Class SIn

java.lang.Object
    SIn

```
public class SIn
extends java.lang.Object
```

Class for simple console input. A class designed primarily for simple keyboard input of the form one input value per line. If the user enters an improper input, i.e., an input of the wrong type or a blank line, then the user is prompted to reenter the input and given a brief explanation of what is required. Also includes some additional methods to input single numbers, words, and characters, without going to the next line.

Constructor Summary

Constructors

Constructor	Description
`SIn()`	

Method Summary

All Methods | **Static Methods** | **Concrete Methods**

Modifier and Type	Method	Description
`static int`	`read()`	Reads the first byte in the input stream and returns that byte as an int.
`static char`	`readChar()`	Reads the next input character and returns that character.
`static double`	`readDouble()`	Precondition: The next input consists of a double value, possibly preceded by white space, but definitely followed by white space.
`static float`	`readFloat()`	Precondition: The next input consists of a float value, possibly preceded by white space, but definitely followed by white space.
`static int`	`readInt()`	Precondition: The next input in the stream consists of an int value, possibly preceded by white space, but definitely followed by white space.
`static java.lang.String`	`readLine()`	Reads a line of text and returns that line as a String value.

Modifier and Type	Method	Description
`static boolean`	`readLineBoolean()`	Input should consists of a single word on a line, possibly surrounded by white space.
`static double`	`readLineDouble()`	Precondition: The user has entered a number of type double on a line by itself, except that there may be white space before and/or after the number.
`static float`	`readLineFloat()`	Precondition: The user has entered a number of type float on a line by itself, except that there may be white space before and/or after the number.
`static int`	`readLineInt()`	Precondition: The user has entered a number of type int on a line by itself, except that there may be white space before and/or after the number.
`static long`	`readLineLong()`	Precondition: The user has entered a number of type long on a line by itself, except that there may be white space before and/or after the number.
`static char`	`readLineNonwhiteChar()`	Reads the first Nonwhite character on a line and returns that character.
`static java.lang.String`	`readLineWord()`	Reads the first string of nonwhite characters on a line and returns that string.
`static long`	`readLong()`	Precondition: The next input consists of a long value, possibly preceded by white space, but definitely followed by white space.
`static char`	`readNonwhiteChar()`	Reads the next nonwhite input character and returns that character.
`static java.lang.String`	`readWord()`	Reads the first string of nonwhite characters and returns that string.

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Details

SIn

```
public SIn()
```

Method Details

readLine

```
public static java.lang.String readLine()
```

Reads a line of text and returns that line as a String value. The end of a line must be indicated either by a new line character '\n' or by a carriage return '\r' followed by a new line character '\n'. (Almost all systems do this automatically. So, you need not worry about this detail.) Neither the '\n', nor the '\r' if present, are part of the string returned. This will read the rest of a line if the line is already partially read.

readLineWord

```
public static java.lang.String readLineWord()
```

Reads the first string of nonwhite characters on a line and returns that string. The rest of the line is discarded. If the line contains only white space, then the user is asked to reenter the line.

readLineInt

```
public static int readLineInt()
```

Precondition: The user has entered a number of type int on a line by itself, except that there may be white space before and/or after the number. Action: Reads and returns the number as a value of type int. The rest of the line is discarded. If the input is not entered correctly, then in most cases, the user will be asked to reenter the input. In particular, this applies to incorrect number formats and blank lines.

readLineLong

```
public static long readLineLong()
```

Precondition: The user has entered a number of type long on a line by itself, except that there may be white space before and/or after the number. Action: Reads and returns the number as a value of type long. The rest of the line is discarded. If the input is not entered correctly, then in most cases, the user will be asked to reenter the input. In particular, this applies to incorrect number formats and blank lines.

readLineDouble

```
public static double readLineDouble()
```

Precondition: The user has entered a number of type double on a line by itself, except that there may be white space before and/or after the number. Action: Reads and returns the number as a value of type double. The rest of the line is discarded. If the input is not entered correctly, then in most cases, the user will be asked to reenter the input. In particular, this applies to incorrect number formats and blank lines.

