

コンパイラ演習資料 (javacc)

担当：佐々木晃

□演習問題 B0 (締め切り = 11/17 演習中の回収時)

問題番号: B0

課題名：コンパイラの作成準備

出題日：11/12

課題：JavaCC の設定と準備

本演習資料にしたがって、中置記法による算術式（四則演算、括弧が使える）を逆ポーランドによる式に変換するプログラム

を JavaCC を用いて作成せよ。

備考：コンパイラ作成演習のガイドラインに従って、アップロードしてください。ssh の設定などが必要です。

(参考、コンパイラ作成 1)

<http://cis.k.hosei.ac.jp/~asasaki/lect/compiler/2008A/problem/problem1.htm>

0. javacc について。

詳しくは、中田先生のページを参照

<http://cis.k.hosei.ac.jp/~nakata/lectureCompiler/JavaCC/index.html>

その他、<http://cis.k.hosei.ac.jp/~asasaki/lect/compiler/>の JavaCC の項目を参考にせよ。

1. javacc のインストール (Eclipse-plugin を利用)

説明は下記にある。

Eclipse JavaCC Plugin Home

<http://eclipse-javacc.sourceforge.net/>

下記からダウンロード。

SourceForge.net: JavaCC Eclipse Plugin

<http://sourceforge.net/projects/eclipse-javacc>

SourceForge.net > Find Software > JavaCC Eclipse Plugin



JavaCC Eclipse Plugin by koutch, mmazas, yarrumretep

Summary | Files | Support | Develop



View screenshots

The JavaCC Eclipse Plugin provides a JavaCC Editor and builder which processes .jj .jdt .jtb files and integrates with Eclipse's incremental build system. Compatible with Eclipse 3.3, 3.4, 3.5 Galileo with Java 1.5.

Download Now!
sf.eclipse.javacc-1.5.16-... (913.9 KB)

OR [View all files >](#)

←Eclipseのバージョンが3.3以降であれば、こちら

←Eclipseのバージョンが3.3以降であれば、こちら
合、古いバージョンでは動作しない場合がある。こちらが正しいバージョンをダウンロードさせる

Show project details

(古いバージョンの場合。)

File/Folder Name	Platform	Size	Date ↓	Downloads	Notes/Subscribe
Newest Files					
sf.eclipse.javacc-1.5.16-updatesite.zip		913.9 KB	2009-10-04 12:42:04 GMT	644	
All Files					
sf.eclipse.javacc-1.5.16-updatesite.zip		913.9 KB	2009-10-04 12:42:04 GMT	644	
eclipse-javacc			2009-05-24 15:36:01 GMT		

