

Scanner and Parsing

(by Nicholas Ruozzi)

Dealing with input and output used to be cumbersome before Java 5, but now Java has a new class to handle file and keyboard I/O called the Scanner class. This class tokenizes the input, i.e. breaks it into smaller pieces called tokens, via pattern matching rules. This allows programmers to easily parse the primitive types from a file or the keyboard and use them in their applications.

By default, white spaces are used as delimiters. Consider the following input from a file:

```
( 34 + -34 wed: -
```

This would be broken into the following string tokens:

```
(  
34  
+  
-34  
wed:  
-
```

Creating a new scanner is easy, for keyboard input:

```
Scanner sc = new Scanner(System.in);
```

for file I/O:

```
Scanner sc = new Scanner(new File("myfile.txt"));
```

The Scanner class provides the following useful methods to parse input:

String next()	returns the next token as a string
int nextInt()	returns the next token as an int
double nextDouble()	returns the next token as a double
String nextLine()	return the next line of the input
void close()	closes the Scanner

However, the above methods cannot be used haphazardly. They can all throw an exception if the input is not in the correct form or there is not any input remaining to be parsed. To account for these possibilities, input should be read only when it is guaranteed to be correct. The following appropriately named methods allow the programmer to check to see what is “next” in the input:

hasNext()	returns true if the next token is as a string
hasNextInt()	returns true if the next token is an int
hasNextDouble()	returns true if the next token is a double
hasNextLine()	return true if there is a next line of the input

Notice that these methods only work with the primitive types and the String class. There are other methods for handling bytes, floats, booleans, etc.

Also, note that Scanner must be imported before we can use it in our Java program (as illustrated in the following example). Let's try an example. We'll read a string of integers from a file "integers.txt" and then display the sum of those integers.

```
import java.util.Scanner;
import java.io.File;

public static void main(String[] args)
{
    int sum = 0;
    Scanner sc = null;

    try
    {
        sc = new Scanner(new File("integers.txt"));
    }
    catch(Exception e)
    {
        System.exit(0);
    }

    while(sc.hasNextInt())    // Check to see that there is a valid integer next
        sum += sc.nextInt(); // Parse the next token as an integer

    sc.close();              // Close the scanner

    System.out.println("The sum is: " + sum);
}
```

Ignore the try-catch construction for the moment since this deals with exceptions which will be covered in a later section. That's really all there is to it. For more information visit the Java API.