第1章　Eclipse の概要

【学習内容とねらい】

これから、Eclipse（エクリプス）を用いた Java 言語プログラミングの学習を始めます。まず、本章では、Eclipse の基本操作の仕方をマスターしましょう。とは言っても現時点では「Eclipse とは何？」と戸惑う人も多いと思います。詳細は、この後すぐに体験してもらうとして、一言で言うと、無償で公開されている（プログラム開発のための）**統合開発環境**（ソフト）と言うことになります。開発プロジェクトの内容や最新版のダウンロードについては **http://www.eclipse.org/**を参照してください。ここでは、統合開発環境という言葉について少し補足しておきましょう。皆は、すでにワープロを用いた文書作成を習得したと思います。具体的には MS Word を用いたと思いますが、このワープロを用いれば、文書の作成・編集そして印刷やファイルへの保存など、文書作成に必要な操作を全て処理できたはずです。あえて言えば、ワープロは文書作成の統合開発環境ソフトと言って良いでしょう。それと同じように、Eclipse は Java 言語プログラム開発に必要な全ての処理が備わったソフトだと捉えて下さい。これで少しはイメージがつかめたでしょうか。

もう少し具体的に説明しましょう。前章で述べた通り、Java 言語のプログラムは、「ソースコードの記述」→「コンパイル（翻訳）によるバイトコードへの変換」→「JVM によるバイトコードの解読・実行」という過程を経て実行結果を得ます。これが「プログラムの作成・実行に必要な操作」ということになります。したがって、Eclipse がこれら操作を行う環境を全て揃えてくれている、ということになります。

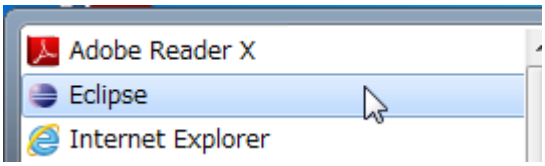
本章で学ぶのは今後（毎回）必要となる基本操作です。慣れれば確実に身に付く簡単な操作ですので、どうかしっかりと身につけて下さい。なお、1-4 節では、統合開発環境を用いずに、**手作業で** Java 言語プログラムを作成、翻訳・実行するという体験を行ってもらいます。こうすることによって本来必要な作業が、Eclipse を用いることによっていかに軽減されているかが理解できると思います。

＜第１章の構成＞

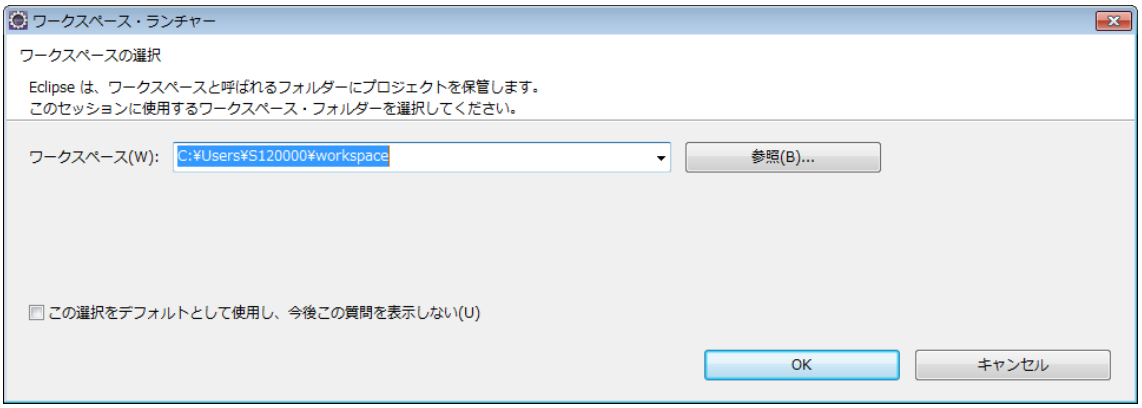
1-1　Eclipse の概観
1-2　アプリケーションの作成・実行
1-3　アプリケーションの編集
1-4　Java 言語プログラムの翻訳・実行（コンソールバージョン）

1-1 Eclipse の概観

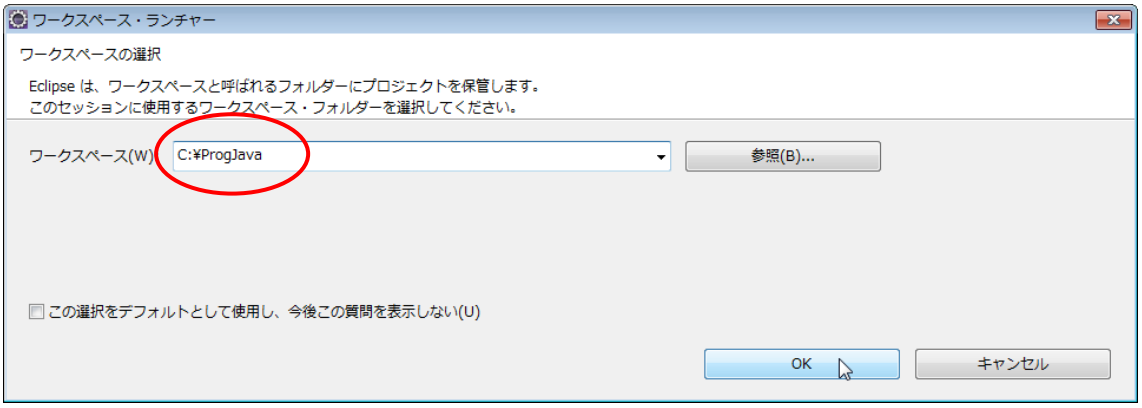
まず、Eclipse を起動しましょう。[スタート]ボタンから[すべてのプログラム]→[Eclipse]を選択します。



すると、下のような、ワークスペース選択の画面が現れます。ワークスペースとは、Eclipse を使って作成するプログラムの保管場所（フォルダ）のことです。



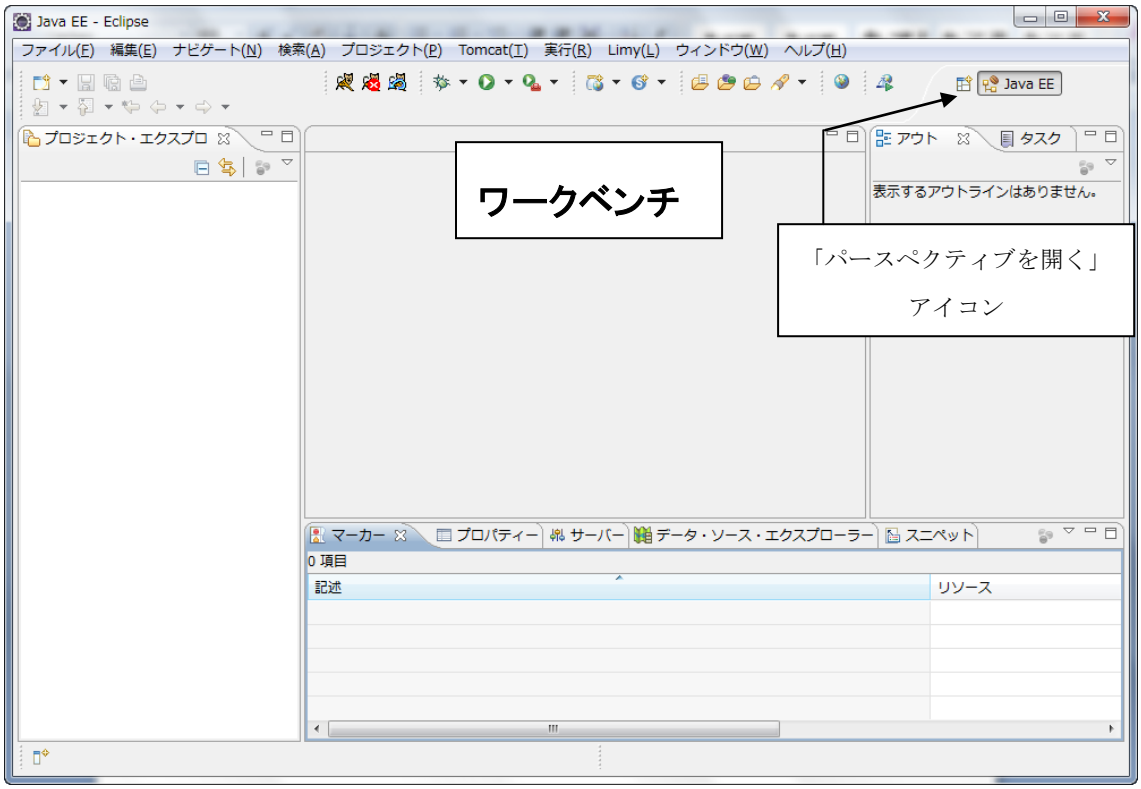
ここでは、下のようにワークスペースの名前を「ProgJava」に変更します（名前は何でも良いのですが、混乱を避けるためにこのように統一してください）。また、保管場所については、下では C ドライブのルートディレクトリにしていますが、これは各自適当な場所に変えて結構です。



指定後［OK］ボタンをクリックすると、Eclipse の起動画面に進みます。なお、次回以降は、ここで指定したワークスペースが表示されますので、そのまま［OK］ボタンをクリックするだけで結構です。授業とは別に趣味で作ったプログラムを保管したい，など保管場所を分けたい場合にだけ、適宜別のワークスペースを指定してください。

ワークスペースの確認が終わると次の初期画面（ようこそ画面）が現れます。ここでは、右上隅にある「ワークベンチ」ボタンをクリックしてください。

すると、次ページのワークベンチ画面が現れます。ワークベンチとは作業台という意味ですが、まさにこれから Eclipse で作業を行う際の作業台という訳です。なお、2 回目以降の起動後は、上の「ようこそ」画面は現れず、直接ワークベンチが現れるようになります。

