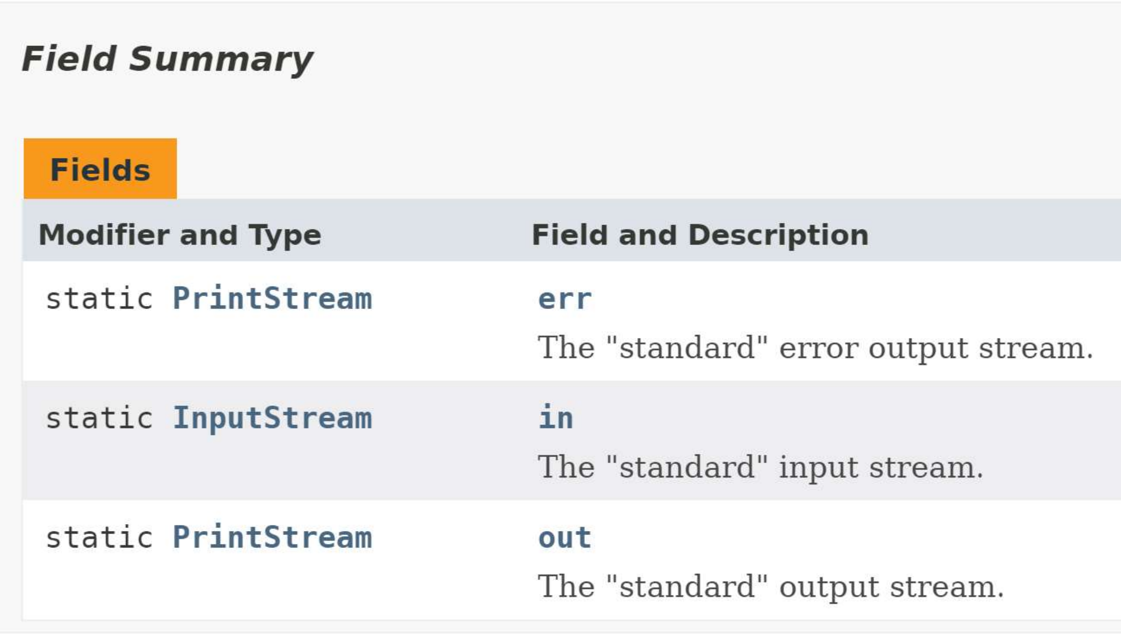
2.3 KEYBOARD AND SCREEN IO = Program I/O

- I/O - Input/Output
- Keyboard is the normal input device
- Screen is the normal output device
- Classes are used for I/O
- They are generally add-on classes (not actually part of Java)
- Some I/O classes are always provided with Java, others are not

Screen output



Field Summary

Fields

Modifier and Type	Field and Description
`static PrintStream`	`err` The "standard" error output stream.
`static InputStream`	`in` The "standard" input stream.
`static PrintStream`	`out` The "standard" output stream.

- Ex)
  - » System.out.println(“Enter a whole number..”);
  - » System.out.println(quarters + “ quarters”);
  - » System.out : an [redacted] that is part of the Java language
  - » an Object System.out has println as one of its [redacted]

Print versus println

- Ex) Same line versus next line
  - System.out.print(“One, two,”);
  - System.out.print(“ buck my shoe.”);
  - System.out.println(“ Three, four,”);
  - System.out.println(“ shut the door.”);
- Output

Screen Output: **`print`** and **`println`**

- Sometimes you want to print part of a line and not go to the next line when you print again
- Two methods, one that goes to a new line and one that does not

```
System.out.println(…);//ends with a new line
System.out.print(…);//stays on the same line
```

- For example:

```
System.out.print("This will all ");
System.out.println("appear on one line");
```

- **`System.out.print()`** works similar to the “+” operator:

```
System.out.println("This will all "
                + "appear on one line, too");
```

Keyboard Input

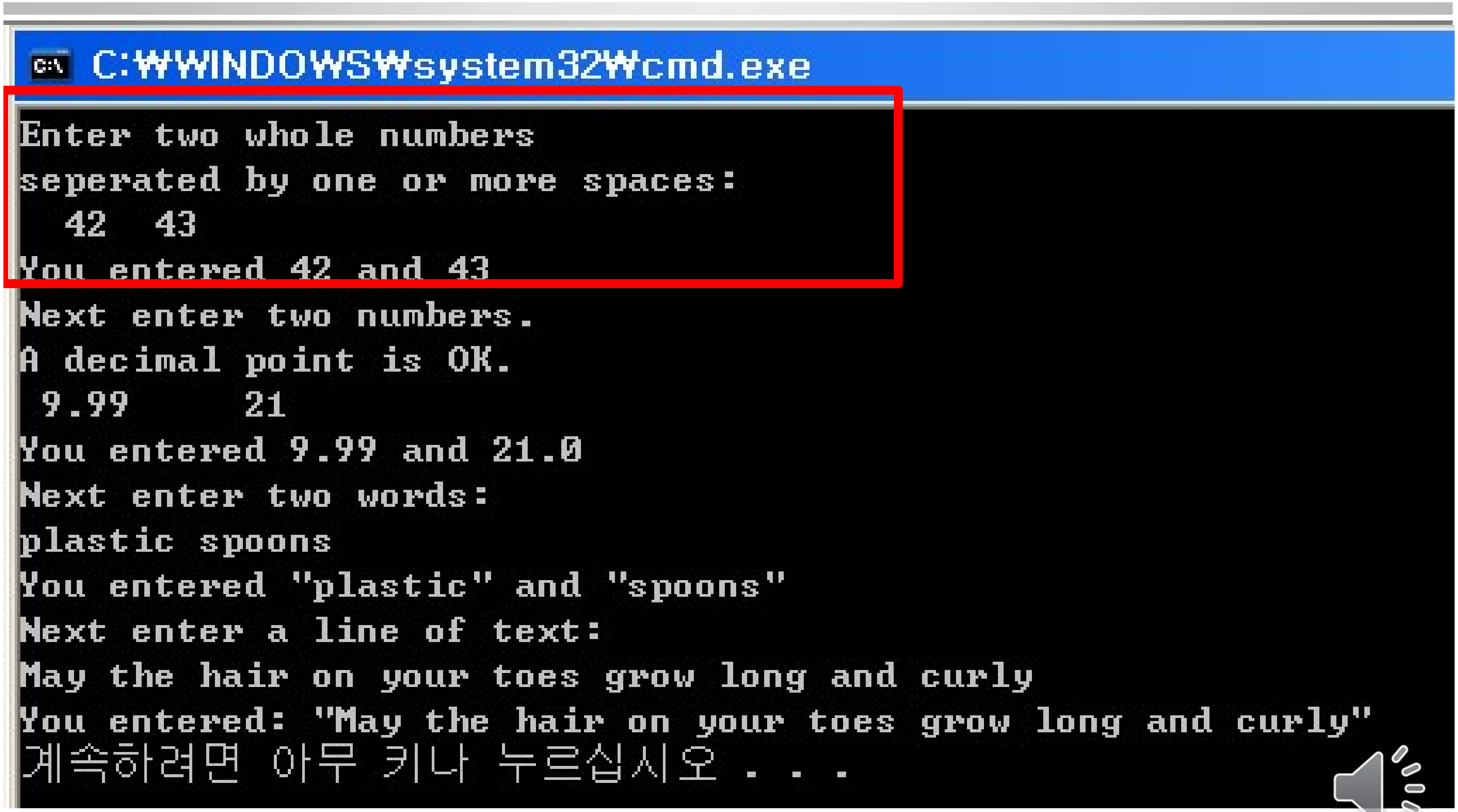
- Scanner [redacted]
  - » Scanner Scanner_Object_Name = new Scanner(System.in);
  - » scannerObject.nextInt()
    - – must be separated by whitespace characters
  - » nextDouble()
  - » next() : read in a word

