Class Scanner and Class String

Mitsu Ogihara

Department of Computer Science
University of Miami

Table of Contents

What is a scanner?

- **Scanner** is a Java class that provides access to text, where the text that is being “scanned” is split into *tokens* with a *token delimiter*

What is a scanner?

- **Scanner** is a Java class that provides access to text, where the text that is being "scanned" is split into *tokens* with a *token delimiter*
- Scanner "can read" from three different types of things
  1. String (either a variable or a literal)
  2. File
  3. Input from keyboard

Using a Scanner - the Basics

To be able to use Scanner, you need to declare an **object data type** (as opposed to the primitive data type).

Using a Scanner - the Basics

To be able to use Scanner, you need to declare an **object data type** (as opposed to the primitive data type).

In particular, for receiving input from user, we need to declare a Scanner variable as:

- `Scanner VARIABLE-NAME;`

and then assign to this variable a *Scanner object*

- `VARIABLE-NAME = new Scanner( PARAMETER );`

This of course can be written in a single line:

- `Scanner VARIABLE-NAME = new Scanner( PARAMETER );`

Examples of a Scanner Construction/Object Creation

1. `new Scanner( "My \t Year 2015 \n GPA is 3.5" );`
   ··· the Scanner object will scan for the tokens in the string
2. `new Scanner( new File( "foo.txt" ) );`
   ··· the Scanner object will scan for tokens in the file by the name of `foo.txt`
3. `new Scanner( System.in );`
   ··· the Scanner object will scan for tokens from the keyboard input

Importing a Java class

To be able to use Scanner, you need to add its code to your code, which can be accomplished by adding either of the following lines at the beginning of your program:

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

The Delimiter

The default delimiter of Scanner is any nonempty sequence of the write space, the tab, and the newline (' ', '\t', '\n')

How to use a Scanner

After creating a Scanner object, you can use it to receive input
Three frequently used methods for receiving input:

- `next()` : returns the next token as a String
- `nextInt()` : returns the value generated by interpreting the next token as than an expression of an integer
- `nextDouble()` : returns the value generated by interpreting the next token as than an expression of a double

In the last two, if the format of the token does not conform to the desired format, a runtime error `InputMismatchException` occurs

Scanner of a String

```
1  import java.util.Scanner;
2  // an example of using Scanner
3  public class StringScanner {
4    public static void main( String[] args ) {
5      ////////////////////////////////////////////////
6      // defining variables
7      ////////////////////////////////////////////////
8      Scanner console;
9      String input, token1, token2, token3, token4;
10     int year;
11     double gpa;
12     ////////////////////////////////////////////////
13     // assigning value to input and to the consolener
14     ////////////////////////////////////////////////
15     input = "My \t Year \t 2015 GPA is    3.5";
16     console  = new Scanner( input );
```

The variables to be used in the program

Scanner of a String

```
1  import java.util.Scanner;
2  // an example of using Scanner
3  public class StringScanner {
4    public static void main( String[] args ) {
5      ////////////////////////////////////////////////
6      // defining variables
7      ////////////////////////////////////////////////
8      Scanner console;
9      String input, token1, token2, token3, token4;
10     int year;
11     double gpa;
12     ////////////////////////////////////////////////
13     // assigning value to input and to the consolener
14     ////////////////////////////////////////////////
15     input = "My \t Year \t 2015 GPA is    3.5";
16     console  = new Scanner( input );
```

Assign a value to the string `input` and construct a scanner out of the string

Scanner of a String

```
17      ////////////////////////////////////////////////
18      // read tokens in the order of appearance
19      ////////////////////////////////////////////////
20      token1 = console.next();
21      token2 = console.next();
22      year = console.nextInt();
23      token3 = console.next();
24      token4 = console.next();
25      gpa = console.nextDouble();
```

The first, second, third, and fourth String tokens

Scanner of a String

```
17      ////////////////////////////////////////////
18      // read tokens in the order of appearance
19      ////////////////////////////////////////////
20      token1 = console.next();
21      token2 = console.next();
22      year = console.nextInt();
23      token3 = console.next();
24      token4 = console.next();
25      gpa = console.nextDouble();
```

The int token The double token

Scanner of a String

```
17      ////////////////////////////////////////////////
18      // read tokens in the order of appearance
19      ////////////////////////////////////////////////
20      token1 = console.next();
21      token2 = console.next();
22      year = console.nextInt();
23      token3 = console.next();
24      token4 = console.next();
25      gpa = console.nextDouble();
```

Scanner of Multiple Data Types

```
26      ////////////////////////////////////////////
27      // produce output
28      ////////////////////////////////////////////
29      System.out.println( "The input is:" );
30      System.out.println( input );
31      System.out.println( "-------------------------------------------" );
32      System.out.println( "The string tokens are:" );
33      System.out.println( token1 );
34      System.out.println( token2 );
35      System.out.println( token3 );
36      System.out.println( token4 );
37      System.out.println( "-------------------------------------------" );
38      System.out.println( "The int token is:" );
39      System.out.println( year );
40      System.out.println( "-------------------------------------------" );
41      System.out.println( "The double token is:" );
42      System.out.println( gpa );
43    }
44  }
```

Produce message for `input`

Scanner of Multiple Data Types

```
26      ////////////////////////////////////////////
27      // produce output
28      ////////////////////////////////////////////
29      System.out.println( "The input is:" );
30      System.out.println( input );
31      System.out.println( "-------------------------------------------" );
32      System.out.println( "The string tokens are:" );
33      System.out.println( token1 );
34      System.out.println( token2 );
35      System.out.println( token3 );
36      System.out.println( token4 );
37      System.out.println( "-------------------------------------------" );
38      System.out.println( "The int token is:" );
39      System.out.println( year );
40      System.out.println( "-------------------------------------------" );
41      System.out.println( "The double token is:" );
42      System.out.println( gpa );
43    }
44  }
```

