1.2 A sip of java

- History
- Application
- First Program
- Class 찾기 ✓

Finally! Now, a taste of Java!

History

- 1991 - James Gosling, Sun Microsystems, Inc.
- originally a language for programming [redacted]
  - » the menufacturer of home appliance would be unwilling to invest large amount of time and money into developing complicated compilers.
  - » intermediate language : [redacted]

- later (1994) used for World Wide Web applications (since byte code can be downloaded and run without compiling it)
  - HotJava browser.
  - Netscape (1995) decided to make the next release of its Web browser capable of running Java Program.
- eventually used as a general-purpose programming language (for the same reason as above plus it is object-oriented)
- Why the name “Java”? Not sure - it may just be a name that came during a coffee break and it had not been copyrighted, yet.

- 2006년 11월 13일
  - » 썬 마이크로시스템즈는 대부분은 자바를 GPL 라이선스로 소스를 오픈
  - » 2007년 5월 8일 이 과정을 마쳤음
- 2009년 4월 20일 제작사인 썬 마이크로시스템즈가 오라클과 인수합병

Java version

- 1 JDK Alpha and Beta (1995)
- 2 JDK 1.0 (January 23, 1996)
- 3 JDK 1.1 (February 19, 1997)
- 4 J2SE 1.2 (December 8, 1998) – Standard edition
- 5 J2SE 1.3 (May 8, 2000)
- 6 J2SE 1.4 (February 6, 2002)
- 7 J2SE 5.0 (September 30, 2004)
- 8 Java SE 6 (December 11, 2006)
- 9 Java SE 7 (July 28, 2011)
- 10 Java SE 8 (March 18, 2014)
- 11 Java SE 9 (September 2017)
- 12 java SE 10 (March 2018)
- 13 java SE 11(September 2018)

Applets(X) vs. Java Applications

●

» Java programs intended to be downloaded via the WWW and run immediately

» “little applications”

» requires a web browser

» 모질라 재단에서 2015년 10월 파이어폭스에서 NPAPI 플러그인 지원을 중단하겠다는 발표를 했고, 곧이어 오라클에서는 2016년 1월 Java 9부터 애플릿을 위한 자바 플러그인 지원을 중단하겠다고 발표했다. 따라서 자바 애플릿은 Java 9 이후 역사 속으로 사라질 예정이며, 이후 자바 애플릿이 했던 역할은 유사한 기술인 Java Web Start가 대신하게 됨

●

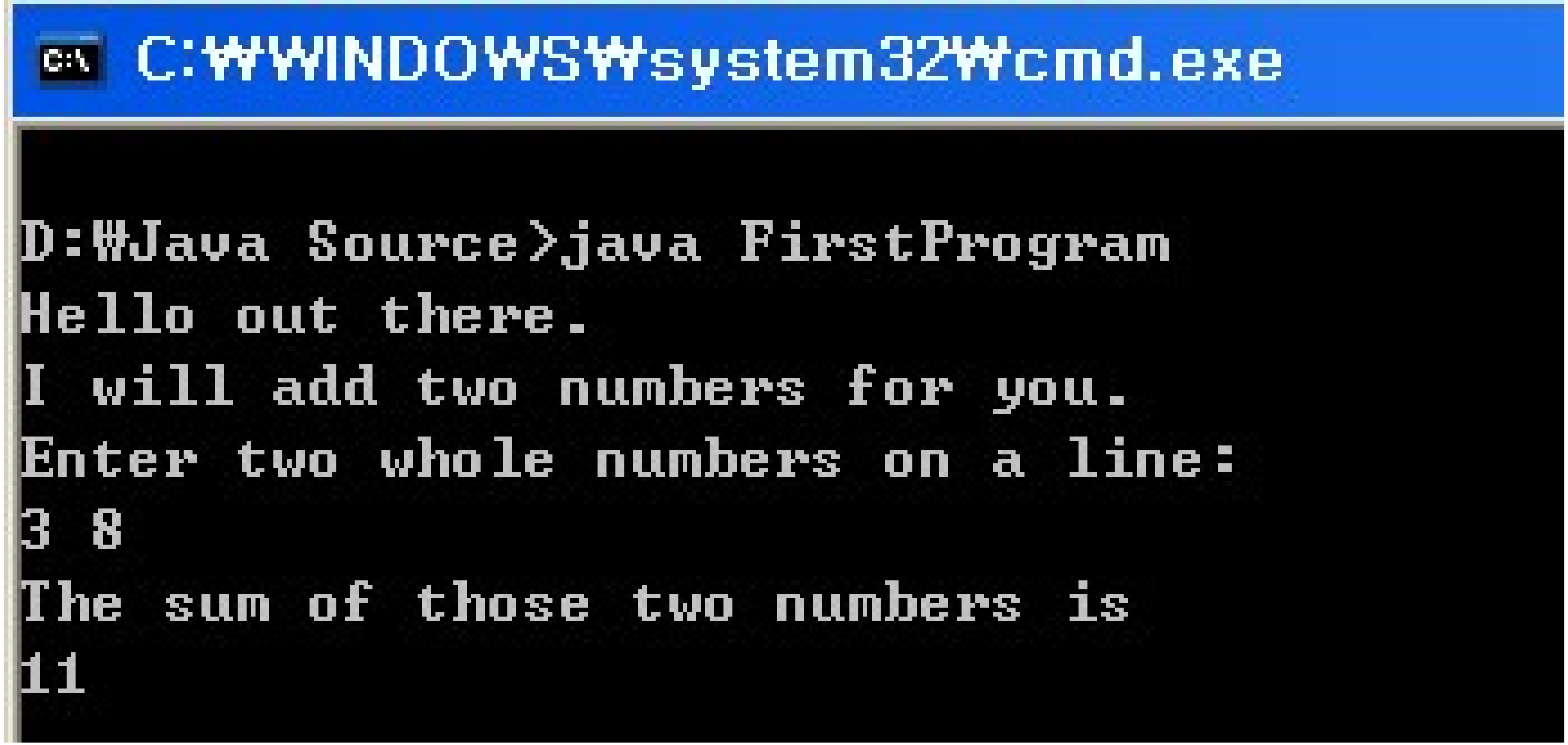
» Java programs intended to be installed then run

» often larger applications

```
// Listing 1.1

import java.util.Scanner;

public class FirstProgram
{
   public static void main (String [] args)
   {
      System.out.println ("Hello out there.");
      System.out.println ("I will add two numbers for you.");
      System.out.println ("Enter two whole numbers on a line:");
      int n1, n2;
      Scanner keyboard = new Scanner (System.in);
      n1 = keyboard.nextInt ();
      n2 = keyboard.nextInt ();
      System.out.println ("The sum of those two numbers is");
      System.out.println (n1 + n2);
   }
}
```



```
D:₩Java Source>java FirstProgram
Hello out there.
I will add two numbers for you.
Enter two whole numbers on a line:
3 8
The sum of those two numbers is
11
```

Explanation of Code ...

