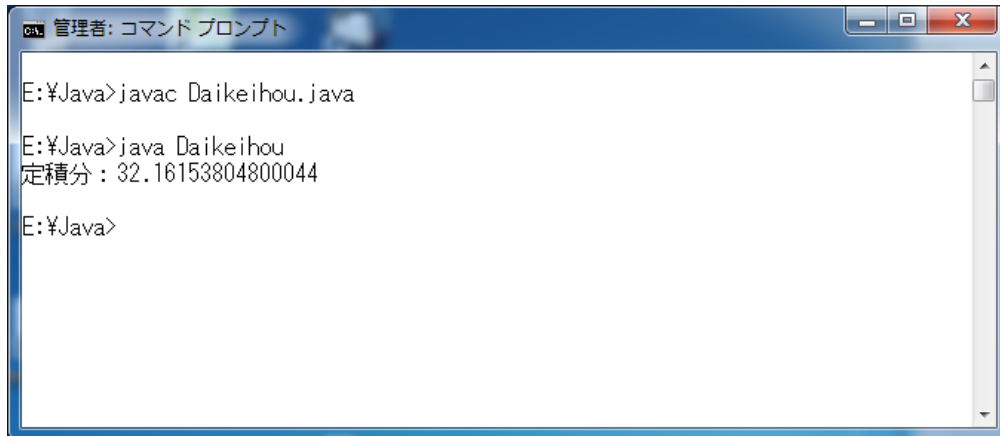




77 ページ：図-11.4 (実行結果の値の変更)



```
管理者: コマンド プロンプト
E:\¥Java>javac Daikeihou.java
E:\¥Java>java Daikeihou
定積分：32.16153804800044
E:\¥Java>
```



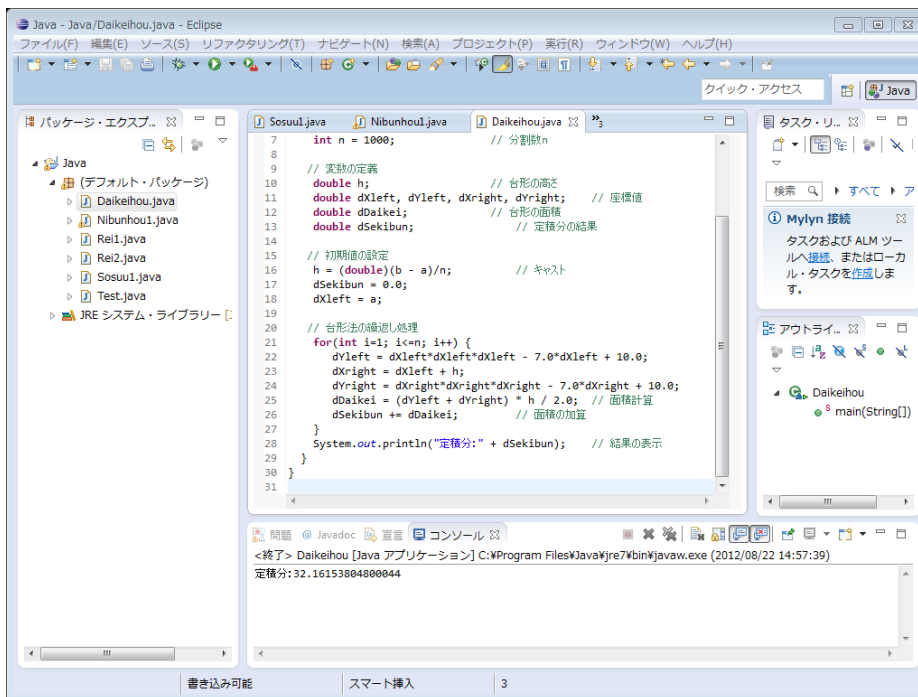
(修正後の実行画面)



```
管理者: コマンド プロンプト
E:\¥Java¥Javaプログラミング原稿>javac Daikeihou.java
E:\¥Java¥Javaプログラミング原稿>java Daikeihou
定積分：2240.001535999951
E:\¥Java¥Javaプログラミング原稿>
```

図-11.4 コマンドプロンプトの実行画面

77 ページ：図-11.5 (実行結果の値の変更)