Print the value of `input` and proceed to the next line

Scanner of Multiple Data Types

```
26      ////////////////////////////////////////////
27      // produce output
28      ////////////////////////////////////////////
29      System.out.println( "The input is:" );
30      System.out.println( input );
31      System.out.println( "-------------------------------------------" );
32      System.out.println( "The string tokens are:" );
33      System.out.println( token1 );
34      System.out.println( token2 );
35      System.out.println( token3 );
36      System.out.println( token4 );
37      System.out.println( "-------------------------------------------" );
38      System.out.println( "The int token is:" );
39      System.out.println( year );
40      System.out.println( "-------------------------------------------" );
41      System.out.println( "The double token is:" );
42      System.out.println( gpa );
43    }
44  }
```

Produce message for the `String` tokens

Scanner of Multiple Data Types

```
26      ////////////////////////////////////////////
27      // produce output
28      ////////////////////////////////////////////
29      System.out.println( "The input is:" );
30      System.out.println( input );
31      System.out.println( "-------------------------------------------" );
32      System.out.println( "The string tokens are:" );
33      System.out.println( token1 );
34      System.out.println( token2 );
35      System.out.println( token3 );
36      System.out.println( token4 );
37      System.out.println( "-------------------------------------------" );
38      System.out.println( "The int token is:" );
39      System.out.println( year );
40      System.out.println( "-------------------------------------------" );
41      System.out.println( "The double token is:" );
42      System.out.println( gpa );
43    }
44  }
```

Print the value of each `String` token, each with a newline character

Scanner of Multiple Data Types

```
26      ////////////////////////////////////////////
27      // produce output
28      ////////////////////////////////////////////
29      System.out.println( "The input is:" );
30      System.out.println( input );
31      System.out.println( "-------------------------------------------" );
32      System.out.println( "The string tokens are:" );
33      System.out.println( token1 );
34      System.out.println( token2 );
35      System.out.println( token3 );
36      System.out.println( token4 );
37      System.out.println( "-------------------------------------------" );
38      System.out.println( "The int token is:" );
39      System.out.println( year );
40      System.out.println( "-------------------------------------------" );
41      System.out.println( "The double token is:" );
42      System.out.println( gpa );
43    }
44  }
```

Produce message for `year`

Scanner of Multiple Data Types

```
26      ////////////////////////////////////////////
27      // produce output
28      ////////////////////////////////////////////
29      System.out.println( "The input is:" );
30      System.out.println( input );
31      System.out.println( "-------------------------------------------" );
32      System.out.println( "The string tokens are:" );
33      System.out.println( token1 );
34      System.out.println( token2 );
35      System.out.println( token3 );
36      System.out.println( token4 );
37      System.out.println( "-------------------------------------------" );
38      System.out.println( "The int token is:" );
39      System.out.println( year );
40      System.out.println( "-------------------------------------------" );
41      System.out.println( "The double token is:" );
42      System.out.println( gpa );
43    }
44  }
```

Print the value of `year` and proceed to the next line

Scanner of Multiple Data Types

```
26      ////////////////////////////////////////////
27      // produce output
28      ////////////////////////////////////////////
29      System.out.println( "The input is:" );
30      System.out.println( input );
31      System.out.println( "-------------------------------------------" );
32      System.out.println( "The string tokens are:" );
33      System.out.println( token1 );
34      System.out.println( token2 );
35      System.out.println( token3 );
36      System.out.println( token4 );
37      System.out.println( "-------------------------------------------" );
38      System.out.println( "The int token is:" );
39      System.out.println( year );
40      System.out.println( "-------------------------------------------" );
41      System.out.println( "The double token is:" );
42      System.out.println( gpa );
43    }
44  }
```

Produce message for `gpa`

Scanner of Multiple Data Types

```
26      ////////////////////////////////////////////
27      // produce output
28      ////////////////////////////////////////////
29      System.out.println( "The input is:" );
30      System.out.println( input );
31      System.out.println( "-------------------------------------------" );
32      System.out.println( "The string tokens are:" );
33      System.out.println( token1 );
34      System.out.println( token2 );
35      System.out.println( token3 );
36      System.out.println( token4 );
37      System.out.println( "-------------------------------------------" );
38      System.out.println( "The int token is:" );
39      System.out.println( year );
40      System.out.println( "-------------------------------------------" );
41      System.out.println( "The double token is:" );
42      System.out.println( gpa );
43    }
44  }
```

Print the value of `gpa` and proceed to the next line

Scanner(System.in) Example

```
1  import java.util.Scanner;
2  // an example of using Scanner
3  public class ScannerExample {
4    public static void main( String[] args ) {
5      Scanner console;
6      String theWord;
7      int theWholeNumber;
8      double theRealNumber;
9      console = new Scanner( System.in );
10     System.out.print( "Enter a string: " );
11     theWord = console.next();
12     System.out.print( "Enter an int: " );
13     theWholeNumber = console.nextInt();
14     System.out.print( "Enter a double: " );
15     theRealNumber = console.nextDouble();
16     System.out.print( "You have entered " );
17     System.out.print( theWord );
18     System.out.print( ", " );
19     System.out.print( theWholeNumber );
20     System.out.print( ", and " );
21     System.out.print( theRealNumber );
22     System.out.println();
23   }
24 }
```

Variables are declared

Scanner(System.in) Example

```
1  import java.util.Scanner;
2  // an example of using Scanner
3  public class ScannerExample {
4    public static void main( String[] args ) {
5      Scanner console;
6      String theWord;
7      int theWholeNumber;
8      double theRealNumber;
9      console = new Scanner( System.in );
10     System.out.print( "Enter a string: " );
11     theWord = console.next();
12     System.out.print( "Enter an int: " );
13     theWholeNumber = console.nextInt();
14     System.out.print( "Enter a double: " );
15     theRealNumber = console.nextDouble();
16     System.out.print( "You have entered " );
17     System.out.print( theWord );
18     System.out.print( ", " );
19     System.out.print( theWholeNumber );
20     System.out.print( ", and " );
21     System.out.print( theRealNumber );
22     System.out.println();
23   }
24 }
```