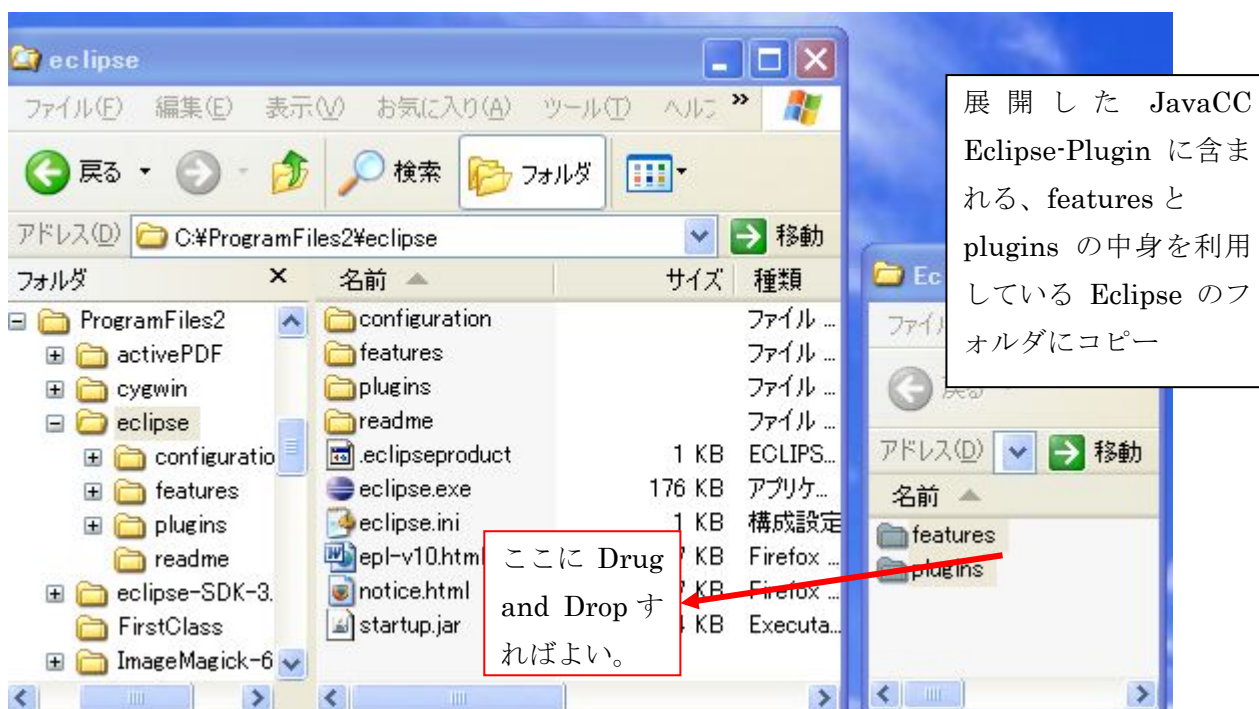
このフォルダをたどると、zipファイルが見つかる

インストール。

アーカイブを展開（解凍）した中には、features と plugins というフォルダが含まれる。

この中身を利用している Eclipse のルートフォルダにある features, plugins のそれぞれに入れてやれば

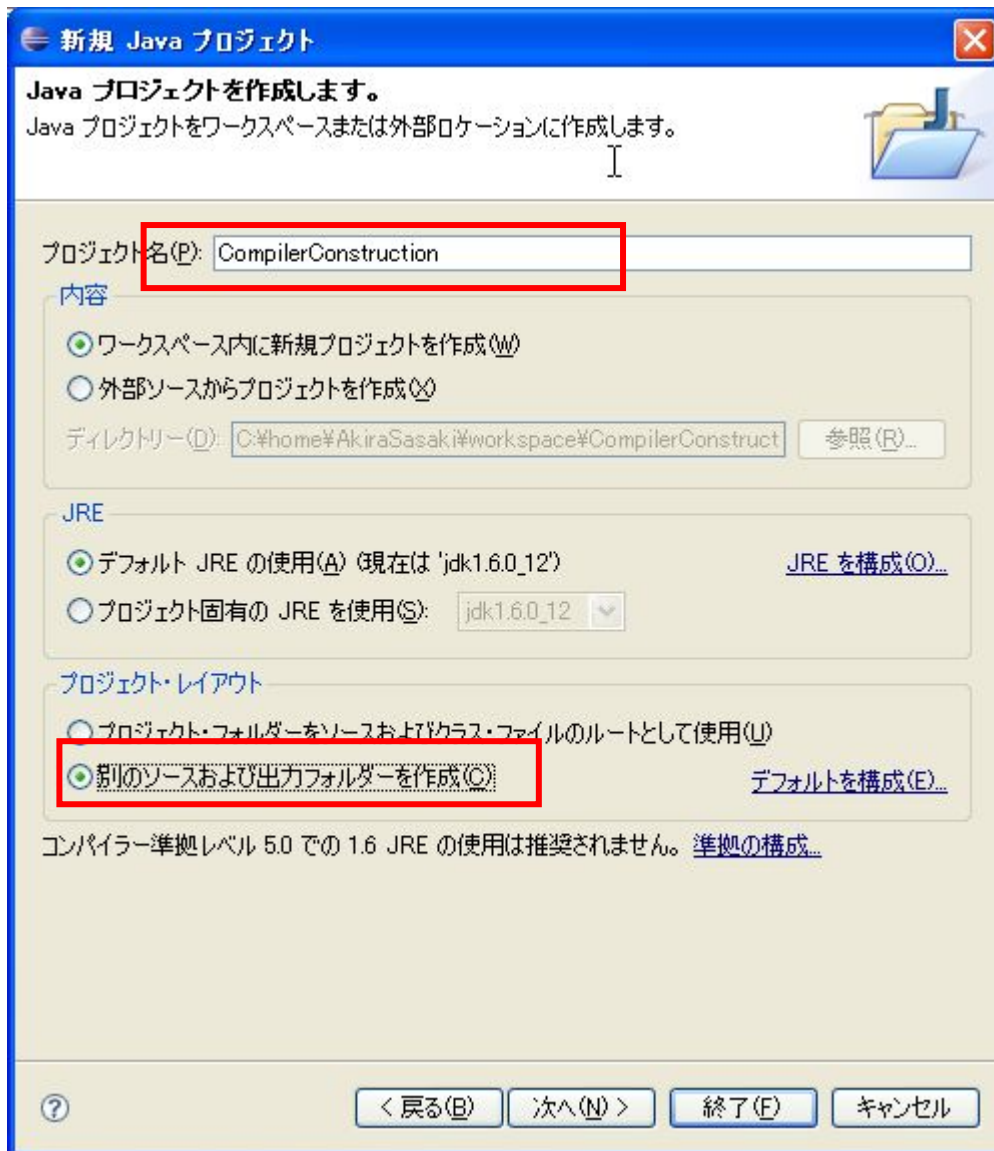
よい。



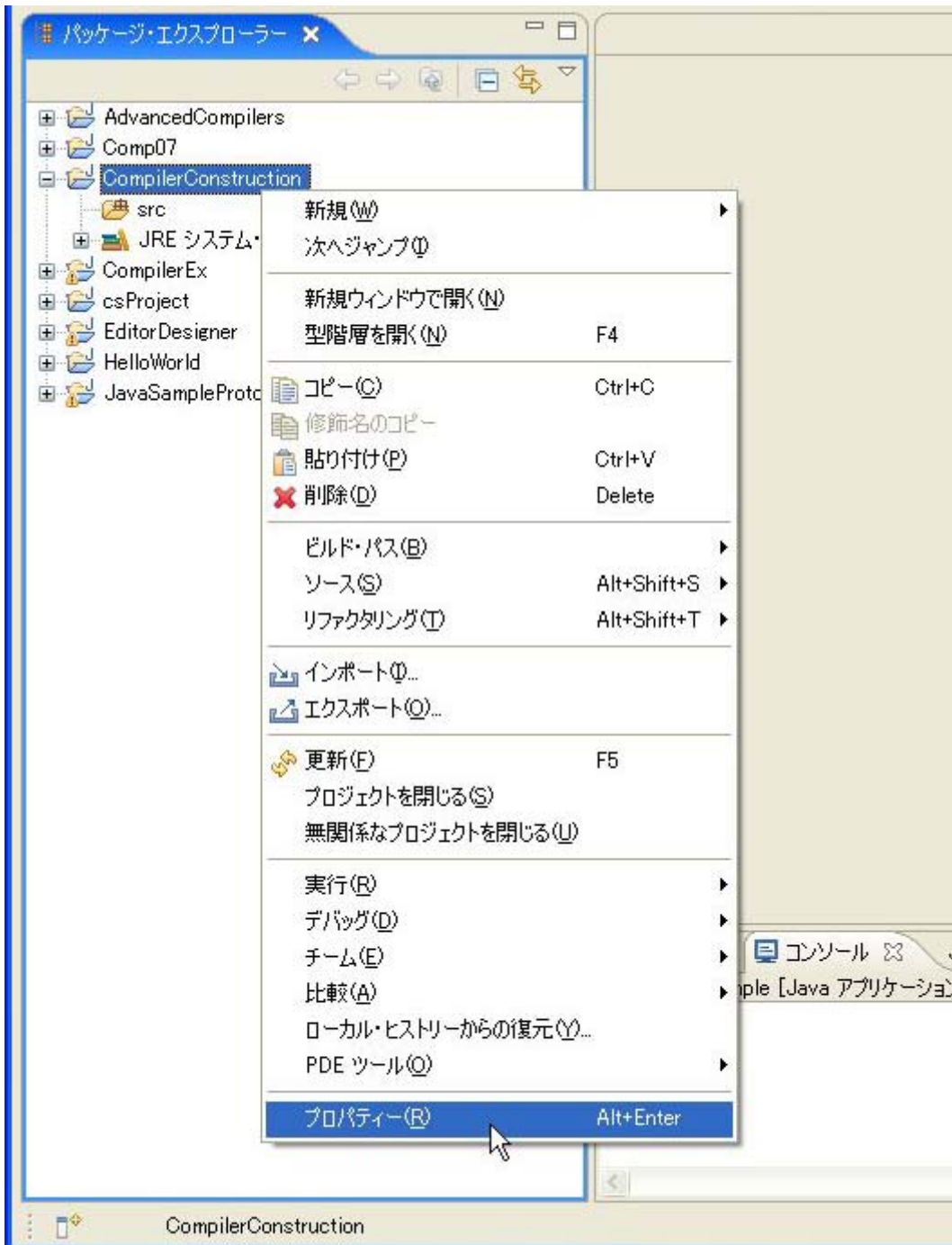
以上で終了

2. プロジェクトと JavaCC オプションの設定

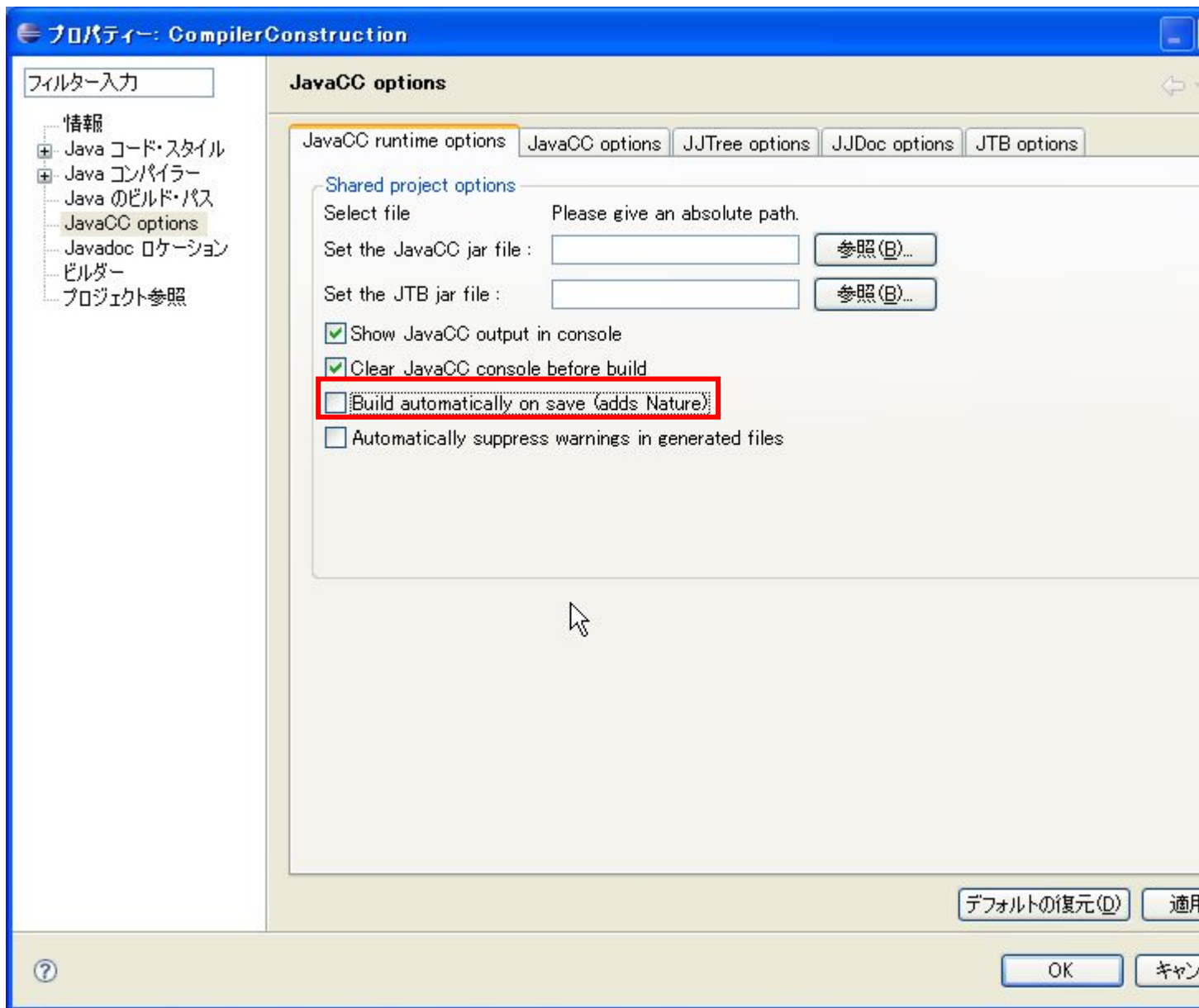
ここでは CompilerConstruction というプロジェクトを設定。通常と同じく Java プロジェクトを指定。



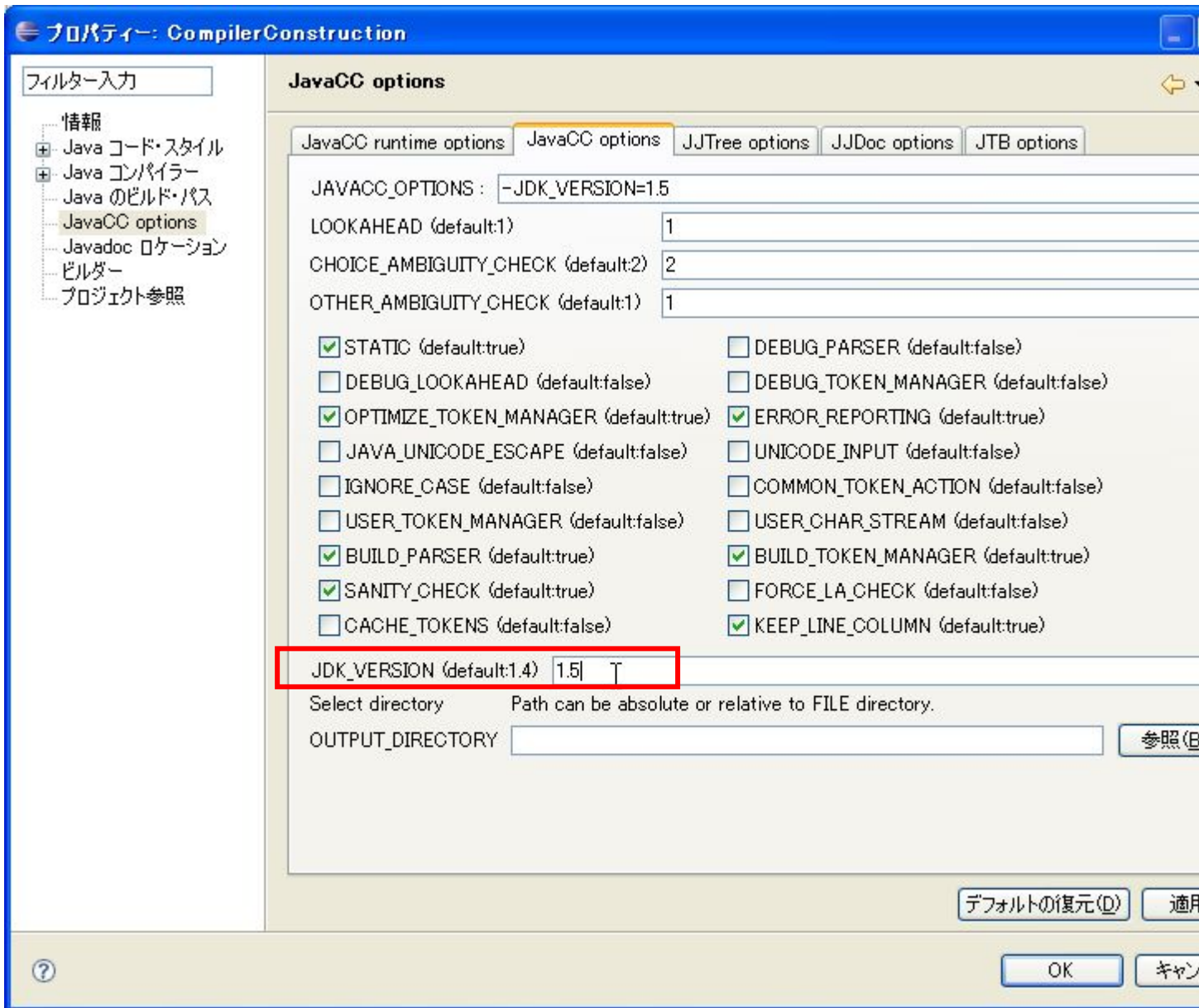
JavaCC のオプションの設定。



自動セーブ機能をOFF(好みで)