- Package

```
Import java.util.Scanner
```

-

  » Tells the compiler that this program uses the class Scanner

  » Scanner class is defined in the package [redacted]

  » Package is a library of classes that have already been defined

Explanation of Code ...

- Code to begin the program (to be explained later):

```
public class FirstProgram
{
    public static void main(String[ ] args)
    {
```

- Java applications all have [redacted] code at the beginning
  - » The name of the class differs from one program to another.
  - » Other information about the class might also be included on the first line.

Explanation of Code ...

- Code to display a text string:

```
System.out.println("Hello out there.");
System.out.println("Want to talk some more?");
System.out.println("Answer y for yes or n for no.");
```

  - Note the “dot” operator
  - **`System.out`** is an [redacted] (PrintStream)
  - **`println`** is a [redacted] that it carries out
  - double-quoted text inside the parentheses is an [redacted] to the method
  - general syntax: ***`Object_Name.Method_Name(`*** [redacted] ***`)`***

… Explanation of Code ...

- Code to create a variable named n1,n2 to contain a single character of data:

```
int n1, n2;
```

- This variable is used to store the user’s response.

… Explanation of Code ...

●

```
   Scanner keyboard = new
    Scanner(System.in)
```

» enables the program to accept, or read, data that a user enters at the keyboard.

… Explanation of Code

-

- n1 = keyboard.nextInt()

» **`Method`** [redacted]

Method Invocation

- A method is an action that an object is capable of performing
- Invoking or [redacted] the method : when you ask an [redacted] to perform the action of a [redacted]
- Object name+ “.” + method name + the arguments enclosed in parentheses.

Compiling a Java Program

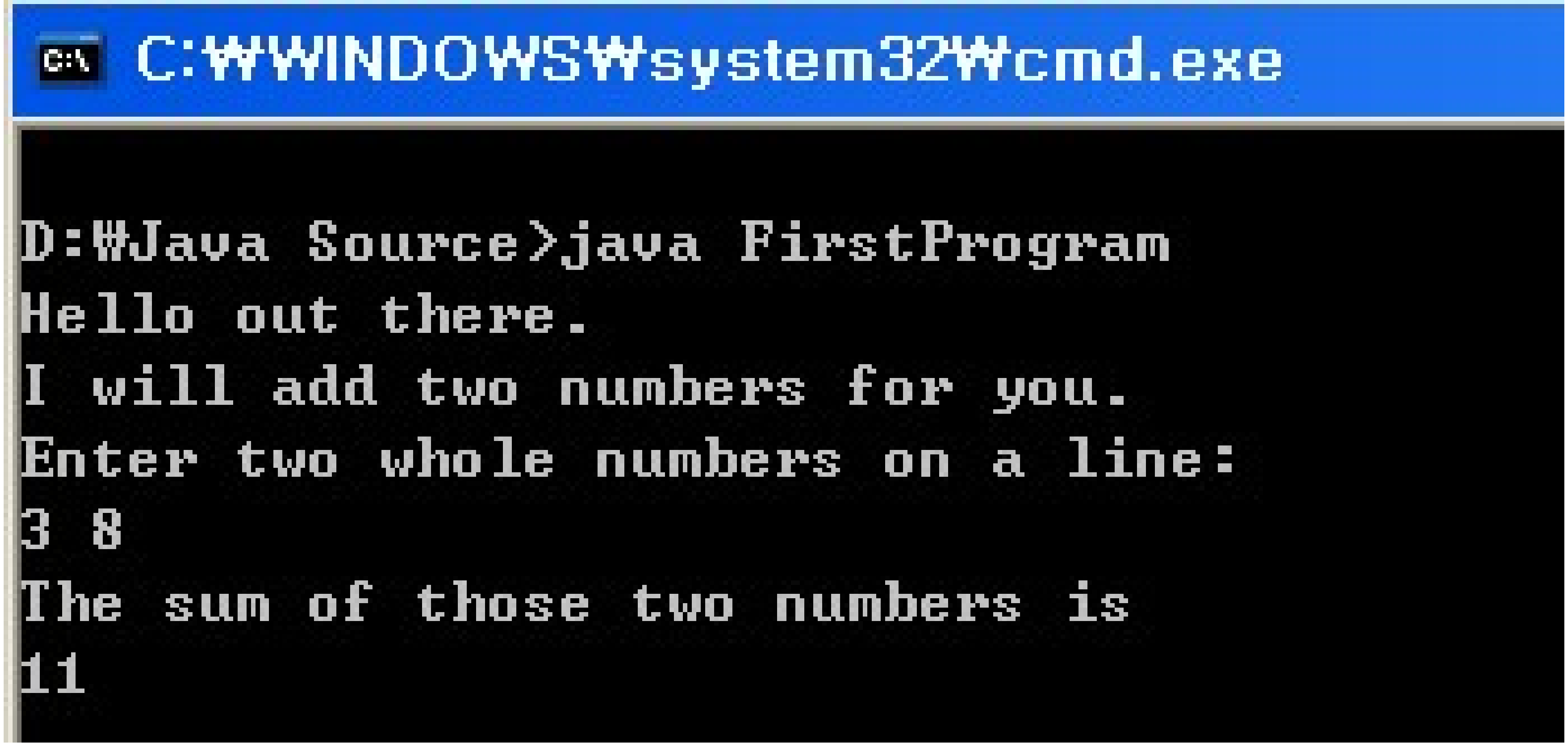
Assuming the java compiler is already set up and all the files are in the same folder (subdirectory):