The class Scanner is imported; a Scanner object is created

Scanner(System.in) Example

```
1  import java.util.Scanner;
2  // an example of using Scanner
3  public class ScannerExample {
4    public static void main( String[] args ) {
5      Scanner console;
6      String theWord;
7      int theWholeNumber;
8      double theRealNumber;
9      console = new Scanner( System.in );
10     System.out.print( "Enter a string: " );
11     theWord = console.next();
12     System.out.print( "Enter an int: " );
13     theWholeNumber = console.nextInt();
14     System.out.print( "Enter a double: " );
15     theRealNumber = console.nextDouble();
16     System.out.print( "You have entered " );
17     System.out.print( theWord );
18     System.out.print( ", " );
19     System.out.print( theWholeNumber );
20     System.out.print( ", and " );
21     System.out.print( theRealNumber );
22     System.out.println();
23   }
24 }
```

A String is read

Scanner(System.in) Example

```
1  import java.util.Scanner;
2  // an example of using Scanner
3  public class ScannerExample {
4    public static void main( String[] args ) {
5      Scanner console;
6      String theWord;
7      int theWholeNumber;
8      double theRealNumber;
9      console = new Scanner( System.in );
10     System.out.print( "Enter a string: " );
11     theWord = console.next();
12     System.out.print( "Enter an int: " );
13     theWholeNumber = console.nextInt();
14     System.out.print( "Enter a double: " );
15     theRealNumber = console.nextDouble();
16     System.out.print( "You have entered " );
17     System.out.print( theWord );
18     System.out.print( ", " );
19     System.out.print( theWholeNumber );
20     System.out.print( ", and " );
21     System.out.print( theRealNumber );
22     System.out.println();
23   }
24 }
```

An int is read

Scanner(System.in) Example

```
1  import java.util.Scanner;
2  // an example of using Scanner
3  public class ScannerExample {
4    public static void main( String[] args ) {
5      Scanner console;
6      String theWord;
7      int theWholeNumber;
8      double theRealNumber;
9      console = new Scanner( System.in );
10     System.out.print( "Enter a string: " );
11     theWord = console.next();
12     System.out.print( "Enter an int: " );
13     theWholeNumber = console.nextInt();
14     System.out.print( "Enter a double: " );
15     theRealNumber = console.nextDouble();
16     System.out.print( "You have entered " );
17     System.out.print( theWord );
18     System.out.print( ", " );
19     System.out.print( theWholeNumber );
20     System.out.print( ", and " );
21     System.out.print( theRealNumber );
22     System.out.println();
23   }
24 }
```

A double is read

Scanner(System.in) Example

```
1  import java.util.Scanner;
2  // an example of using Scanner
3  public class ScannerExample {
4    public static void main( String[] args ) {
5      Scanner console;
6      String theWord;
7      int theWholeNumber;
8      double theRealNumber;
9      console = new Scanner( System.in );
10     System.out.print( "Enter a string: " );
11     theWord = console.next();
12     System.out.print( "Enter an int: " );
13     theWholeNumber = console.nextInt();
14     System.out.print( "Enter a double: " );
15     theRealNumber = console.nextDouble();
16     System.out.print( "You have entered " );
17     System.out.print( theWord );
18     System.out.print( ", " );
19     System.out.print( theWholeNumber );
20     System.out.print( ", and " );
21     System.out.print( theRealNumber );
22     System.out.println();
23   }
24 }
```

An output is generated
Six items are printed in order and then a newline

Things to Note

- No output is generated until the "return key" (or "enter key") is entered (because space and tab can be erased)
  - Until the “return key” is entered, the tokens typed are pooled
  - It is possible to enter multiple inputs at a time with the space character as the delimiter
- The keys typed by the user are printed on screen (*echoed*) and so the output from the program execution (`System.out`), the error output (`System.err`), and the keyboard echo share the same single screen!

Example 2: Receiving input and calculate

```
1  import java.util.Scanner;
2  // an example of using Scanner
3  public class ScannerMath {
4    public static void main( String[] args ) {
5      Scanner console;
6      int int1, int2, product;
7      double real1, real2, quotient;
8
9      console = new Scanner( System.in );
```

Variable declarations

Example 2: Receiving input and calculate

```
1  import java.util.Scanner;
2  // an example of using Scanner
3  public class ScannerMath {
4    public static void main( String[] args ) {
5      Scanner console;
6      int int1, int2, product;
7      double real1, real2, quotient;
8
9      console = new Scanner( System.in );
```

Scanner import and creation

Example 2: Receiving input and calculate

```
11      System.out.print( "Enter int no. 1: " );
12      int1 = console.nextInt();
13      System.out.print( "Enter int no. 2: " );
14      int2 = console.nextInt();
15      product = int1 * int2;
16
17      System.out.print( "Received " );
18      System.out.print( int1 );
19      System.out.print( " and " );
20      System.out.println( int2 );
21      System.out.print( "The product is " );
22      System.out.println( int1 * int2 );
```

Prompt the user and receive `int1`

Example 2: Receiving input and calculate

```
11      System.out.print( "Enter int no. 1: " );
12      int1 = console.nextInt();
13      System.out.print( "Enter int no. 2: " );
14      int2 = console.nextInt();
15      product = int1 * int2;
16
17      System.out.print( "Received " );
18      System.out.print( int1 );
19      System.out.print( " and " );
20      System.out.println( int2 );
21      System.out.print( "The product is " );
22      System.out.println( int1 * int2 );
```

Prompt the user and receive `int2`

Example 2: Receiving input and calculate

```
11      System.out.print( "Enter int no. 1: " );
12      int1 = console.nextInt();
13      System.out.print( "Enter int no. 2: " );
14      int2 = console.nextInt();
15      product = int1 * int2;
16
17      System.out.print( "Received " );
18      System.out.print( int1 );
19      System.out.print( " and " );
20      System.out.println( int2 );
21      System.out.print( "The product is " );
22      System.out.println( int1 * int2 );
```

Compute the product

Example 2: Receiving input and calculate

```
11      System.out.print( "Enter int no. 1: " );
12      int1 = console.nextInt();
13      System.out.print( "Enter int no. 2: " );
14      int2 = console.nextInt();
15      product = int1 * int2;
16
17      System.out.print( "Received " );
18      System.out.print( int1 );
19      System.out.print( " and " );
20      System.out.println( int2 );
21      System.out.print( "The product is " );
22      System.out.println( int1 * int2 );
```