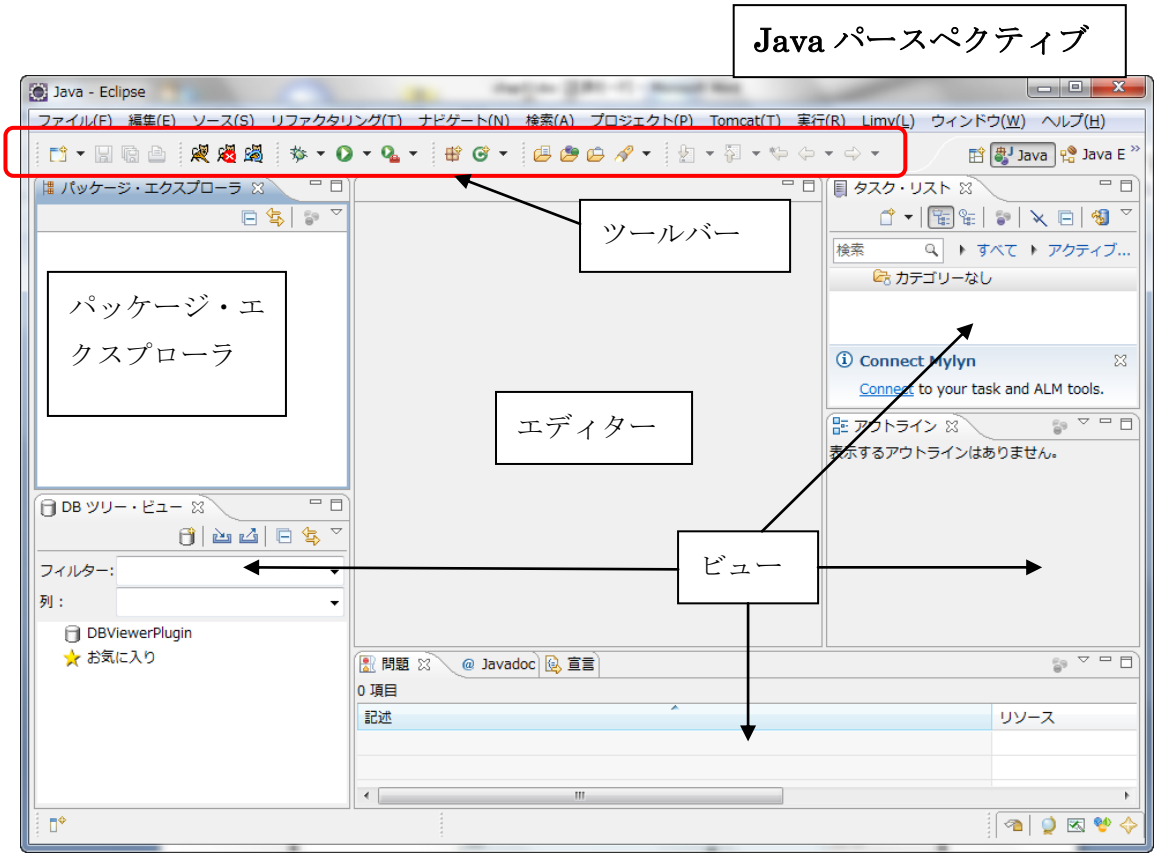


パースペクティブの切り替え

ここで、パースペクティブを切り替えます。パースペクティブとは、「視点」という意味ですが、ここでは、作業の種類によって設定されているワークベンチのレイアウト、という意味です。例えば一つの作業台でも本と筆記用具が置かれていれば勉強机になり、それを取り払って箸や食器を置けば食卓になるようなものです。

Eclipse には幾つかのパースペクティブが用意されていますが、当面用いるのは **Java パースペクティブ**です（ちなみに、最初は Java EE パースペクティブになっています）。このパースペクティブに切り替えるため、ワークベンチ右上にある「パースペクティブを開く」アイコンをクリックしてください。そうして現れたメニューから「Java」を選択してください。

すると、次のように、Java パースペクティブに切り替わります。

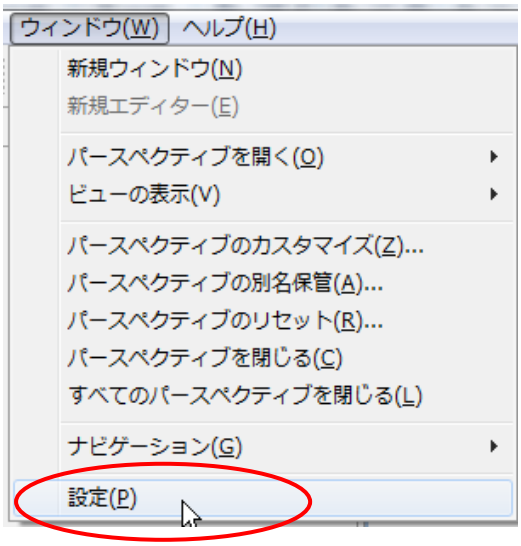


ワークベンチの特徴は、ビューと呼ばれる区画に区切られていることです。これらビューはそれぞれの役割を持っており、例えば中央にあるエディターは、プログラムの編集を行う領域であり、「パッケージ・エクスプローラ」ビューには、ワークスペース内のプログラムリストが表示されます。これらを含めた各ビューの機能については、以下、実際にプログラムを作成しながら把握して行くことにしましょう。次へ進んでください。

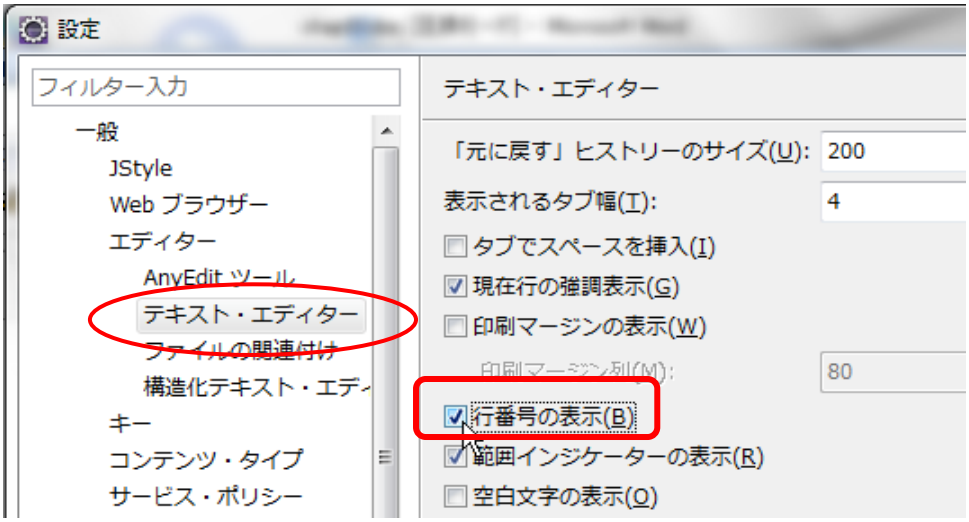
準備－Eclipse の設定

実際に Java プログラムの作成に入る前に、Eclipse の動作環境に関する設定を 3 点ほど行いましょう。

ワークベンチの「ウィンドウ」メニューから「設定」を選択してください。すると、次ページのように、Eclipse の動作環境に関する設定画面が現れます。

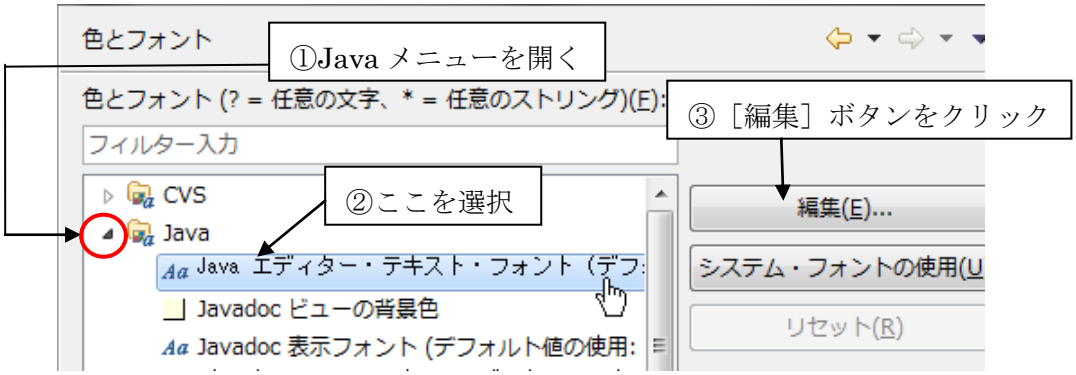


設定画面の左側にあるリストから「一般」→「エディター」→「テキストエディター」とたどり、現れた選択項目の中から「行番号の表示」を選択します。これにより、プログラム編集画面に行番号が表示されるようになり大変便利です。