(修正後の実行画面)

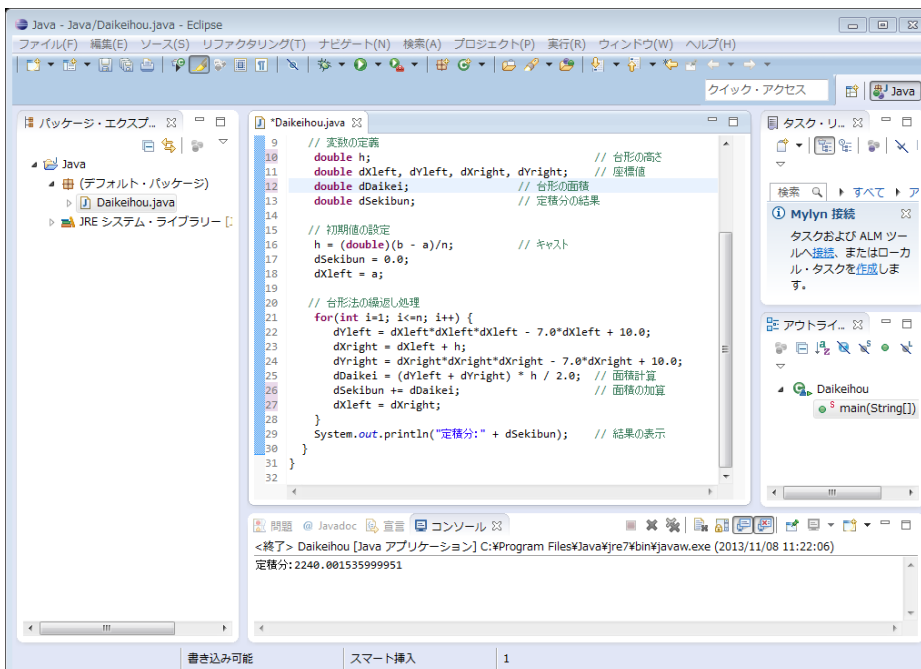


図-11.5 Eclipse の実行画面

103 ページ：図-15.3 (入替処理の終了条件を変更)

