Supplement E: Teaching/Learning Java Effectively with Eclipse

For Introduction to Java Programming
By Y. Daniel Liang

0 Introduction

Supplement II.D, "Eclipse Tutorial," gives a brief tutorial on how to use Eclipse. Eclipse is not only a powerful Java program development tool, but it is also a valuable pedagogical tool for teaching and learning Java programming. This supplement shows how to use Eclipse effectively with the text.

The supplement is written for instructors, but it is also useful to students.

1 Important Tips

The objective of the course is to teach Java, not Eclipse. Eclipse is a complex and powerful tool. All you need for this course, however, is a minimum set of features that enable students to create, compile, run, and debug programs. So students should avoid exploring unnecessary features.

If your students follow the instructions in Supplement II.D, "Eclipse Tutorial," or the instructions from you, students can master all essential skills in sixty minutes. It is important that your students adhere to the instructions to avoid frustrating mistakes. If a mistake is made, simply read the instructions and restart from scratch.

2 Eclipse as a Valuable Pedagogical Tool

The following sections demonstrate how to utilize Eclipse in the first eight chapters.

2.1 Using Eclipse in Chapter 1

After Listing 1.1, you can start to cover how to create, compile, and run a program in Eclipse. You may also introduce how to use Eclipse online help.

2.2 Using Eclipse in Chapter 2

You may start to introduce debugging when you cover variables. You can use debug to show the value of a variable in the memory and show the change of the value during

execution. Figure 1 shows a simple test program with variable *i*.

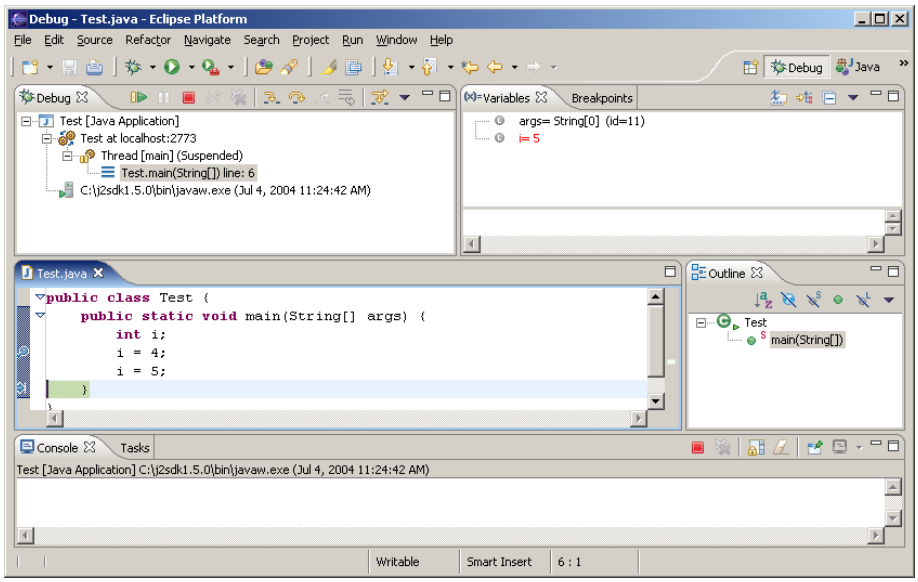

Figure 1

Displaying values of variables in Eclipse debugger.

2.3 Using Eclipse in Chapter 3

Use the debugger to trace the if statements in Common Error 4 in Section 3.7, "Common Errors in Selection Statements," as shown in Figure 2.

