Java プログラムの開発手順
(eclipse を利用して)

システム情報科学演習第 2

田中文基

1. 概要

この資料は、システム情報演習第２を行う上で、使用する計算機（コンピュータ）とソフトウェアに関して記述します。この演習では、計算機室のコンピュータ、WindowsXP と Java プログラミング環境 eclipse を用いますので、その使い方に関して詳しく説明します。なお、LINUX でも同じように使えます。この章では、注意事項に関してまとめて記述しますので、よく読んでください。また、すべての演習でプログラムを使用するわけではありませんので、その点も注意してください。

演習手順は、WindowsXP (LINUX)に、**各自のユーザ ID で**ログインしたあと、以下のとおりになります。

（1）eclipse を起動する。
（2）プログラムを作成する。
（3）コンパイル、実行し動作を確かめる。
（4）提出用のデータを作成し、印刷をする。

各自のプログラムは、それぞれホームディレクトリ下にディレクトリを作成して、保存してください。デフォルトでは、ホームディレクトリ下に workspace ディレクトリが作成され，保存されることになります。

レポート提出は、**各先生の指示に従ってください**。

最後に、何かトラブルがあった場合は、TA や担当の先生に相談してください。また、システムに関するトラブルに関しては、向かいの部屋の斉藤技術職員にお願いしてください。

2. eclipse を用いた Java プログラムの作り方

ここでは eclipse の開発環境を用いて Java プログラムを作る手順を説明します。

コンピュータに作業を行わせたい場合、コンピュータにわかる形式で作業を記述する必要があります。しかし、コンピュータにわかる形式は、人間にとって入力や理解がしにくいため、人間がわかりやすい形式で記述し、コンピュータがわかる形式に翻訳する必要があります。人間がわかる形式の命令を（高級言語による）ソースプログラムといい、この翻訳作業のことをコンパイル、翻訳結果を実行プログラムといいます。以下に、ソースプログラムの作成からコンパイル、実行の作業を説明します。

2.1　Java プログラムの作成（新しく作る場合）

ここではプログラムを作って走らせるための一連の流れを説明します。基本的に流れは、
eclipse の起動→プロジェクトの作成→プログラム入力 →コンパイル・実行
となります。

2.1.1　eclipse の起動

WindowsXP では、デストクップ上のアイコンをダブルクリックする（図１）。LINUX では、GONE ウインドを開き、ウインド上で、eclipse と入力し”eclipse”を起動します(図２)。その後、図３のようにワークスペースを選択する画面になっているはずです。ここで、ホームディレクトリ＋eclipse¥workspace になっていることを確認し OK をクリックしてください。ホームディレクトリは、LINUX では図４のようになります。ホームディレクトリ上の workspace ディレクトリ以下にプログラムなどが保存されます。また、“この選択を...” にチェックを入れると、再度入力する必要はありません。

図１　デスクトップと eclipse アイコン

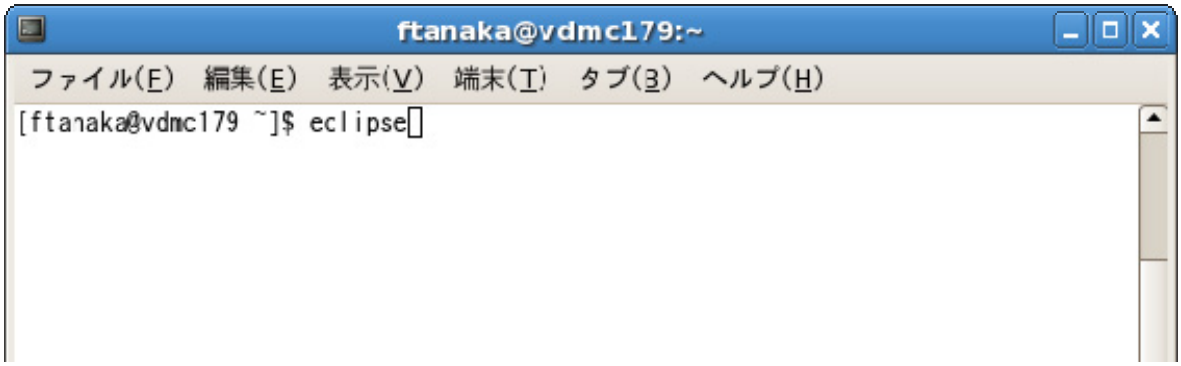

図２　eclipse コマンドを入力する

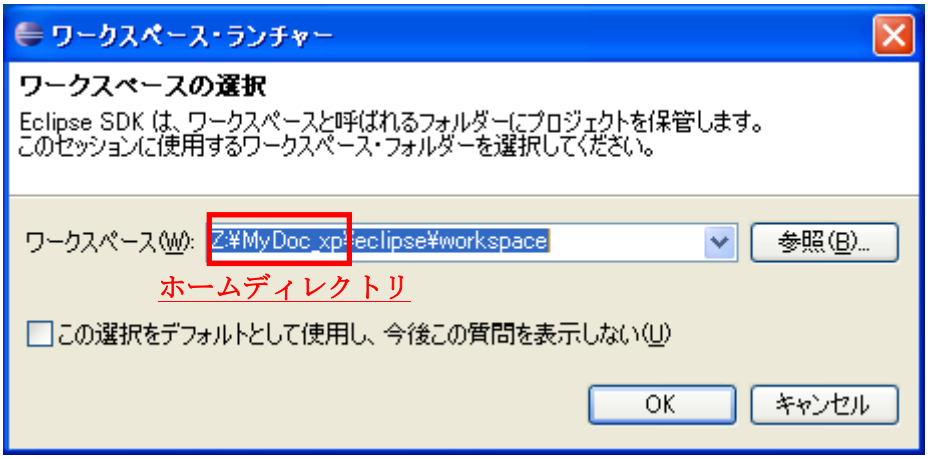

図３　メイン画面（WindowsXP）

WindowsXP を使っている人は，ここで，必ず，**ドライブが Z:**になっていることを確認してください．**C:¥program files¥eclipse やデスクトップ**には，**データを保存しない**ように注意してください．