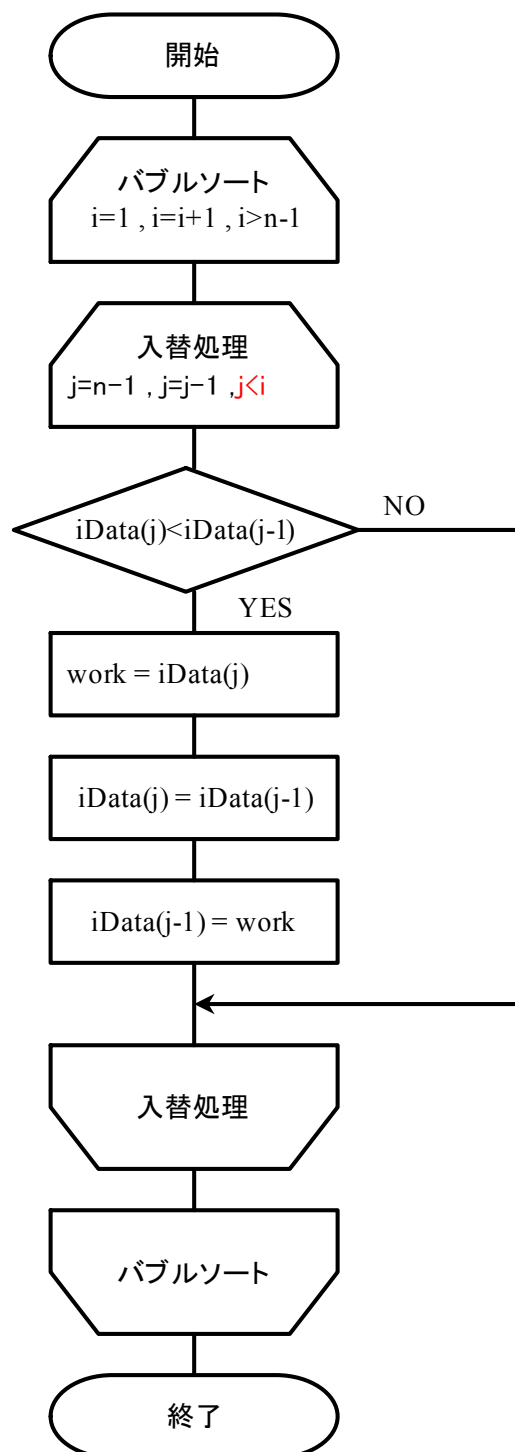


図-15.3 バブルソートのフローチャート

129 ページ：リスト 17.1 (11 行のキーワード“static”を取る)

リスト 17.1 コンストラクタを使ったクラス Total4

行	プログラムの内容
1	class Total4 {
2	// フィールド
3	long lTotal;
4	
5	// コンストラクタ
6	Total4() {
7	lTotal = 0;
8	}
9	
10	// メソッド getTotal
11	public <del>static</del> long getTotal(int iData[]) {
12	for (int i=0; i<iData.length; i++) {
13	lTotal += iData[i];
14	}
15	return lTotal;
16	}
17	}

130 ページ：リスト 17.3 (11 行のキーワード“static”を取る)

リスト 17.3 コンストラクタに引数を使ったクラス Total5

行	プログラムの内容
1	class Total5 {
2	// フィールド
3	long lTotal;
4	
5	// コンストラクタ
6	Total5(long lSet) {
7	lTotal = lSet;
8	}
9	
10	// メソッド getTotal
11	public <del>static</del> long getTotal(int iData[]) {
12	for (int i=0; i<iData.length; i++) {
13	lTotal += iData[i];
14	}
15	return lTotal;
16	}
17	}

145 ページ：リスト 20.2 (8 行に文字列"%3d"を挿入する)

リスト 20.2 オーバーライドのクラス Total5

行	プログラムの内容
1	// バブルソート (オーバーライド)
2	class Total5 extends Total4 {
3	
4	// データ表示 : int 型
5	public void doDisp(int iData[]) {
6	System.out.println(" [ ソートデータ ] int 型");
7	for(int i=0; i<iData.length; i++) {
8	System.out.printf(" %2d 番目 : %3d\n", i+1, iData[i]);
9	}
10	} // メソッド終端
11	
12	} // クラス終端

55 ページ：リスト 22.3 ①1,5 行のコメントを修正する

②7 行の計算式を修正する

リスト 22.3 インターフェースを使ったクラス Square

行	プログラムの内容
1	// <b>四角形</b> の面積計算
2	class Square implements Area {
3	
4	// メソッド getArea
5	// <b>縦</b> : dData[0]、 <b>横</b> : dData[1]
6	public double getArea(double dData[]) {
7	double dArea = <b>dData[0] * dData[1]</b> ;
8	return dArea;
9	}
10	
11	} // クラス終端

168 ページ：リスト 24.1 (23 行を追加)

リスト 24.1 テキストデータの出カクラス Tool9

行	プログラムの内容
1	// テキストデータ出力
2	import java.io.*; // ファイル出カクラスのパッケージ
3	
4	class Tool9 {
5	// フィールド
6	FileWriter fw;
7	BufferedWriter bw;
8	String errorMessage; // エラーメッセージ
9	
10	// コンストラクタ
11	Tool9(String fileName) {
12	try {
13	fw = new FileWriter(fileName); // インスタンス生成
14	bw = new BufferedWriter(fw); // インスタンス生成
15	// エラー処理
16	} catch(IOException e) {
17	errorMessage = "ファイルを作成できません。";
18	}
19	}
20	
21	// メソッド
22	public String writeString(int iData[]) {
23	<b>String sWork;</b>
24	errorMessage = null;
25	try {
26	for(int i=0; i<iData.length; i++) {
27	sWork = Integer.toString(iData[i]); // テキスト変換
28	bw.write(sWork); // テキストデータ出力
29	bw.newLine(); // 改行
30	}
31	bw.close(); // クローズ
32	// エラー処理
33	} catch(IOException e) {
34	errorMessage = "ファイル書き込みエラー";
35	}
36	return errorMessage;
37	}
38	}