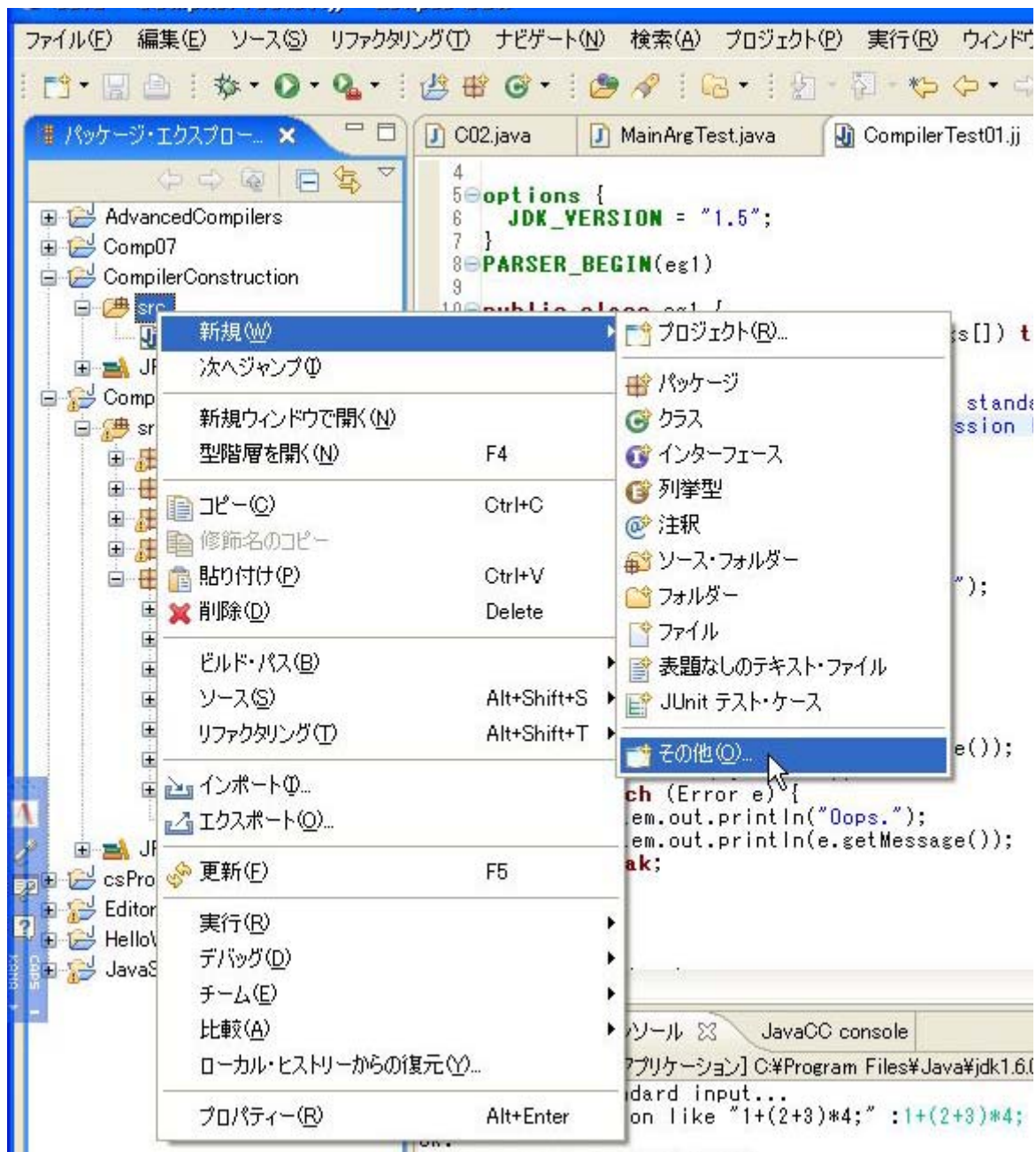
Java のバージョンを 1.5 (必須)