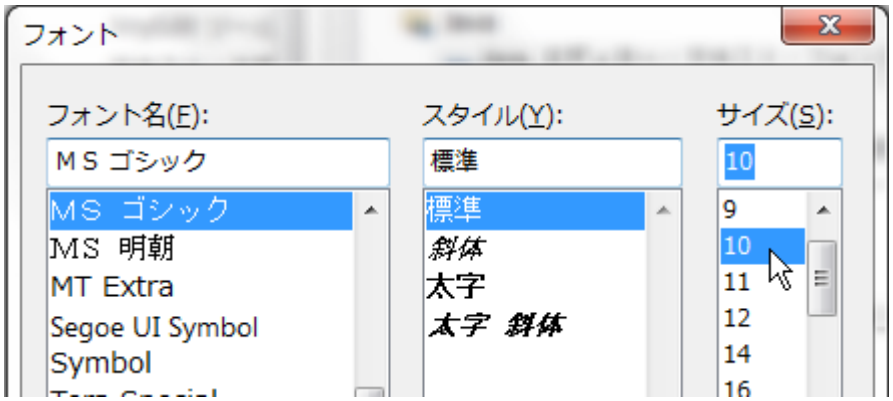


続いて、同じ画面の下方にある「Colors and Fonts」設定へのリンクをクリックします。

そして現れた「色とフォント」設定画面から、「Java」→「Java エディター・テキスト・フォント…」欄を選択し、［編集］ボタンをクリックします。

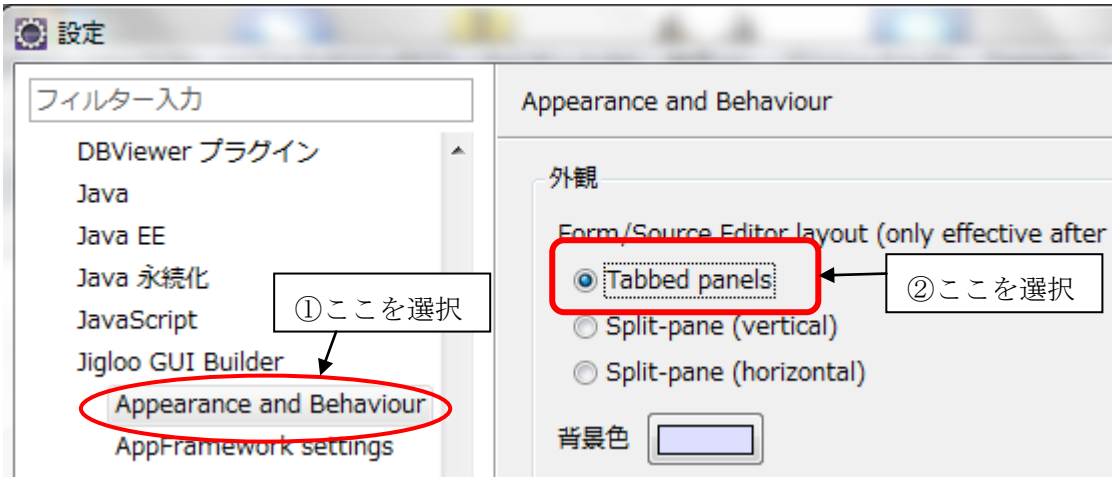


すると、Java プログラミング記述時に使用する文字フォントの設定画面が現れます。ここで、次のようにサイズを（当初の 9 ポイントから）10 に変更して、画面下方の[OK]ボタンをクリックして下さい。これでフォント設定が完了し、全体の設定画面に戻ります。



このように文字を 1 ポイント大きくしたのは、デフォルトで設定している 9 ポイントの文字サイズでは小さくて見にくいためです。実際に使用してみて、さらに大きくしたい場合は、同じ要領で変更して下さい。

最後に、設定画面のメニューから今度は「Jigloo GUI Builder」の中にある「Appearance and Behaviour」を開き、外観という項目の「Tabbed panels」を選択します。これにより、プログラム編集時に使用する複数の画面をタブクリックで切り替えられるようになります。



以上の設定後、設定画面下の［OK］ボタンを押して、設定を確定させてください。

これで、準備が完了したので、次節のプログラム作成の手順に進むことにしましょう。いったん、ワークベンチの「ファイル」→「終了」メニューを選んで Eclipse を閉じてください。

1-2　アプリケーションの作成・実行

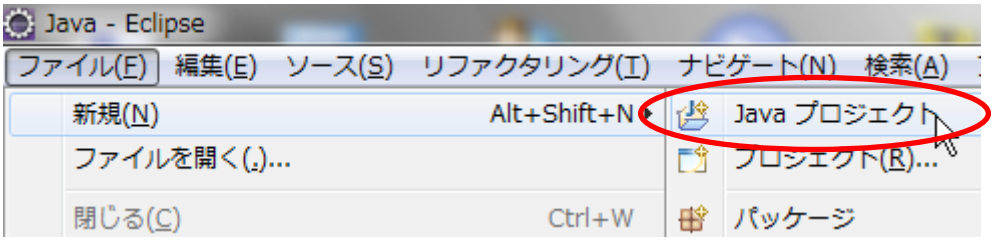
本節では、新たにプログラム（Windows アプリケーションを想定しているので、今後**アプリケーション**と呼びます）を作成する際の手順を学習します。前節で説明した通り、Eclipse を起動させてください。前節で指定したワークスペースが「ワークスペースの選択」画面に現れるので、そのまま［OK］ボタンを押すと、ワークベンチが直接現れるはずです。

【基礎課題 1-2-1】

以下のⅠ～Ⅲの指示に従って、アプリケーションの作成・実行を行ってください。

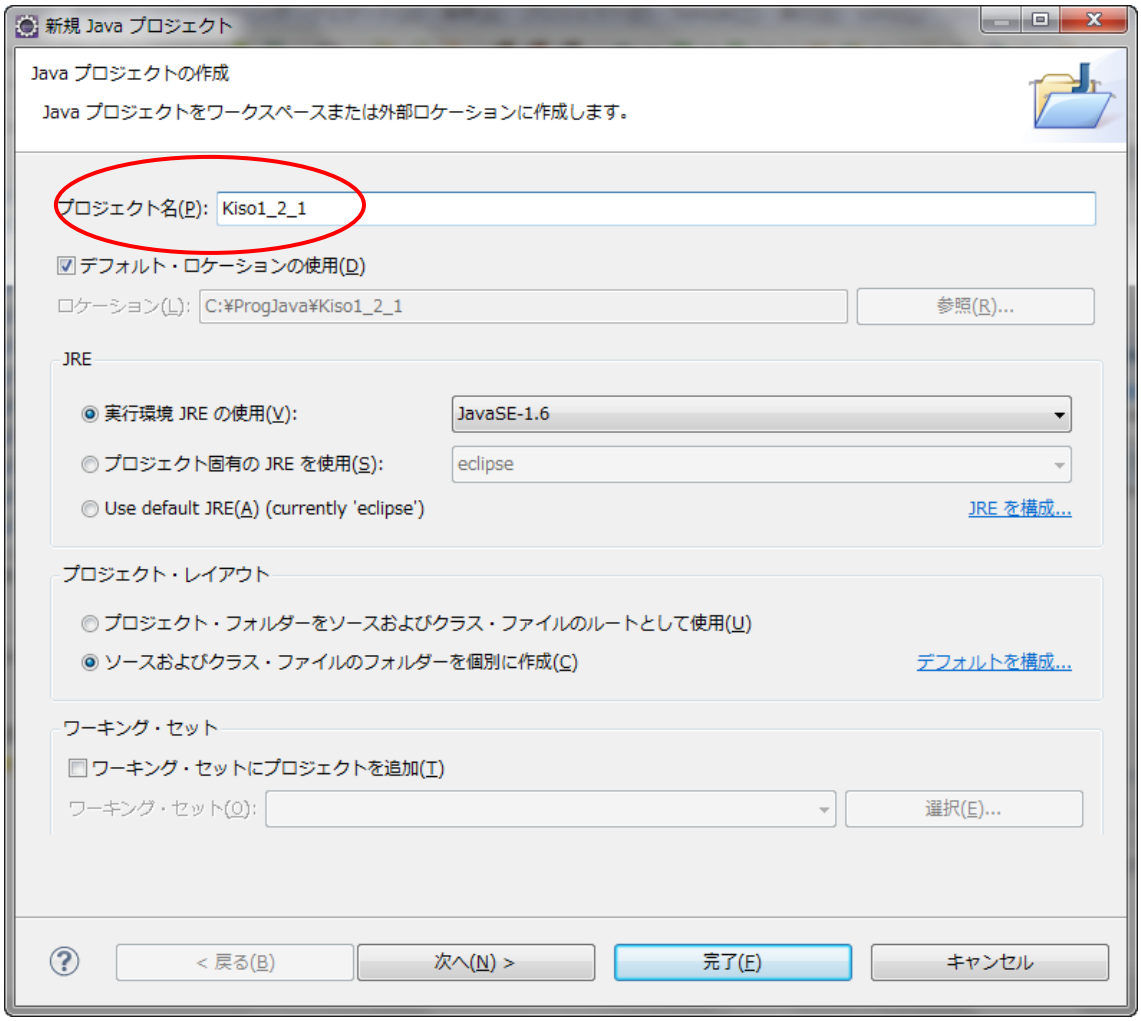
Ⅰ．プロジェクトの作成

ワークベンチのメニューから「ファイル」→「新規」→「Java プログジェクト」と選択してください。

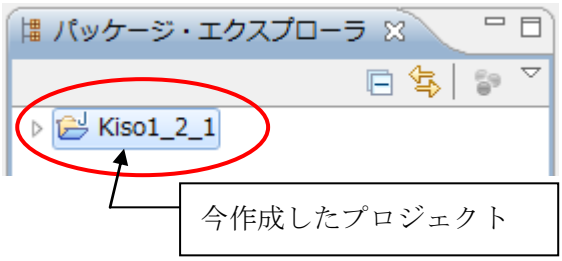


すると、次ページのように新規 Java プロジェクト設定画面が現れます。

ここでは、プロジェクト名を「Kiso1_2_1」と指定して［完了］ボタンをクリックしてください。今後の課題も同じ要領でプロジェクト名を指定します。

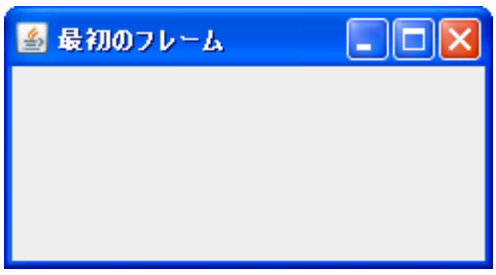


プロジェクトの設定後、ワークベンチ左側のパッケージ・エクスプローラを見ると、今作成したプログジェクト「Kiso1_2_1」が表示されているはずです。



さて、Java プロジェクトとは何でしょうか？それは、一言で言うと、Java プログラムを入れる“入れ物”ということになります。この中に、Eclipse がプログラムを実行するのに必要なファイルが格納されます。ですから、Eclipse ではまず、プロジェクトを作成することから始めなければならないのです。