Fig 2.7 Methods in the Class Scanner

- nextLine()
- nextLong()
- nextInt()
- nextByte()
- nextShort()
- nextDouble()
- nextFloat()
- nextBoolean()
- useDelimiter(Delimiter_word)

Listing 2.5 A Demonstration of Keyboard input

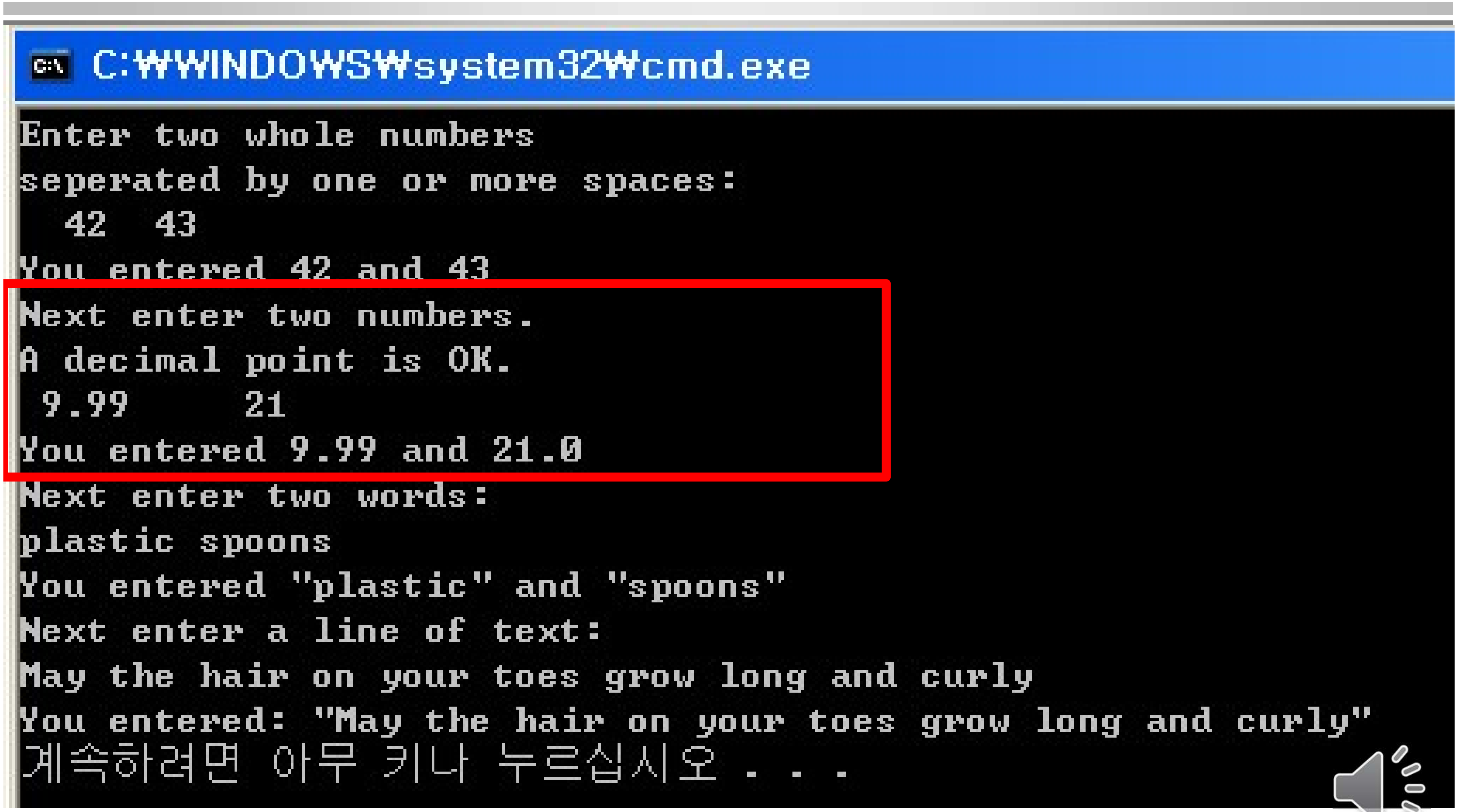
```
import java.util.*;
//import java.util.Scanner;

public class ScannerDemo
{
    public static void main(String[] args)
    {
        int n1, n2;
        Scanner scannerObject = new Scanner(System.in);

        System.out.println("Enter two whole numbers");
        System.out.println("seperated by one or more spaces:");

        n1 = scannerObject.nextInt( );
        n2 = scannerObject.nextInt( );
        System.out.println("You entered " + n1 + " and " + n2);
```