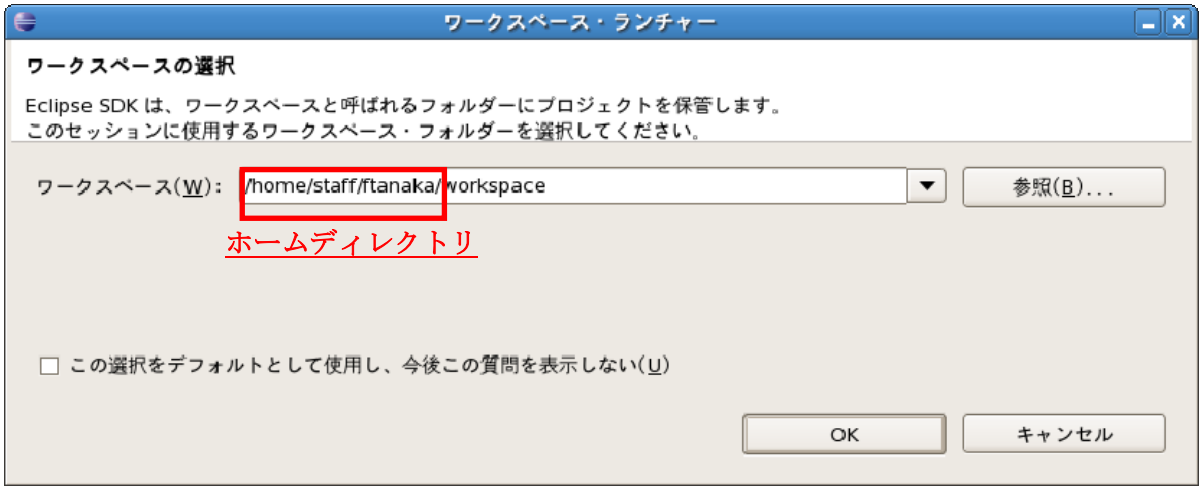

図４　メイン画面（LINUX）

そのあと、図５に示すような“ようこそ画面”になります。そこで“ワークベンチ”をクリックすることで図６に示すような“ワークベンチ画面”になります。

図５　ようこそ画面

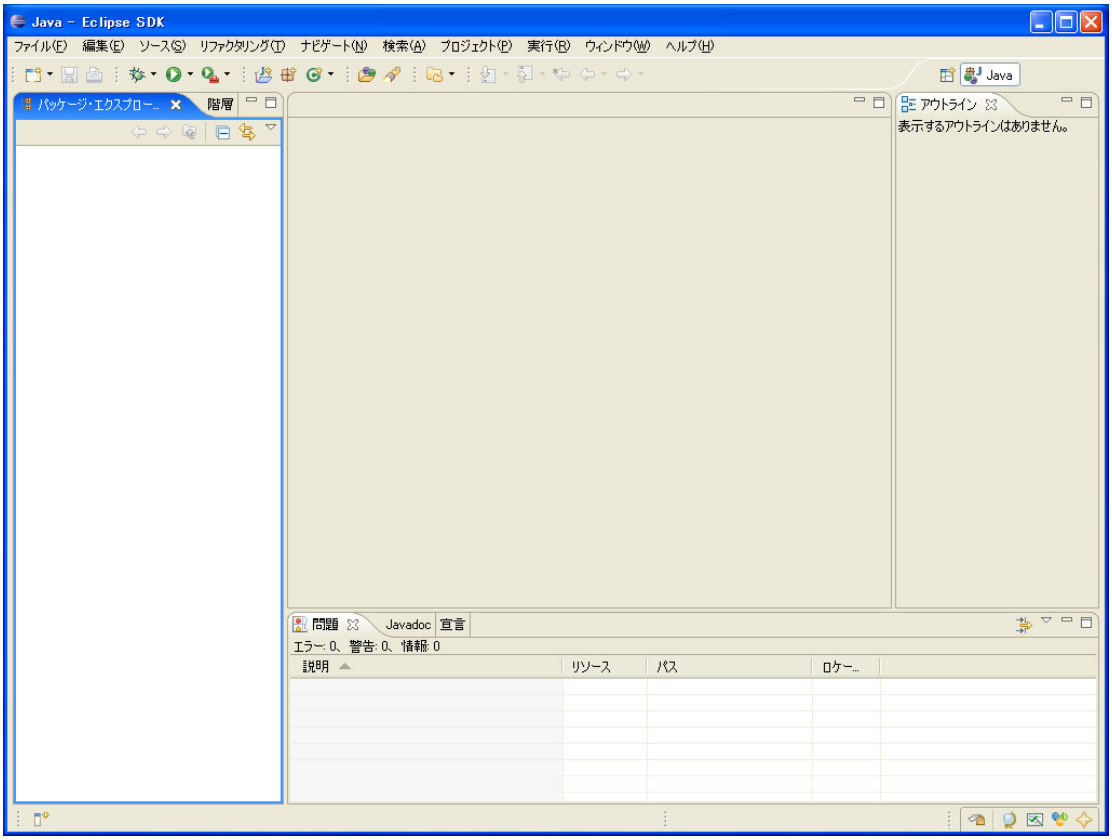

図６　ワークベンチ画面

2.1.2 プロジェクトの作成

最初に、プロジェクトを作成します。プロジェクトとは、ソースプログラム及び実行プログラムなどを管理する一連の単位のことを指します。プロジェクトには名前を付ける必要があり、その名前がついたフォルダ内に、ソースプログラムや実行プログラムが保存されます。従って、プロジェクトの名前は、わかりやすい名前にしましょう。たとえば、例題を入力する場合には、rei1_1等とすると例題との対応関係がつきやすく、わかりやすいプロジェクト名となります。

それでは、プロジェクトの作成方法について説明しましょう。

現在、図６のような画面になっているはずです。ここで、“ファイル(F)“から”新規(N)“を選びさらに”プロジェクト(R)“を選択します（図７）。次に、”ウィザードを選択“（図８）が出てくるので、ここで、“Java プロジェクト”を選び、 “次へ”をクリックします。次の“Java プロジェクトを作成します”では“プロジェクト名” を入力します。ここでは、“sample01”とし“次へ”をクリックします（図９）。次の“Java 設定”ではプロジェクト名とデフォルト出力フォルダとが一致しているか確認し“終了”をクリックします（図１０）。その後、図１１のような画面になっているはずです。