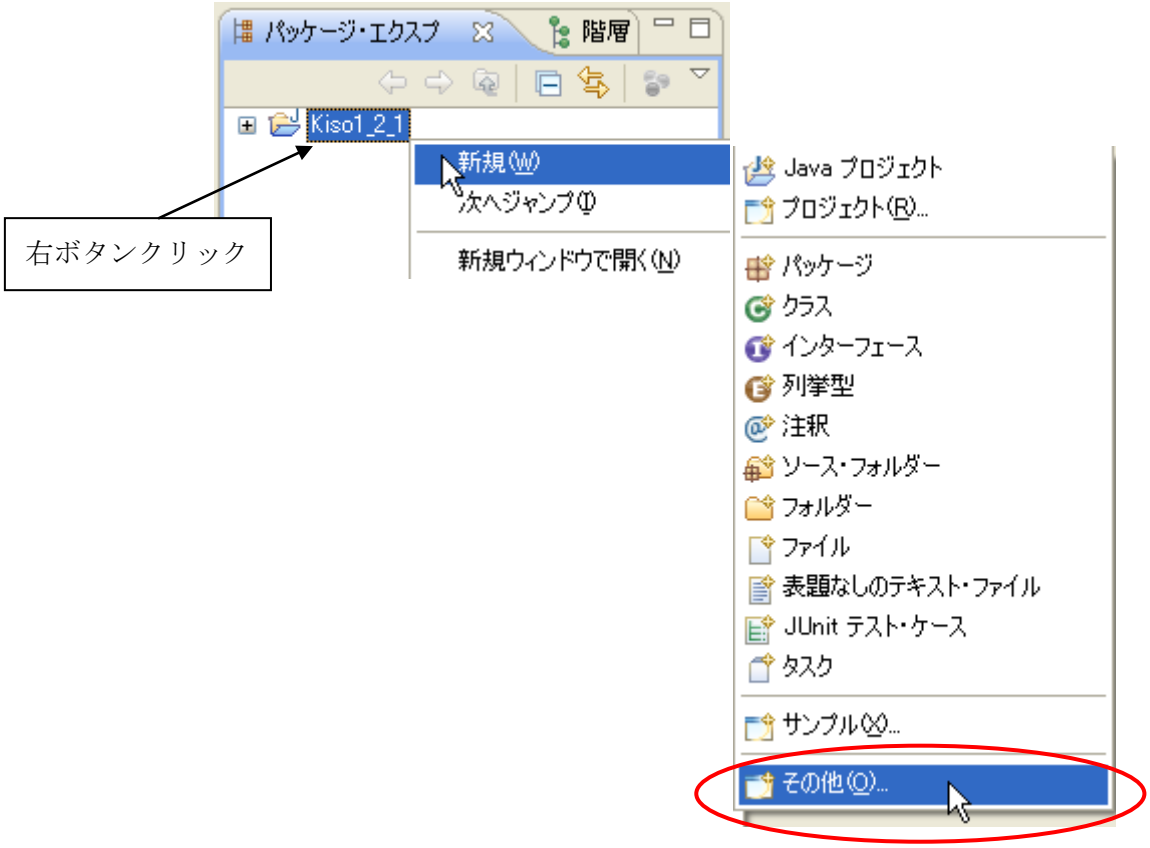
Ⅱ．アプリケーションのひな形の作成



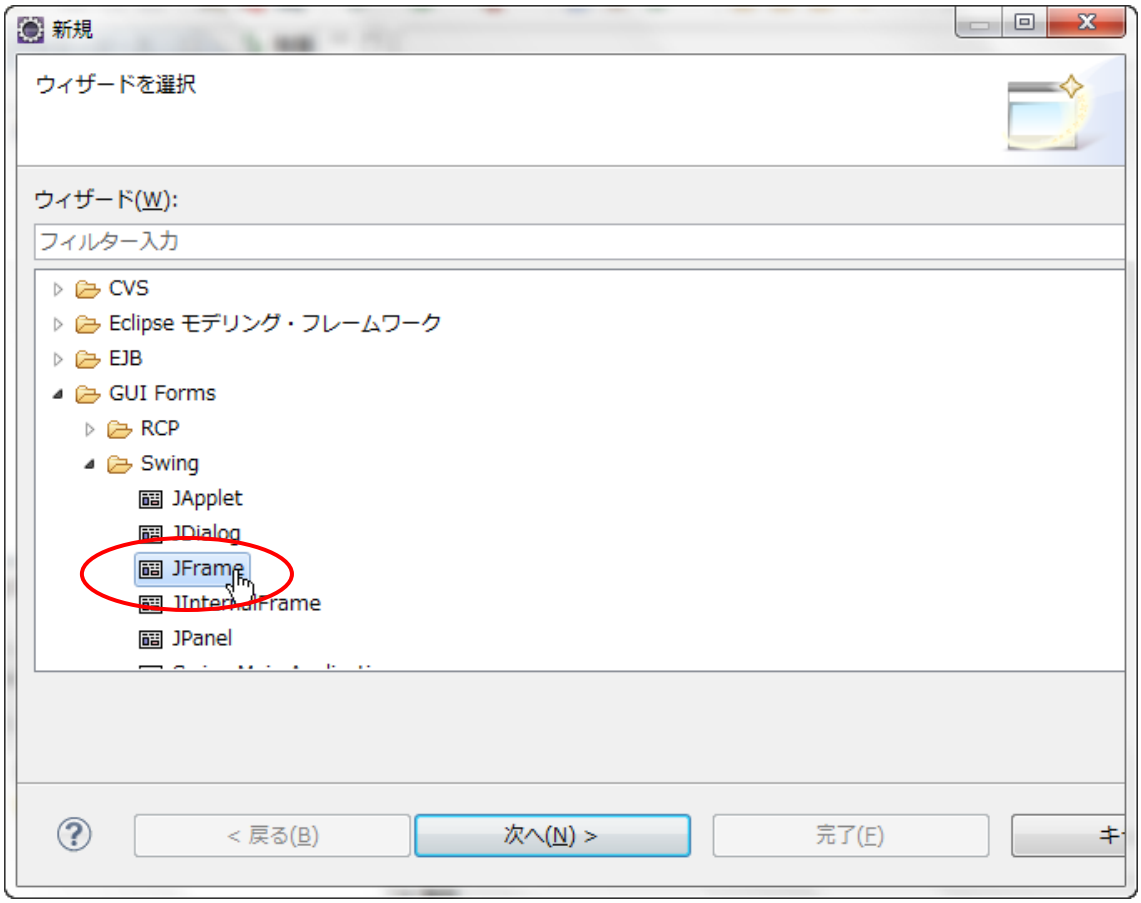
プログラムの“入れ物”が出来たので、その中にプログラムを作成します。これから作成するのは、次のような Windows が現れるプログラムです。

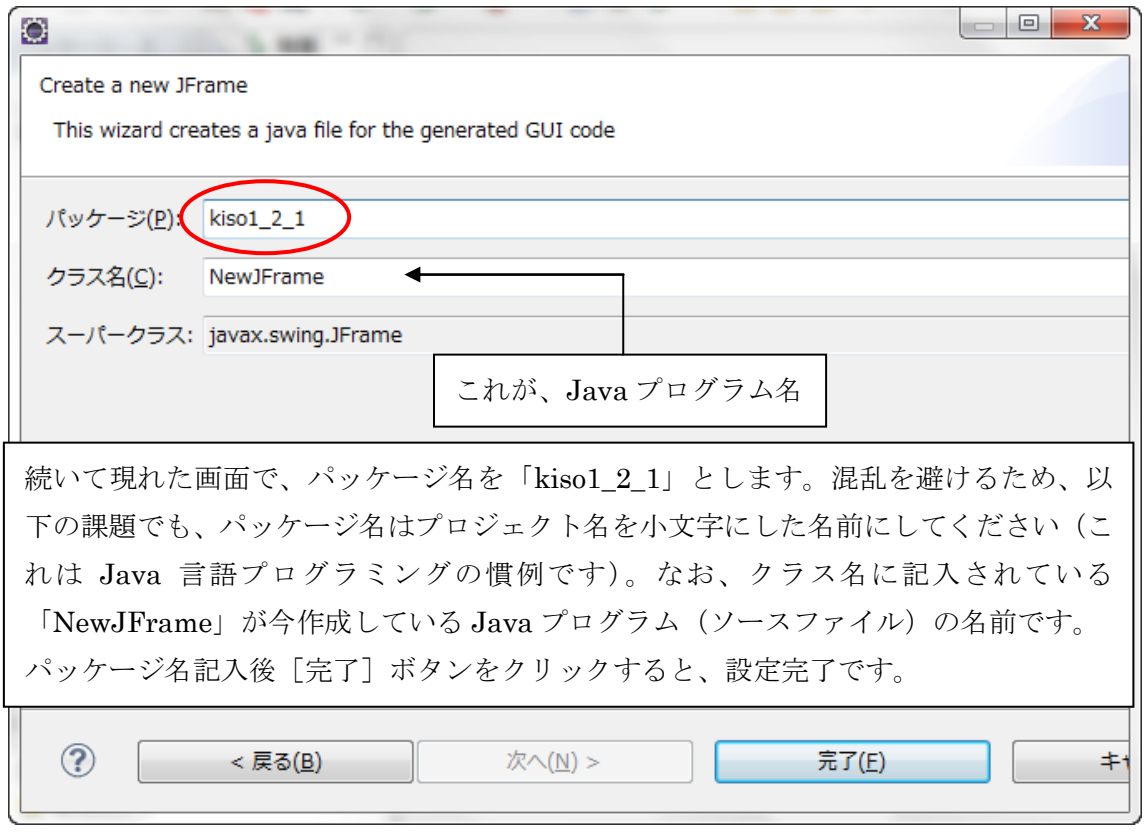
このようなプログラムを一から記述するのは初心者には大変です（第 10 章で体験します）。ところが、Eclipse には Windows（Java 言語では**フレーム**と呼びます）を表示する部分を予め記述したプログラムがひな形として用意されています（正確には Jigloo GUI Builder というプラグインソフトを組み込んだ後で使用可能になります。皆の Eclipse にはプラグイン済みです）。それを利用しましょう。