Generate the output to print the inputs and the result

Example 2: Receiving input and calculate

```
24      System.out.print( "Enter double no. 1: " );
25      real1 = console.nextDouble();
26      System.out.print( "Enter double no. 2: " );
27      real2 = console.nextDouble();
28      quotient = real1 / real2;
29
30      System.out.print( "Received " );
31      System.out.print( real1 );
32      System.out.print( " and " );
33      System.out.println( real2 );
34      System.out.print( "The quotient is " );
35      System.out.println( quotient );
36    }
37  }
```

Prompt the user and receive `real1`

Example 2: Receiving input and calculate

```
24      System.out.print( "Enter double no. 1: " );
25      real1 = console.nextDouble();
26      System.out.print( "Enter double no. 2: " );
27      real2 = console.nextDouble();
28      quotient = real1 / real2;
29
30      System.out.print( "Received " );
31      System.out.print( real1 );
32      System.out.print( " and " );
33      System.out.println( real2 );
34      System.out.print( "The quotient is " );
35      System.out.println( quotient );
36    }
37  }
```

Prompt the user and receive `real12`

Example 2: Receiving input and calculate

```
24      System.out.print( "Enter double no. 1: " );
25      real1 = console.nextDouble();
26      System.out.print( "Enter double no. 2: " );
27      real2 = console.nextDouble();
28      quotient = real1 / real2;
29
30      System.out.print( "Received " );
31      System.out.print( real1 );
32      System.out.print( " and " );
33      System.out.println( real2 );
34      System.out.print( "The quotient is " );
35      System.out.println( quotient );
36    }
37  }
```

Compute the quotient

Example 2: Receiving input and calculate

```
24      System.out.print( "Enter double no. 1: " );
25      real1 = console.nextDouble();
26      System.out.print( "Enter double no. 2: " );
27      real2 = console.nextDouble();
28      quotient = real1 / real2;
29
30      System.out.print( "Received " );
31      System.out.print( real1 );
32      System.out.print( " and " );
33      System.out.println( real2 );
34      System.out.print( "The quotient is " );
35      System.out.println( quotient );
36    }
37  }
```

Generate the output to print the inputs and the result

Computing the Area for a Number of Trapezoids

- The user tells the program how many trapezoids are there.
- The program receives from the user the bases and the height of each trapezoid and prints its area.
- Use a method for calculating the area given the two bases and height

TrapezoidScanner (part 1)

```
1  import java.util.*;
2  public class TrapezoidScanner {
3    /*
4     * compute the area of a trapezoid given
5     * the two bases and the height
6     */
7    public static double calculateArea
8        ( double base1, double base2, double height ) {
9      System.out.println( "....Entered calculateArea...." );
10     double area = ( base1 + base2 ) * height / 2;
11     System.out.println( "....Returning: " + area );
12     return area;
13   }
```

The header

TrapezoidScanner (part 1)

```
1  import java.util.*;
2  public class TrapezoidScanner {
3    /*
4     * compute the area of a trapezoid given
5     * the two bases and the height
6     */
7    public static double calculateArea
8        ( double base1, double base2, double height ) {
9      System.out.println( "....Entered calculateArea...." );
10     double area = ( base1 + base2 ) * height / 2;
11     System.out.println( "....Returning: " + area );
12     return area;
13   }
```

Print a message at the beginning

TrapezoidScanner (part 1)

```
1  import java.util.*;
2  public class TrapezoidScanner {
3    /*
4     * compute the area of a trapezoid given
5     * the two bases and the height
6     */
7    public static double calculateArea
8        ( double base1, double base2, double height ) {
9      System.out.println( "....Entered calculateArea...." );
10     double area = ( base1 + base2 ) * height / 2;
11     System.out.println( "....Returning: " + area );
12     return area;
13   }
```

Compute the area

TrapezoidScanner (part 1)

```
1  import java.util.*;
2  public class TrapezoidScanner {
3    /*
4     * compute the area of a trapezoid given
5     * the two bases and the height
6     */
7    public static double calculateArea
8        ( double base1, double base2, double height ) {
9      System.out.println( "....Entered calculateArea...." );
10     double area = ( base1 + base2 ) * height / 2;
11     System.out.println( "....Returning: " + area );
12     return area;
13   }
```

Print a message at the end and return

TrapezoidScanner (part 2)

```
15    public static void main( String[] args ) {
16      //---- console is a Scanner object out of the keyboard
17      Scanner console = new Scanner( System.in );
18
19      // the bases, the height (called altitude), and area
20      double baseOne, baseTwo, altitude, area;
21      int count;          // the # of repetitions
22
23      System.out.print( "Enter the # of trapezoids: " );
24      count = console.nextInt();
25      for ( int i = 1; i <= count; i ++ ) {
26        System.out.print( i + ": Enter bases and altitude: " );
27        baseOne = console.nextDouble();
28        baseTwo = console.nextDouble();
29        altitude = console.nextDouble();
30        area = calculateArea( baseOne, baseTwo, altitude );
31        System.out.println( "The area is: "
32            + Math.round( area * 10000 ) / 10000.0 );
33      }
34    }
35  }
```

Create a console

TrapezoidScanner (part 2)

```
15    public static void main( String[] args ) {
16      //---- console is a Scanner object out of the keyboard
17      Scanner console = new Scanner( System.in );
18
19      // the bases, the height (called altitude), and area
20      double baseOne, baseTwo, altitude, area;
21      int count;          // the # of repetitions
22
23      System.out.print( "Enter the # of trapezoids: " );
24      count = console.nextInt();
25      for ( int i = 1; i <= count; i ++ ) {
26        System.out.print( i + ": Enter bases and altitude: " );
27        baseOne = console.nextDouble();
28        baseTwo = console.nextDouble();
29        altitude = console.nextDouble();
30        area = calculateArea( baseOne, baseTwo, altitude );
31        System.out.println( "The area is: "
32            + Math.round( area * 10000 ) / 10000.0 );
33      }
34    }
35  }
```