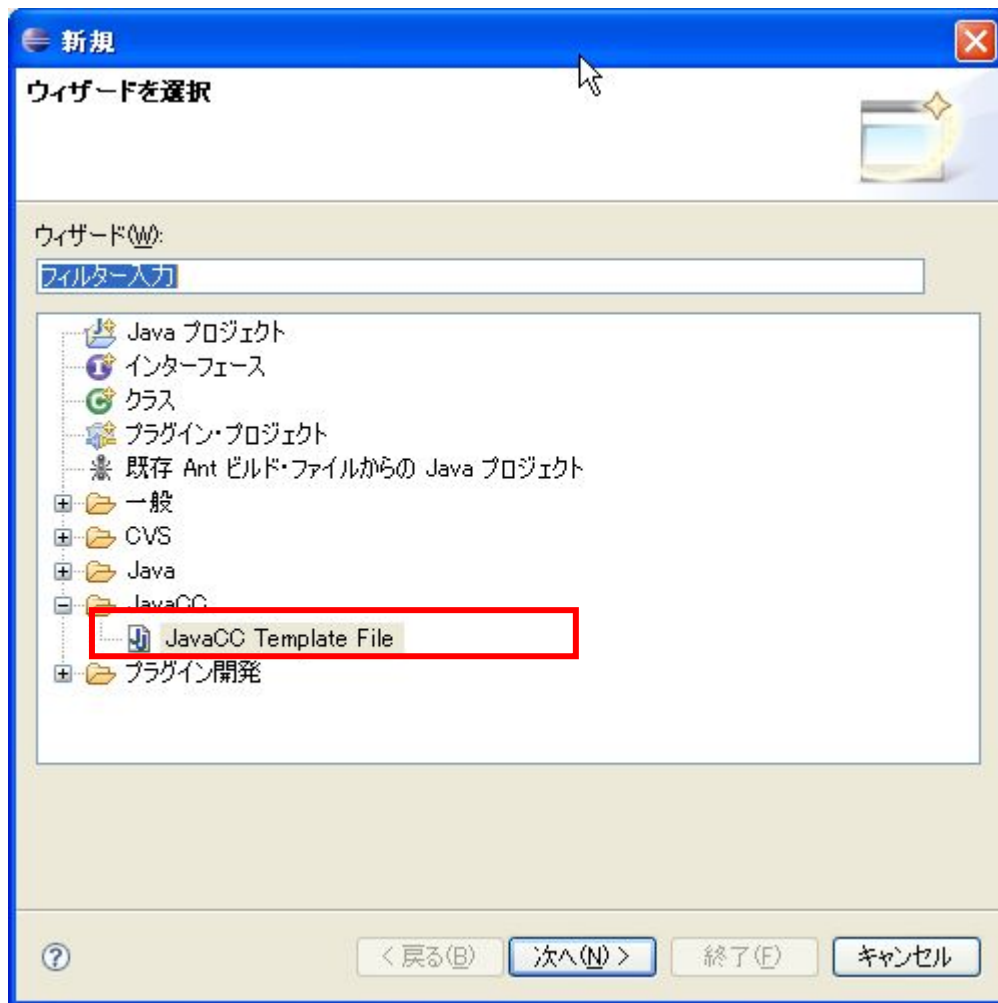


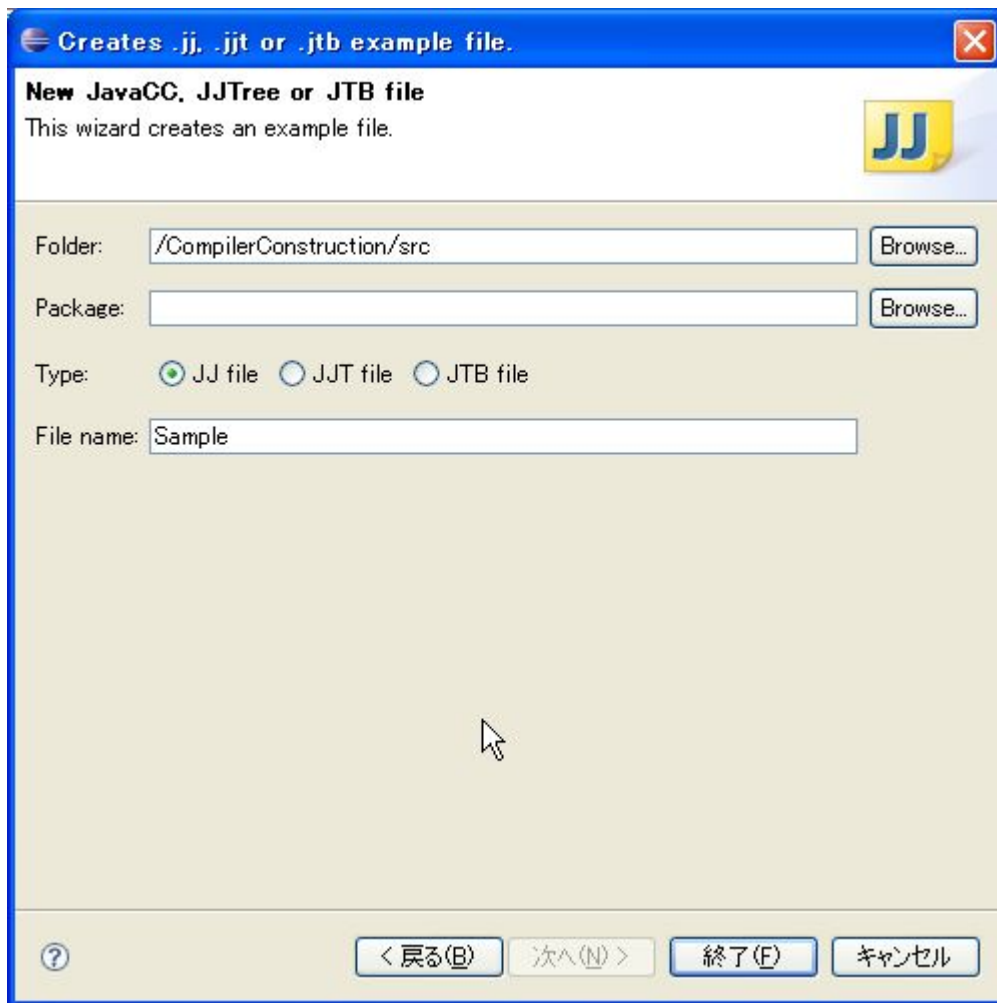
3. コンパイラ（構文解析器）の作成の流れ

準備：テンプレート作成

（ただし、ここでは実際には作成されたテンプレートは全く使わない）





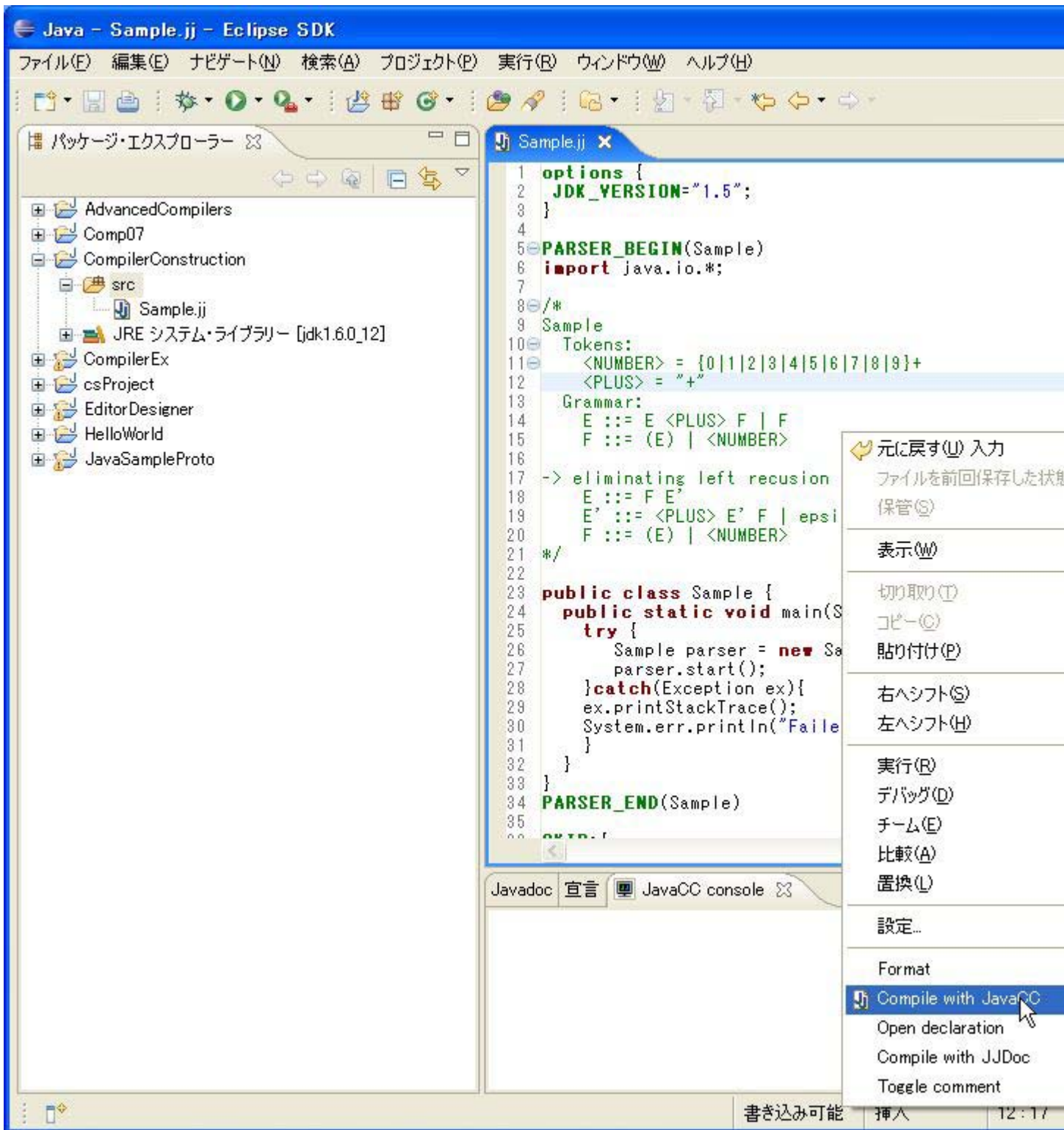


A: JJ ファイルを編集 :

ここでは、テンプレートを完全に書き換えてプログラムを書く。

(本資料の「5 : 開発の流れ」のサンプルプログラム、あるいは配布プログラムの `Sample.jj` の内容で説明する。)

B: コンパイラの自動生成 : JavaCC でコンパイラ (構文解析器) を自動生成する。



C:生成されたコンパイラ（構文解析器）のコンパイル：

実際には、Sample.jj にはバグがあるので、自動生成の段階でエラーがでるので、直さないと先には進めない。(後述)

Sample.jj のバグを取り除いたあと、Sample.java（自動生成されたコンパイラ）をコンパイルする。下

記のようなエラーになればコンパイルは一応成功している。(コンパイルすべきプログラムが main の引数に与えられていないために出ているエラーである。)

The screenshot shows the Eclipse IDE interface. On the left, the Package Explorer displays a project structure with a folder named 'src' containing several Java files. The file 'Sample.java' is highlighted with a red box. A black arrow points from this box to a text box containing the following text:

Sample.jj から自動生成されたコンパイラ(構文解析器)の Java ファイル。うまく、自動生成できないときは、refresh する意味で、以前に生成されたものを削除すると良い。

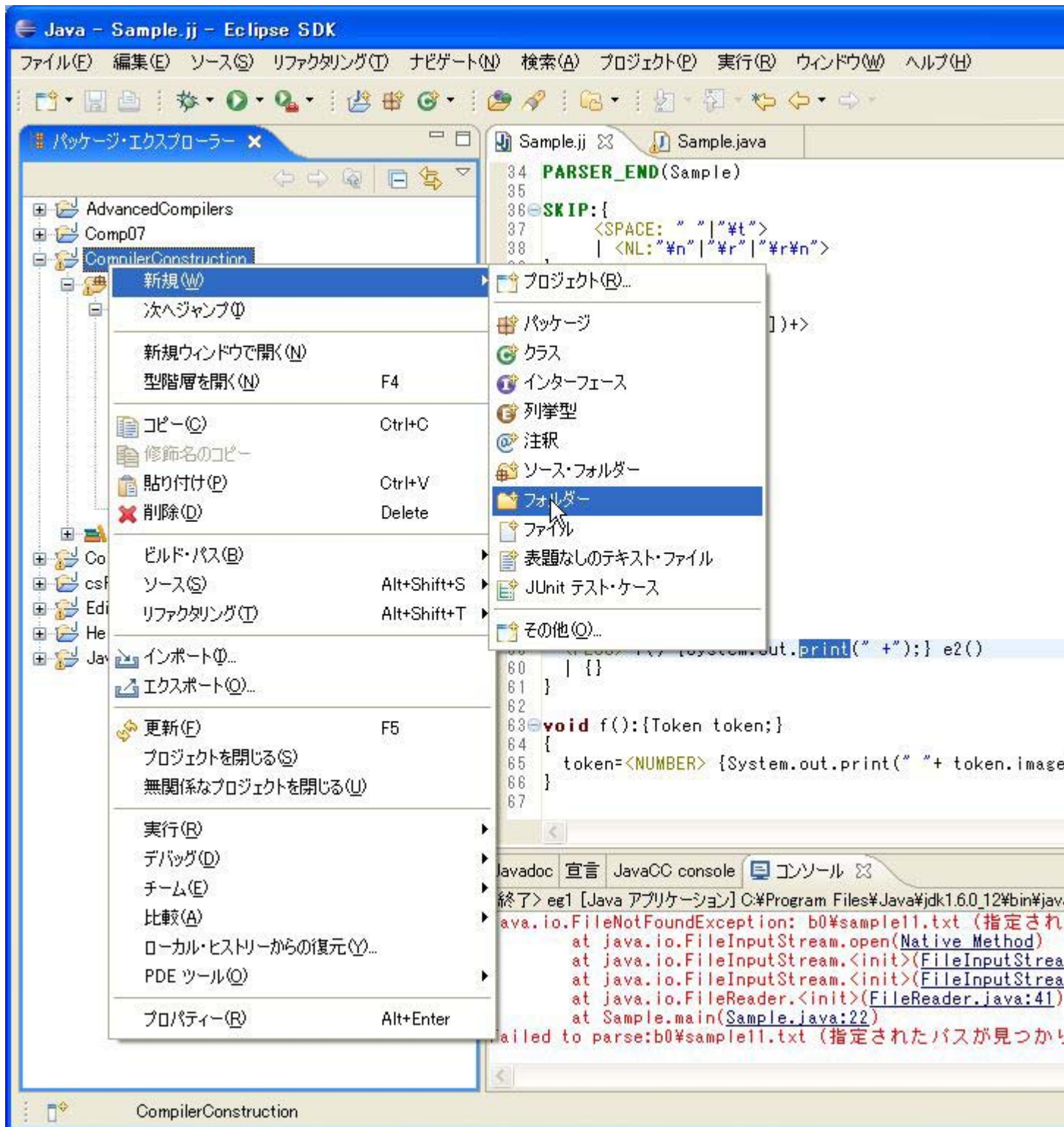
On the right, the editor shows the source code of 'Sample.java'. The code includes a main method that attempts to parse an input string. At the bottom, the Console window shows the following error message:

```
<終了> Sample [Java アプリケーション] C:\Program Files\Java\jdk1.6.0_12\bin\java.lang.ArrayIndexOutOfBoundsException: 0
    at Sample.main(Sample.java:22)
Failed to parse:0
```