C:₩WINDOWS₩system32₩cmd.exe

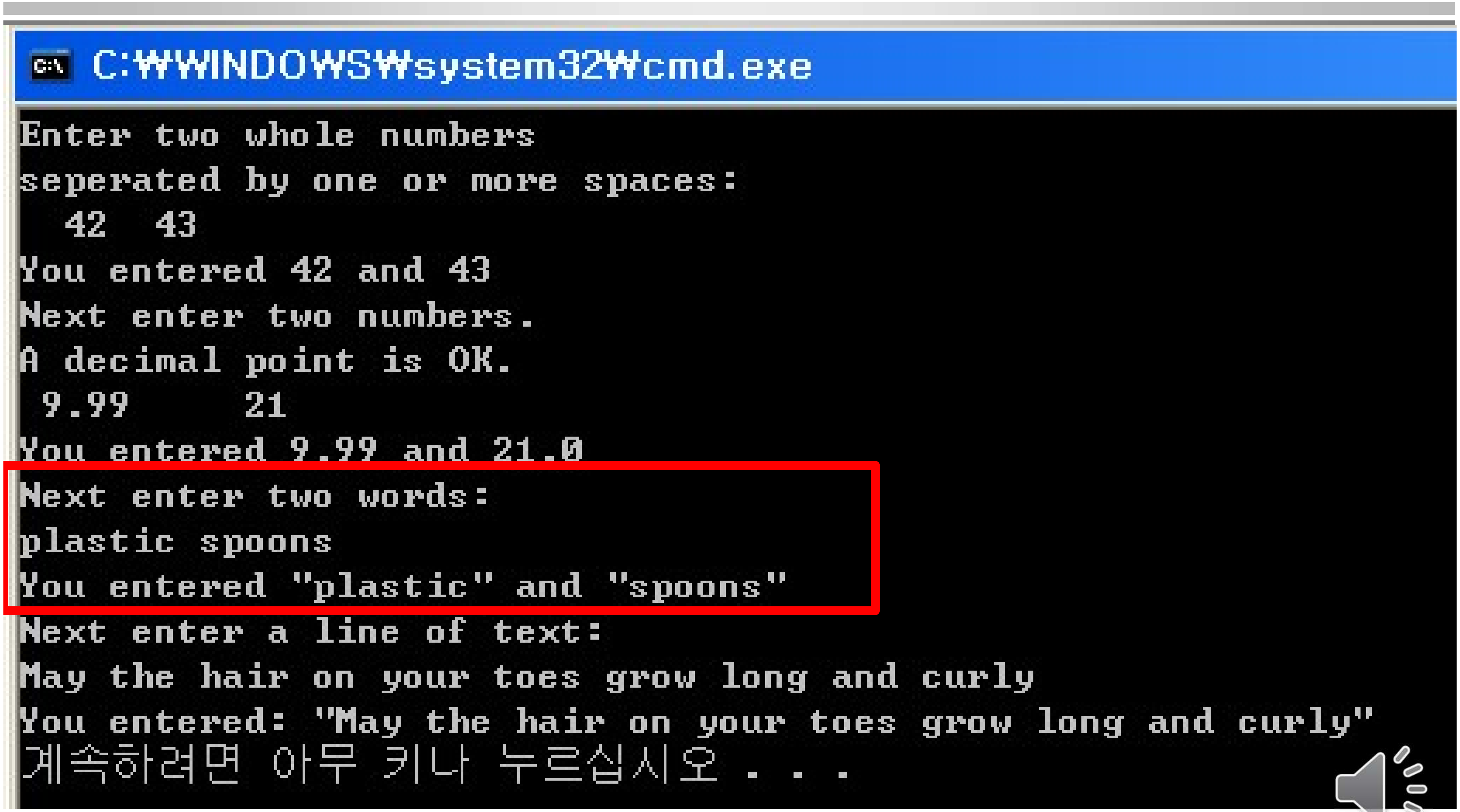
```
Enter two whole numbers
seperated by one or more spaces:
  42   43
You entered 42 and 43
Next enter two numbers.
A decimal point is OK.
 9.99      21
You entered 9.99 and 21.0
Next enter two words:
plastic spoons
You entered "plastic" and "spoons"
Next enter a line of text:
May the hair on your toes grow long and curly
You entered: "May the hair on your toes grow long and curly"
계속하려면 아무 키나 누르십시오 . . .
```

```
System.out.println("Next enter two numbers.");
        System.out.println("A decimal point is OK.");

        double d1, d2;
        d1 = scannerObject.nextDouble( );
        d2 = scannerObject.nextDouble( );
        System.out.println("You entered " + d1 + " and " + d2);

        System.out.println("Next enter two words:");

        String s1, s2;
        s1 = scannerObject.next( );
        s2 = scannerObject.next( );
        System.out.println("You entered \"" +
                             s1 + "\" and \"" + s2 + "\"");

        s1 = scannerObject.nextLine( ); //To get rid of '\n'

        System.out.println("Next enter a line of text:");
        s1 = scannerObject.nextLine( );
        System.out.println("You entered: \"" + s1 + "\"");
    }
}
```