Define variables

TrapezoidScanner (part 2)

```
15    public static void main( String[] args ) {
16      //---- console is a Scanner object out of the keyboard
17      Scanner console = new Scanner( System.in );
18
19      // the bases, the height (called altitude), and area
20      double baseOne, baseTwo, altitude, area;
21      int count;          // the # of repetitions
22
23      System.out.print( "Enter the # of trapezoids: " );
24      count = console.nextInt();
25      for ( int i = 1; i <= count; i ++ ) {
26        System.out.print( i + ": Enter bases and altitude: " );
27        baseOne = console.nextDouble();
28        baseTwo = console.nextDouble();
29        altitude = console.nextDouble();
30        area = calculateArea( baseOne, baseTwo, altitude );
31        System.out.println( "The area is: "
32            + Math.round( area * 10000 ) / 10000.0 );
33      }
34    }
35  }
```

Receive the number of trapezoids

TrapezoidScanner (part 2)

```
15    public static void main( String[] args ) {
16      //---- console is a Scanner object out of the keyboard
17      Scanner console = new Scanner( System.in );
18
19      // the bases, the height (called altitude), and area
20      double baseOne, baseTwo, altitude, area;
21      int count;          // the # of repetitions
22
23      System.out.print( "Enter the # of trapezoids: " );
24      count = console.nextInt();
25      for ( int i = 1; i <= count; i ++ ) {
26        System.out.print( i + ": Enter bases and altitude: " );
27        baseOne = console.nextDouble();
28        baseTwo = console.nextDouble();
29        altitude = console.nextDouble();
30        area = calculateArea( baseOne, baseTwo, altitude );
31        System.out.println( "The area is: "
32            + Math.round( area * 10000 ) / 10000.0 );
33      }
34    }
35  }
```

For loop

TrapezoidScanner (part 2)

```
15    public static void main( String[] args ) {
16      //---- console is a Scanner object out of the keyboard
17      Scanner console = new Scanner( System.in );
18
19      // the bases, the height (called altitude), and area
20      double baseOne, baseTwo, altitude, area;
21      int count;          // the # of repetitions
22
23      System.out.print( "Enter the # of trapezoids: " );
24      count = console.nextInt();
25      for ( int i = 1; i <= count; i ++ ) {
26        System.out.print( i + ": Enter bases and altitude: " );
27        baseOne = console.nextDouble();
28        baseTwo = console.nextDouble();
29        altitude = console.nextDouble();
30        area = calculateArea( baseOne, baseTwo, altitude );
31        System.out.println( "The area is: "
32            + Math.round( area * 10000 ) / 10000.0 );
33      }
34    }
35  }
```

Prompt the user for information and receive data

TrapezoidScanner (part 2)

```
15    public static void main( String[] args ) {
16      //---- console is a Scanner object out of the keyboard
17      Scanner console = new Scanner( System.in );
18
19      // the bases, the height (called altitude), and area
20      double baseOne, baseTwo, altitude, area;
21      int count;          // the # of repetitions
22
23      System.out.print( "Enter the # of trapezoids: " );
24      count = console.nextInt();
25      for ( int i = 1; i <= count; i ++ ) {
26        System.out.print( i + ": Enter bases and altitude: " );
27        baseOne = console.nextDouble();
28        baseTwo = console.nextDouble();
29        altitude = console.nextDouble();
30        area = calculateArea( baseOne, baseTwo, altitude );
31        System.out.println( "The area is: "
32            + Math.round( area * 10000 ) / 10000.0 );
33      }
34    }
35  }
```

Calculate area and generate output

Table of Contents

Class String

String is an object data type, but since String is a data type that is quite often used, Java provides some short-cuts

```
String w = new String("ABC");
  String x = new String(w);
```

String Methods

- As we have seen earlier, Strings can be concatenated with the "+" operator
- Java provides many methods for dealing with String objects
  All such methods take the form of:

```
STRING-OBJECT.METHOD-NAME(PARAMETERS);
```

- The things that can be accomplished by such operations are:
  - Obtaining information about the String object, e.g., the length, substrings, and a symbol at a particular position
  - Comparison with another String object
  - Result of pattern search
  - Transformation by pattern replacement

  All these methods preserve the String object to which they are applied

The Indices Inside a String

- The letters in a String object are assigned positions $0, 1, 2, \ldots$
- Suppose we are searching for a pattern (which is also a String) w having length L in a String type s
- We say that w occurs at position p, if the pattern in s of L letters starting at position p matches w

String Methods for Information Extraction

Let `s` be a String

- `s.length()` : returns as int the number of symbols in `s`
- `s.charAt(int index)` : returns the `char` at position `index`, where `index` is between 0 and `s.length()-1`

Primitive Data Type char

char is the type that represents a letter that can be typed using a keyboard

Primitive Data Type char

char is the type that represents a letter that can be typed using a keyboard

- The numbers assigned to '0' - '9' are consecutive (48-57)
- The numbers assigned to 'A' - 'Z' are consecutive (65 - 90)
- The numbers assigned to 'a' - 'z' are consecutive (97 - 122)

Primitive Data Type char

char is the type that represents a letter that can be typed using a keyboard

- The numbers assigned to '0' - '9' are consecutive (48-57)
- The numbers assigned to 'A' - 'Z' are consecutive (65 - 90)
- The numbers assigned to 'a' - 'z' are consecutive (97 - 122)

Given a `char c`:

- Condition `c >= '0' && c<='9'` tests whether `c` is a numeral
- Condition `c >= 'A' && c<='Z'` tests whether `c` is an uppercase letter
- Condition `c >= 'a' && c<='z'` tests whether `c` is a lowercase letter

Primitive Data Type char

char is the type that represents a letter that can be typed using a keyboard

- The numbers assigned to '0' - '9' are consecutive (48-57)
- The numbers assigned to 'A' - 'Z' are consecutive (65 - 90)
- The numbers assigned to 'a' - 'z' are consecutive (97 - 122)

Given a `char c`:

- Condition `c >= '0' && c<='9'` tests whether `c` is a numeral
- Condition `c >= 'A' && c<='Z'` tests whether `c` is an uppercase letter
- Condition `c >= 'a' && c<='z'` tests whether `c` is a lowercase letter

Given a `char c`:

- Given two chars `c` and `d` you can calculate by `c - d` their positional difference

