラスを作成します。作成するのは、7-2節と同じく、氏名と年齢を与えると、それに対応したメッセージを作成するメソッドを有する **Meibo** というクラスです。いつも通り、Java プロジェクトを新規作成して下さい。プロジェクト名は「**ClassSample**」とします

続いて、プロジェクト内にクラスを新規作成しましょう。7-2節でやったように、プロジェクト「**ClassSample**」を右ボタンクリックして「新規」→「クラス」と選択してください。



すると、次ページの新規クラス設定画面が現れるので、ここでパッケージ名を「**classsample**」、クラス名を「**Meibo**」に指定します。



以下では、パッケージ名を「classsample」クラス名を「Meibo」と指定したものとして説明します。

続いて現れたクラス定義の編集画面で次のようにプログラムを記述して下さい。

```
package classsample;

public class Meibo {
    private int Age;
    private String Name;

    public Meibo() {    //コンストラクタ
    }

    public void setName(String Shimei) {
        Name=Shimei;
    }

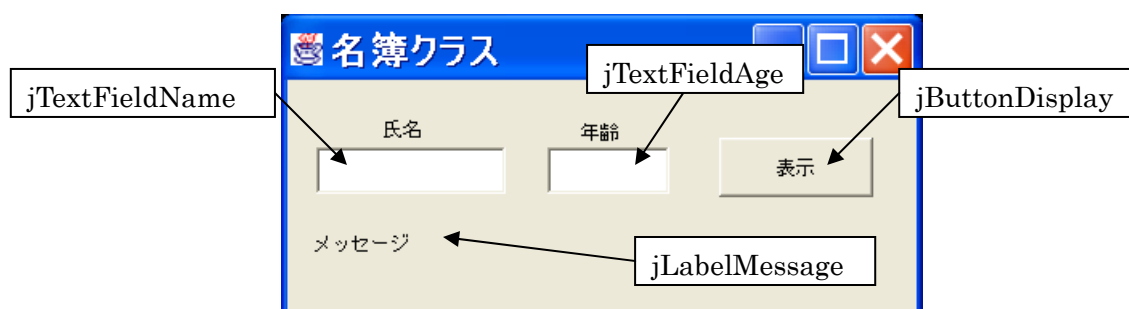
    public void setAge(int Nenrei) {
        Age=Nenrei;
    }

    public String getMessage() {
        String Message="私は"+Name+"と申します。年齢は"+Age+"歳です。";
        return Message;
    }
}
```

第7章の【基礎課題 7-2-1】と違って、ここではコンストラクタには何も定義していません。その代わり氏名と年齢を設定する、`setName()`および`setAge()`というメソッドを定義しています。どちらでも良いのですが、ここでは、氏名と年齢をオブジェクト作成時のみではなく、生成後も変更可能とするためこのようにしました。メッセージを作成する`getMessage()`メソッドは第7章と同じです。ここで、`Meibo.java`を保存してこのプロジェクトを閉じて下さい。

【例題 A-2 パッケージのインポート】

次に、新たに（上のクラスを呼び出して用いる）アプリケーションを作成します。いつも通り `JFrame` クラスを基にアプリケーションを作成してください。その際、プロジェクト名を「`ClassApp`」、パッケージ名を「`classapp`」とします。そして、フレームは以下の通り設計して下さい。



さて、この `NewJFrame.java` から先ほど定義したクラス `Meibo` を利用したいのですが、それには、`Meibo` が含まれているパッケージ「`classsample`」を取り込まなければなりません。つまり **import** する必要があります。そこで、次の波線部をプログラムの冒頭部に加えてみましょう。

```
package classapp;

import javax.swing.JButton;
import javax.swing.JLabel;
import javax.swing.JTextField;
import classsample.*;
```

もし、このパッケージ「`classsample`」が今のプロジェクト内にあれば、それが利用可能になるのですが、今は「インポートされた `classsample` は見つかりません。」というエラーが出ると思います。これは、指定したパッケージ「`classsample`」がどこに保管されているか（Eclipse が）分からないために起こるエラーです。そこで、次に、パッケージ「`classsample`」

を今作成しているプロジェクト内に取り込みます。それには以下の①～⑦の手順をとります。

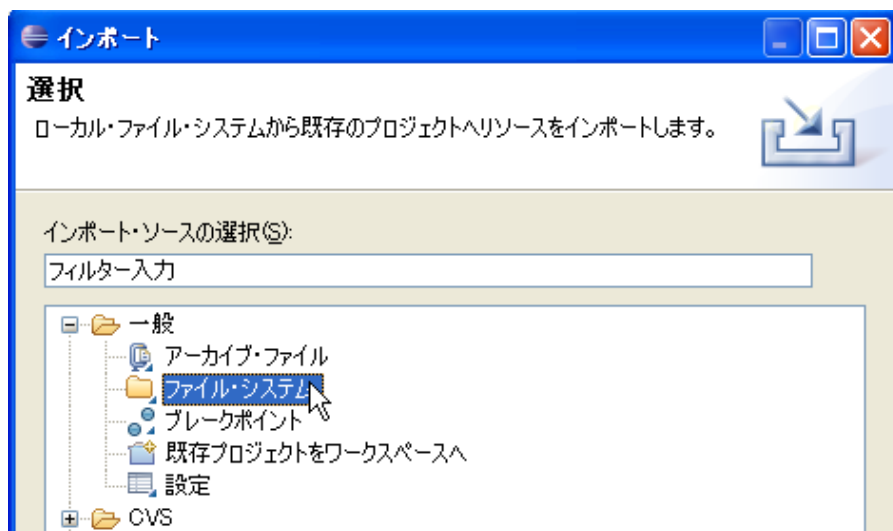
- ① パッケージ・エクスプローラからプロジェクト「ClassApp」の「src」フォルダを選択して右ボタンクリックして下さい。