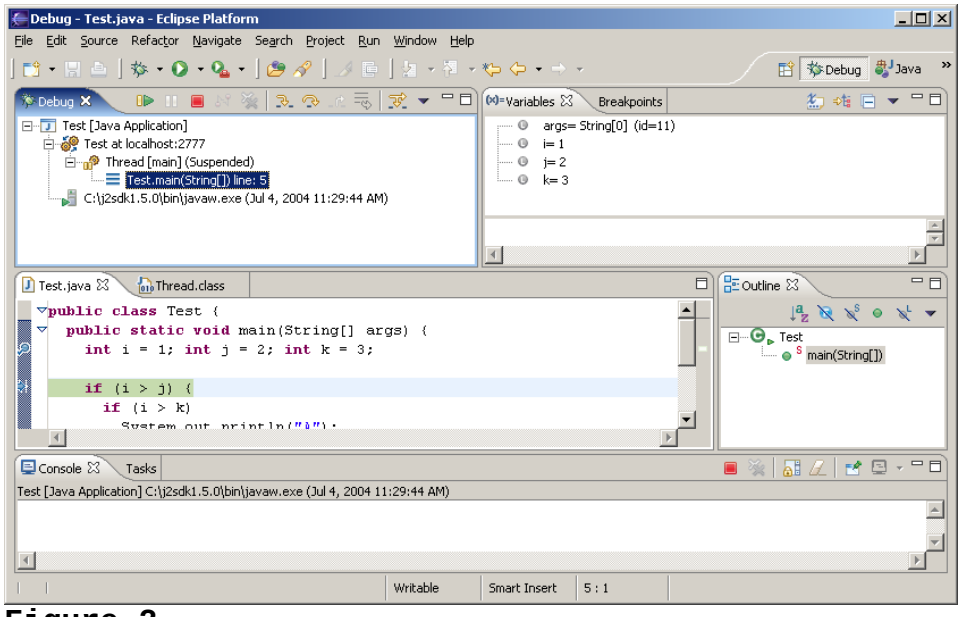

Figure 2

Trace the execution of an if statement.

2.4 Using Eclipse in Chapter 4

Use the debugger to trace the while loop in Listing 4.1 RepeatAdditionQuiz.java," as shown in Figure 3.

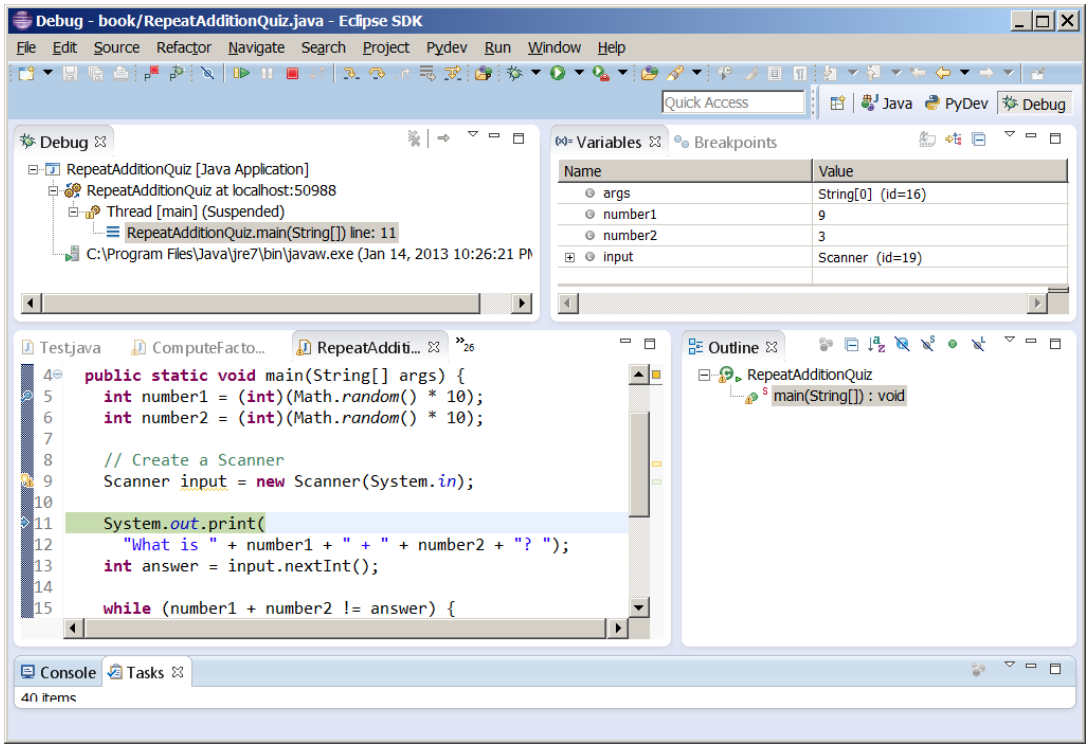

Figure 3

Trace the execution of a loop statement.

2.5 Using Eclipse in Chapter 5

You can use the debugger to show the call stack, which is very effective to help understand method invocation. Let us use Listing 5.1 to demonstrate method invocation. Set a breakpoint at Line 6. Start debugger, and the debugger pauses at Line 6. Choose Step into to step into the max method, as shown in Figure 4. Now in the Call Stack tab of the Debugger window, you will see method max to be invoked.