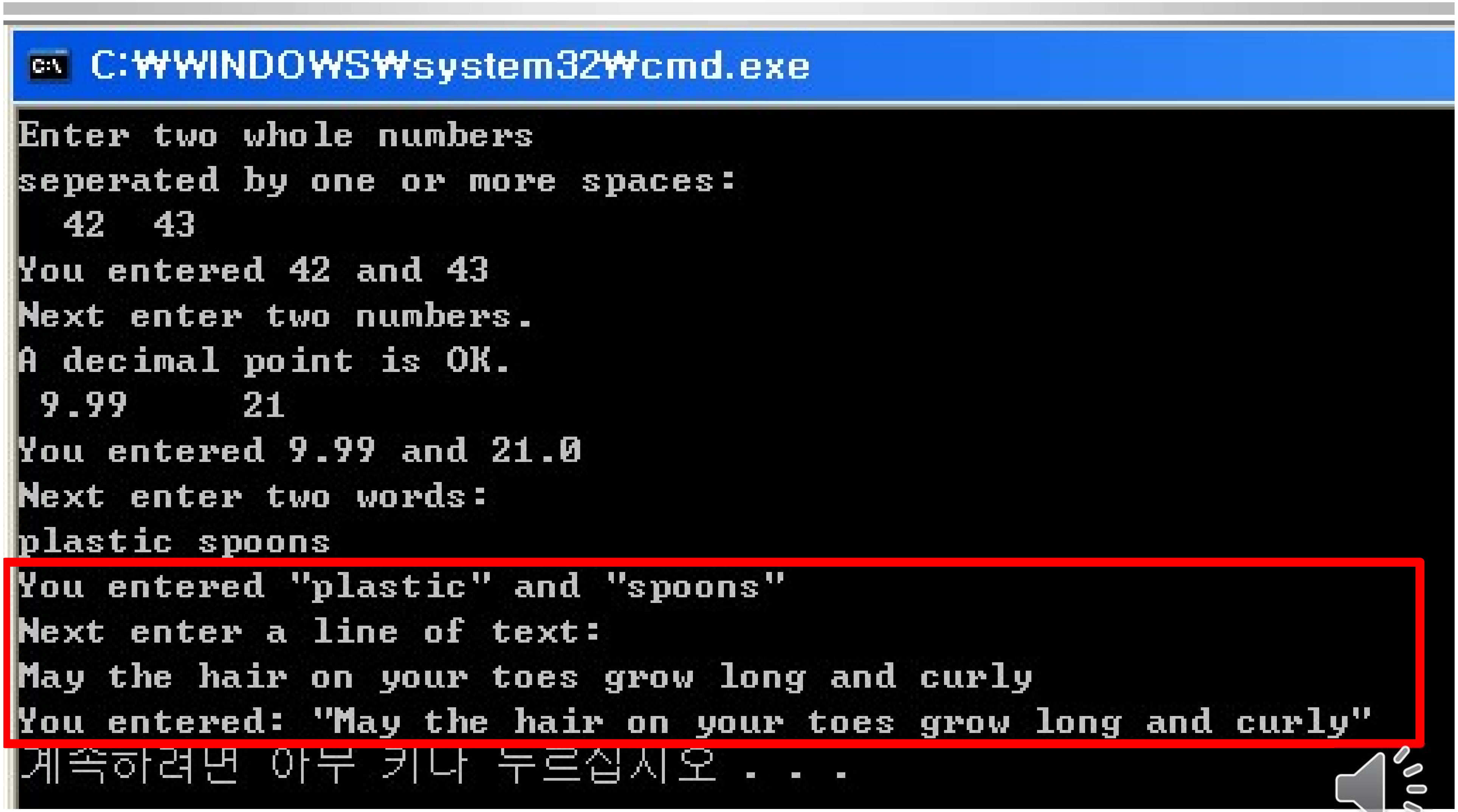
```
Enter two whole numbers
seperated by one or more spaces:
  42  43
You entered 42 and 43
Next enter two numbers.
A decimal point is OK.
 9.99     21
You entered 9.99 and 21.0
Next enter two words:
plastic spoons
You entered "plastic" and "spoons"
Next enter a line of text:
May the hair on your toes grow long and curly
You entered: "May the hair on your toes grow long and curly"
계속하려면 아무 키나 누르십시오 . . .
```

```
 System.out.println("Next enter two numbers.");
     System.out.println("A decimal point is OK.");

     double d1, d2;
     d1 = scannerObject.nextDouble( );
     d2 = scannerObject.nextDouble( );
     System.out.println("You entered " + d1 + " and " + d2);

     System.out.println("Next enter two words:");

     String s1, s2;
     s1 = scannerObject.next( );
     s2 = scannerObject.next( );
     System.out.println("You entered \"" +
                      s1 + "\" and \"" + s2 + "\"");
     s1 = scannerObject.nextLine( ); //To get rid of '\n'

     System.out.println("Next enter a line of text:");
     s1 = scannerObject.nextLine( );
     System.out.println("You entered: \"" + s1 + "\"");
  }
}
```



```
Enter two whole numbers
seperated by one or more spaces:
  42  43
You entered 42 and 43
Next enter two numbers.
A decimal point is OK.
 9.99     21
You entered 9.99 and 21.0
Next enter two words:
plastic spoons
You entered "plastic" and "spoons"
Next enter a line of text:
May the hair on your toes grow long and curly
You entered: "May the hair on your toes grow long and curly"
계속하려면 아무 키나 누르십시오 . . .
```

```
 System.out.println("Next enter two numbers.");
      System.out.println("A decimal point is OK.");

      double d1, d2;
      d1 = scannerObject.nextDouble( );
      d2 = scannerObject.nextDouble( );
      System.out.println("You entered " + d1 + " and " + d2);

      System.out.println("Next enter two words:");

      String s1, s2;
      s1 = scannerObject.next( );
      s2 = scannerObject.next( );
      System.out.println("You entered \"" +
                        s1 + "\" and \"" + s2 + "\"");

      s1 = scannerObject.nextLine( ); //To get rid of '\n'

      System.out.println("Next enter a line of text:");
      s1 = scannerObject.nextLine( );
      System.out.println("You entered: \"" + s1 + "\"");
   }
}
```