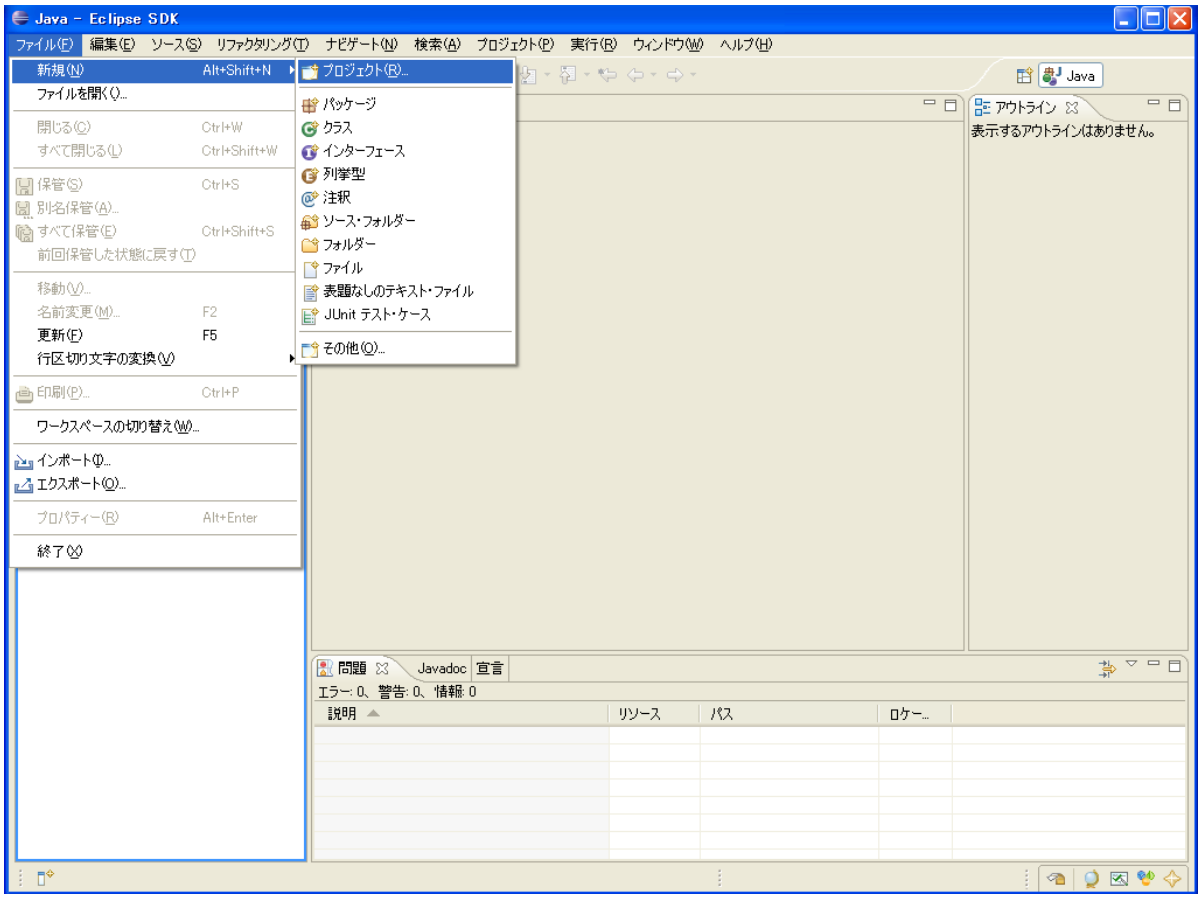

図７　新規作成選択

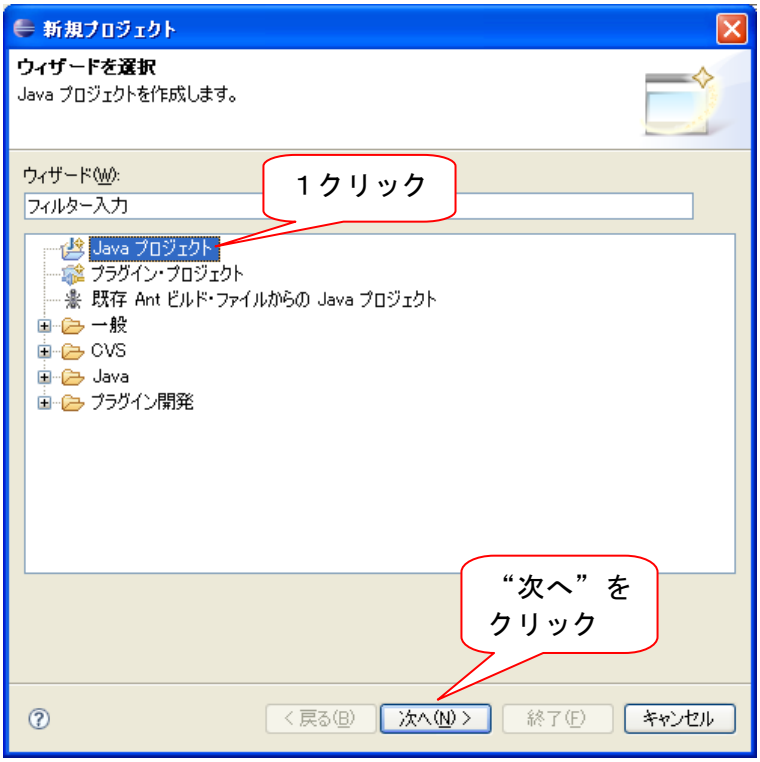

図８　新しいプロジェクトダイアログ（その１）

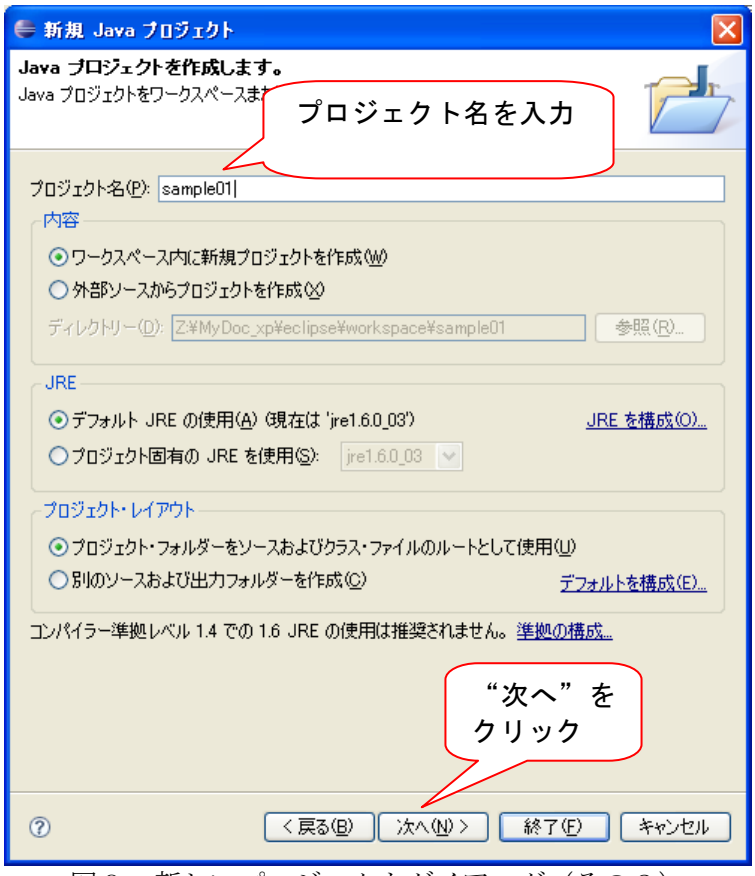

図９　新しいプロジェクトダイアログ（その２）

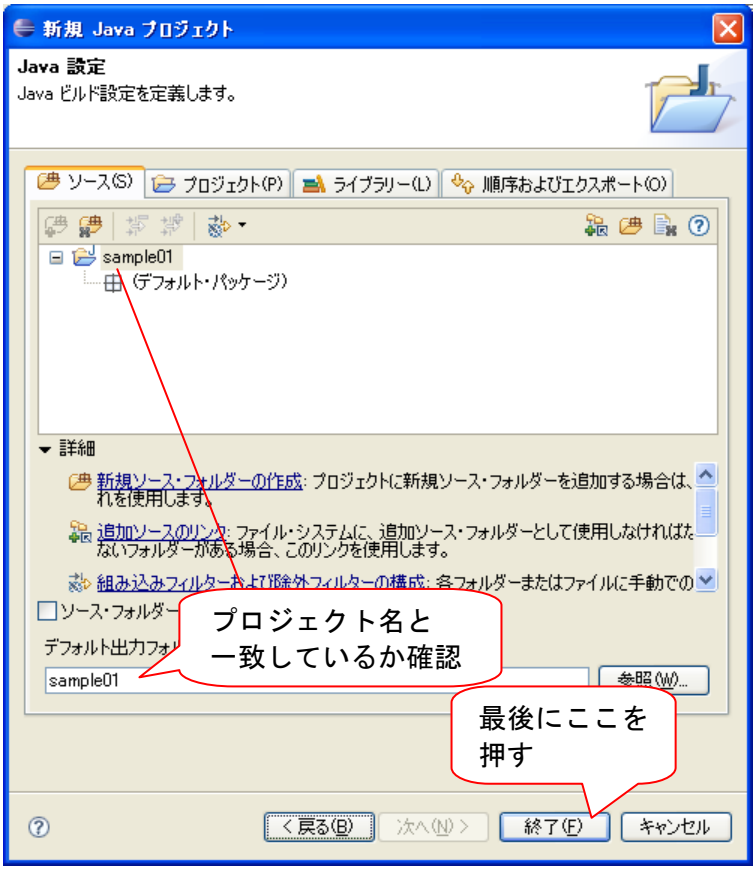

図１０　新しいプロジェクトダイアログ（その３）

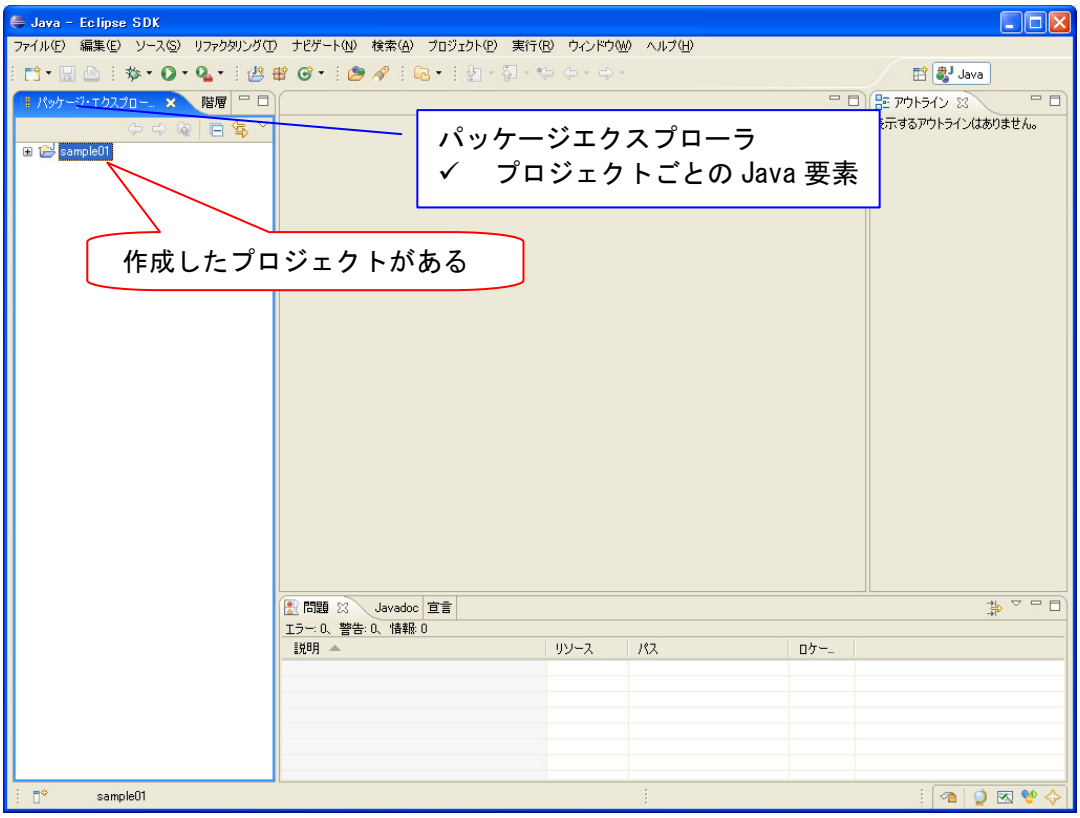

図１１　新しいプロジェクトが作成された画面

2.1.3 クラスの作成

次に、プログラムを作成します。プログラムは、プロジェクト名がついたディレクトリ内のファイルに保存されます。Java では、すべてのプログラムはクラス単位に管理されるため、ここではクラスを作成することになります。

現在、図１１のような画面になっているはずです。図６との違いは、右のパッケージエクスプローラの中にプロジェクト名が入っていることだと思います。この画面がプログラム作成のための基本画面となります。ここでプロジェクトをクリックすると、そのプロジェクトに属するファイルが一覧されます。

ここで、該当するプロジェクト（ここでは sample01）を選んで右クリックするとメニューが出てきますので、そのなかから“新規(W)“から”クラス“を選択します（図１２）。

次に、“Java クラス”が出てくるので、ここで、パッケージの欄にパッケージ名、名前の欄にクラス名を入力します。ここでは、パッケージ名として sample01、クラス名として HelloJava を入力します(図１３)。また、図１３に示すすべてのチェックをはずします。現在、これらをチェックしてはエラーが生じます。最後に“終了”を押しましょう。

これで、プログラム作成の準備ができました。画面は図１４のようになっていると思います。これで後は空白の部分にどんどんプログラムを打っていきましょう。このエディタの使い方は、普通のワープロの使い方と、ほぼ同じです。しかも、プログラム作成が容易になるように、Java 言語の予約語は赤紫色文字になります。ちょっとした間違いは、赤い波線がでるのですぐわかります。また、関数を入力した場合、引数が出てきたりもします。わからないことは、右上メニューのヘルプを参照してください。また、プログラム本体は、半角英数字で書いてください。全角でプログラムを書くと、コンパイルエラーを起こしてしまいます。