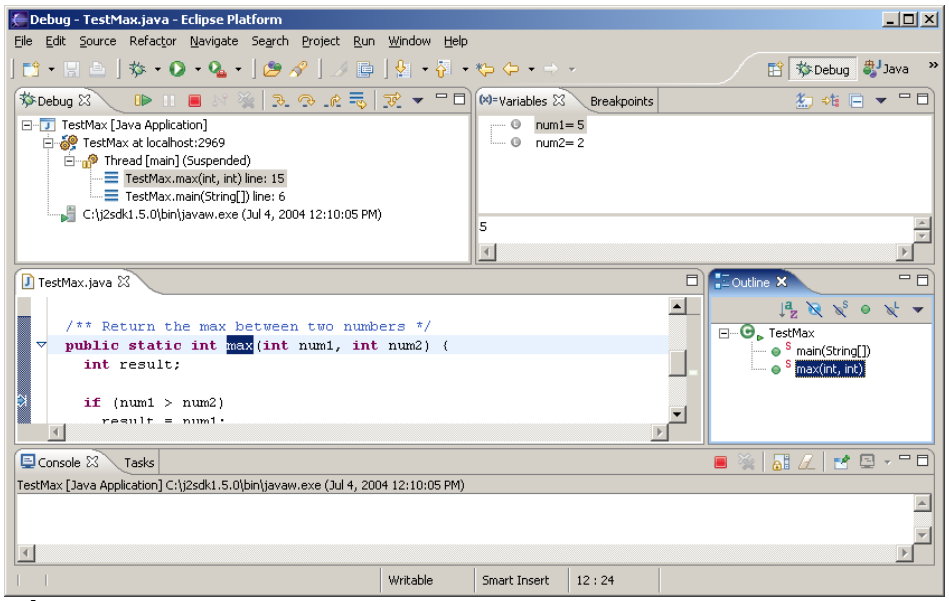

Figure 4

Trace method invocation.

Figure 5 shows tracing recursive execution of the factorial method.

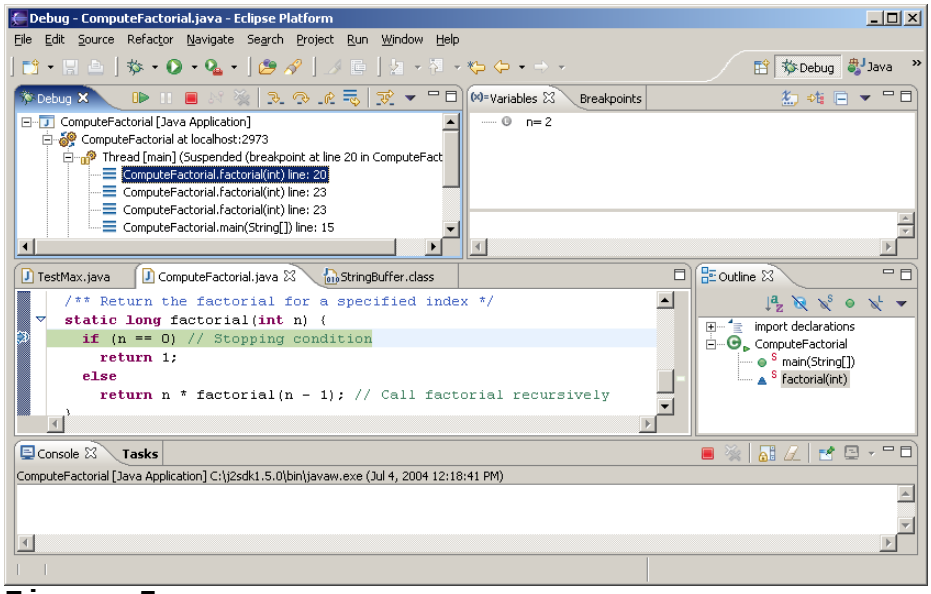

Figure 5

Trace a recursive method invocation.

2.6 Using Eclipse in Chapter 6

You can use the debugger to show the values of all the elements in an array. Figure 6 shows debugging Listing 6.2.