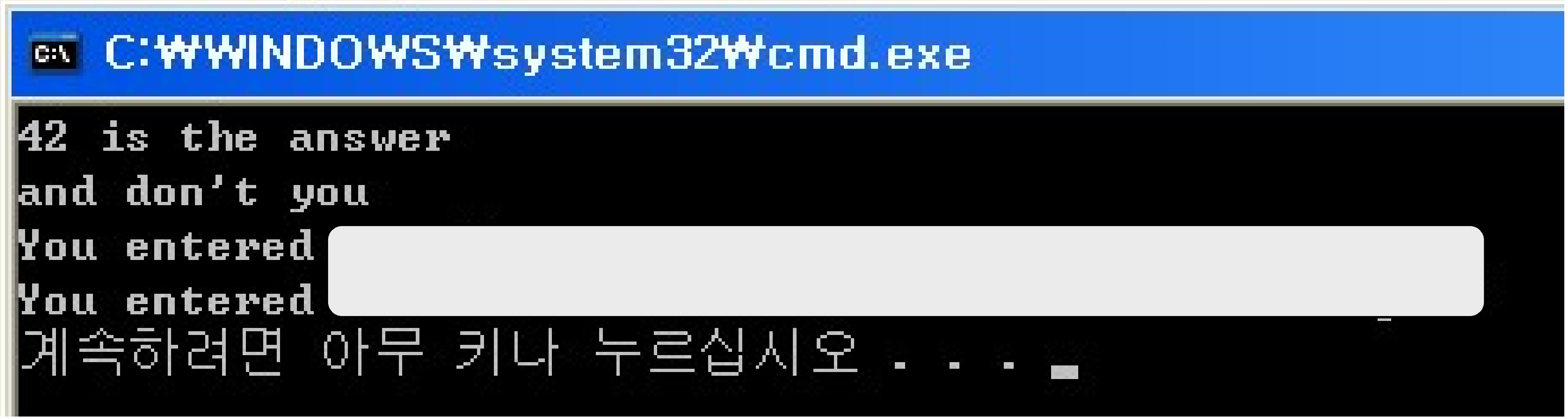
```
Enter two whole numbers
seperated by one or more spaces:
  42   43
You entered 42 and 43
Next enter two numbers.
A decimal point is OK.
 9.99      21
You entered 9.99 and 21.0
Next enter two words:
plastic spoons
You entered "plastic" and "spoons"
Next enter a line of text:
May the hair on your toes grow long and curly
You entered: "May the hair on your toes grow long and curly"
계속하려면 아무 키나 누르십시오 . . .
```

Problems with the nextLine method

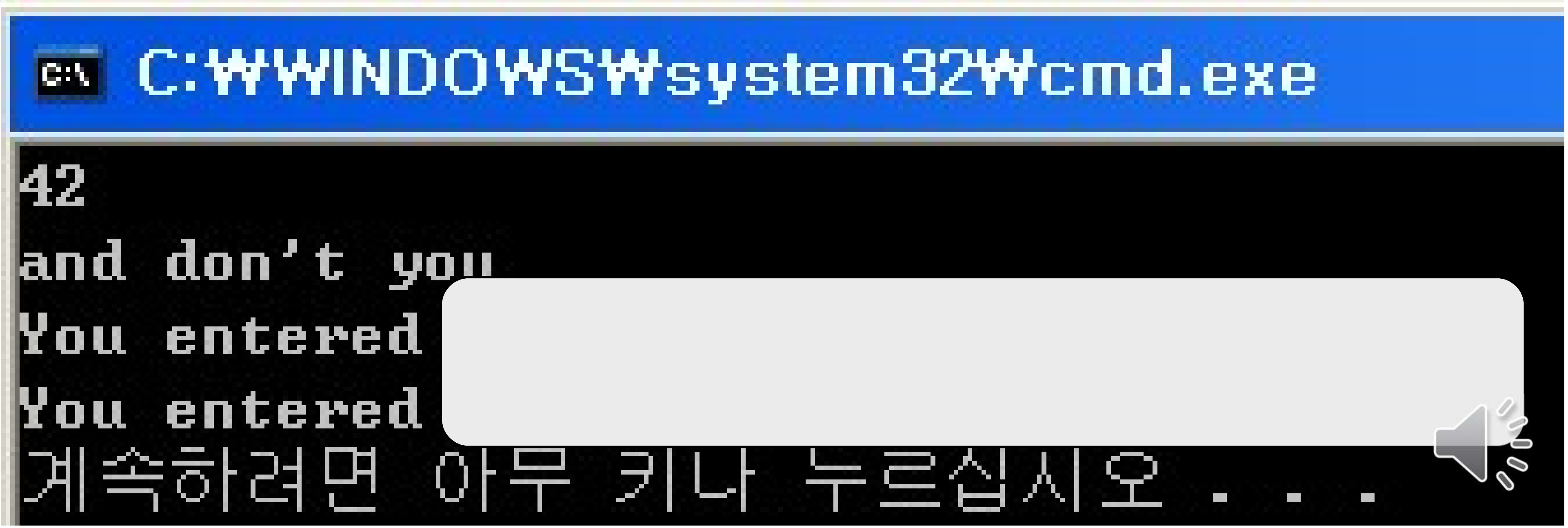
- The method nextLine of the class Scanner reads the remainder of a line of text starting wherever the last keyboard reading left off.

Example

```
import java.util.*;

public class ScannerDemo2
{
    public static void main(String[] args)
    {
        int n1;
        Scanner scannerObject = new Scanner(System.in);

        n1 = scannerObject.nextInt( );
        String s1, s2;
        s1 = scannerObject.nextLine( );
        s2 = scannerObject.nextLine( );
        System.out.println("You entered n = " +n1 );

        System.out.println("You entered s1=\"" +
                        s1 + "\" and s2=\"" + s2 + "\"");

    }
}
```