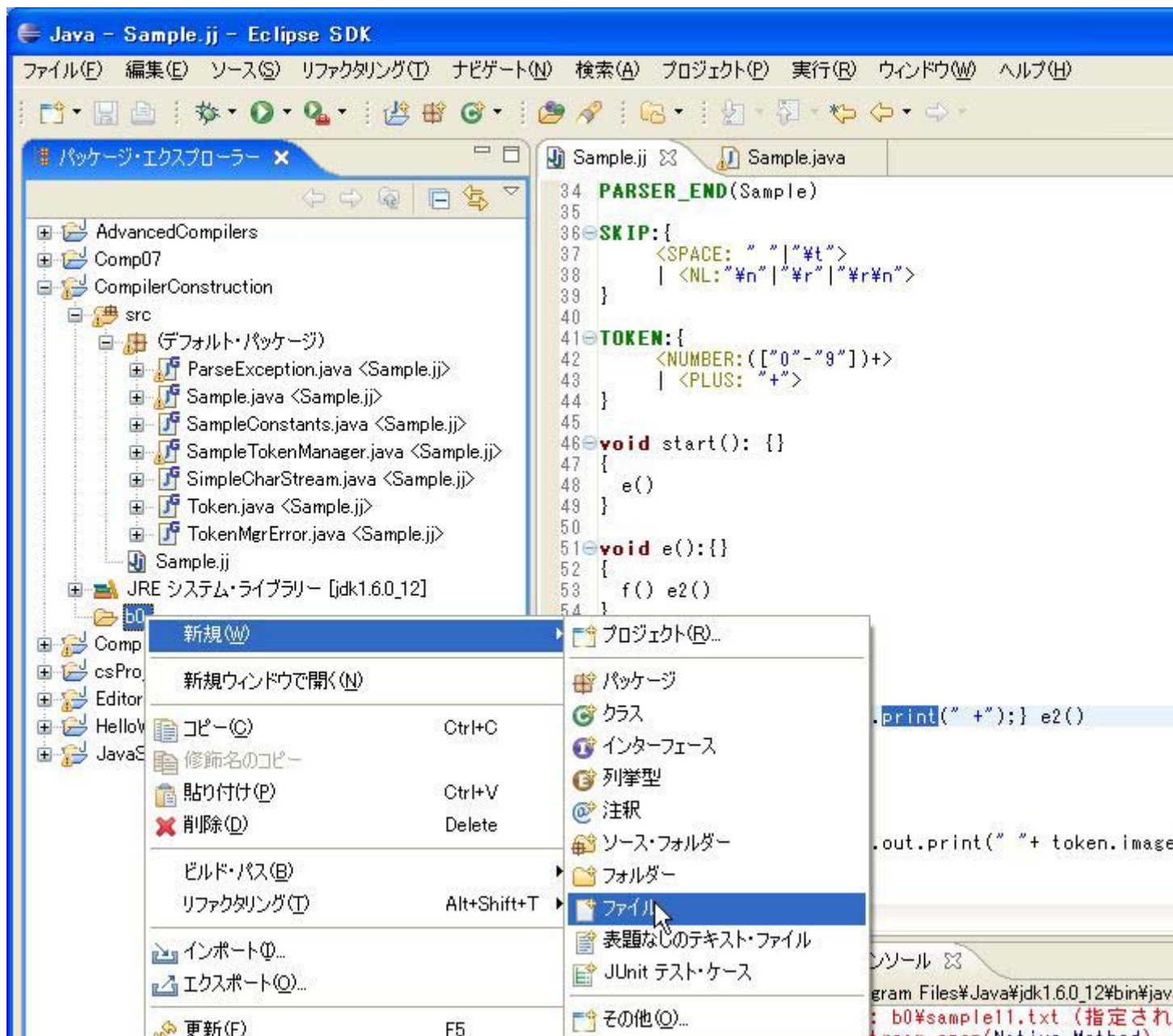
D(1):ソースコードの作成

作成したコンパイラに読み込ませるソースコードを用意する。

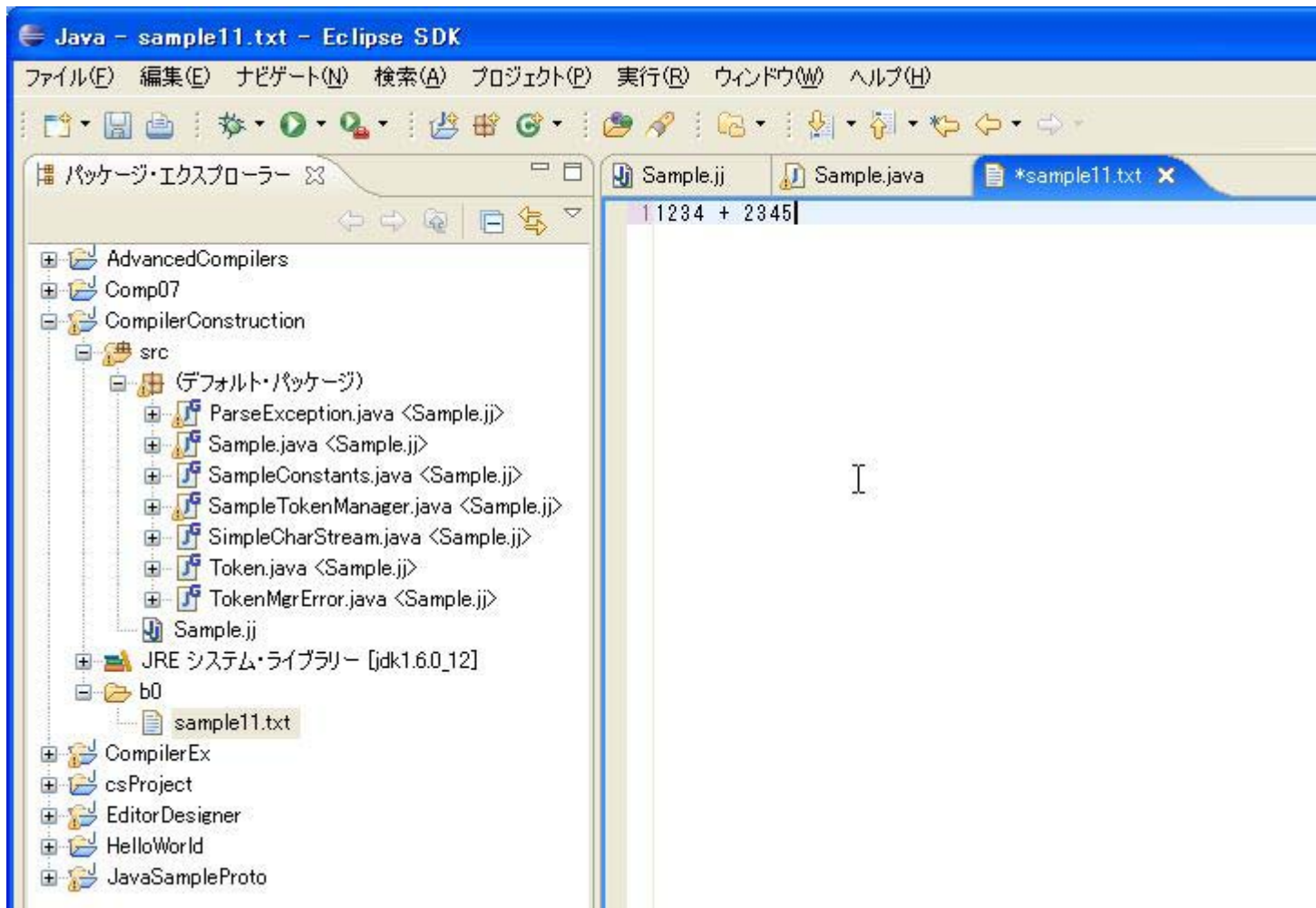
ここでは b0 というフォルダを用意して、sample1.txt にソースコードを書く。