ワークベンチのパッケージ・エクスプローラにある、今作成したプロジェクトにカーソルを合わせ、右ボタンクリックしてください。そして、現れたメニューから「新規」→「その他」を選択します。



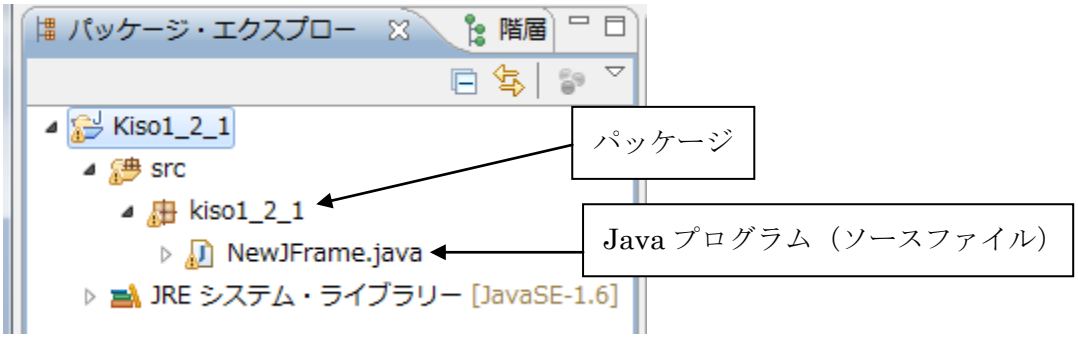
すると、次ページのような新規画面が現れます。ここで、「GUI Forms」→「Swing」→「JFrame」を選択して、［次へ］ボタンをクリックしてください。



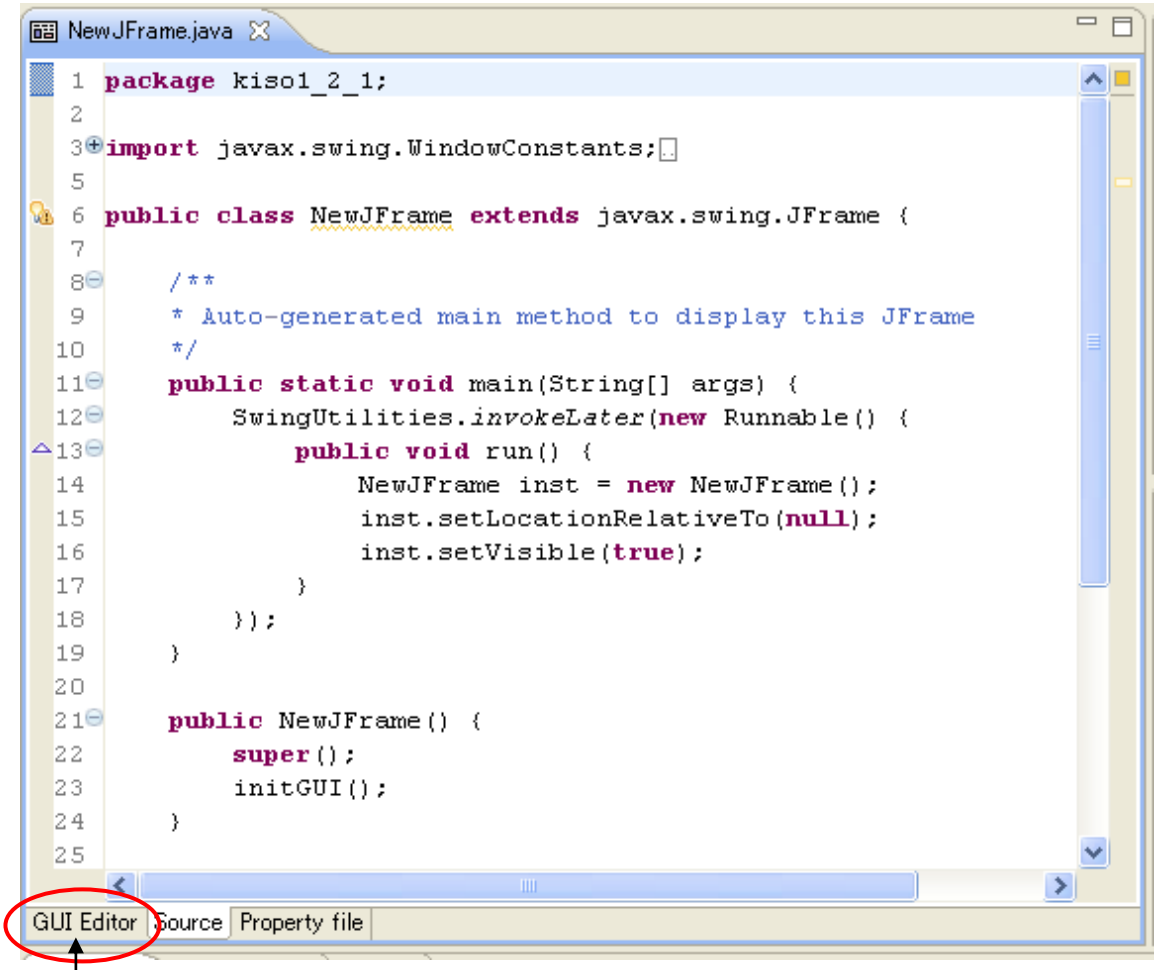


続いて現れた画面で、パッケージ名を「kiso1_2_1」とします。混乱を避けるため、以下の課題でも、パッケージ名はプロジェクト名を小文字にした名前にしてください（これは Java 言語プログラミングの慣例です）。なお、クラス名に記入されている「NewJFrame」が今作成している Java プログラム（ソースファイル）の名前です。パッケージ名記入後［完了］ボタンをクリックすると、設定完了です。

上の設定後、パッケージ・エクスプローラを見ると、次のように今作成した Java プログラム（ソースファイル）が表示されているはずです（適宜 ▷ ボタンをクリックしてフォルダを展開し、確認してください）。

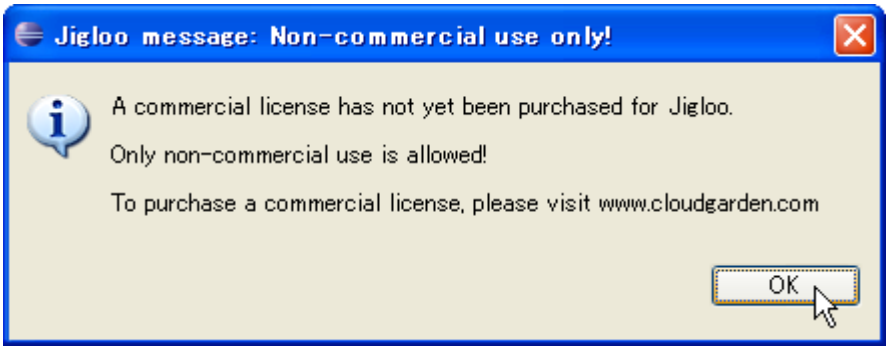


また、このとき中央のエディターには、次のようなプログラムが表示されています。これが、今自動生成したプログラムです。フレームを表示させる Java プログラムですが、今は内容を気にする必要はありません。

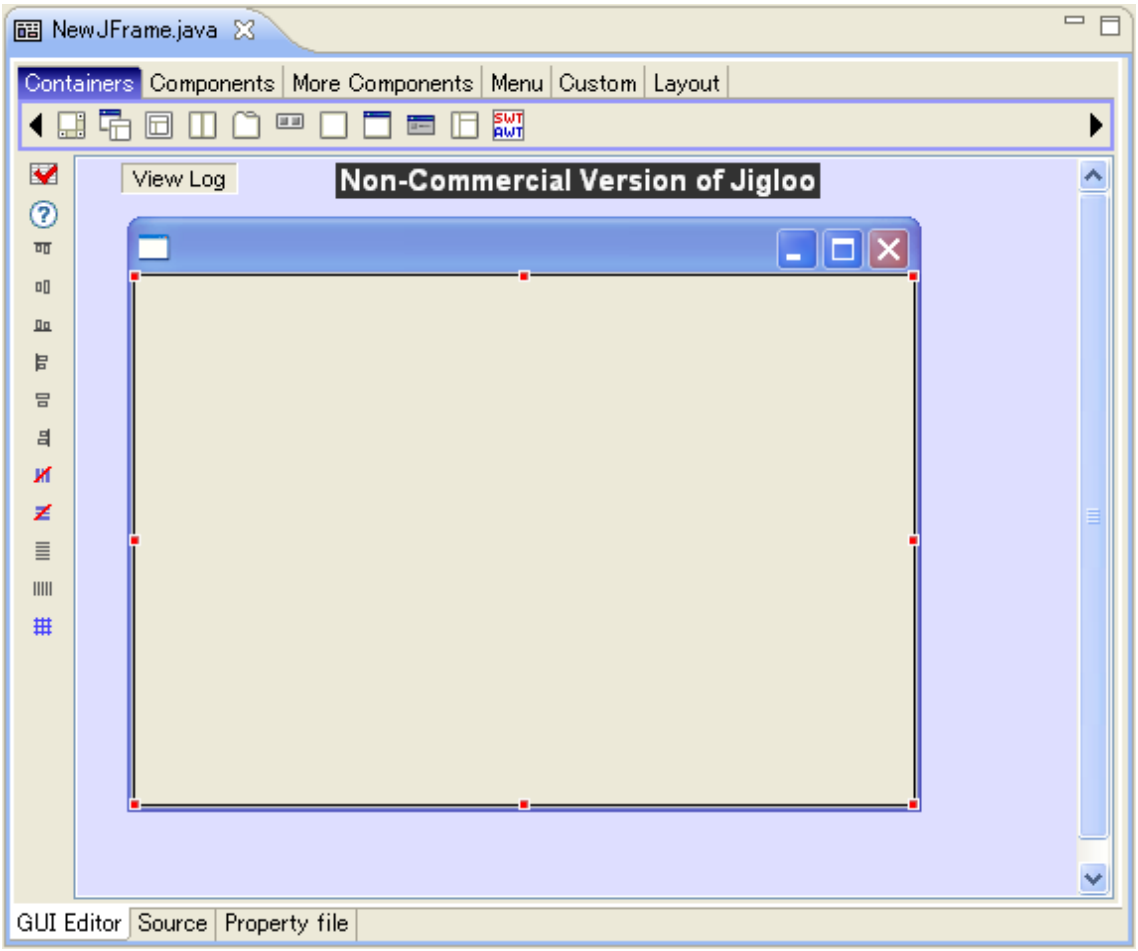


次に、エディターの「GUI Editor」タブをクリックしてください。

すると次のような画面が現れますが、ここでは［OK］をクリックしてください。これは、今使っている Jigloo GUI Builder というソフトのライセンスが非商用のライセンスであることを告げるものです（今は教育に使っているのでこのライセンスで問題ありません）。