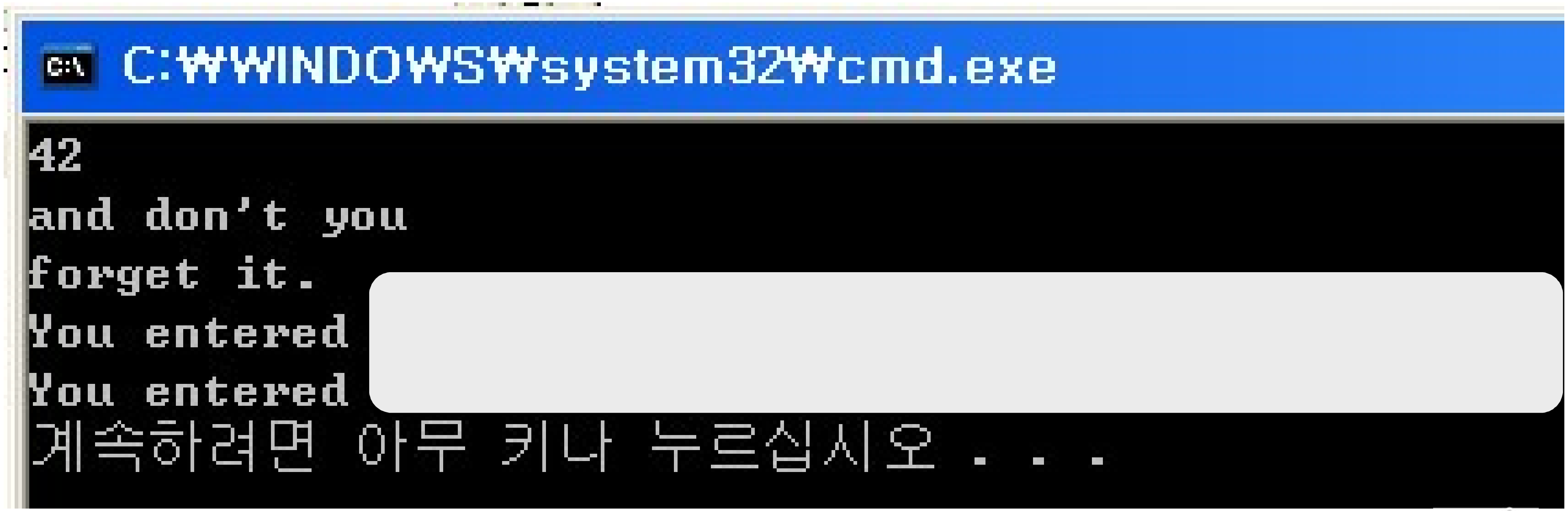
```
42 is the answer
and don't you
You entered
You entered
계속하려면 아무 키나 누르십시오 . . .
```



```
42
and don't you
You entered
You entered
계속하려면 아무 키나 누르십시오 . . .
```

Example(Modified)

```
import java.util.*;

public class ScannerDemo2
{
    public static void main(String[] args)
    {
        int n1;
        Scanner scannerObject = new Scanner(System.in);

        n1 = scannerObject.nextInt( );
        String s1, s2;
        s1 = scannerObject.nextLine( ); // To get rid of '\n'
        s1 = scannerObject.nextLine( );
        s2 = scannerObject.nextLine( );
        System.out.println("You entered n = " +n1 );

        System.out.println("You entered s1=\"" +
                        s1 + "\" and s2=\"" + s2 + "\"");

    }
}
```



```
42
and don't you
forget it.
You entered
You entered
계속하려면 아무 키나 누르십시오 . . .
```

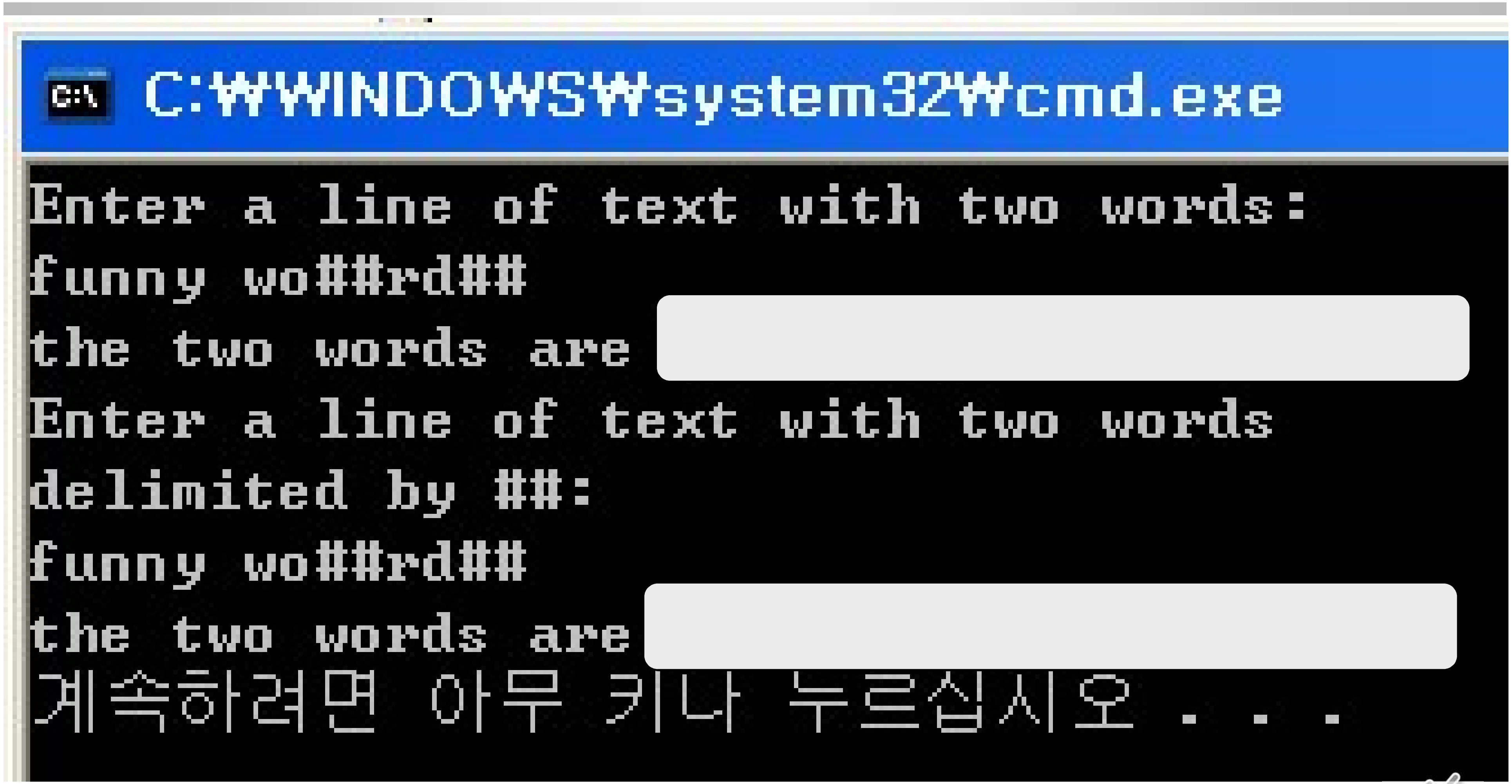
Other input Delimiters

- With Scanner the default delimiters are the whitespace characters.
- You can change delimiters
- Ex) keyboard2.[redacted](“##”);