図１５にサンプルプログラムを示しますので入力してみてください。

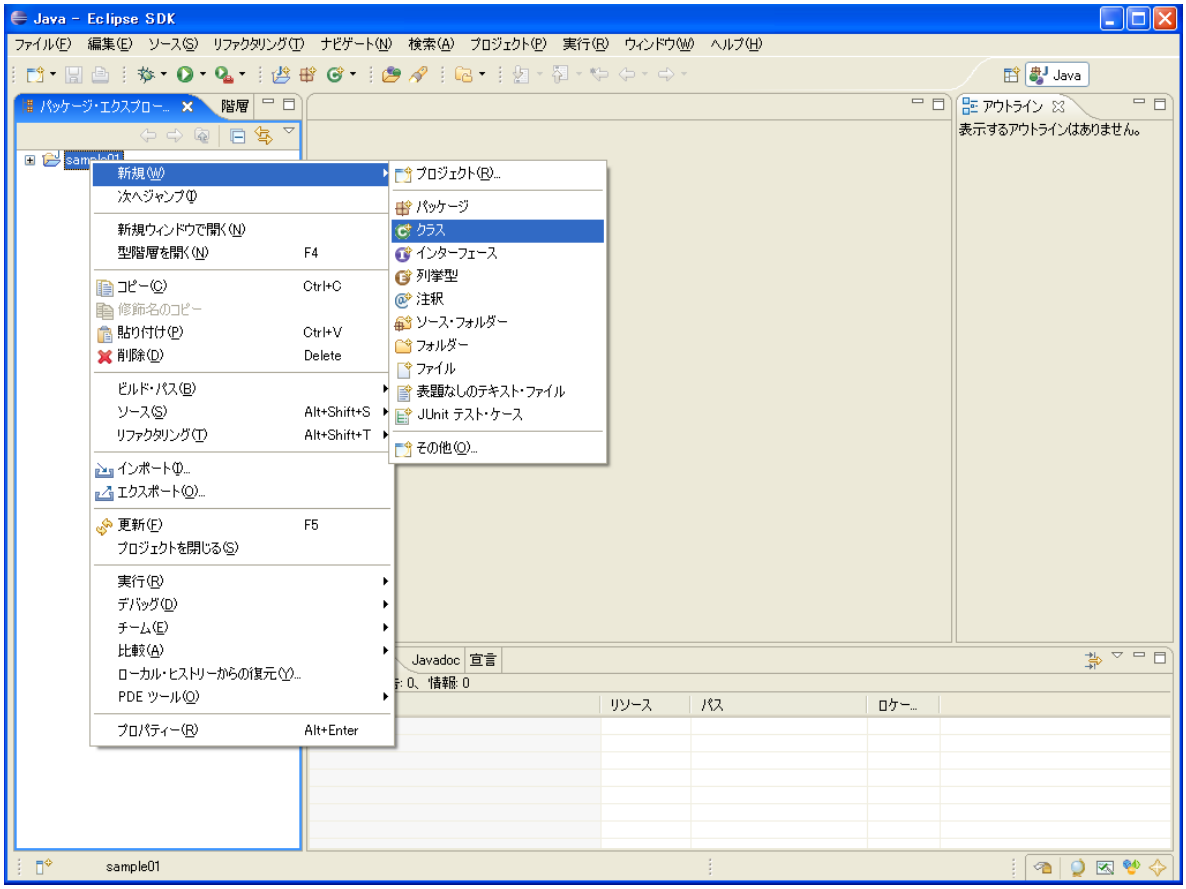

図１２　クラスの新規作成（その１）

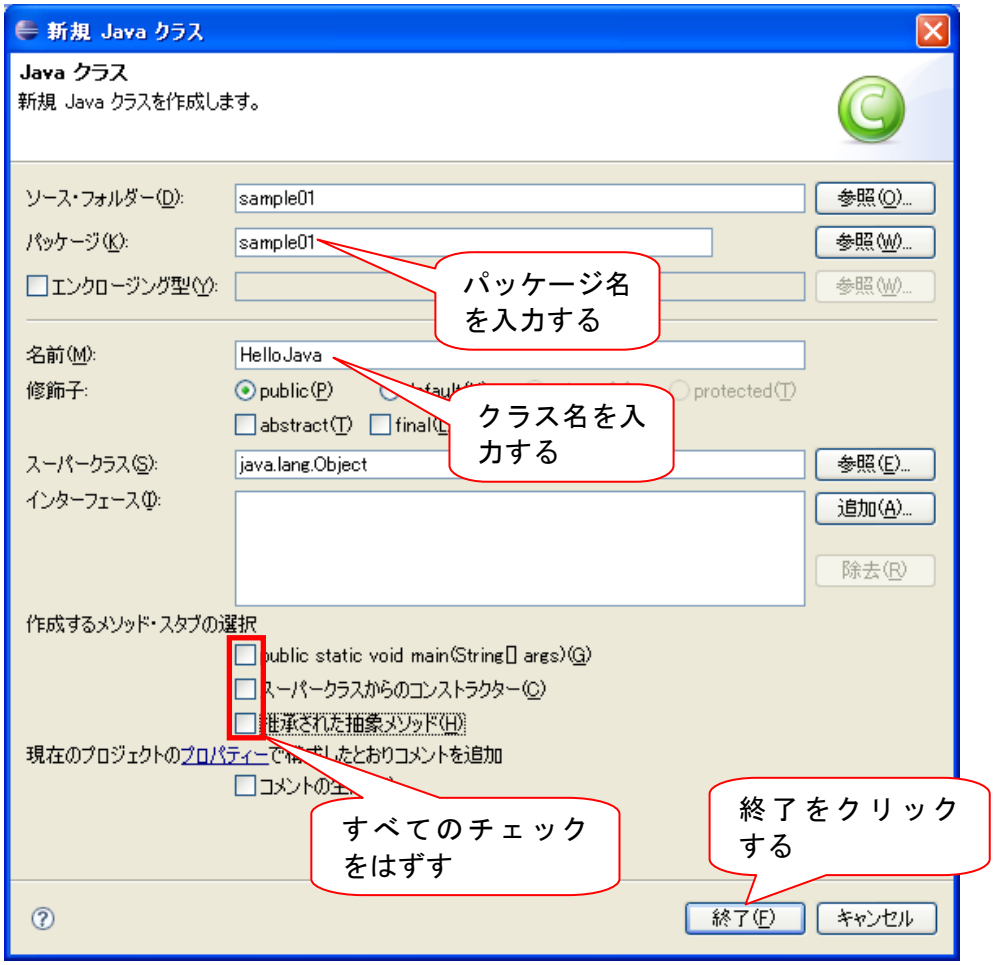

図１３　クラスの新規作成（その２）

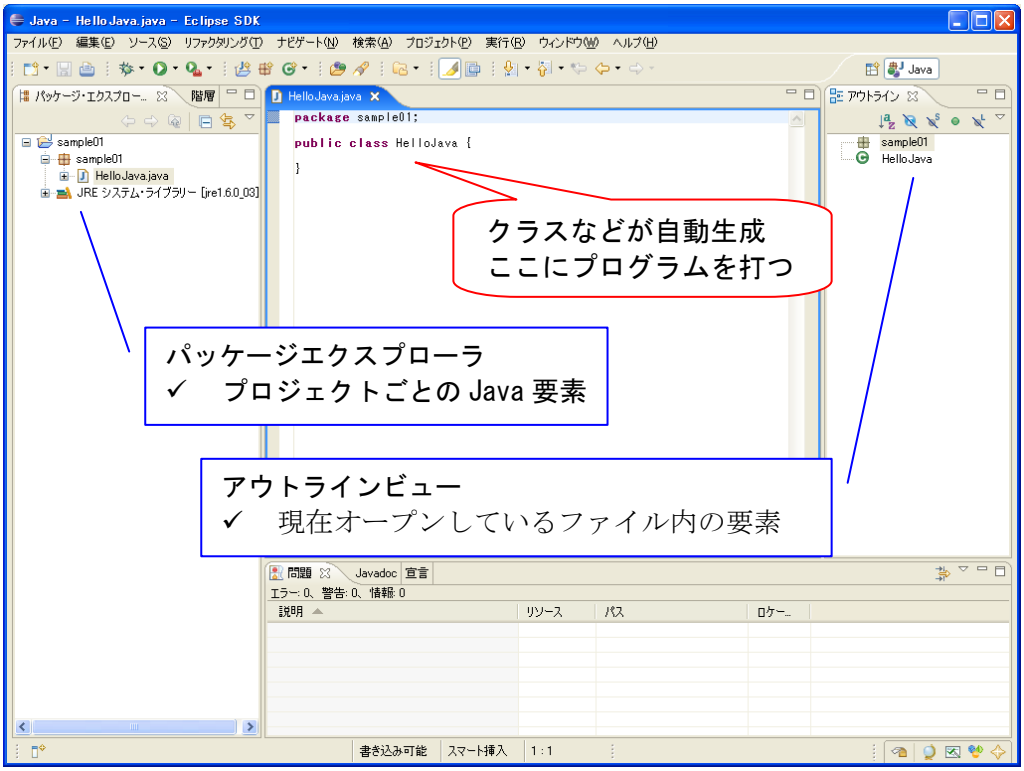

図１４　プログラム新規作成（その３）

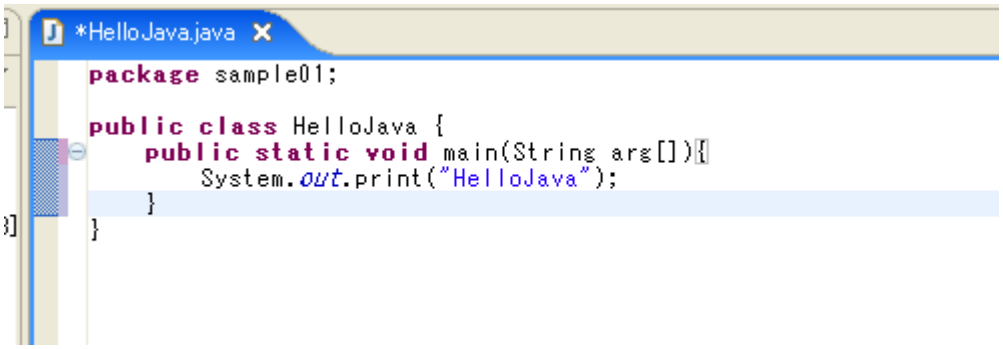

図１５　サンプルプログラム

2.1.4　コンパイルとプログラムの実行

プログラムを打ち終わったならば、今度はそれを実行してみましょう。

パッケージエクスプローラのソースファイルをクリックしたあと，メニューバーの中の“実行(R)”から“実行(S)”から“Java アプリケーション”を選択しましょう（図１６）。プログラムが保存済みの場合は、そのまま実行されますが、保存がまだの場合は、図１７の画面が出て、“OK”を押すと、保存されコンパイルされることとなります。もしもエラーがなくなったならば、下のコンソールに、プログラムの実行結果が出力されます（図１８）。これで一連の作業は終了です。