Methods indexOf and lastIndexOf

Let `s` and `w` be Strings and let `p` be an int

- `s.indexOf(w)` : returns the position of the first match of `w` in `s`; if there is no match, it returns -1
- `s.lastIndexOf(w)` : returns the position of the last match of `w` in `s`; if there is no match, it returns -1
- `s.indexOf(w, p)` : returns the position of the first match of `w` in `s` at or after position `p`; if there is no such a match, it returns -1
- `s.lastIndexOf(w, p)` : returns the position of the last match of `w` in `s` under the restriction that the matches have to be at position `p` or lower; if there is no such a match, it returns -1

Methods indexOf and lastIndexOf

Let `s` and `w` be Strings and let `p` be an int

- `s.indexOf(w)` : returns the position of the first match of `w` in `s`; if there is no match, it returns -1
- `s.lastIndexOf(w)` : returns the position of the last match of `w` in `s`; if there is no match, it returns -1
- `s.indexOf(w, p)` : returns the position of the first match of `w` in `s` at or after position `p`; if there is no such a match, it returns -1
- `s.lastIndexOf(w, p)` : returns the position of the last match of `w` in `s` under the restriction that the matches have to be at position `p` or lower; if there is no such a match, it returns -1

Note that:
`s.indexOf(w)` is equivalent to `s.indexOf(w,0)` and
`s.lastIndexOf(w)` is equivalent to `s.lastIndexOf(w,0)`

IndexOf Example

- Receive from the user an input sentence `input` (use `nextLine()` to receive a sequence possibly with whitespace)
- Present the input with the character position values
- Receive from the user a pattern `pattern` to search for (use `nextLine()` to receive a pattern possibly with whitespace)
- Receive from the user an index value `p`
- Prints the result of four methods:
  - `input.indexOf( pattern, p)`
  - `input.indexOf( pattern)`
  - `input.lastIndexOf( pattern, p)`
  - `input.lastIndexOf( pattern )`

IndexOf.java

```
1  import java.util.*;
2  public class IndexOf {
3    public static void main( String[] args ) {
4      Scanner console = new Scanner( System.in );
5      System.out.print( "Enter the input: " );
6      String input = console.nextLine();
7      System.out.print( "Enter the pattern: " );
8      String pattern = console.nextLine();
9      System.out.print( "Enter the position " );
10     int p = console.nextInt();
11
12     System.out.print( "input.indexOf( pattern, p ) is " );
13     System.out.println( input.indexOf( pattern, p ) );
14     System.out.print( "input.indexOf( pattern ) is " );
15     System.out.println( input.indexOf( pattern ) );
16     System.out.print( "input.lastIndexOf( pattern, p ) is " );
17     System.out.println( input.lastIndexOf( pattern, p ) );
18     System.out.print( "input.lastIndexOf( pattern ) is " );
19     System.out.println( input.lastIndexOf( pattern ) );
20   }
21 }
```

Keyboard `Scanner`

IndexOf.java

```
1  import java.util.*;
2  public class IndexOf {
3    public static void main( String[] args ) {
4      Scanner console = new Scanner( System.in );
5      System.out.print( "Enter the input: " );
6      String input = console.nextLine();
7      System.out.print( "Enter the pattern: " );
8      String pattern = console.nextLine();
9      System.out.print( "Enter the position " );
10     int p = console.nextInt();
11
12     System.out.print( "input.indexOf( pattern, p ) is " );
13     System.out.println( input.indexOf( pattern, p ) );
14     System.out.print( "input.indexOf( pattern ) is " );
15     System.out.println( input.indexOf( pattern ) );
16     System.out.print( "input.lastIndexOf( pattern, p ) is " );
17     System.out.println( input.lastIndexOf( pattern, p ) );
18     System.out.print( "input.lastIndexOf( pattern ) is " );
19     System.out.println( input.lastIndexOf( pattern ) );
20   }
21 }
```

Receive input after prompting using `nextLine()`

IndexOf.java

```
1  import java.util.*;
2  public class IndexOf {
3    public static void main( String[] args ) {
4      Scanner console = new Scanner( System.in );
5      System.out.print( "Enter the input: " );
6      String input = console.nextLine();
7      System.out.print( "Enter the pattern: " );
8      String pattern = console.nextLine();
9      System.out.print( "Enter the position " );
10     int p = console.nextInt();
11
12     System.out.print( "input.indexOf( pattern, p ) is " );
13     System.out.println( input.indexOf( pattern, p ) );
14     System.out.print( "input.indexOf( pattern ) is " );
15     System.out.println( input.indexOf( pattern ) );
16     System.out.print( "input.lastIndexOf( pattern, p ) is " );
17     System.out.println( input.lastIndexOf( pattern, p ) );
18     System.out.print( "input.lastIndexOf( pattern ) is " );
19     System.out.println( input.lastIndexOf( pattern ) );
20   }
21 }
```

Receive pattern after prompting using `nextLine()`

IndexOf.java

```
1  import java.util.*;
2  public class IndexOf {
3    public static void main( String[] args ) {
4      Scanner console = new Scanner( System.in );
5      System.out.print( "Enter the input: " );
6      String input = console.nextLine();
7      System.out.print( "Enter the pattern: " );
8      String pattern = console.nextLine();
9      System.out.print( "Enter the position " );
10     int p = console.nextInt();
11
12     System.out.print( "input.indexOf( pattern, p ) is " );
13     System.out.println( input.indexOf( pattern, p ) );
14     System.out.print( "input.indexOf( pattern ) is " );
15     System.out.println( input.indexOf( pattern ) );
16     System.out.print( "input.lastIndexOf( pattern, p ) is " );
17     System.out.println( input.lastIndexOf( pattern, p ) );
18     System.out.print( "input.lastIndexOf( pattern ) is " );
19     System.out.println( input.lastIndexOf( pattern ) );
20   }
21 }
```