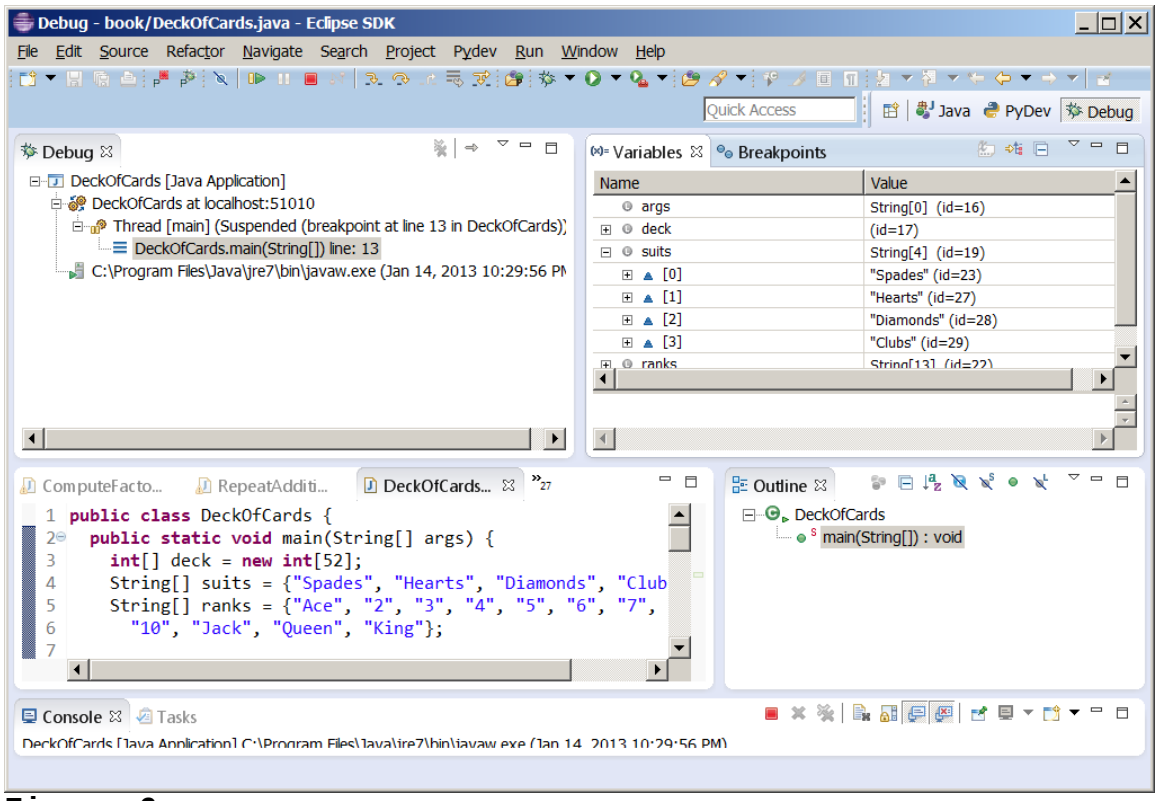

Figure 6

You can see the change of values in an array.

You can use the debugger to demonstrate how arguments are passed and to see the differences between passing primitive type values and arrays.

2.7 Using Eclipse in Chapter 8

You can use the debugger to show the contents of an object. Figure 7 shows debugging Listing 7.1.

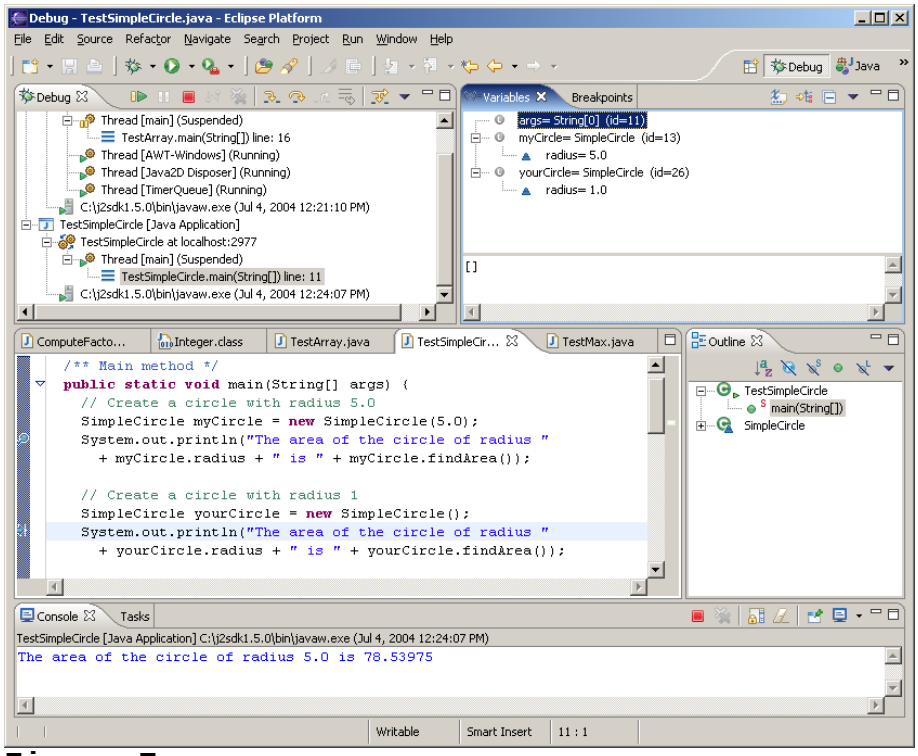

Figure 7

You can see the change of values in an object.

You can use the debugger to demonstrate how arguments are passed and to see the differences between passing primitive type values and objects.