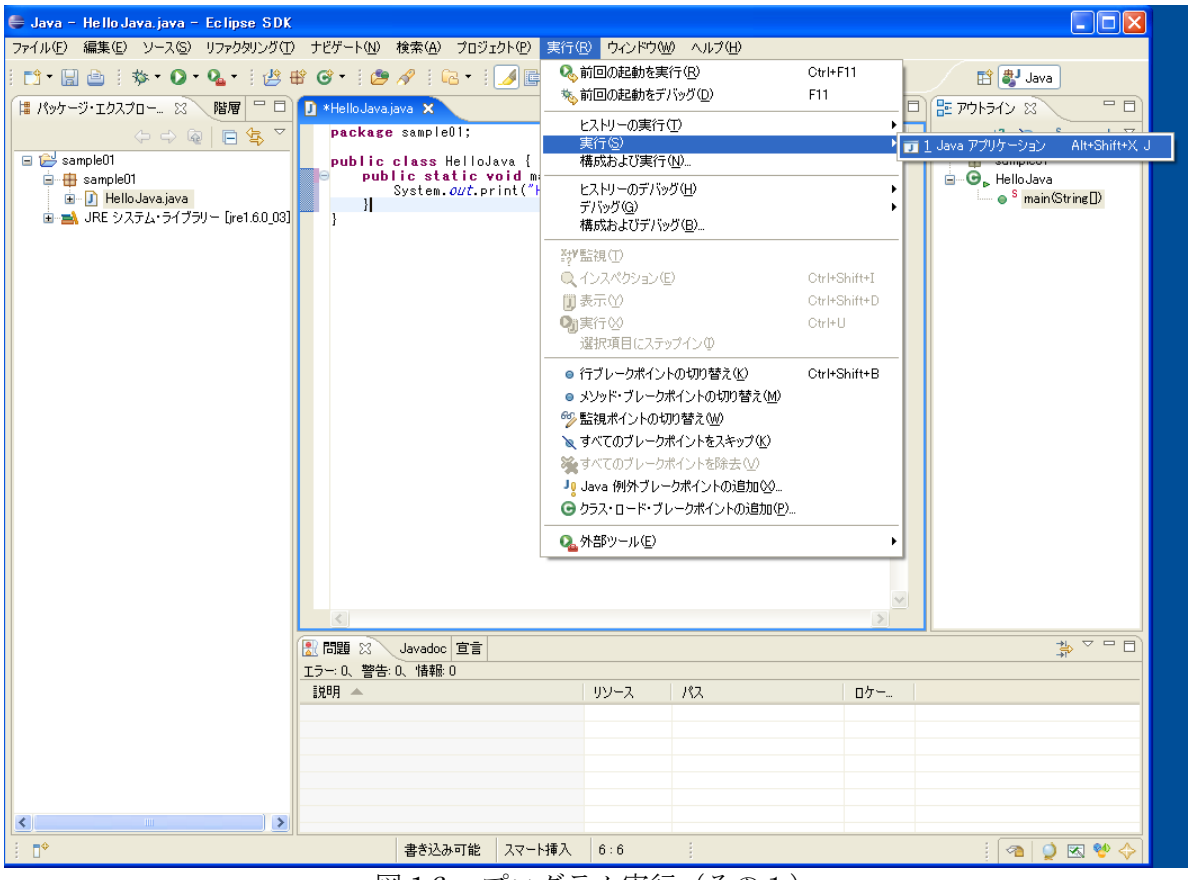

図１６　プログラム実行（その１）

図１７　プログラム実行（その２）

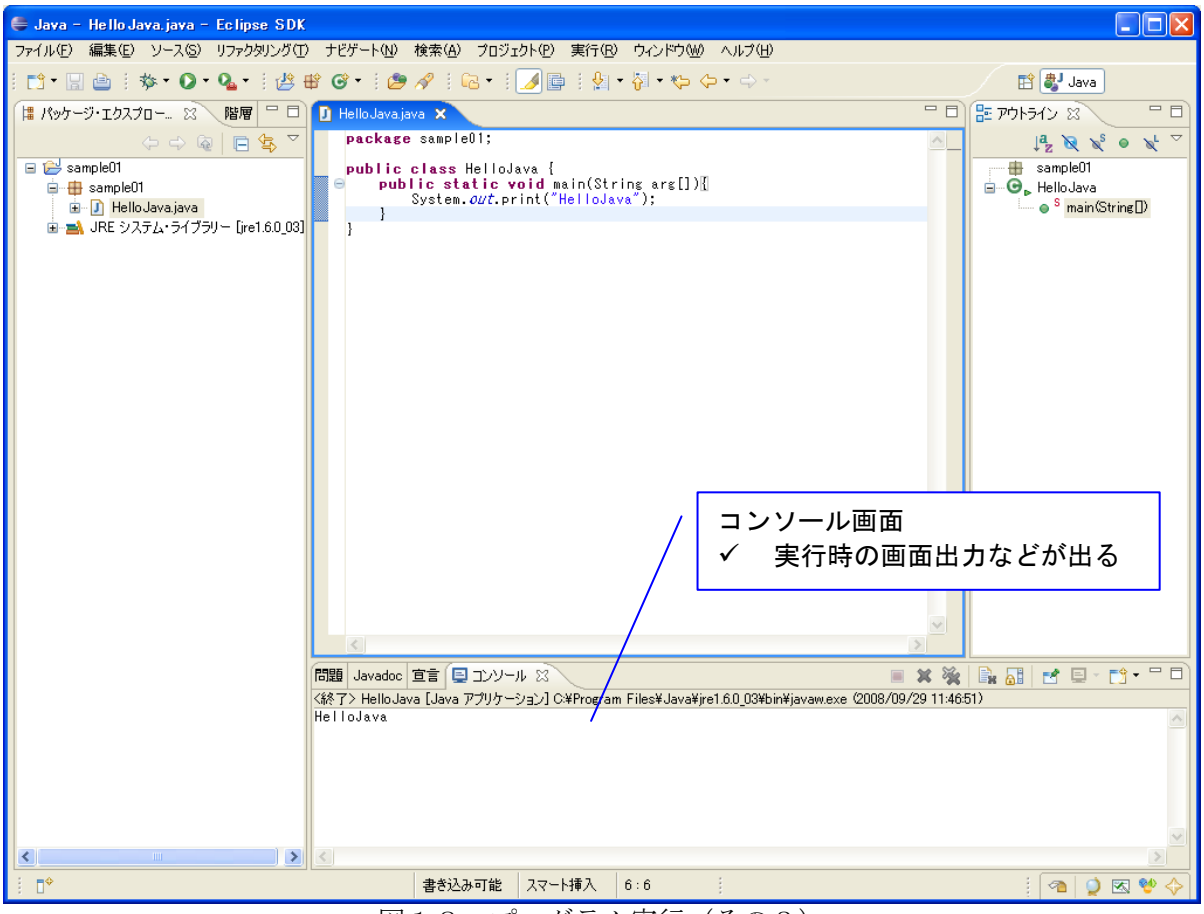

図１８　プログラム実行（その３）

グラフィックを使用する場合は、Web からの実行が可能なアプレットを作成するほうが容易です。本演習でもアプレットを作成します。メニューバーの中の“実行(R)”から“実行(S)”を選択すると、“Java アプリケーション”のかわりに“Java アプレット”と出てくるのでそれを選択しましょう（図１９）。