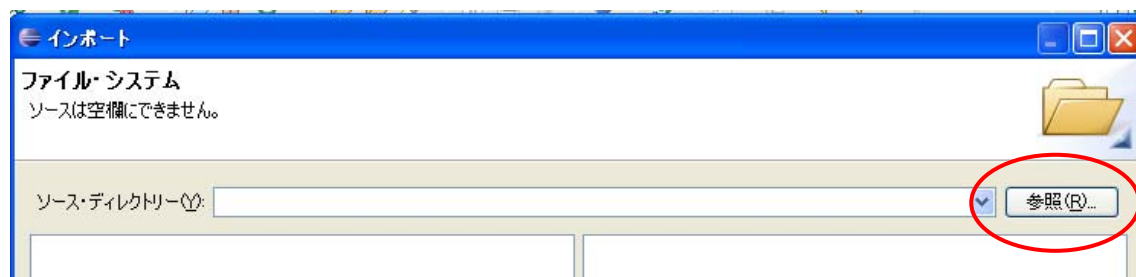
- ② 現れたメニューから、次の様に「インポート」を選択します。



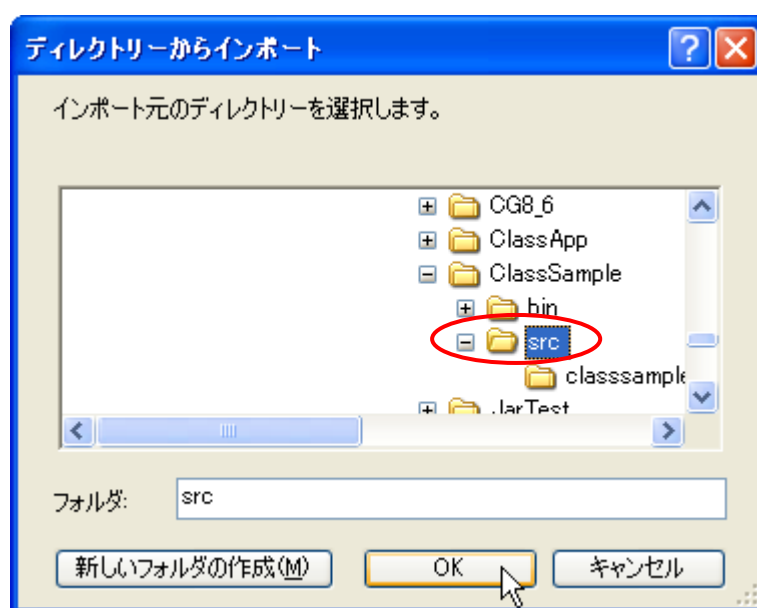
- ③ 「インポート」画面から「一般」→「ファイルシステム」を選択します。選択後「次へ」ボタンをクリックします。



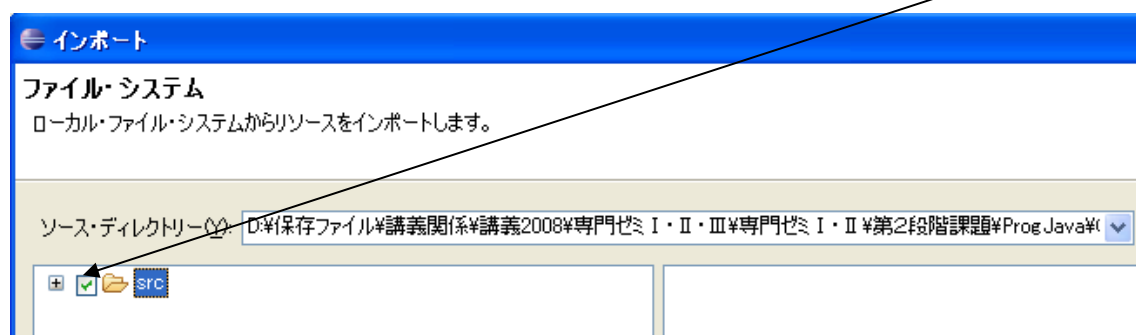
- ④ すると、次の「ファイルシステム」設定画面が現れます。



- ⑤ ここで、[参照] ボタンをクリックし、取り込むクラスが入っているフォルダを指定します。具体的には、次の様に、「ClassSample」→「src」フォルダを選択します。



- ⑥ すると、次の様に「ファイルシステム」設定画面に戻るので、ここで、「src」欄にチェックを入れて下さい。



- ⑦ その後 [終了] ボタンをクリックすると、次の様にパッケージ「classsample」が今のプロジェクト内に取り込まれます。



これで、今作成しているプロジェクト内でパッケージ「classsample」（内の全てのクラス）が利用可能になります。したがって、この時点で先ほどのエラーは消えているはずです。

それでは、ボタンをクリックしたときのイベントハンドラを作成しましょう。次のように記述して下さい。プログラムの意味は説明しなくても分かります。

<ボタンクリック時のイベントハンドラ>

```
private void jButtonDisplayActionPerformed(ActionEvent evt) {  
    String Name=jTextFieldName.getText();  
    int Age=Integer.parseInt(jTextFieldAge.getText());  
    Meibo Meibo1=new Meibo(); //名簿オブジェクトの生成  
    Meibo1.setName(Name); //氏名の設定  
    Meibo1.setAge(Age); //年齢の設定  
    jLabelMessage.setText(Meibo1.getMessage()); //メッセージの表示  
}
```

このポイントは、アプリケーションを作っているプログラマは使用するクラス Meibo の中身（実装）の詳細を知る必要がない、ということです。このように、他の人から（必要なクラスが入った）パッケージをもらって、それを自分のプログラムに利用することが容易にできます。