readLineFloat

```
public static float readLineFloat()
```

Precondition: The user has entered a number of type float on a line by itself, except that there may be white space before and/or after the number. Action: Reads and returns the number as a value of type float. The rest of the line is discarded. If the input is not entered correctly, then in most cases, the user will be asked to reenter the input. In particular, this applies to incorrect number formats and blank lines.

readLineNonwhiteChar

```
public static char readLineNonwhiteChar()
```

Reads the first Nonwhite character on a line and returns that character. The rest of the line is discarded. If the line contains only white space, then the user is asked to reenter the line.

readLineBoolean

```
public static boolean readLineBoolean()
```

Input should consists of a single word on a line, possibly surrounded by white space. The line is read and discarded. If the input word is "true" or "t", then true is returned. If the input word is "false" or "f", then false is returned. Uppercase and lowercase letters are considered equal. If the user enters anything else (e.g. multiple words or different words), then, the user is asked to reenter the input.

readChar

```
public static char readChar()
```

Reads the next input character and returns that character. The next read takes place on the same line where this one left off.

readNonwhiteChar

```
public static char readNonwhiteChar()
```

Reads the next nonwhite input character and returns that character. The next read takes place immediately after the character read.

readInt

```
public static int readInt()
                   throws java.lang.NumberFormatException
```

Precondition: The next input in the stream consists of an int value, possibly preceded by white space, but definitely followed by white space. Action: Reads the first string of nonwhite characters and returns the int value it represents. Discards the first whitespace character after the word. The next read takes place immediately after the discarded whitespace. In particular, if the word is at the end of a line, the next reading will take place starting on the next line. If the next word does not represent an int value, a NumberFormatException is thrown.

Throws:

`java.lang.NumberFormatException`

readLong

```
public static long readLong()
                     throws java.lang.NumberFormatException
```

Precondition: The next input consists of a long value, possibly preceded by white space, but definitely followed by white space. Action: Reads the first string of nonwhite characters and returns the long value it represents. Discards the first whitespace character after the string read. The next read takes place immediately after the discarded whitespace. In particular, if the string read is at the end of a line, the next reading will take place starting on the next line. If the next word does not represent a long value, a NumberFormatException is thrown.

Throws:

`java.lang.NumberFormatException`

readDouble

```
public static double readDouble()
                         throws java.lang.NumberFormatException
```

Precondition: The next input consists of a double value, possibly preceded by white space, but definitely followed by white space. Action: Reads the first string of nonwhitespace characters and returns the double value it represents. Discards the first whitespace character after the string read. The next read takes place immediately after the discarded whitespace. In particular, if the string read is at the end of a line, the next reading will take place starting on the next line. If the next word does not represent a double value, a NumberFormatException is thrown.

Throws:

`java.lang.NumberFormatException`

readFloat

```
public static float readFloat()
                       throws java.lang.NumberFormatException
```

Precondition: The next input consists of a float value, possibly preceded by white space, but definitely followed by white space. Action: Reads the first string of nonwhite characters and returns the float value it represents. Discards the first whitespace character after the string read. The next read takes place immediately after the discarded whitespace. In particular, if the string read is at the end of a line, the next reading will take place starting on the next line. If the next word does not represent a float value, a NumberFormatException is thrown.

Throws:

```
java.lang.NumberFormatException
```

readWord

```
public static java.lang.String readWord()
```

Reads the first string of nonwhite characters and returns that string. Discards the first whitespace character after the string read. The next read takes place immediately after the discarded whitespace. In particular, if the string read is at the end of a line, the next reading will take place starting on the next line. Note, that if it receives blank lines, it will wait until it gets a nonwhitespace character.

read

```
public static int read()
```

Reads the first byte in the input stream and returns that byte as an int. The next read takes place where this one left off. This read is the same as System.in.read(), except that it catches IOExceptions.