Receive the position value

IndexOf.java

```
1  import java.util.*;
2  public class IndexOf {
3    public static void main( String[] args ) {
4      Scanner console = new Scanner( System.in );
5      System.out.print( "Enter the input: " );
6      String input = console.nextLine();
7      System.out.print( "Enter the pattern: " );
8      String pattern = console.nextLine();
9      System.out.print( "Enter the position " );
10     int p = console.nextInt();
11
12     System.out.print( "input.indexOf( pattern, p ) is " );
13     System.out.println( input.indexOf( pattern, p ) );
14     System.out.print( "input.indexOf( pattern ) is " );
15     System.out.println( input.indexOf( pattern ) );
16     System.out.print( "input.lastIndexOf( pattern, p ) is " );
17     System.out.println( input.lastIndexOf( pattern, p ) );
18     System.out.print( "input.lastIndexOf( pattern ) is " );
19     System.out.println( input.lastIndexOf( pattern ) );
20   }
21 }
```

Print what is going to be executed

IndexOf.java

```
1  import java.util.*;
2  public class IndexOf {
3    public static void main( String[] args ) {
4      Scanner console = new Scanner( System.in );
5      System.out.print( "Enter the input: " );
6      String input = console.nextLine();
7      System.out.print( "Enter the pattern: " );
8      String pattern = console.nextLine();
9      System.out.print( "Enter the position " );
10     int p = console.nextInt();
11
12     System.out.print( "input.indexOf( pattern, p ) is " );
13     System.out.println( input.indexOf( pattern, p ) );
14     System.out.print( "input.indexOf( pattern ) is " );
15     System.out.println( input.indexOf( pattern ) );
16     System.out.print( "input.lastIndexOf( pattern, p ) is " );
17     System.out.println( input.lastIndexOf( pattern, p ) );
18     System.out.print( "input.lastIndexOf( pattern ) is " );
19     System.out.println( input.lastIndexOf( pattern ) );
20   }
21 }
```

Print the result of execution

String Methods for Creating a New String

- `toUpperCase()` : returns a new string by changing each lowercase letter to the corresponding uppercase letter
- `toLowerCase()` : returns a new string by changing each uppercase letter to the corresponding lowercase letter
- `substring(int startIndex)` : returns a new string by eliminating all the letters at position less than `startIndex`
- `substring(int startIndex, int endIndex)` : returns a new string by eliminating all the letters at position less than `startIndex` and all the letters at position greater than or equal to `endIndex`
- `s.replace(String x, String y)` : returns the String created from `s` by concurrently replacing all occurrences of String `x` with String `y`

ModifyString.java

```
1  import java.util.*;
2  public class ModifyString {
3    public static void main(String[] args) {
4      Scanner console;
5      String input, pat1, pat2;
6      int pos1, pos2;
7      console = new Scanner(System.in);
8
9      System.out.print( "Enter the input String : " );
10     input = console.nextLine();
11     System.out.print( "Enter pattern 1: " );
12     pat1 = console.nextLine();
13     System.out.print( "Enter pattern 2: " );
14     pat2 = console.nextLine();
15
16     System.out.print( "Enter positions 1 and 2: " );
17     pos1 = console.nextInt();
18     pos2 = console.nextInt();
```

Keyboard `Scanner` setup

ModifyString.java

```
1  import java.util.*;
2  public class ModifyString {
3    public static void main(String[] args) {
4      Scanner console;
5      String input, pat1, pat2;
6      int pos1, pos2;
7      console = new Scanner(System.in);
8
9      System.out.print( "Enter the input String : " );
10     input = console.nextLine();
11     System.out.print( "Enter pattern 1: " );
12     pat1 = console.nextLine();
13     System.out.print( "Enter pattern 2: " );
14     pat2 = console.nextLine();
15
16     System.out.print( "Enter positions 1 and 2: " );
17     pos1 = console.nextInt();
18     pos2 = console.nextInt();
```

Other variables

ModifyString.java

```
1  import java.util.*;
2  public class ModifyString {
3    public static void main(String[] args) {
4      Scanner console;
5      String input, pat1, pat2;
6      int pos1, pos2;
7      console = new Scanner(System.in);
8
9      System.out.print( "Enter the input String : " );
10     input = console.nextLine();
11     System.out.print( "Enter pattern 1: " );
12     pat1 = console.nextLine();
13     System.out.print( "Enter pattern 2: " );
14     pat2 = console.nextLine();
15
16     System.out.print( "Enter positions 1 and 2: " );
17     pos1 = console.nextInt();
18     pos2 = console.nextInt();
```

Prompt the user to enter the input and receive it using `nextLine`

ModifyString.java

```
1  import java.util.*;
2  public class ModifyString {
3    public static void main(String[] args) {
4      Scanner console;
5      String input, pat1, pat2;
6      int pos1, pos2;
7      console = new Scanner(System.in);
8
9      System.out.print( "Enter the input String : " );
10     input = console.nextLine();
11     System.out.print( "Enter pattern 1: " );
12     pat1 = console.nextLine();
13     System.out.print( "Enter pattern 2: " );
14     pat2 = console.nextLine();
15
16     System.out.print( "Enter positions 1 and 2: " );
17     pos1 = console.nextInt();
18     pos2 = console.nextInt();
```

Prompt the user to enter pattern 1 and receive it using `nextLine`

ModifyString.java

```
1  import java.util.*;
2  public class ModifyString {
3    public static void main(String[] args) {
4      Scanner console;
5      String input, pat1, pat2;
6      int pos1, pos2;
7      console = new Scanner(System.in);
8
9      System.out.print( "Enter the input String : " );
10     input = console.nextLine();
11     System.out.print( "Enter pattern 1: " );
12     pat1 = console.nextLine();
13     System.out.print( "Enter pattern 2: " );
14     pat2 = console.nextLine();
15
16     System.out.print( "Enter positions 1 and 2: " );
17     pos1 = console.nextInt();
18     pos2 = console.nextInt();
```

Prompt the user to enter pattern 2 and receive it using `nextLine`

ModifyString.java

```
1  import java.util.*;
2  public class ModifyString {
3    public static void main(String[] args) {
4      Scanner console;
5      String input, pat1, pat2;
6      int pos1, pos2;
7      console = new Scanner(System.in);
8
9      System.out.print( "Enter the input String : " );
10     input = console.nextLine();
11     System.out.print( "Enter pattern 1: " );
12     pat1 = console.nextLine();
13     System.out.print( "Enter pattern 2: " );
14     pat2 = console.nextLine();
15
16     System.out.print( "Enter positions 1 and 2: " );
17     pos1 = console.nextInt();
18     pos2 = console.nextInt();
```