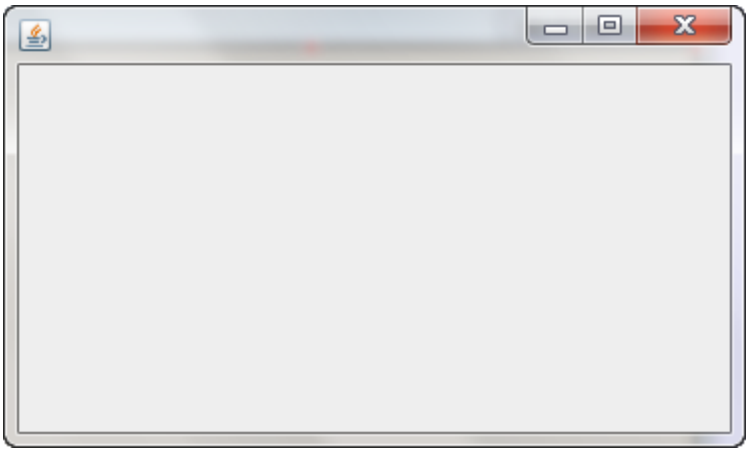
すると、エディターが次のようにフレーム設計画面に切り替わります。この画面上で、フレームの大きさを変えたり、またフレームの上にボタンや入力欄を配置したりする、という設計操作が容易にできます。そのやり方は次章で学習する事にしましょう。画面を確認したら次に進んでください。



Ⅲ. アプリケーション・プログラムの実行

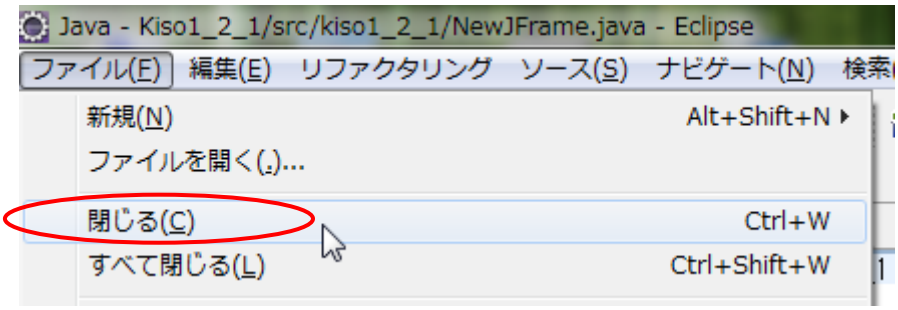
ここで、今作成した（と言ってもまだ何もやっていませんが･･･）プログラムを実行してみましょう。ワークベンチのツールバーから実行ボタンをクリックしてください。

すると、次のようなフレームが現れます。これが今のプログラムの実行結果です。画面を確認したら×をクリックしてフレームを閉じて（プログラムを終了させて）ください。



以上が、Eclipse で Windows アプリケーション（フレームを持つプログラム）を作成し実行する一連の手順です。

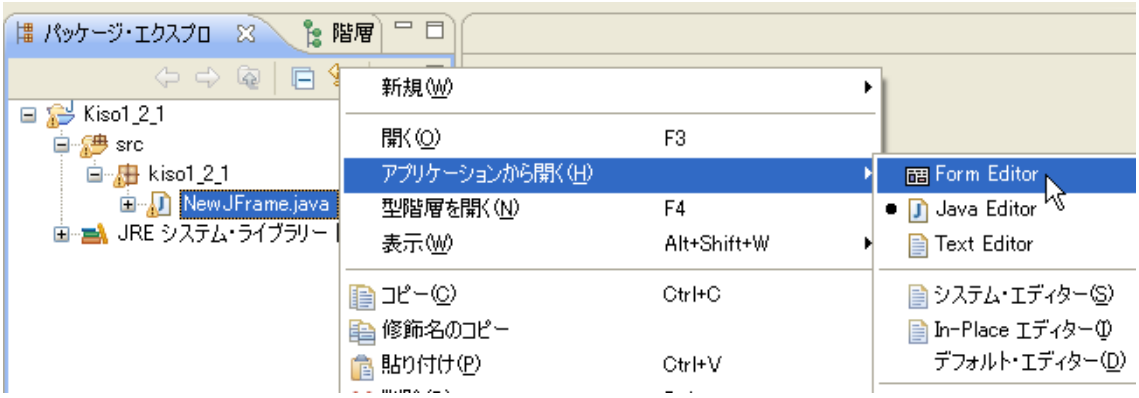
ここで、いったん今作成したプログラムを閉じましょう。ワークベンチのメニューから、「ファイル」→「閉じる」を選択してください。このように、今後も作業が終わったプログラムは閉じるようにしましょう。



1-3 アプリケーションの編集

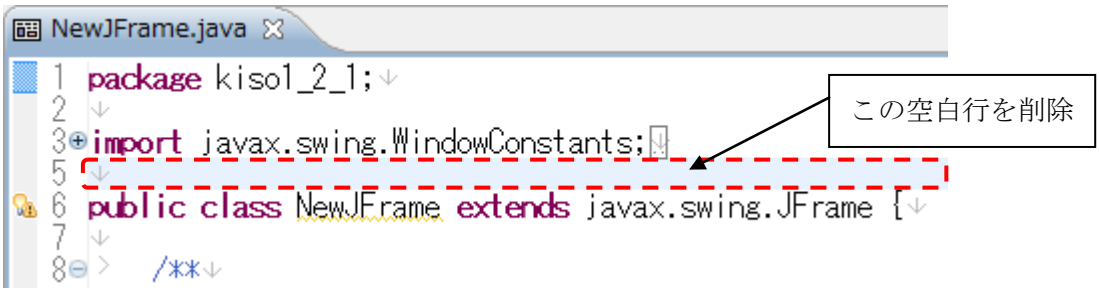
以降の学習では、一度作成したプログラムを開き、追加・修正等の編集を行う場面が出てきます。ここでは、そのやり方を確認しておきましょう。

先ほど【基礎課題 1_2_1】で作成したプログラムを開くには、パッケージ・エクスプローラから、当該 Java プログラム(今の場合 NewJFrame.java）を右ボタンクリックします。そして現れたメニューから「アプリケーションから開く」→「Form Editor」を選択します。



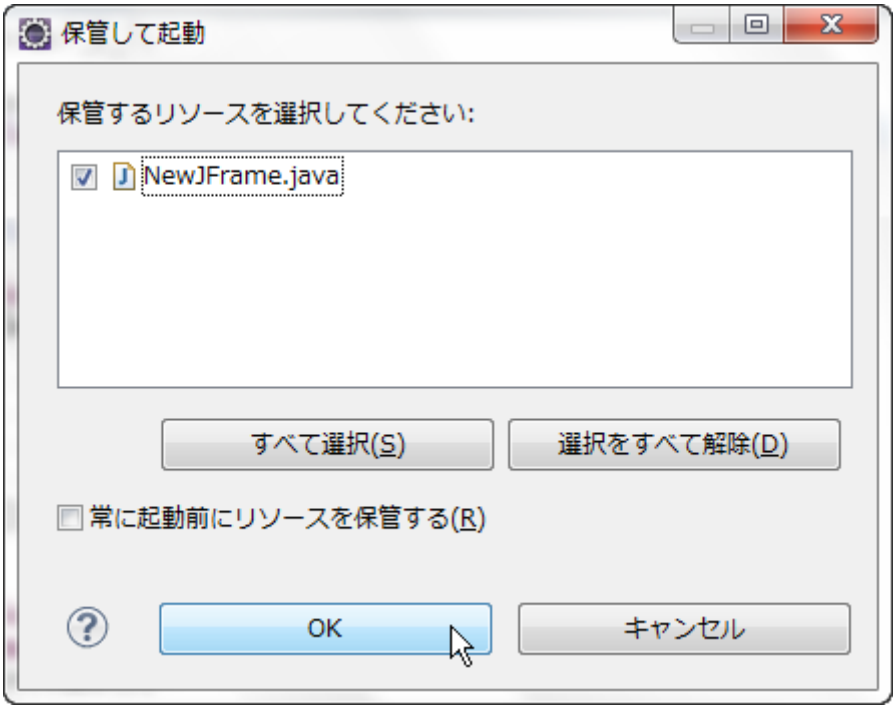
すると、エディターに先ほどのプログラムが現れます。なお、上で「Java Editor」を選択すると、「GUI Editor」タブが現れないので注意してください。

ここでちょっとだけ、プログラムを変更しましょう。と言ってもまだ内容を学習していないので、中身をいじることはしません。次のように空白の行を 1 行だけ削除することにします。実際上の意味はありませんが、これも立派な編集作業です。