- Each class used in a program should be in a separate file
- The name of the file should be the same as the class except with ".java" added to it
- First compile each class definition used in the program
  - » for Sun Microsystems' JDK (Java Development Kit), type **`javac`** `FirstProgram`**`.java`**
  - » a [redacted] file is created with the name FirstProgram[redacted]
- Next compile the program file:
  - » **`javac <file>.java`** (which creates **`<file>.class`**)

```
// Listing 1.1

import java.util.Scanner;

public class FirstProgram
{
   public static void main (String [] args)
   {
      System.out.println ("Hello out there.");
      System.out.println ("I will add two numbers for you.");
      System.out.println ("Enter two whole numbers on a line:");
      int n1, n2;
      Scanner keyboard = new Scanner (System.in);
      n1 = keyboard.nextInt ();
      n2 = keyboard.nextInt ();
      System.out.println ("The sum of those two numbers is");
      System.out.println (n1 + n2);
   }
}
```



```
D:₩Java Source>java FirstProgram
Hello out there.
I will add two numbers for you.
Enter two whole numbers on a line:
3 8
The sum of those two numbers is
11
```

(Java API Documentation)

- http://docs.oracle.com/javase/8/docs/api/index.html
- https://docs.oracle.com/en/java/javase/12/docs/api/index.html

Method/Class 찾기

```
// Listing 1.1
import java.util.Scanner;

public class FirstProgram
{
    public static void main (String [] args)
    {
        System.out.println ("Hello out there.");
        System.out.println ("I will add two numbers for you.");
        System.out.println ("Enter two whole numbers on a line:");
        int n1, n2;
        Scanner keyboard = new Scanner (System.in);
        n1 = keyboard.nextInt ();
        n2 = keyboard.nextInt ();
        System.out.println ("The sum of those two numbers is");
        System.out.println (n1 + n2);
    }
}
```

- Scanner Class 찾기
- nextInt() method 찾기
- System 찾기
- System.out 찾기
- PrintStream 넘어가기
- println() 찾기

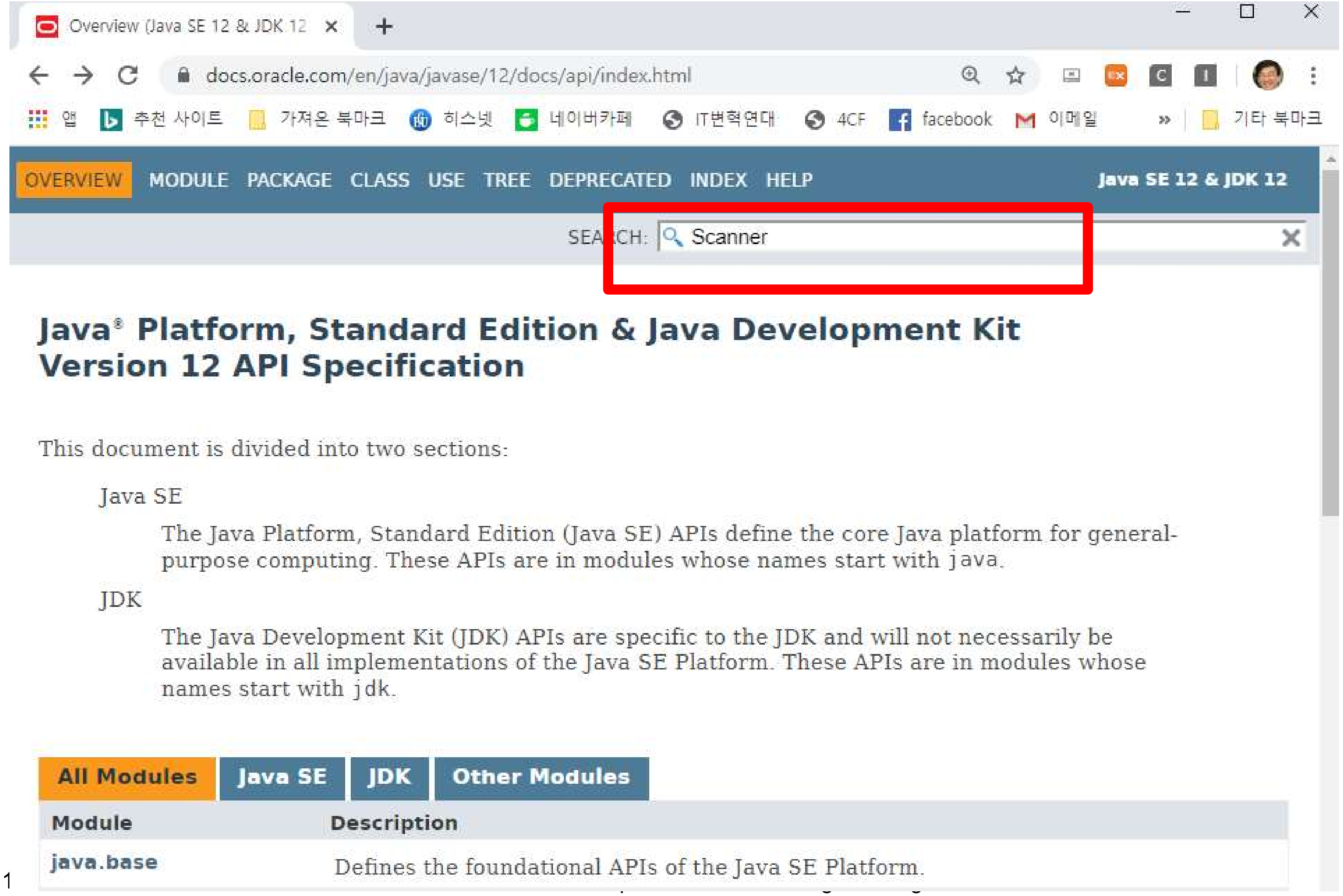
Method/Class 찾기

```
// Listing 1.1

import java.util.Scanner;

public class FirstProgram
{
    public static void main (String [] args)
    {
        System.out.println ("Hello out there.");
        System.out.println ("I will add two numbers for you.");
        System.out.println ("Enter two whole numbers on a line:");
        int n1, n2;
        Scanner keyboard = new Scanner (System.in);
        n1 = keyboard.nextInt ();
        n2 = keyboard.nextInt ();
        System.out.println ("The sum of those two numbers is");
        System.out.println (n1 + n2);
    }
}
```

Scanner 검색



Java® Platform, Standard Edition & Java Development Kit Version 12 API Specification

This document is divided into two sections:

Java SE

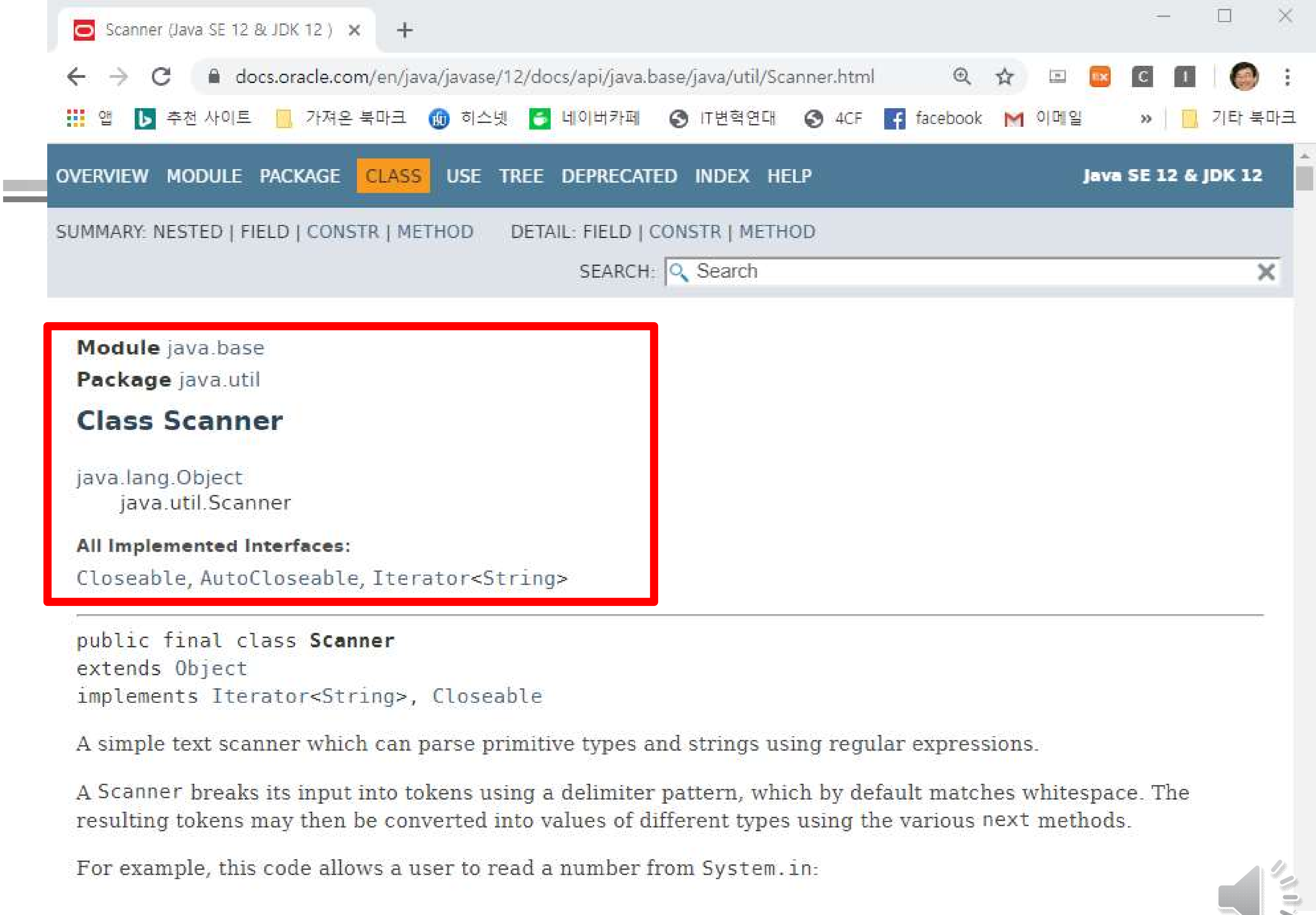
The Java Platform, Standard Edition (Java SE) APIs define the core Java platform for general-purpose computing. These APIs are in modules whose names start with java.

JDK

The Java Development Kit (JDK) APIs are specific to the JDK and will not necessarily be available in all implementations of the Java SE Platform. These APIs are in modules whose names start with jdk.

All Modules | Java SE | JDK | Other Modules

Module	Description
java.base	Defines the foundational APIs of the Java SE Platform.


Scanner (Java SE 12 & JDK 12)
docs.oracle.com/en/java/javase/12/docs/api/java.base/java/util/Scanner.html
앱
추천 사이트
가져온 북마크
히스넷
네이버카페
IT변혁연대
4CF
facebook
이메일
기타 북마크
OVERVIEW
MODULE
PACKAGE
CLASS
USE
TREE
DEPRECATED
INDEX
HELP
Java SE 12 & JDK 12
SUMMARY: NESTED | FIELD | CONSTR | METHOD
DETAIL: FIELD | CONSTR | METHOD
SEARCH:
Search

Module java.base

Package java.util

Class Scanner

java.lang.Object
    java.util.Scanner

All Implemented Interfaces:

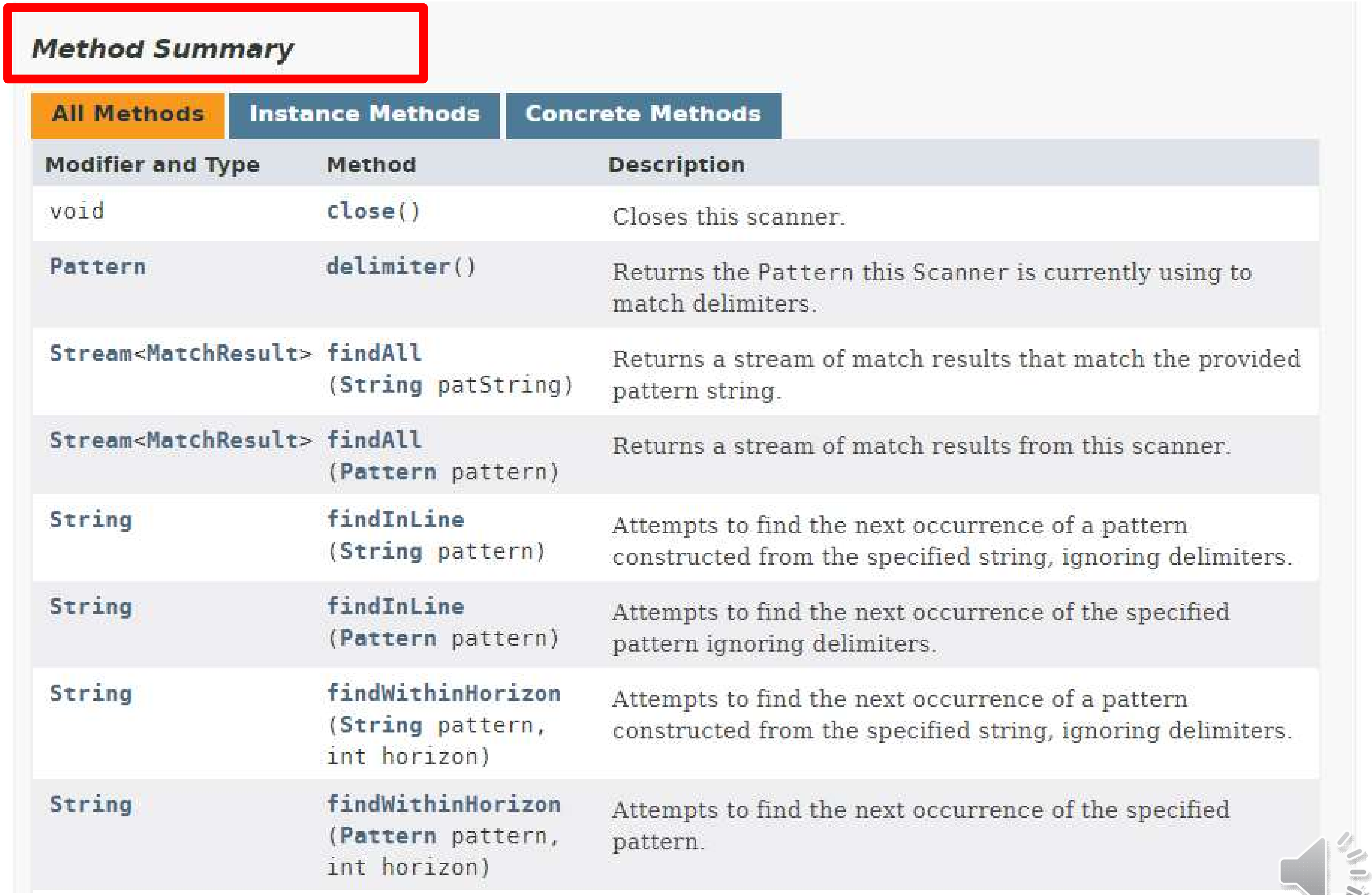
Closeable, AutoCloseable, Iterator<String>

```
public final class Scanner
extends Object
implements Iterator<String>, Closeable
```

A simple text scanner which can parse primitive types and strings using regular expressions.

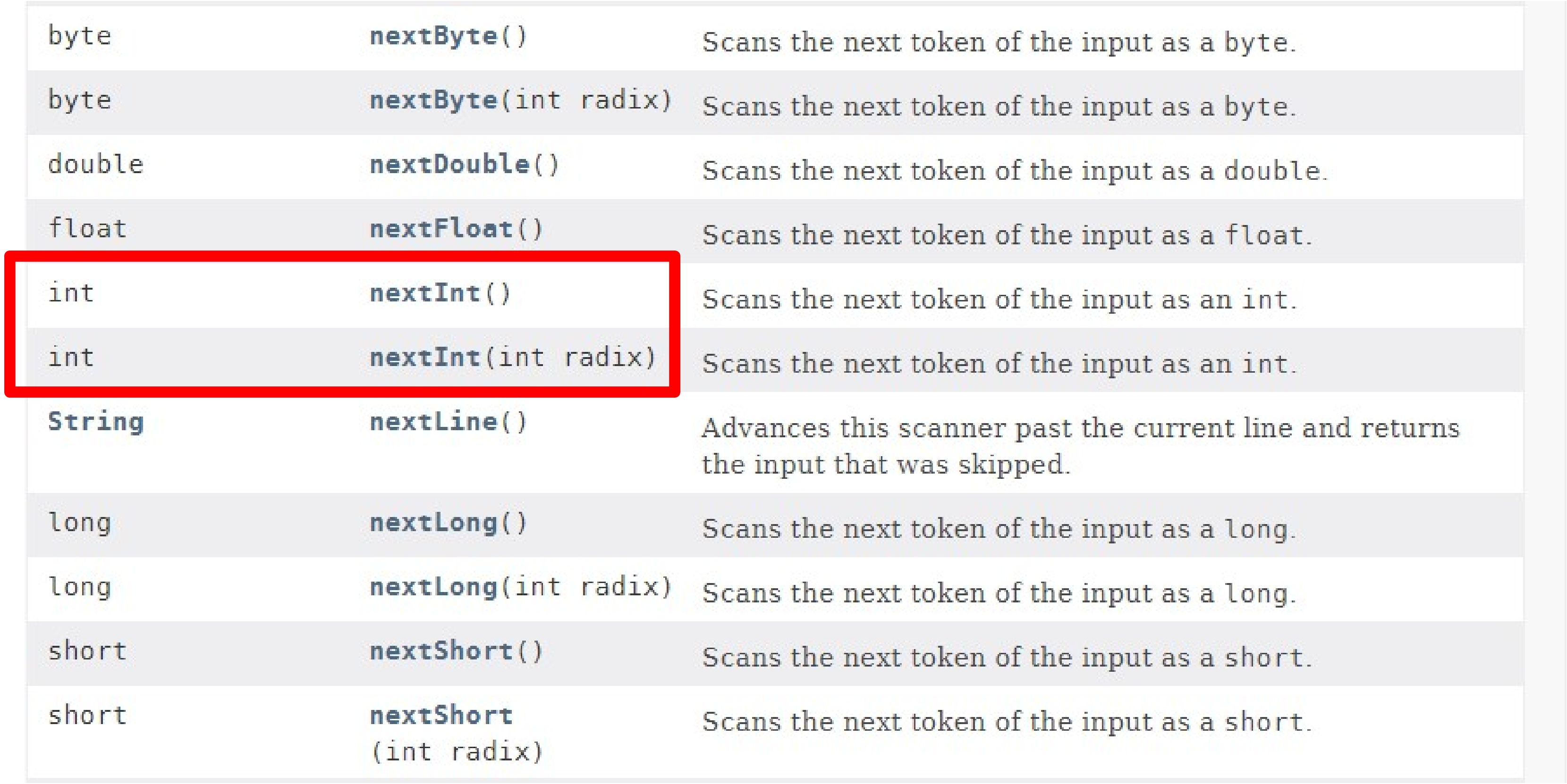
A `Scanner` breaks its input into tokens using a delimiter pattern, which by default matches whitespace. The resulting tokens may then be converted into values of different types using the various `next` methods.