Prompt the user to enter two numbers and receive them using `nextInt` twice

ModifyString.java

```
19
20      System.out.println();
21      System.out.println( "input.toLowerCase() is " );
22      System.out.println( input.toLowerCase() );
23      System.out.println();
24      System.out.println( "input.toUpperCase() is " );
25      System.out.println( input.toUpperCase() );
26      System.out.println();
27      System.out.println( "input.substring( pos1 ) is " );
28      System.out.println( input.substring( pos2 ) );
29      System.out.println();
30      System.out.println( "input.substring( pos1, pos2 ) is " );
31      System.out.println( input.substring( pos1, pos2 ) );
32      System.out.println();
33      System.out.println( "input.replace( pat1, pat2 ) is " );
34      System.out.println( input.replace( pat1, pat2 ) );
35    }
36  }
```

Inform which method to execute

ModifyString.java

```
19
20      System.out.println();
21      System.out.println( "input.toLowerCase() is " );
22      System.out.println( input.toLowerCase() );
23      System.out.println();
24      System.out.println( "input.toUpperCase() is " );
25      System.out.println( input.toUpperCase() );
26      System.out.println();
27      System.out.println( "input.substring( pos1 ) is " );
28      System.out.println( input.substring( pos2 ) );
29      System.out.println();
30      System.out.println( "input.substring( pos1, pos2 ) is " );
31      System.out.println( input.substring( pos1, pos2 ) );
32      System.out.println();
33      System.out.println( "input.replace( pat1, pat2 ) is " );
34      System.out.println( input.replace( pat1, pat2 ) );
35    }
36  }
```

Announce the outcome

String Methods for Comparison

Let `s` be a String

- `s.compareTo(String otherString)` : Compares `s` and `otherString` in the dictionary order and returns the result:
  - a strictly negative integer if `s` is strictly smaller than `otherString`,
  - a strictly positive integer if `s` is strictly greater than `otherString`,
  - 0 if they are equal to each other
- `s.equals(String otherString)` : compares `s` and `otherString` for equality and returns `true` if they are equal to each other and `false` otherwise
- `s.startsWith(String otherString)` : returns `true` if `s` starts with `otherString` and `false` otherwise
- `s.endsWith(String otherString)` : returns `true` if `s` ends with `otherString` and `false` otherwise

Comparison Example: Receive Input from User and Perform Tests

```
1  import java.util.Scanner;
2  public class OtherString {
3    public static void main( String[] args ) {
4      Scanner console = new Scanner( System.in );
5      System.out.print( "Enter input1: " );
6      String input1 = console.nextLine();
7      System.out.print( "Enter input2: " );
8      String input2 = console.nextLine();
9
10     System.out.print( "input1.compareTo( input2 ) is " );
11     System.out.println( input1.compareTo( input2 ) );
12     System.out.print( "input1.equals( input2 ) is " );
13     System.out.println( input1.equals( input2 ) );
14     System.out.print( "input1.startsWith( input2 ) is " );
15     System.out.println( input1.startsWith( input2 ) );
16     System.out.print( "input1.endsWith( input2 ) is " );
17     System.out.println( input1.endsWith( input2 ) );
18   }
19 }
```

Receive input 1 using `nextLine`

Comparison Example: Receive Input from User and Perform Tests

```
1  import java.util.Scanner;
2  public class OtherString {
3    public static void main( String[] args ) {
4      Scanner console = new Scanner( System.in );
5      System.out.print( "Enter input1: " );
6      String input1 = console.nextLine();
7      System.out.print( "Enter input2: " );
8      String input2 = console.nextLine();
9
10     System.out.print( "input1.compareTo( input2 ) is " );
11     System.out.println( input1.compareTo( input2 ) );
12     System.out.print( "input1.equals( input2 ) is " );
13     System.out.println( input1.equals( input2 ) );
14     System.out.print( "input1.startsWith( input2 ) is " );
15     System.out.println( input1.startsWith( input2 ) );
16     System.out.print( "input1.endsWith( input2 ) is " );
17     System.out.println( input1.endsWith( input2 ) );
18   }
19 }
```

Receive input 2 using `nextLine`

Comparison Example: Receive Input from User and Perform Tests

```
1  import java.util.Scanner;
2  public class OtherString {
3    public static void main( String[] args ) {
4      Scanner console = new Scanner( System.in );
5      System.out.print( "Enter input1: " );
6      String input1 = console.nextLine();
7      System.out.print( "Enter input2: " );
8      String input2 = console.nextLine();
9
10     System.out.print( "input1.compareTo( input2 ) is " );
11     System.out.println( input1.compareTo( input2 ) );
12     System.out.print( "input1.equals( input2 ) is " );
13     System.out.println( input1.equals( input2 ) );
14     System.out.print( "input1.startsWith( input2 ) is " );
15     System.out.println( input1.startsWith( input2 ) );
16     System.out.print( "input1.endsWith( input2 ) is " );
17     System.out.println( input1.endsWith( input2 ) );
18   }
19 }
```

Announce what is going to be executed

Comparison Example: Receive Input from User and Perform Tests

```
1  import java.util.Scanner;
2  public class OtherString {
3    public static void main( String[] args ) {
4      Scanner console = new Scanner( System.in );
5      System.out.print( "Enter input1: " );
6      String input1 = console.nextLine();
7      System.out.print( "Enter input2: " );
8      String input2 = console.nextLine();
9
10     System.out.print( "input1.compareTo( input2 ) is " );
11     System.out.println( input1.compareTo( input2 ) );
12     System.out.print( "input1.equals( input2 ) is " );
13     System.out.println( input1.equals( input2 ) );
14     System.out.print( "input1.startsWith( input2 ) is " );
15     System.out.println( input1.startsWith( input2 ) );
16     System.out.print( "input1.endsWith( input2 ) is " );
17     System.out.println( input1.endsWith( input2 ) );
18   }
19 }
```

Print the value generated