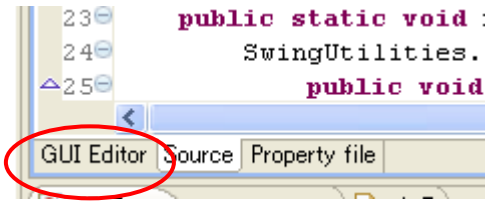
その上で、1-2 節で行ったようにプログラムを実行してください。

すると今度は、次のように保管するプログラムを指定する画面が現れます。ここで、[OK]ボタンをクリックすると、今編集したプログラムが保管され、プログラムが実行されます（もちろん、実行結果は前節と同じです）。

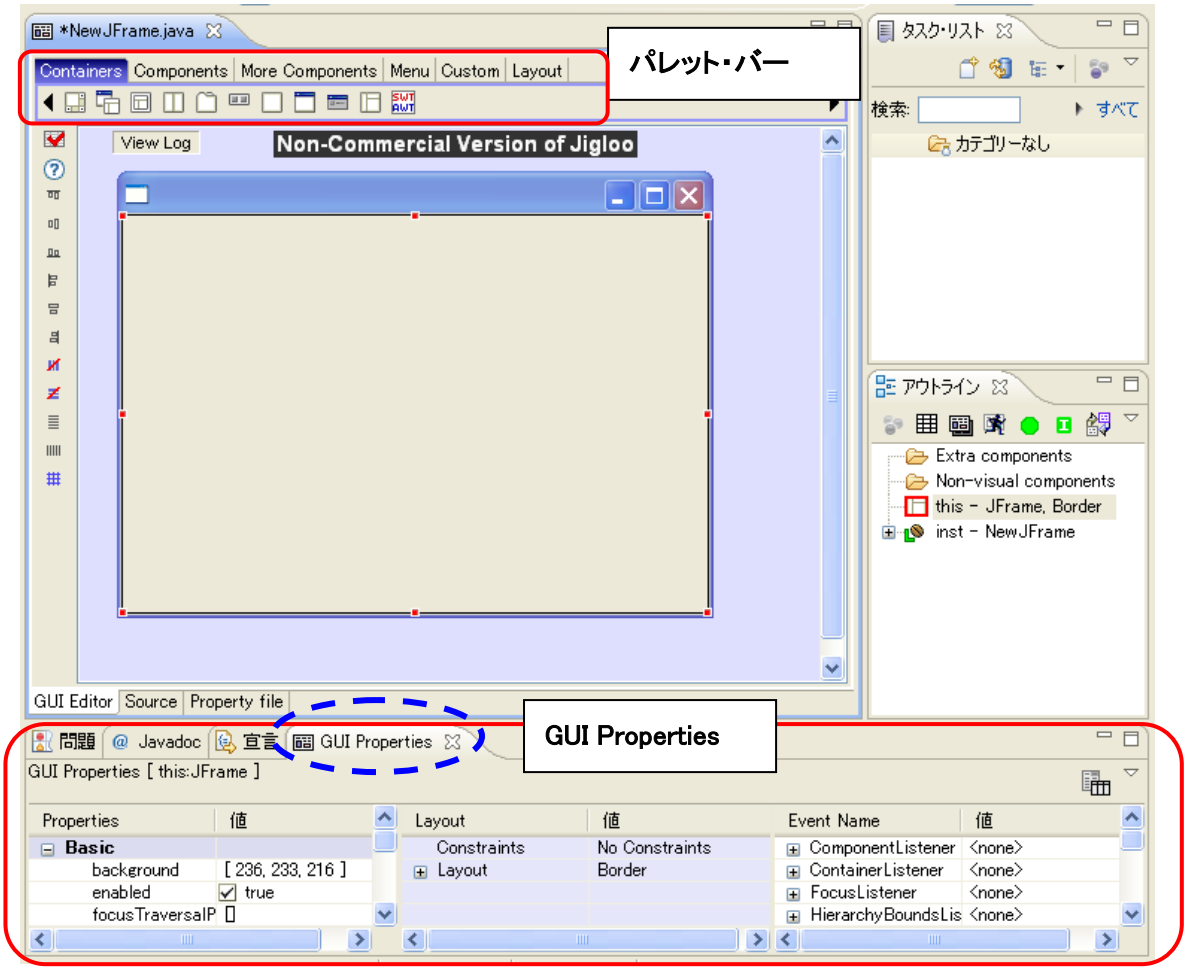


このように、Eclipse では、プログラムを実行する際、当該プログラムの変更箇所があると、まずそれを保管してから実行するようになっています。

さて、プログラムの編集と言っても、実質的にプログラムに何の変更もないのでは実感がわかないと思いますから、第 2 章の学習を少し先取りしてフレームに変更を加えてみましょう。エディターの「GUI Editor」タブをクリックしてフレーム設計画面を開いてください。



ここで、改めてフレーム設計画面を眺めてみましょう。使い方の詳細は第 2 章に譲るとして、ここでは大まかな構成を説明しておきます。

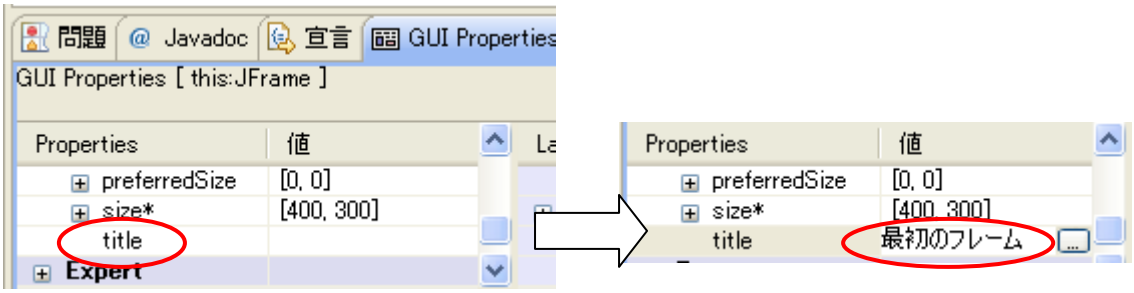


まず、中央にあるのがフレームで、言うまでもなくこれが起動後に表示される Windows になります。この上に、ボタンやテキスト入力欄など、Windows アプリケーションに必要な“部品”（**コンポーネント**と言います）を配置するのですが、それら部品は、上方のパレット・バーに納められています。

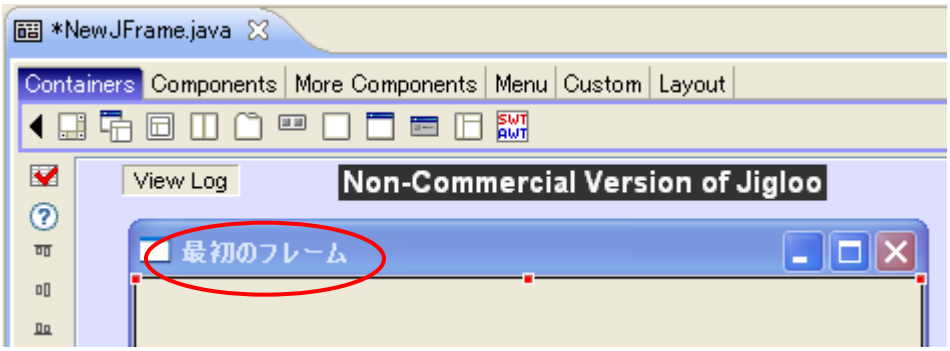
そして配置したボタンなどのコンポーネントはその色やボタン上に記される表記文字など、いくつかの性質を持っています。これらは**プロパティ**と呼ばれます。各コンポーネントについて定義可能なプロパティが、下方の「GUI Properties」ビューに納められています。そして、このビュー内の左端にある Properties 欄の値を変更するだけでコンポーネントの該当するプロパティを変えられるようになっています。中央のレイアウト（Layout）欄は、コンポーネントのレイアウトを指定する部分で、第 2 章で設定の仕方を学習します。最後に、右端に「**イベント**（Event）」という欄がありますが、これについては、第 3 章で詳しく説明する事にしましょう。

さて、それでは、フレームのタイトルを指定しましょう。上で説明した「GUI Properties」

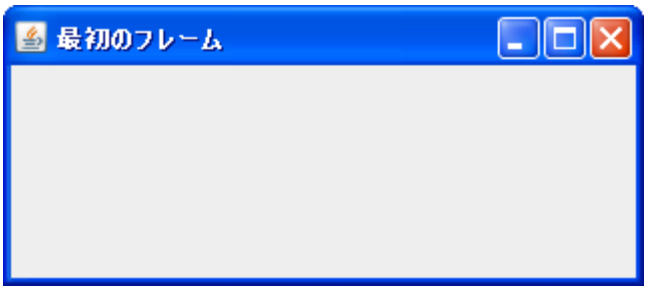
ビューのプロパティ欄をスクロールして、title 欄を見つけてください。そして、「値」欄に「最初のフレーム」と入力してください。



すると、フレームのタイトルバーに今指定した「最初のフレーム」と表示されているはずです。



この状態でプログラムを実行すると、次のようにフレームタイトルが表示されます。



実行結果を確認したら、このプログラムを終了してください。

このように、エディターの「GUI Editor」タブ内にあるフレーム設計画面では、フレームを自在に設計する機能が備わっています。詳細は第 2 章で学習します。一方、Java 言語を用いたプログラムの記述は「Source」タブ内にあるソース画面で行うことになります。したがって、皆はこれから以下の 2 点を学習することになります（きわめて大まかに言うと、ということですが・・・）。

① フレーム設計画面によるフレームの設計 → 主に第 2 章

② ソース（Source）画面による Java 言語プログラミング → 第 3 章以降

1-4 Java 言語プログラムの翻訳・実行（コンソールバージョン）

Eclipseはプログラミングの手間を省くために実は背後で様々な作業を行ってくれています。そのため、このようなアプリケーション・プログラム開発環境を **RAD**（Rapid Application Development）と呼んでいます。プログラムを迅速に（Rapid）開発できる環境という訳です。これは大変便利な開発環境なのですが、Java 言語プログラミング開発の流れあるいは仕組みを理解するためには、本来どのような作業が必要なのかを理解しておく必要があります。そこで、本節では、Eclipse を用いずに”**手作業**”で簡単なプログラムを作ってみましょう。前章で説明した、プログラム（ソースコード）の作成、翻訳・実行のプロセスを全て（手動で）行ってみるわけです。これらは、Eclipse などの RAD が開発される以前は全てのプログラマが行っていた作業です。以下の手順にしたがって、プログラムの作成・実行を行い、”プログラマ”の気分を味わってください。そうすれば、Eclipse の便利さ、有り難さも分かってくるでしょう。