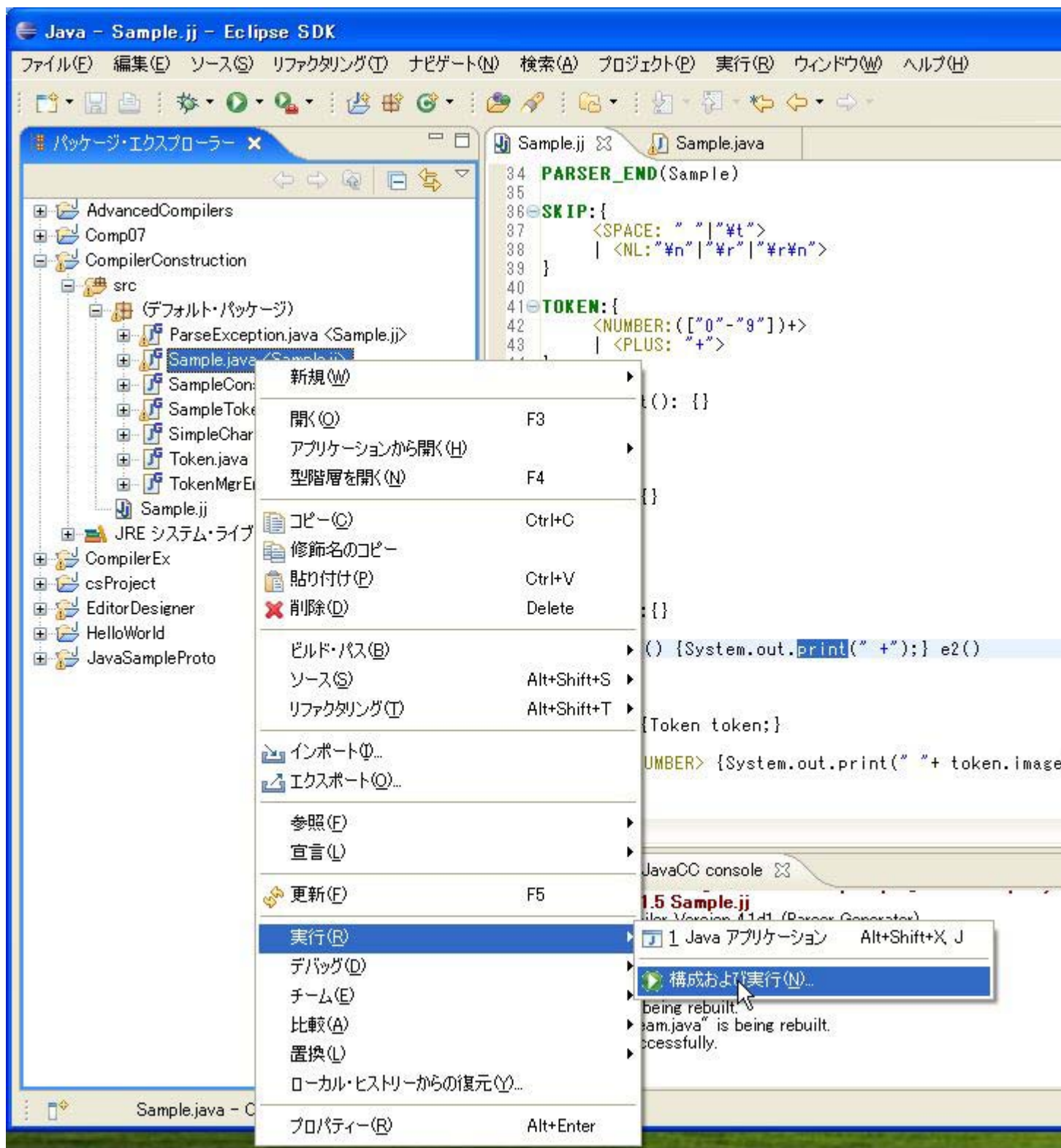


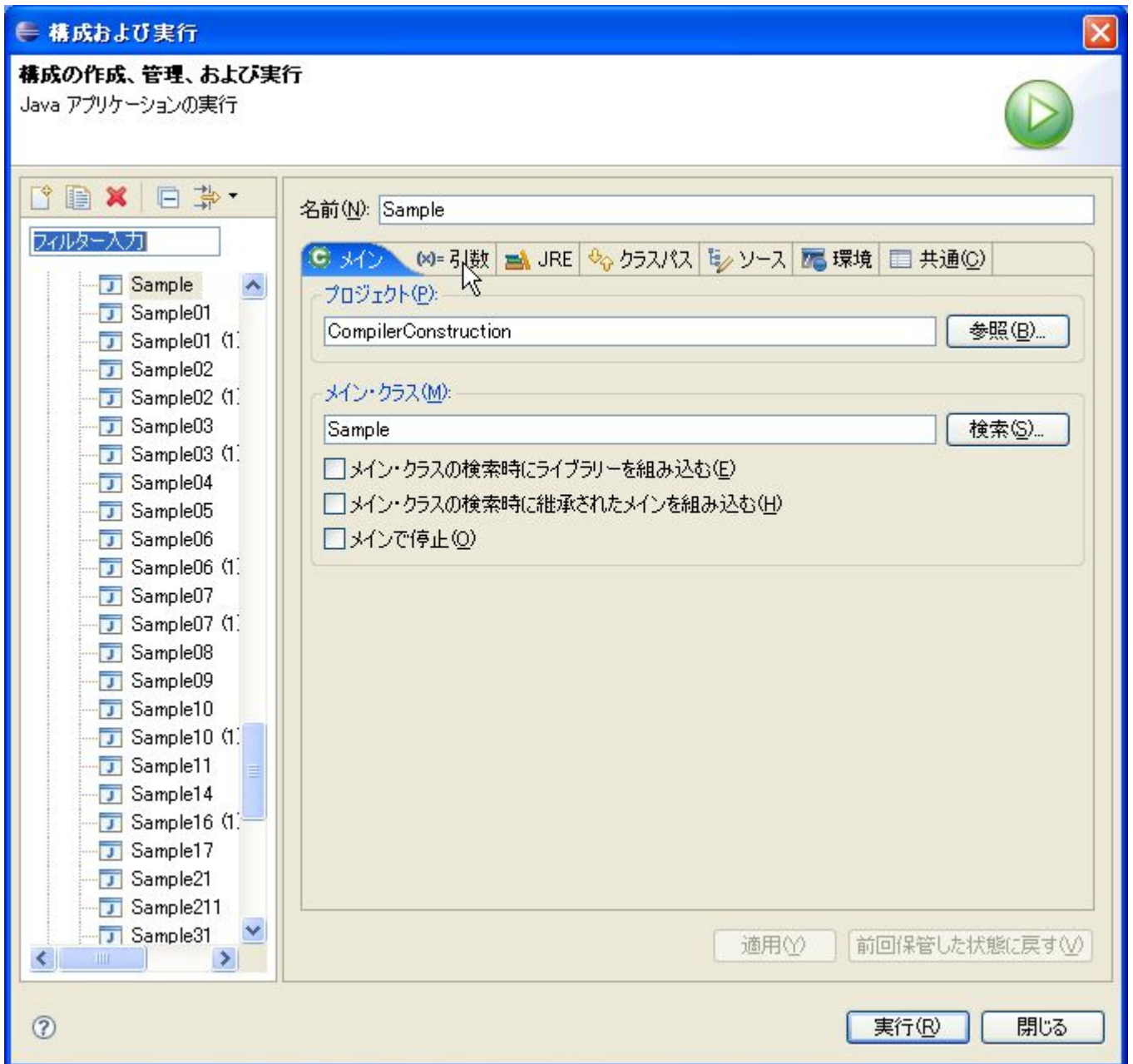




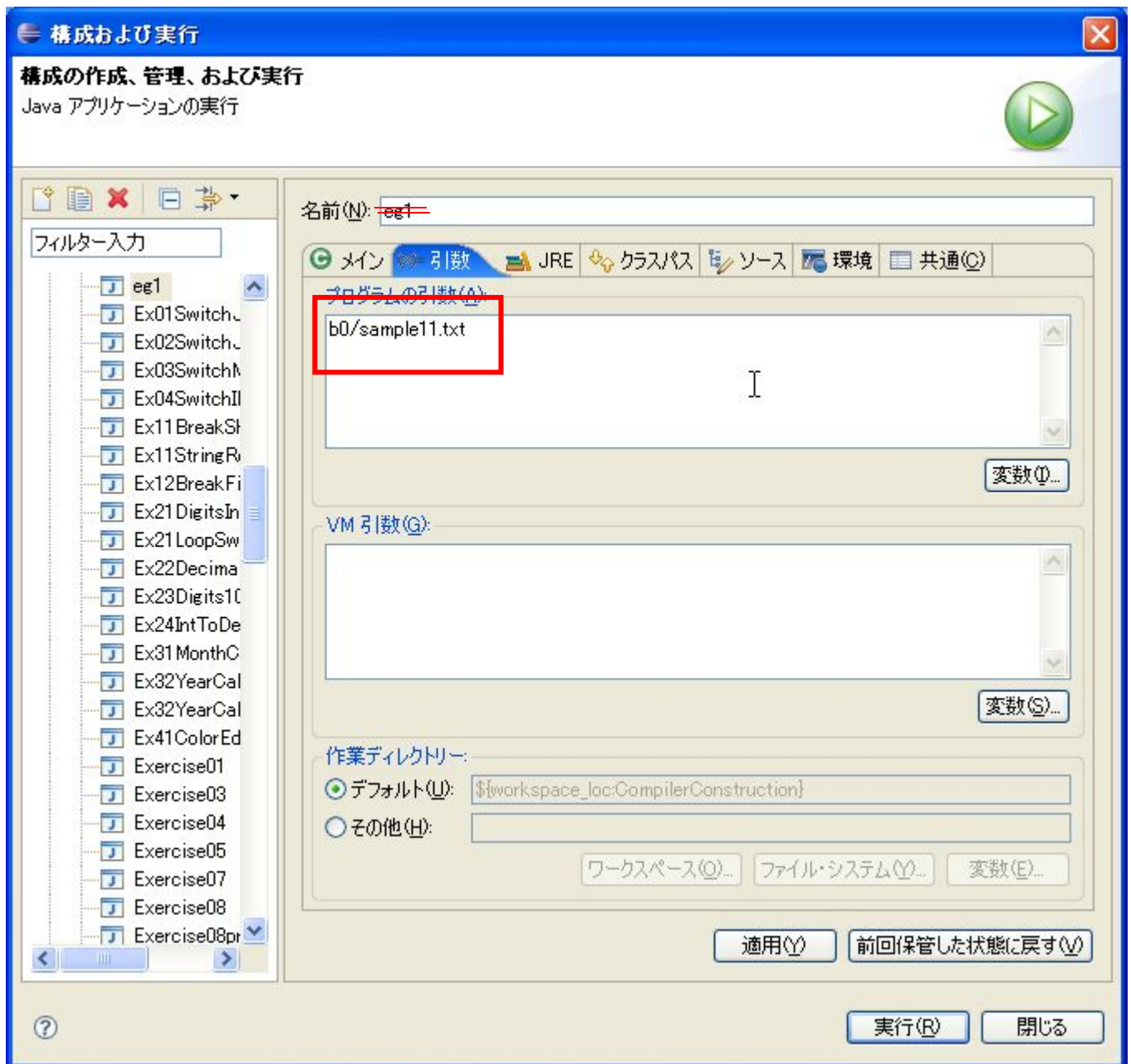


D(2) 作成したコンパイラを実際に動かす。





引数にコンパイルしたいファイル名を書く。(さきほどの 1234 +2345 をコンパイルする。)