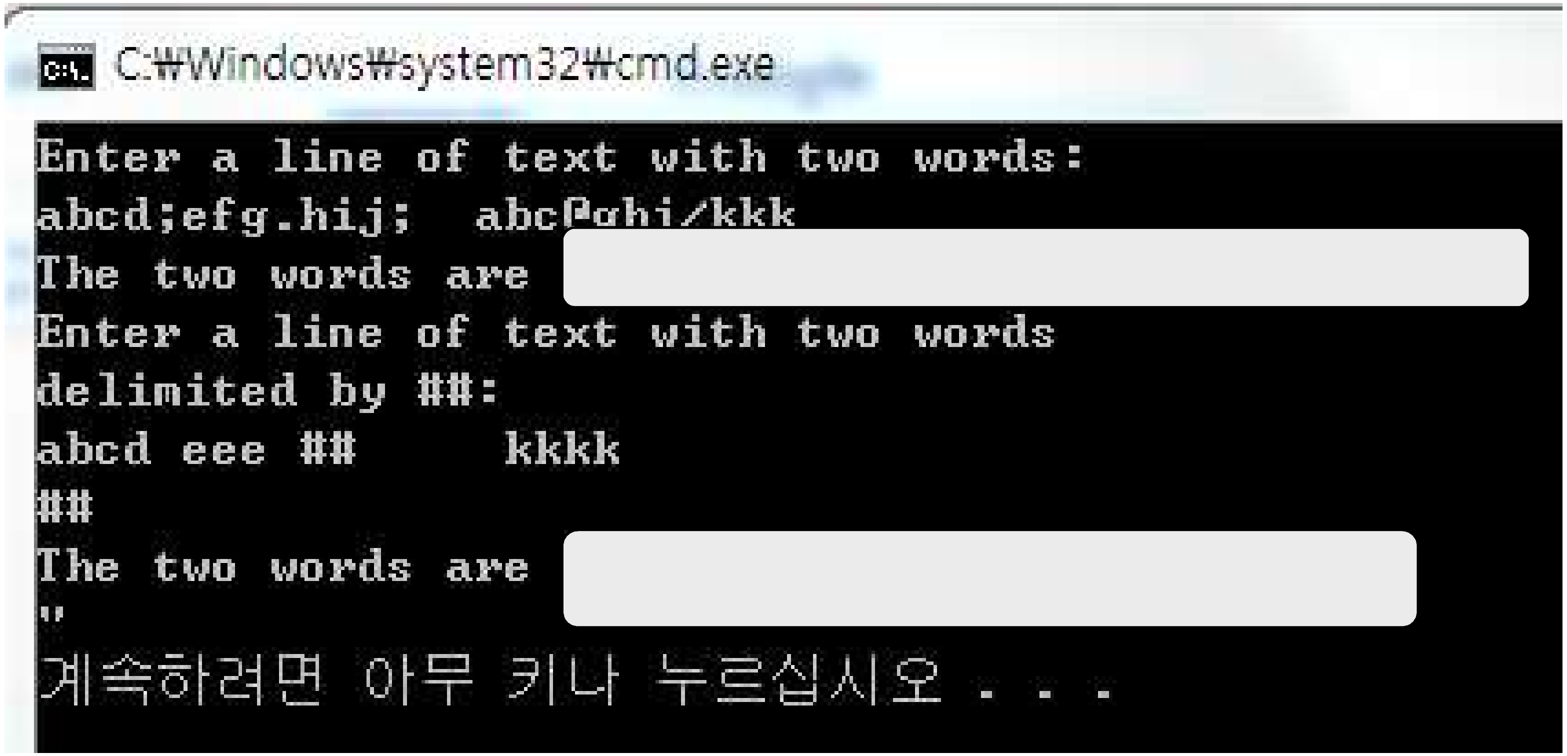
Listing 2.6 Changing Delimiters

```
import java.util.*;

public class DelimitersDemo
{
    public static void main(String[] args)
    {
        Scanner keyboard1 = new Scanner(System.in);
        Scanner keyboard2 = new Scanner(System.in);

        keyboard2.useDelimiter("##");
        //The delimiters for keyboard1 are the white space characters.
        //The only delimiter for keyboard2 is ##.
```

```
 String s1, s2;

        System.out.println("Enter a line of text with two words:");
        s1 = keyboard1.next( );
        s2 = keyboard1.next( );
        System.out.println("the two words are \"" + s1
                              + "\" and \"" + s2 + "\"");

        System.out.println("Enter a line of text with two words");
        System.out.println("delimited by ##:");
        s1 = keyboard2.next( );
        s2 = keyboard2.next( );
        System.out.println("the two words are \"" + s1
                              + "\" and \"" + s2 + "\"");
   }
}
```