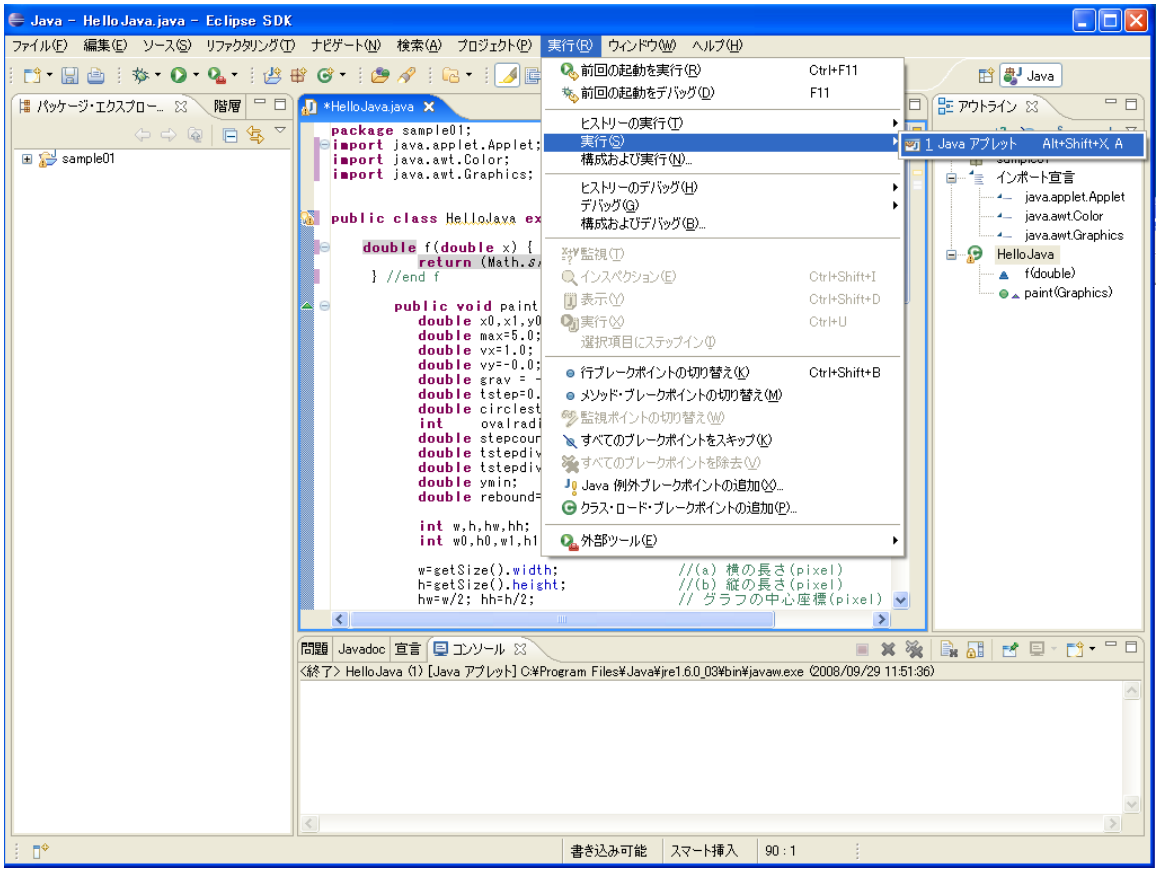

図１９　プログラム実行（その４）

もしもエラー等があるときは、図２０のようになるので、“先行（P)”を押し、起動し、コンソール画面にエラーメッセージが出るのでそれを見て対処しましょう（図２１）。キャンセルで起動を停止してもかまいませんが、コンソール画面にエラーメッセージは出てきません。エラーメッセージの画面で、メッセージをクリックすると、対応する行が上の窓に矢印で示されます。エラーの箇所にカーソルを近づけると、エラーに関する情報が出てきます（図２２）。修正して再度コンパイルしてください。