For example, this code allows a user to read a number from `System.in`:

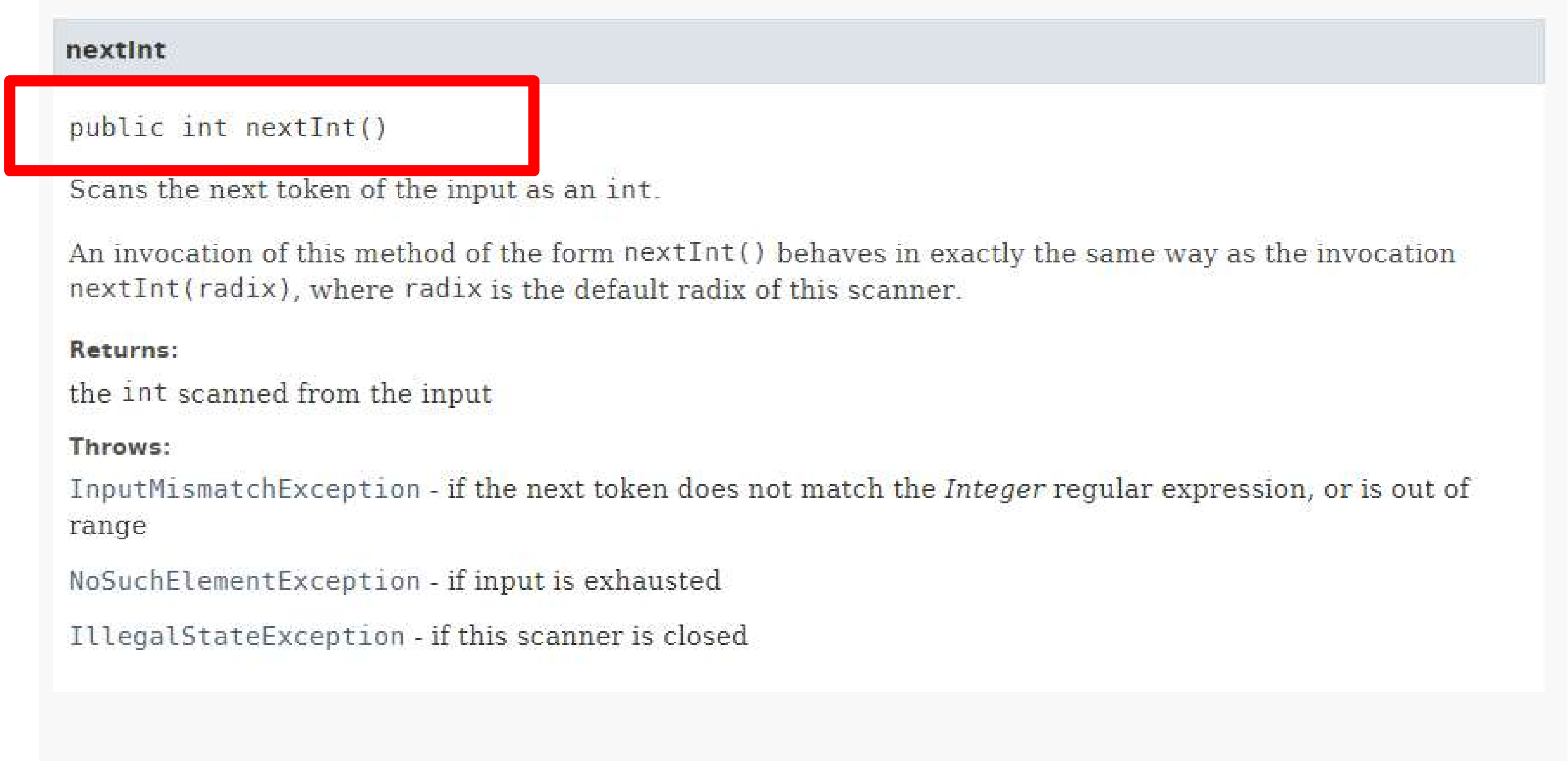


Method Summary

Modifier and Type	Method	Description
`void`	`close()`	Closes this scanner.
`Pattern`	`delimiter()`	Returns the `Pattern` this `Scanner` is currently using to match delimiters.
`Stream<MatchResult>`	`findAll (String patString)`	Returns a stream of match results that match the provided pattern string.
`Stream<MatchResult>`	`findAll (Pattern pattern)`	Returns a stream of match results from this scanner.
`String`	`findInLine (String pattern)`	Attempts to find the next occurrence of a pattern constructed from the specified string, ignoring delimiters.
`String`	`findInLine (Pattern pattern)`	Attempts to find the next occurrence of the specified pattern ignoring delimiters.
`String`	`findWithinHorizon (String pattern, int horizon)`	Attempts to find the next occurrence of a pattern constructed from the specified string, ignoring delimiters.
`String`	`findWithinHorizon (Pattern pattern, int horizon)`	Attempts to find the next occurrence of the specified pattern.



byte	**nextByte**()	Scans the next token of the input as a byte.
byte	**nextByte**(int radix)	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	**nextInt**()	Scans the next token of the input as an int.
int	**nextInt**(int radix)	Scans the next token of the input as an int.
String	**nextLine**()	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	**nextShort**()	Scans the next token of the input as a short.
short	**nextShort** (int radix)	Scans the next token of the input as a short.



nextInt

```
public int nextInt()
```

Scans the next token of the input as an `int`.

An invocation of this method of the form `nextInt()` behaves in exactly the same way as the invocation `nextInt(radix)`, where `radix` is the default radix of this scanner.

Returns:

the `int` scanned from the input

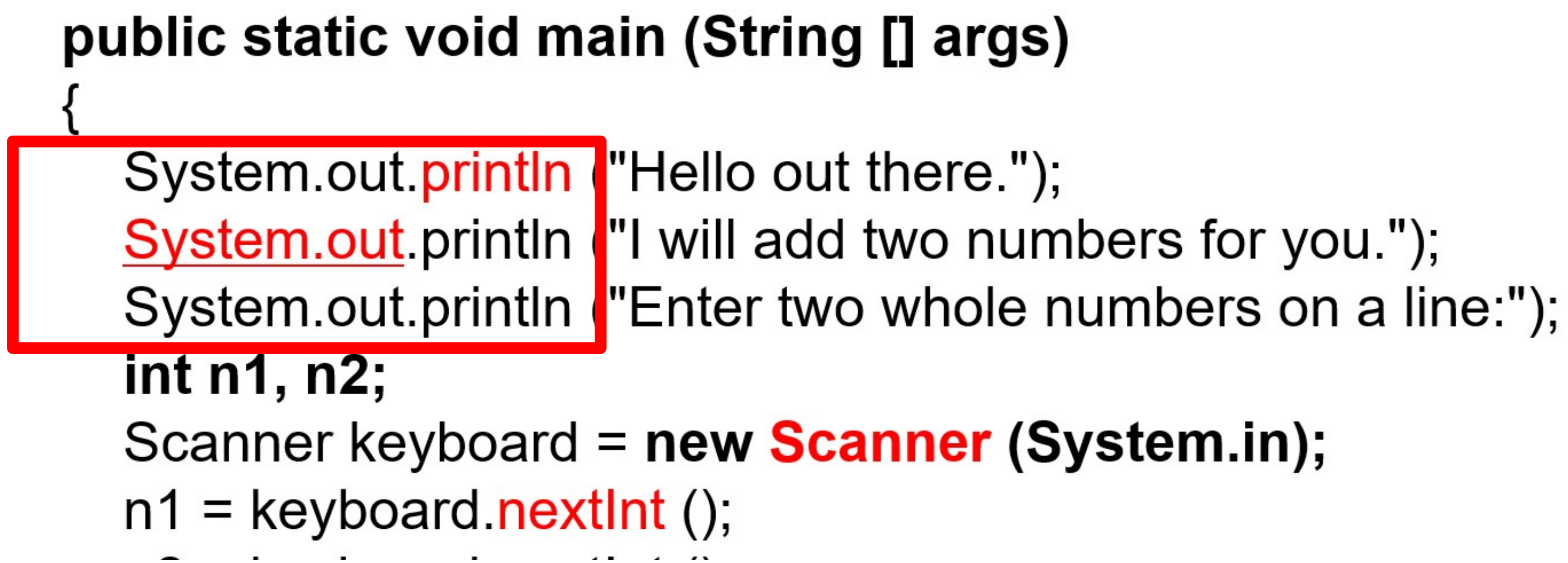
Throws:

`InputMismatchException` - if the next token does not match the *Integer* regular expression, or is out of range

`NoSuchElementException` - if input is exhausted

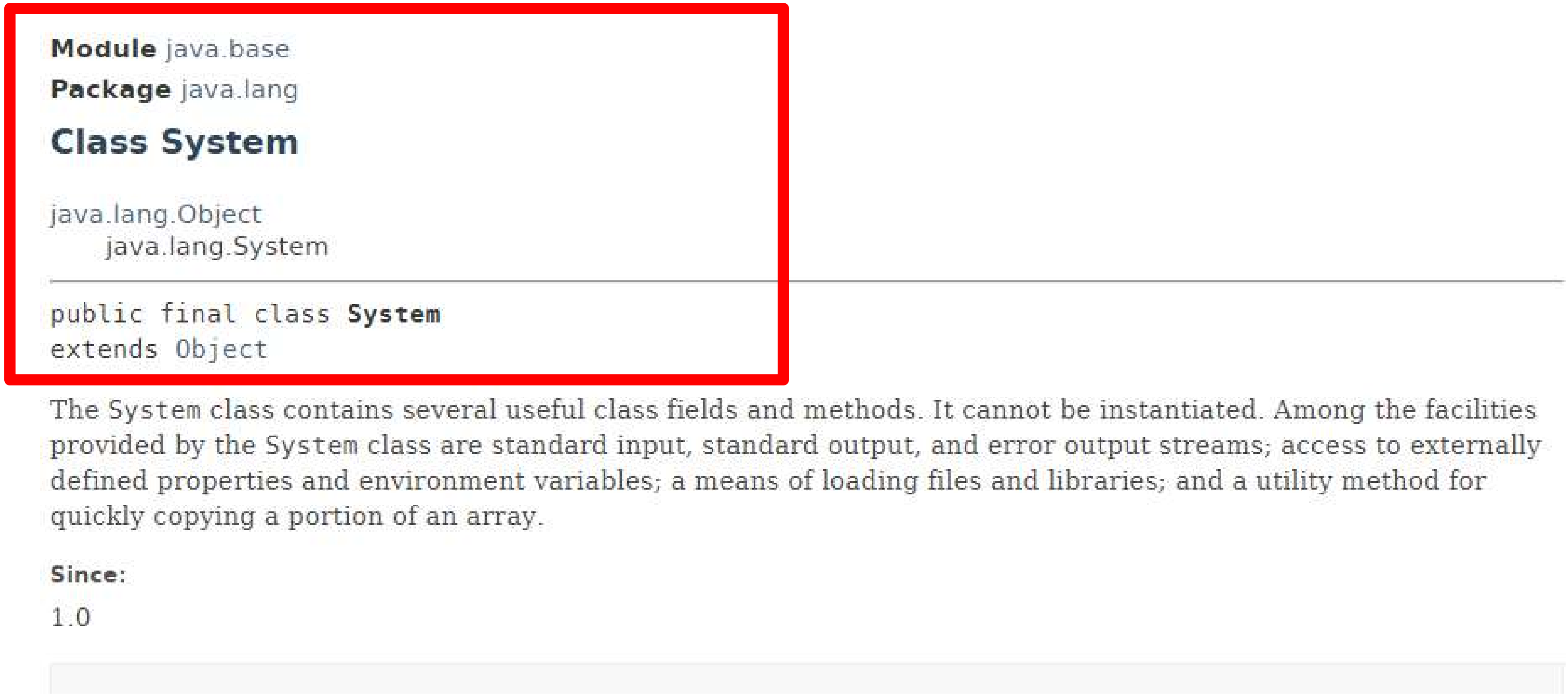
`IllegalStateException` - if this scanner is closed

println()찾기-System Class



```
public static void main (String [] args)
{
    System.out.println ("Hello out there.");
    System.out.println ("I will add two numbers for you.");
    System.out.println ("Enter two whole numbers on a line:");
    int n1, n2;
    Scanner keyboard = new Scanner (System.in);
    n1 = keyboard.nextInt ();
```