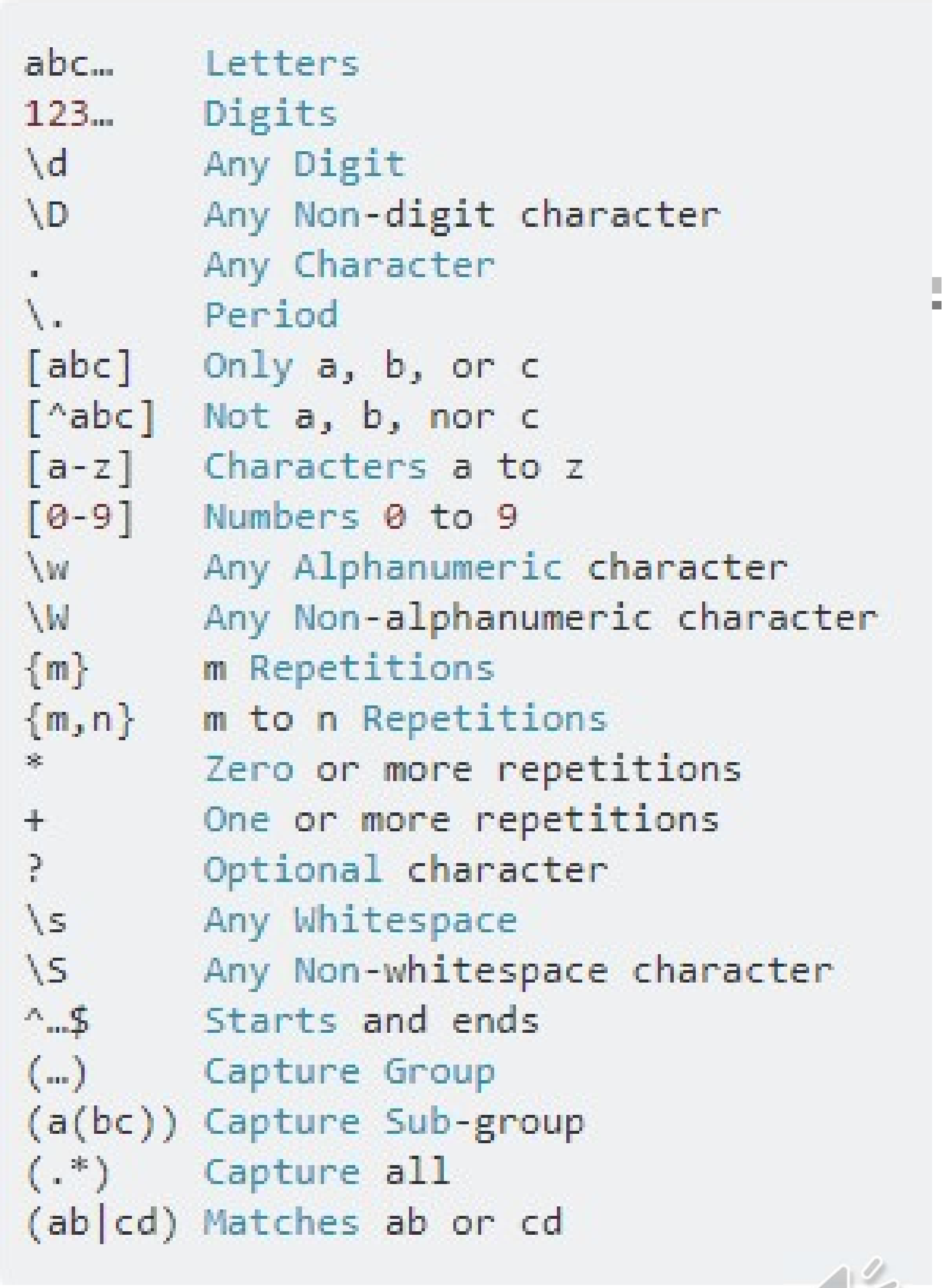


```
Enter a line of text with two words:
funny wo##rd##
the two words are
Enter a line of text with two words
delimited by ##:
funny wo##rd##
the two words are
계속하려면 아무 키나 누르십시오 . . .
```



```
Enter a line of text with two words:
abcd;efg.hij;  abc@ghi/kkk
The two words are [redacted]
Enter a line of text with two words
delimited by ##:
abcd eee ##     kkkk
##
The two words are [redacted]
"
계속하려면 아무 키나 누르십시오 . . .
```

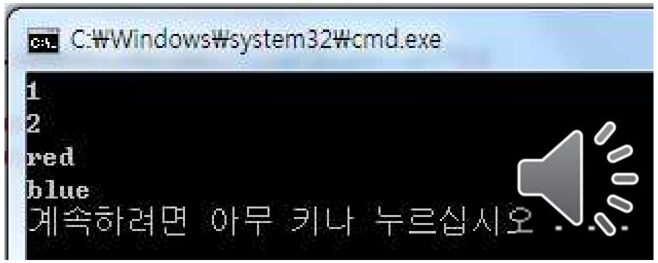
- Regular expression의 활용



abc…	Letters
123…	Digits
\d	Any Digit
\D	Any Non-digit character
.	Any Character
\\.	Period
[abc]	Only a, b, or c
[^abc]	Not a, b, nor c
[a-z]	Characters a to z
[0-9]	Numbers 0 to 9
\w	Any Alphanumeric character
\W	Any Non-alphanumeric character
{m}	m Repetitions
{m,n}	m to n Repetitions
*	Zero or more repetitions
+	One or more repetitions
?	Optional character
\s	Any Whitespace
\S	Any Non-whitespace character
^…$	Starts and ends
(…)	Capture Group
(a(bc))	Capture Sub-group
(.*)	Capture all
(ab|cd)	Matches ab or cd

Regular expression의 활용

```
import java.util.Scanner;

public class DelimitersDemo2
{
    public static void main(String[] args)
    {
  String input = "1 fish 2 fish red fish blue fish";

  // \\s* means 0 or more repetitions of any whitespace character
  // fish is the pattern to find
  Scanner s = new Scanner(input).useDelimiter("[redacted]");

  System.out.println(s.nextInt());   // prints: 1
  System.out.println(s.nextInt());   // prints: 2
  System.out.println(s.next());      // prints: red
  System.out.println(s.next());      // prints: blue

  // don't forget to close the scanner!!
 s.close();
    }
}
```

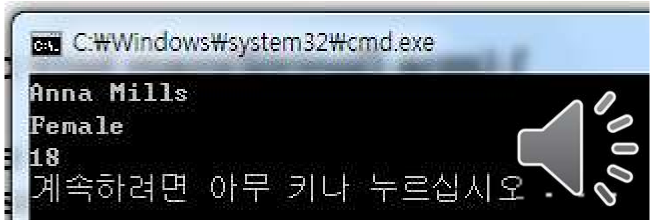


```
1
2
red
blue
계속하려면 아무 키나 누르십시오 . . .
```

useDelimiter()

```
import java.util.Scanner;

public class DelimitersDemo3
{

public static void main(String[] args) {

    // Initialize Scanner object
    Scanner scan = new Scanner("Anna Mills/Female/18");
    // initialize the string delimiter
    scan.useDelimiter("/");
    // Printing the tokenized Strings
    while(scan[redacted]){
        System.out.println(scan.next());
    }
    // closing the scanner stream
    scan.close();
}

}
```

✓



```
C:\Windows\system32\cmd.exe
Anna Mills
Female
18
계속하려면 아무 키나 누르십시오 . . .
```

끝