

How to scan values using Scanner in java?

Introduction

The **java.util.Scanner** class is a simple text scanner which can parse primitive types and strings using regular expressions. Following are the important points about Scanner:

- A Scanner breaks its input into tokens using a delimiter pattern, which by default matches whitespace.
- A scanning operation may block waiting for input.
- A Scanner is not safe for multithreaded use without external synchronization.

Java Scanner Method Examples

The following are the list of Scanner methods which we can use to work out in complicated parsing of input.

return	Method	Description
void	<code>close()</code>	Closes the scanner object.
Pattern	<code>delimiter()</code>	Returns the Pattern the Scanner object is currently using to match delimiters.
String	<code>findInLine(Pattern pattern)</code>	This method returns a String object that satisfies the Pattern object specified as method argument.
String	<code>findInLine(String pattern)</code>	Attempts to find the next occurrence of a pattern constructed from the specified string, ignoring delimiters.
String	<code>findWithinHorizon(Pattern pattern, int horizon)</code>	Attempts to find the next occurrence of the specified pattern.
String	<code>findWithinHorizon(String pattern, int horizon)</code>	This method simply attempts to find the next occurrence of a pattern input ignoring delimiter
boolean	<code>hasNext()</code>	Returns true if this scanner has another token in its input.
boolean	<code>hasNext(Pattern pattern)</code>	Returns true if the next complete token matches the specified pattern.
boolean	<code>hasNext(String pattern)</code>	Returns true if the next token matches the pattern constructed from the specified string.
boolean	<code>hasNextBigDecimal()</code>	Returns true if the next token in this scanner's input can be interpreted as a BigDecimal using the <code>nextBigDecimal()</code> method.

return	Method	Description
boolean	hasNextBigInteger()	Returns true if the next token in this scanner's input can be interpreted as a BigInteger in the default radix using the nextBigInteger() method.
boolean	hasNextBigInteger(int radix)	Returns true if the next token in this scanner's input can be interpreted as a BigInteger in the specified radix using the nextBigInteger() method.
boolean	<u>hasNextBoolean()</u>	This method checks if the Scanner object has boolean data type on its buffer.
boolean	<u>hasNextByte()</u>	This method returns true if the next byte on the scanner buffer can be translated to byte data type otherwise false.
boolean	<u>hasNextByte(int radix)</u>	Returns true if the next token in this scanner's input can be interpreted as a byte value in the specified radix using the nextByte() method.
boolean	hasNextDouble()	Returns true if the next token in this scanner's input can be interpreted as a double value using the nextDouble() method.
boolean	hasNextFloat()	Returns true if the next token in this scanner's input can be interpreted as a float value using the nextFloat() method.
boolean	<u>hasNextInt()</u>	Returns true if the next token in this scanner's input can be interpreted as an int value in the default radix using the nextInt() method.
boolean	<u>hasNextInt(int radix)</u>	This method returns boolean, true if the token can be interpreted as int data type with respect to the radix used by the scanner object otherwise false.
boolean	<u>hasNextLine()</u>	This method returns a boolean data type which corresponds to the existence of new line on the String tokens which the Scanner object holds.
boolean	<u>hasNextLong()</u>	Returns true if the next token in this scanner's input can be interpreted as a long value in the default radix using the nextLong() method.
boolean	<u>hasNextLong(int radix)</u>	Returns true if the next token in this scanner's input can be interpreted as a long value in the specified radix using the nextLong() method.
boolean	<u>hasNextShort()</u>	Returns true if the next token in this scanner's input can be interpreted as a short value in the default radix using the nextShort() method.

return	Method	Description
boolean	<u>hasNextShort(int radix)</u>	This method returns boolean, true if the token can be interpreted as short data type with respect to the radix used by the scanner object otherwise false.
IOException	ioException()	Returns the IOException last thrown by this Scanner's underlying Readable.
Locale	<u>locale()</u>	This method returns a Locale which the Scanner class is using.
MatchResult	<u>match()</u>	This method returns a MatchResult object which corresponds to the result of the last operation by the scanner object.
String	<u>next()</u>	Finds and returns the next complete token from this scanner.
String	<u>next(Pattern pattern)</u>	Returns the next token if it matches the specified pattern.
String	<u>next(String pattern)</u>	Returns the next token if it matches the pattern constructed from the specified string.
BigDecimal	nextBigDecimal()	Scans the next token of the input as a BigDecimal.
BigInteger	nextBigInteger()	Scans the next token of the input as a BigInteger.
BigInteger	nextBigInteger(int radix)	Scans the next token of the input as a BigInteger.
boolean	<u>nextBoolean()</u>	Scans the next token of the input into a boolean value and returns that value.
byte	<u>nextByte()</u>	Scans the next token of the input as a byte.
byte	<u>nextByte(int radix)</u>	Scans the next token of the input as a byte.
double	nextDouble()	Scans the next token of the input as a double.
float	nextFloat()	Scans the next token of the input as a float.
int	<u>nextInt()</u>	Scans the next token of the input as an int.
int	<u>nextInt(int radix)</u>	Scans the next token of the input as an int.
String	<u>nextLine()</u>	Advances this scanner past the current line and returns the input that was skipped.
long	nextLong()	Scans the next token of the input as a long.
long	nextLong(int radix)	Scans the next token of the input as a long.
short	<u>nextShort()</u>	Scans the next token of the input as a short.
short	<u>nextShort(int radix)</u>	Scans the next token of the input as a short.
int	<u>radix()</u>	Returns this scanner's default radix.

return	Method	Description
void	remove()	The remove operation is not supported by this implementation of Iterator.
Scanner	reset()	Resets this scanner.
Scanner	<u>skip(Pattern pattern)</u>	Skips input that matches the specified pattern, ignoring delimiters.
Scanner	<u>skip(String pattern)</u>	Skips input that matches a pattern constructed from the specified string.
String	toString()	Returns the string representation of this Scanner.
Scanner	<u>useDelimiter(Pattern pattern)</u>	Sets this scanner's delimiting pattern to the specified pattern.
Scanner	<u>useDelimiter(String pattern)</u>	Sets this scanner's delimiting pattern to a pattern constructed from the specified String.
Scanner	<u>useLocale(Locale locale)</u>	Sets this scanner's locale to the specified locale.
Scanner	<u>useRadix(int radix)</u>	Sets this scanner's default radix to the specified radix.

References:

- 1) <http://javatutorialhq.com/java/util/scanner-class-tutorial/>
- 2) http://www.tutorialspoint.com/java/util/java_util_scanner.htm

Example:

```
import java.util.Scanner;
class Example4 {
public static void main(String args[]) {

    Scanner sc = new Scanner(System.in);

    byte b;
    short s;
    int i;
    long l;
    float f;
    Double d;
    char c;
    String str1, str2;

    System.out.println("Enter the values:");
    b=sc.nextByte();
    s=sc.nextShort();
    i=sc.nextInt();
    l=sc.nextLong();
    f=sc.nextFloat();
    d=sc.nextDouble();

    c=sc.next().charAt(0);

    str1=sc.next(); //w/o white space

    Scanner texts = new Scanner(System.in);
    str2=texts.nextLine();

    System.out.println("");
    System.out.println("Entered values:");
    System.out.println(b);
    System.out.println(s);
    System.out.println(i);
    System.out.println(l);
    System.out.println(f);
    System.out.println(d);
    System.out.println(c);
    System.out.println(str1);
```

```
System.out.println(str2);

sc.close();
}
```