コンソールに下記が表示されたら成功。

```
1234 2345 +
```

4. コンパイラ作成の流れ

- jj ファイルを編集、作成。(上の例では Sample.jj)
- コンパイラの生成 (jjファイルに誤りがあると、先には進まないのでAにもどる。)
成功すると Sample.java が作られる。
- Sample.java のコンパイルと実行

Sample.java のコンパイルに成功すると、コンパイラが起動する。

ただし、Sample.jjのアクションの部分などに誤りがあった場合、Sample.javaをコンパイルすることはできないので、**Aに戻る。**

D. 実際にコンパイラの動きを試す。動きがおかしいときは**Aに戻る。**

5. 開発の例 :

□version 1

○4A. サンプルプログラム Sample.jj (わざと誤りが入れている) の作成

```
options {
  JDK_VERSION="1.5";
}

PARSER_BEGIN(Sample)
import java.io.*;
/*
Sample
  Tokens:
    <NUMBER> = {0|1|2|3|4|5|6|7|8|9}+
    <PLUS> = "+"
  Grammar:
    E ::= E <PLUS> F | F
    F ::= (E) | <NUMBER>

-> eliminating left recursion of the grammar
    E ::= F E'
    E' ::= <PLUS> E' F | epsilon
    F ::= (E) | <NUMBER>
*/

public class Sample {
  public static void main(String args[]) {
    try {
      Sample parser = new Sample( new FileReader(args[0] ));
      parser.start();
    }catch(Exception ex){
      ex.printStackTrace();
      System.err.println("Failed to parse:"+ex.getMessage());
    }
  }
}
```

```

    }
  }
}
PARSER_END(Sample)

SKIP:{
  <SPACE: " "|"¥t">
  | <NL:"¥n"|"¥r"|"¥r¥n">
}

TOKEN:{
  <NUMBER:("[0"- "9"])+>
  <PLUS: "+">
}

void start(): {}
{
  e()
}

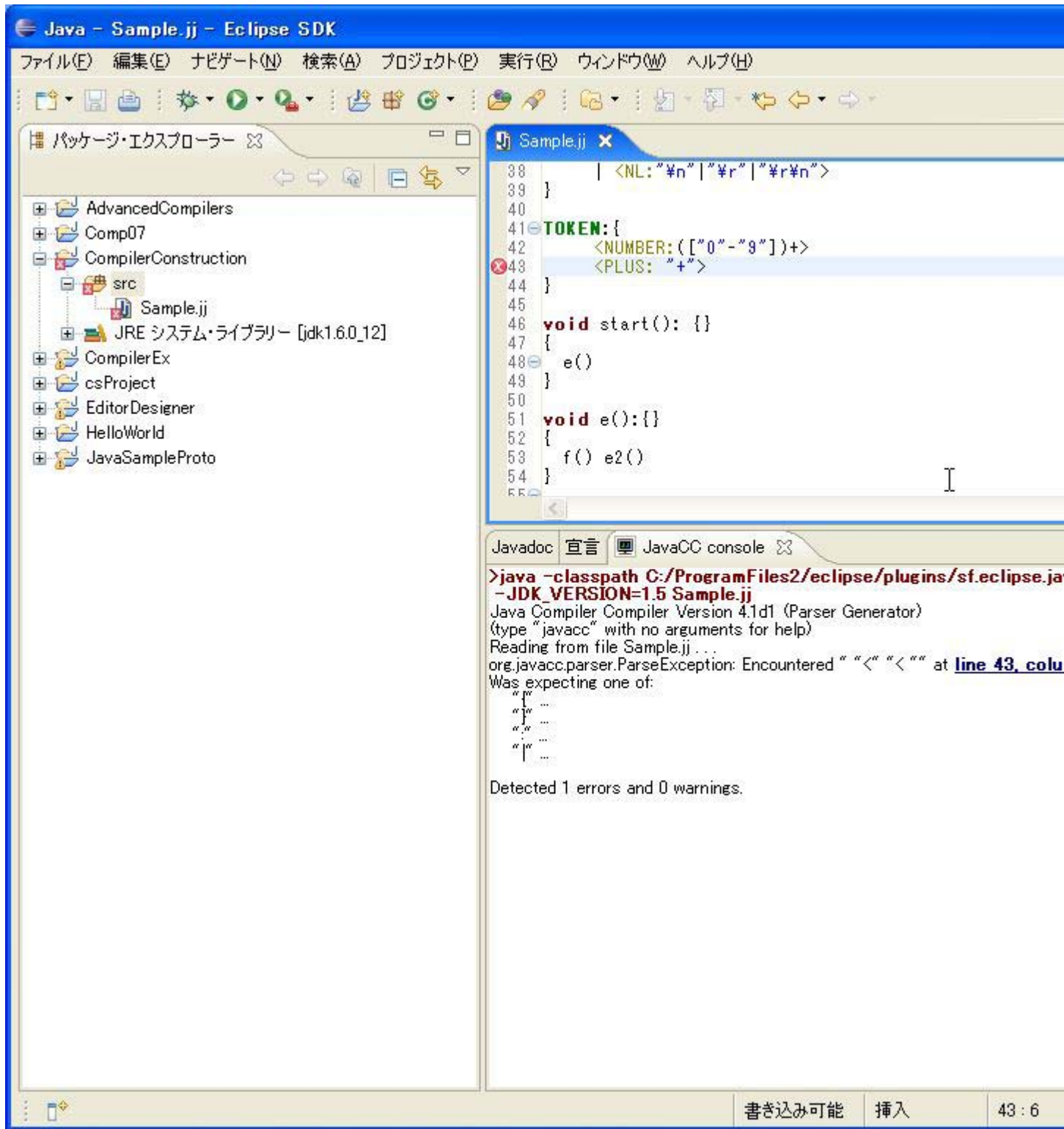
void e():{}
{
  f() e2()
}

void e2():{}
{
  <PLUS> f() {System.out.pritn(" +");} e2() /* 誤り */
  | {}
}

void f():{Token token;}
{
  token=<NUMBER> {System.out.print(" "+ token.image);}
}