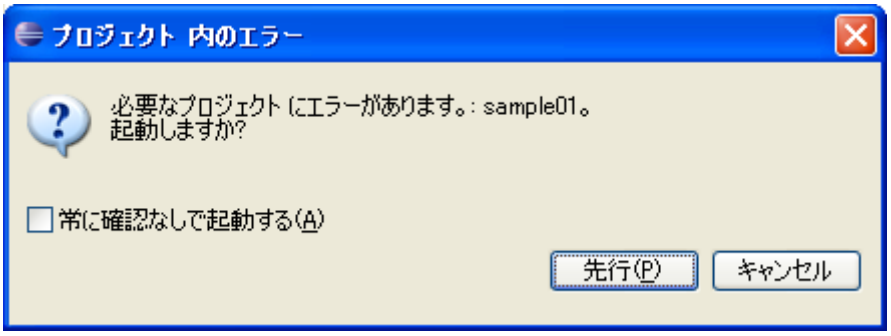

図２０　プログラム実行（その５）

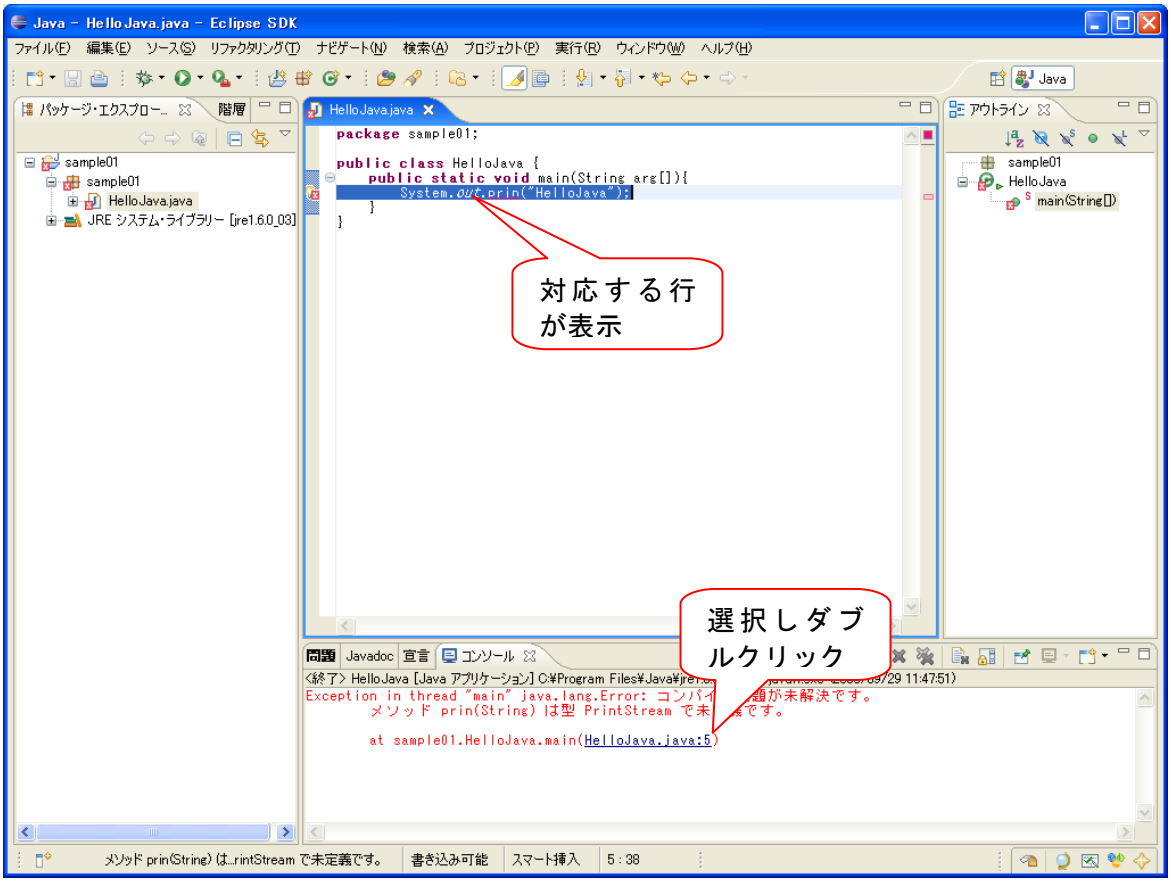

図２１　プログラム実行（その６）

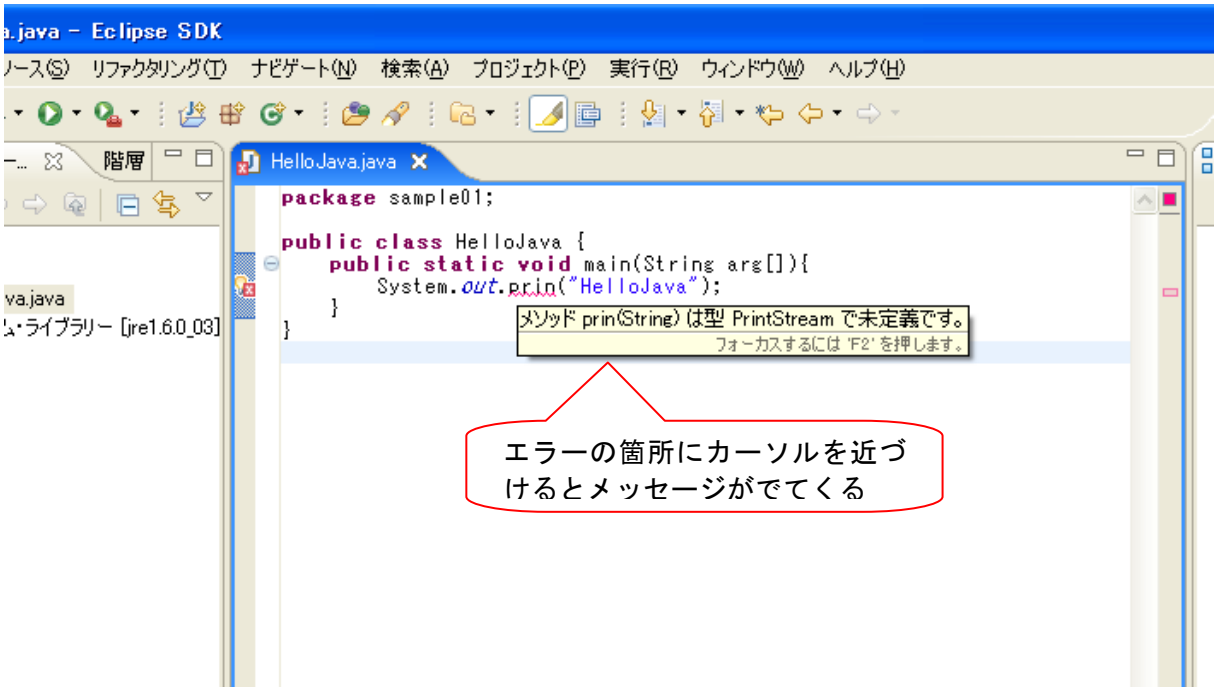

図２２　プログラム実行（その７）

終了は、メニューバーの中の“ファイル(F)”から“アプリケーションの終了(X)”を選択、あるいは、右上の×マークをクリックしてください。

2.2　既存のプログラムを開く時

以前に作成したプログラムを開くときは、eclipse 起動時に指定したワークスペース内にプロジェクトがある場合パッケージエクスプローラ内にそのプロジェクトが出てきます（図２３）。関連するファイルをダブルクリックすることで、編集できます。実行は実行したいファイルをクリックした後、メニューバーの中の“実行(R)”から“実行(S)”から“Java アプリケーション”を選択してください。

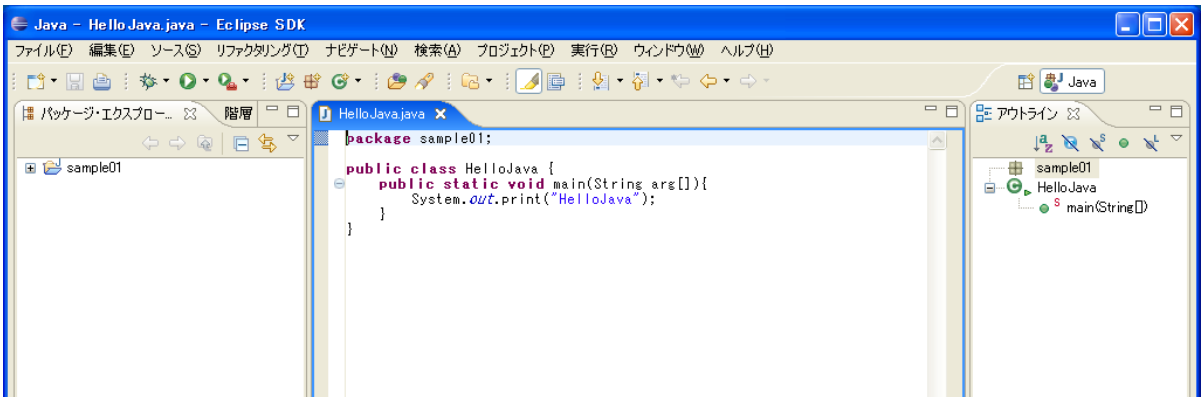

図２３　既存のプログラムを開く

2.3　Java と C との比較

２年生及び３年前期実験で使った C 言語と Java 言語の違いについて簡単にまとめます。

大きな違いとしては、Java 言語はオブジェクト指向言語であり、基本要素はクラスになります。また入出力の方法が大きく異なります。したがって、以下に示すサンプルプログラムにあるように、まずクラス定義が必要になり、クラスの中に main 関数を定義する必要があり、printf のかわりに System.out.print と打つことが必要になります。しかしながら、変数の型、プログラムの制御構造、算術演算子等似通った部分も多いので理解しやすいのではないかと思います。

以下、簡単な比較事項を書きます。

項目	C	Java
コメント	/* … */	/* … */, //行末までコメント
整数	int	Int
単精度浮動小数点	float	Float
倍精度浮動小数点	double	Double
文字	char	Char
文字列	char 型の配列	String クラス

制御構造（if 文）	if （論理式） {
	　　文１（複数の文も可）
	}
	else {
	　　文２（複数の文も可）
	}
		if （論理式） {
		　　文１（複数の文も可）
		}
		else {
		　　文２（複数の文も可）
		}

反復制御（for 文）	for (式１; 式２; 式３) {
	　　文（複数の文も可）
	}
		for (式１; 式２; 式３) {
		　　文（複数の文も可）
		}

算術演算子 （加減乗除と余り）	+,-,*,/, %	+,-,*,/, %

三角関数	sin, cos,tan	Math.sin,Math.cos,Math.tan
平方根	sqrt	Math.sqrt
画面入力	scanf	処理が複雑
画面出力	printf	System.out.print()（改行なし） System.out.println()（改行）

Java の print や println 関数では、整数や実数をフォーマットして出力する機能はありません。
たとえば C だと printf(“%d¥n”,var);と書くところを Java では System.out.println(var);と書くことで変数 var の中身を出力することができます。入力は難しく下記のようになります。

```
import java.io.*;
public class Test {
       public static void main(String args[]) throws IOException
       {
       BufferedReader in =
           new BufferedReader(new InputStreamReader(System.in));
       double d;
       int i;
       System.out.print("Input Integer data ");
       i = Integer.parseInt(in.readLine());
       System.out.print("Input Real data ");
       d = Double.parseDouble(in.readLine());
       System.out.println(" Integer " + i + " Real " + d);
       }
  }
```

図２４　サンプルプログラム２

2.4　終わりに

まずは、この資料どおり実行してみてください。不明な点は、TA にもどしどし質問しましょう。
より詳しく理解したいとか、わからないことがあったら、参考書がたくさん出ているので、まず調べましょう。ここで使った参考書のリストは下に載せます。

参考文献

[1] 高橋 麻奈、やさしい Java 第 3 版 (やさしいシリーズ)、ソフトバンククリエイティブ、
比較的簡単に記述している。ただし教科書的な記述である。
[2] 水島 和憲、Eclipse3 による Java アプリケーション開発、秀和システム、
Eclipse の記述の際に参考にした。