Ⅰ．プログラム（ソースコード）の作成

秀丸エディタやメモ帳などのエディターを起動させ、以下のプログラムを記述して下さい（デスクトップ上に秀丸エディタのショートカットがない場合は、［スタート］→［すべてのプログラム］→［秀丸］とたどって秀丸エディタを起動できます）。

```
class ApplicationFirst {
    public static void main(String[] args) {
        System.out.println("Hello Java!");
    }
}
```

プログラム入力は全て半角英数文字で、空白の位置などは大体上の通りであれば結構です。また、Java 言語では大文字と小文字を区別しますので、上の通り入力して下さい。

何も意味が分からなくては釈然としないでしょうから簡単に解説しておきます。Java 言語では、class（クラス）というものを定義し、その中に特定の処理を記述します。つまり、

Java 言語プログラムはクラスの集まり

なのです。そして**クラス**は次の形で定義します。

```
class クラス名 {
  クラスの定義部分
}
```

上の例の場合は、

◆ クラス名が「ApplicationFirst」である。

◆ 「ApplicationFirst」クラスの定義部分には、「**main**」という名前の”**処理**”がある。

というようになっています。実は、この時点でオブジェクト指向という点について解説すると包括的な理解に到達できるのですが、それは別の機会に学習する事にして、今は全体の流れをできるだけ早くつかむことに重点を置き、先に進むことにします。

当面は以下の点を頭に入れておけば十分です。

◆ 上の”**処理**”を**メソッド**と呼ぶ。

◆ 一つのアプリケーション・プログラムは一般に複数のクラスからなる。

◆ その中のどれか一つのクラスには**main()**メソッドがなければならない。

◆ main()メソッドには、「public static void」という修飾子がつく（今の段階では、そういう約束だと思っておいてください)。

さて、最後にmain()メソッドの中身を見てみましょう。命令らしきものとして、

```
System.out.println("Hello Java");
```

がありますが、この命令文には次の４つの文法事項が含まれています。

① System.out　：標準出力（次ページの「Ⅲ．プログラムの実行」で用いるコマンドプロンプトの黒い画面のことです)。

② println( )　:(　)内を表示するという命令。

③ ""で囲まれた部分は文字列となる。

④ 一つの文（命令）の終わりには区切り記号「；」をつける。

したがって、上の命令文は

標準出力に対して" "内の文字列を出力(表示)せよ。

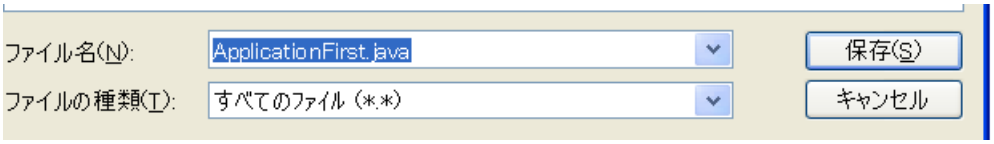
という指示を送った事になります。ですから、プログラムを実行すると画面に「Hello Java!」という文字が表示されるはずです。

Ⅱ．プログラムの保存

それでは、プログラムを保存しましょう。このプログラムは「ApplicationFirst.java」という名前で保存する必要があります。と言うのは、Java では、main メソッドを持つクラスの名前をプログラムのファイル名とする事になっているからです。

保存場所は、どこでも良いのですが、後の説明の都合上、「ローカルディスク（C:)」→「ユーザー」→「S12･･･」と選択してください（最後の「S12･･･」の部分はログインした時のユーザ名で 2012 年度入学生の場合は、学籍番号になっているはずです。もし別のユーザ名でログインした場合は、自分が使用しているユーザ名のフォルダに読み替えて下さい)。保存は、エディターのメニューから［ファイル］→［名前を付けて保存］を選び、上記の保存場所に「ApplicationFirst.java」という名前で保存してください。このとき、ファイ

ルの種類をテキストとしたままだと、「.txt」が自動的につけられることがあります。そこで、次のように、ファイルの種類を「すべてのファイル」としてからファイル名を記入するようにしましょう。

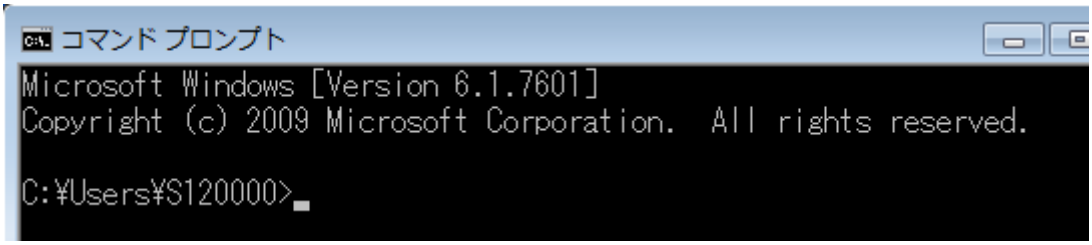


Ⅲ. プログラムの実行

プログラムを実行しましょう。Java 言語プログラムの実行には

ソースコード（*.java）の翻訳（コンパイル）　→　バイトコードの生成
→　バイトコードの実行

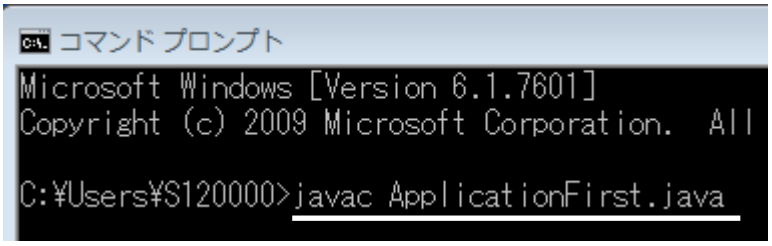
という手順を踏まねばなりません。そのためには、翻訳あるいは実行する命令をコマンド形式で入力しなければなりません。その命令を入力するためには、「コマンドプロンプト」画面を用います。Windows の「スタート」メニューから［スタート］→［すべてのプログラム］→［アクセサリ］→［コマンドプロンプト］を選択してください。すると、次のような黒いウィンドウが現れるはずです。



上の例で言うと「C:¥Users¥S120000>」の部分を**プロンプト**と言います（最後の「S12・・・」の部分は PC にログインした時のユーザ名になっています）。プロンプトは、現在のディレクトリ、つまり作業を行うフォルダを表しています。コマンド入力を行う際にはプロンプトの右側に入力します。次の第一段階に進んで下さい。

★　第一段階：ソースコードの翻訳（クラスファイルの生成）

続いて、次のようにコマンド（白線部）を入力してください。ここで、入力後［Enter］キーを押してください（白線部が入力）。コマンド入力の際は、コマンドを記述した後、［Enter］キーを押すことで初めてコマンドがシステム側に受け付けられます。



もし、正常に翻訳されれば、カーソルが次のプロンプトに移ります。何らかのエラーメッセージが表示された場合は、ソースコードのタイプミスが原因と思われます。エディターを用いて、もう一度タイプミスがないか確認し、修正後再び保存した上で、上の「javac」命令を実行して下さい。

実は、javac 命令により、「*.java」というソースコードから「*.class」というクラスファイルが生成されます。これが前章で説明した**バイトコード**です。

★　第二段階：クラスファイルの実行

さて、上の第一段階により、「ApplicationFirst.class」というクラスファイル（バイトコード）が生成されました。これを実行するには、次のように「**java**」命令を用います。白線部を入力して下さい。この場合、下のように「.class」をつける必要がないことに注意してください。コマンド入力後［Enter］キーを押すと、「Hello Java!」と表示されます。これが今のプログラムの実行結果です。

ここに、「java」命令はクラスファイルの内容を解読し、実行するという機能を持っています。バイトコードを解釈しながら実行するので、**インタプリタ**（interpreter：解釈者の意味）と呼ばれます。そしてすでに気づいた人もいると思いますが、このインタプリタ「java」は、前章で説明した JVM（Java 仮想マシン）に当たります。もっとも、Java 仮想マシンはもっと広い概念で、このインタプリタ以外にも種類があります。しかし、これ以上の理解はここでは必要ないのでこれ以上はふれません。

さて、以上が、通常の Java 言語アプリケーション開発の過程です。改めて行程を振り返ってみると、

- ◆ エディターを用いてのソースコードの作成・保存
- ◆ コマンドプロンプトを用いてのソースコードの翻訳・実行（**2段階**）

という手順を踏まねばならないことが分かります。

一方 Eclipse では、アプリケーションブラウザ内でそれを全て作業できるようになっています。このような意味で、Eclipse のような開発ツールを**統合開発環境**と呼びます。さらに、Eclipse における「実行」ボタン  は上の２段階の命令を一度に行ってくれていたことが分かるでしょう。このように、作業を効率化あるいは高速化できるという意味で **RAD（Rapid Application Development）環境**とも呼ばれます。これは本節の冒頭で述べたことです。実は、Eclipse にはこの他に、

① 以下のような、定型部分（変更の必要のない、決まり切った部分）を自動生成してくれる。

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


}
```

② Windows アプリケーション作成の際に不可欠になるフレームのデザインを(フレーム設計画面により)視覚的に行うことができる。

などの、作業の効率化に関する各種の機能が用意されています。それらの詳細については、次章以降の学習で実際に体験することにしましょう。

それでは、コマンドプロンプトを閉じてください。コマンドプロンプト・ウィンドウ右上隅の×をクリックすれば閉じます。