```

→jj ファイルの記述誤りがあるのでエラーがでる。

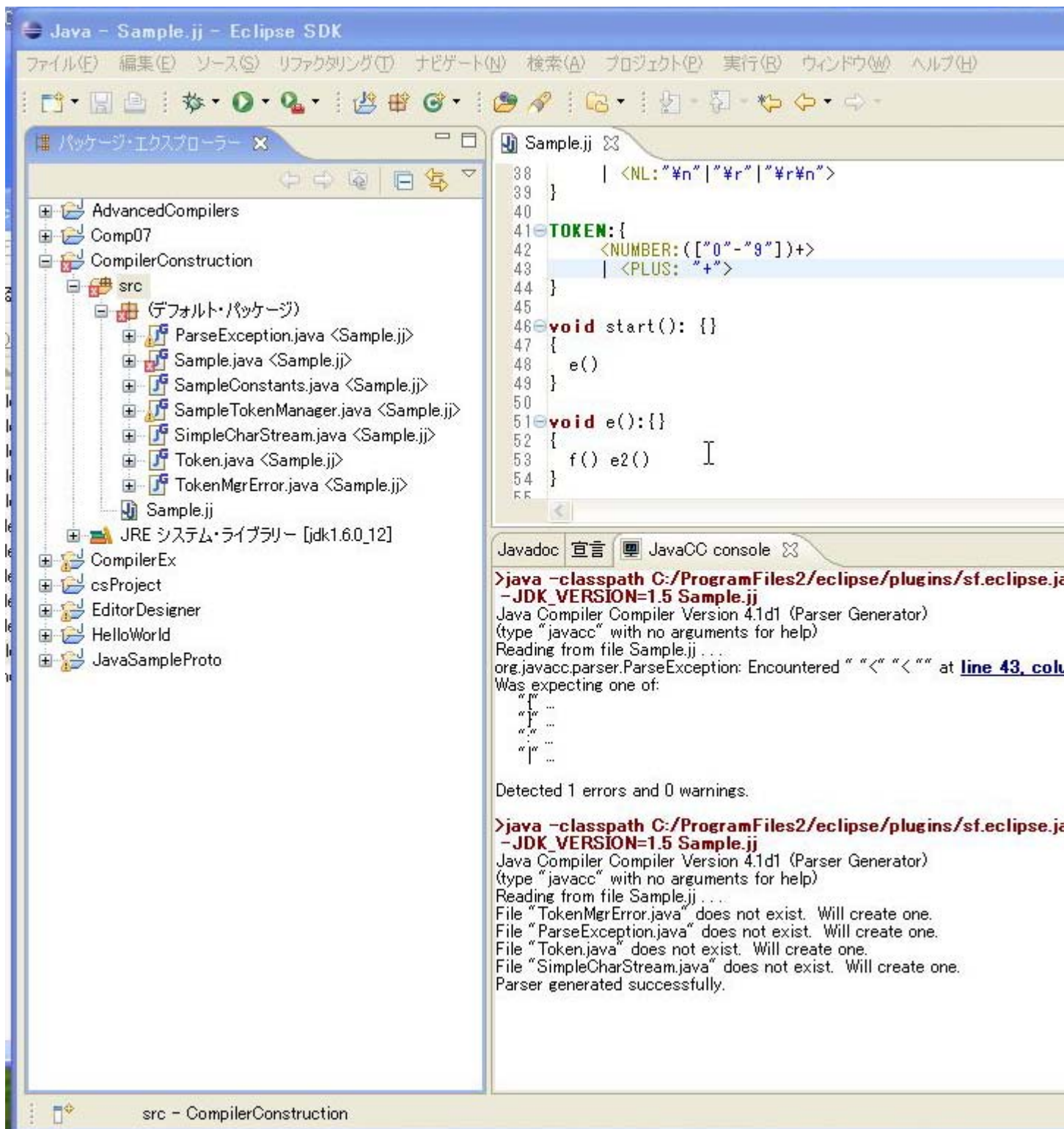


□version 2

○4A→jj ファイルを修正し Version 2 を作る。

「|」を加えると直る。上記 43 行目の `<Plus>`の前。

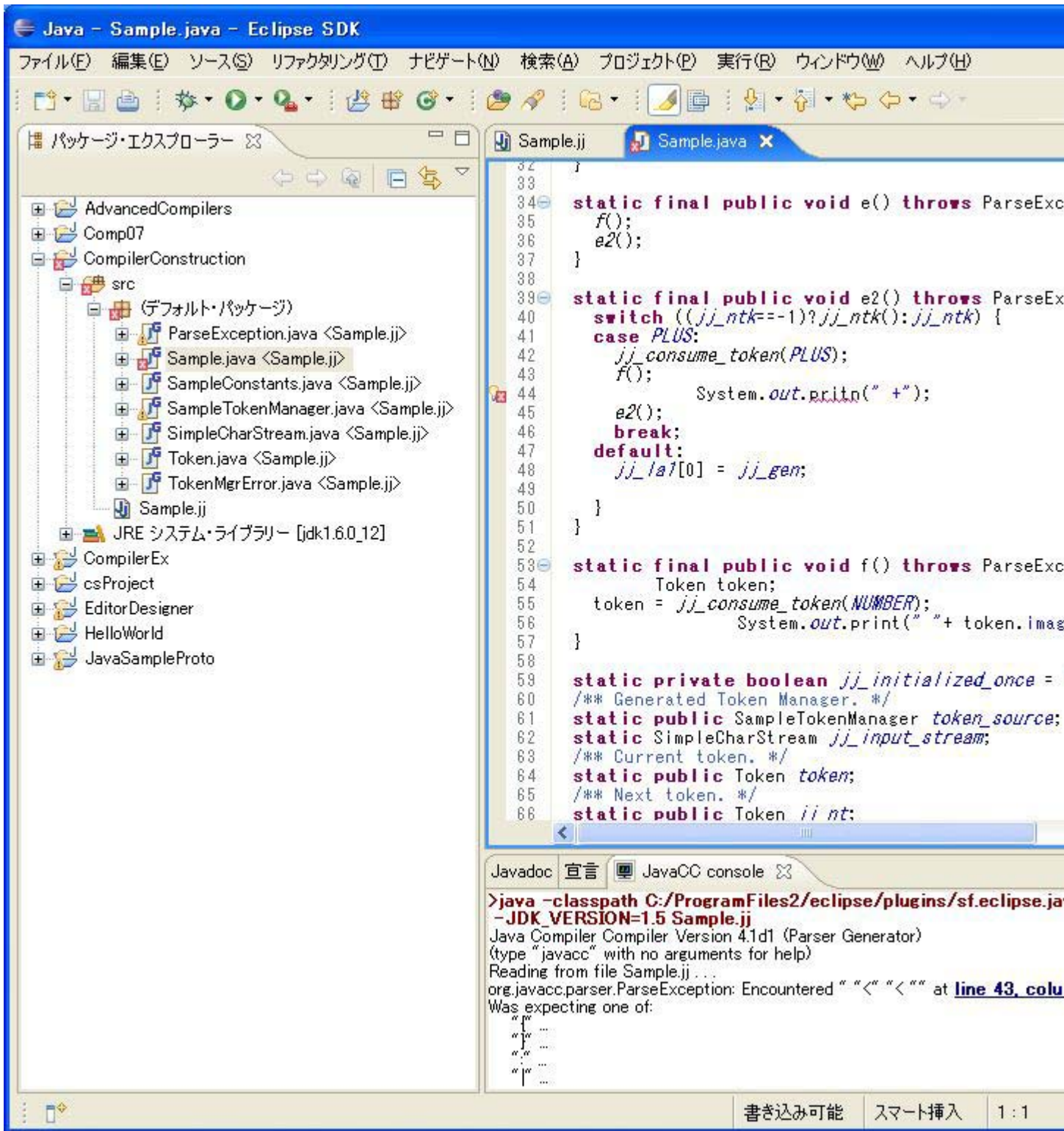
○4B:コンパイラの生成→成功



4C 生成されたコンパイラのコンパイルと実行

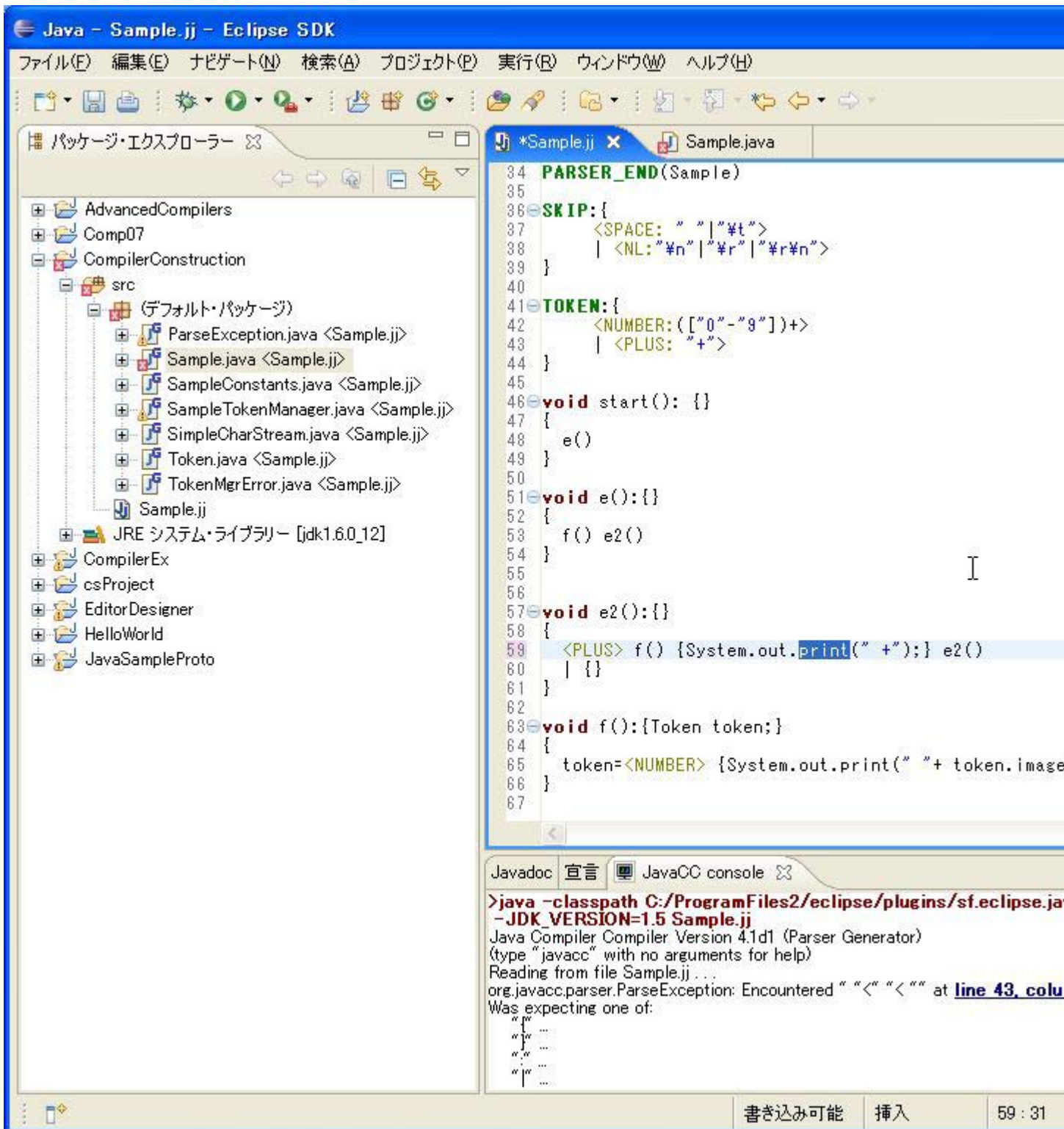
→ Sample.java に赤×印がついているが、生成された Java ファイルに誤りがある。print のつづりあや

まり。もとの jj ファイルを直す必要がある。



□version 3

○4A→jj ファイルを修正



○4B: コンパイラの自動生成→成功

→成功

○4C、4D 生成されたコンパイラのコンパイルと実行

「3.コンパイラ（構文解析器）の作成の流れ」のDの項に書いた方法で、ソースプログラムを用意した後、実際に作成したコンパイラを動かしてみる。

サンプルテキスト、sample1.txt

```
1234+2345
```

実行結果(コンソールに出力される)

```
1234 2345 +
```

6. 配布したサンプル

配布したサンプルに含まれる javaccsamples 以下の jj ファイルおよび txt ファイルの説明

□字句解析のテスト→ScannerSample1.jj

このテストは、標準入力(System.in)から入力するので、実行する際には引数は必要なし。

実行するとコンソールで入力待ちになるので、テキストを入力する。(Enter を押さないと入力は処理されない。)

□算術式を逆ポーランドに翻訳するプログラム

- Sample.jj 上述のエラーのある例。修正したものが Sample1.jj
- Sample1.jj 足し算その1
サンプルソースプログラムは、sample11.txt, sample12.txt
- Sample2.jj 足し算その2
サンプルソースプログラムは、sample11.txt, sample12.txt
- Sample3.jj 括弧を扱えるようにする。
サンプルソースプログラムは、sample11.txt, sample31.txt
- Sample4.jj 引き算を扱えるようにする。
サンプルソースプログラムは、sample11.txt, sample41.txt
- Sample5.jj 掛け算を扱えるようにする。
サンプルソースプログラムは、sample11.txt, sample51.txt