System Class 찾기



Module java.base

Package java.lang

Class System

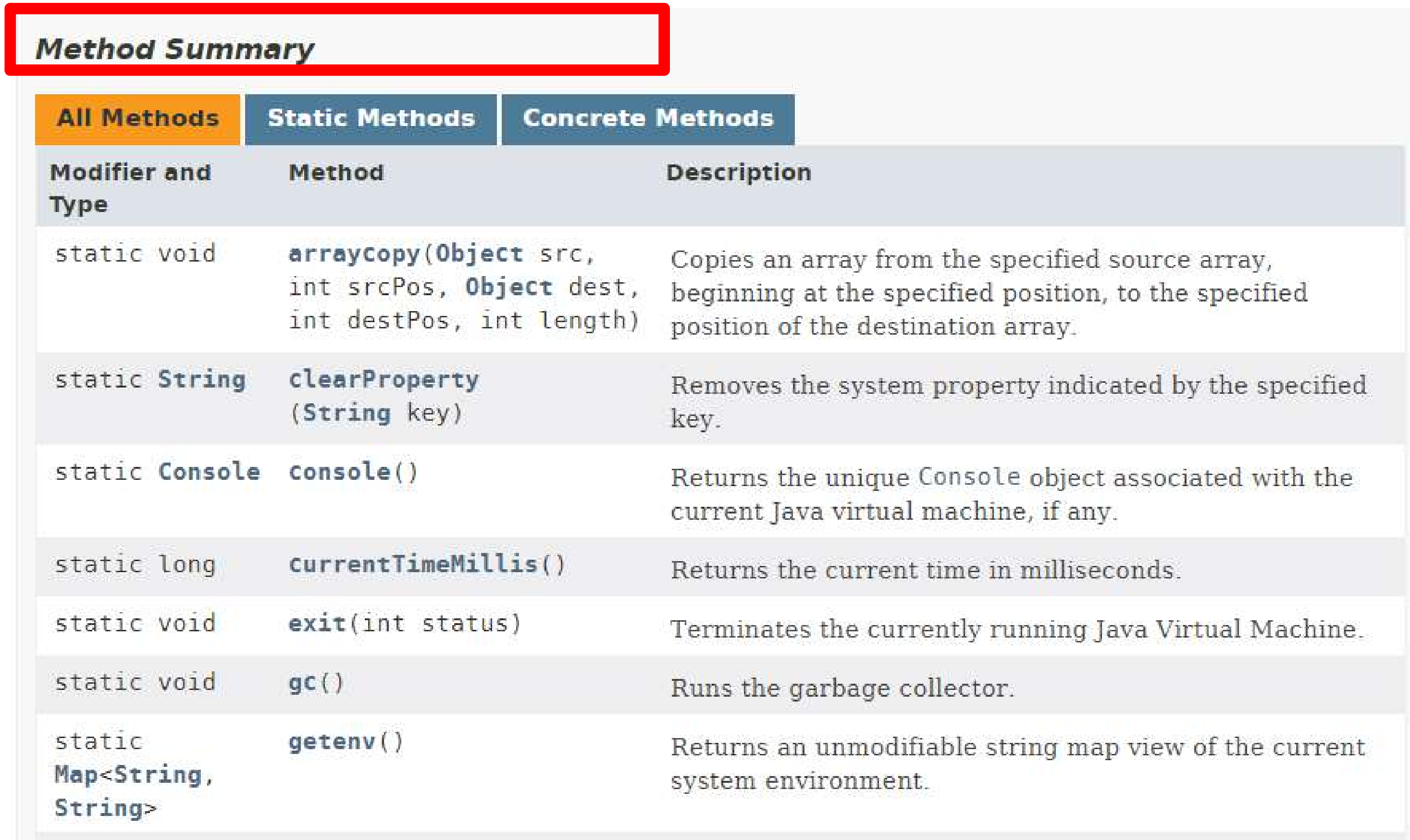
java.lang.Object
    java.lang.System

```
public final class System
extends Object
```

The System class contains several useful class fields and methods. It cannot be instantiated. Among the facilities provided by the System class are standard input, standard output, and error output streams; access to externally defined properties and environment variables; a means of loading files and libraries; and a utility method for quickly copying a portion of an array.

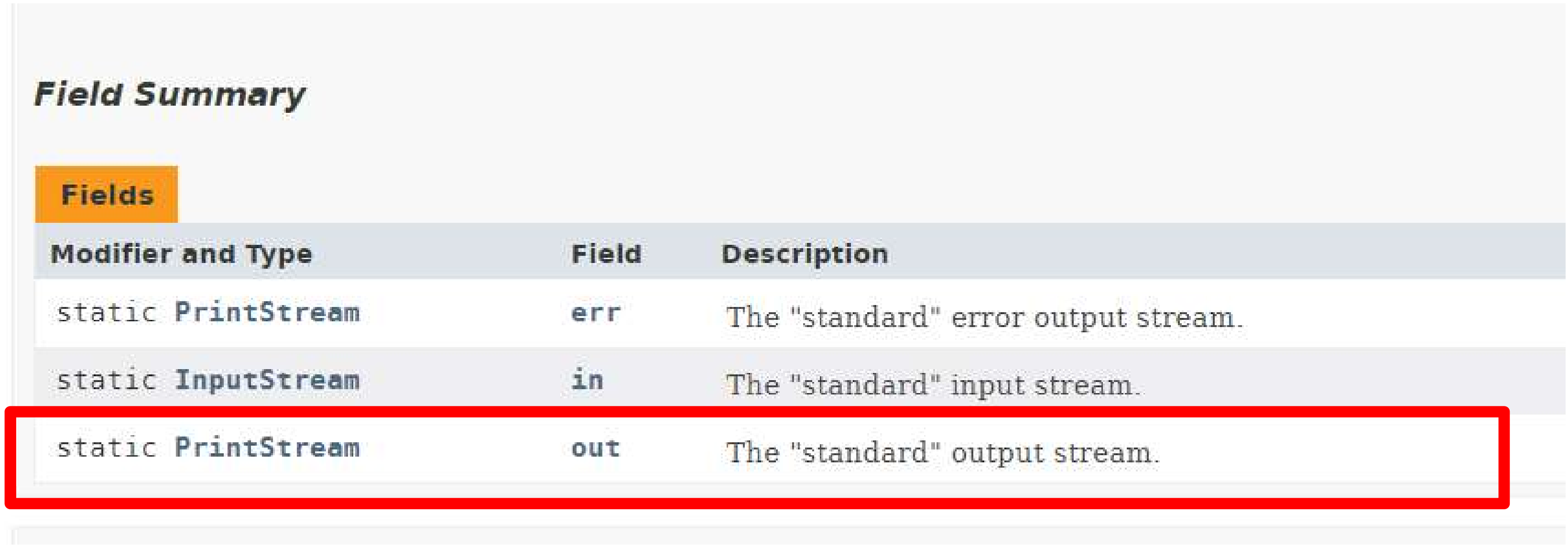
Since:

1.0



Method Summary

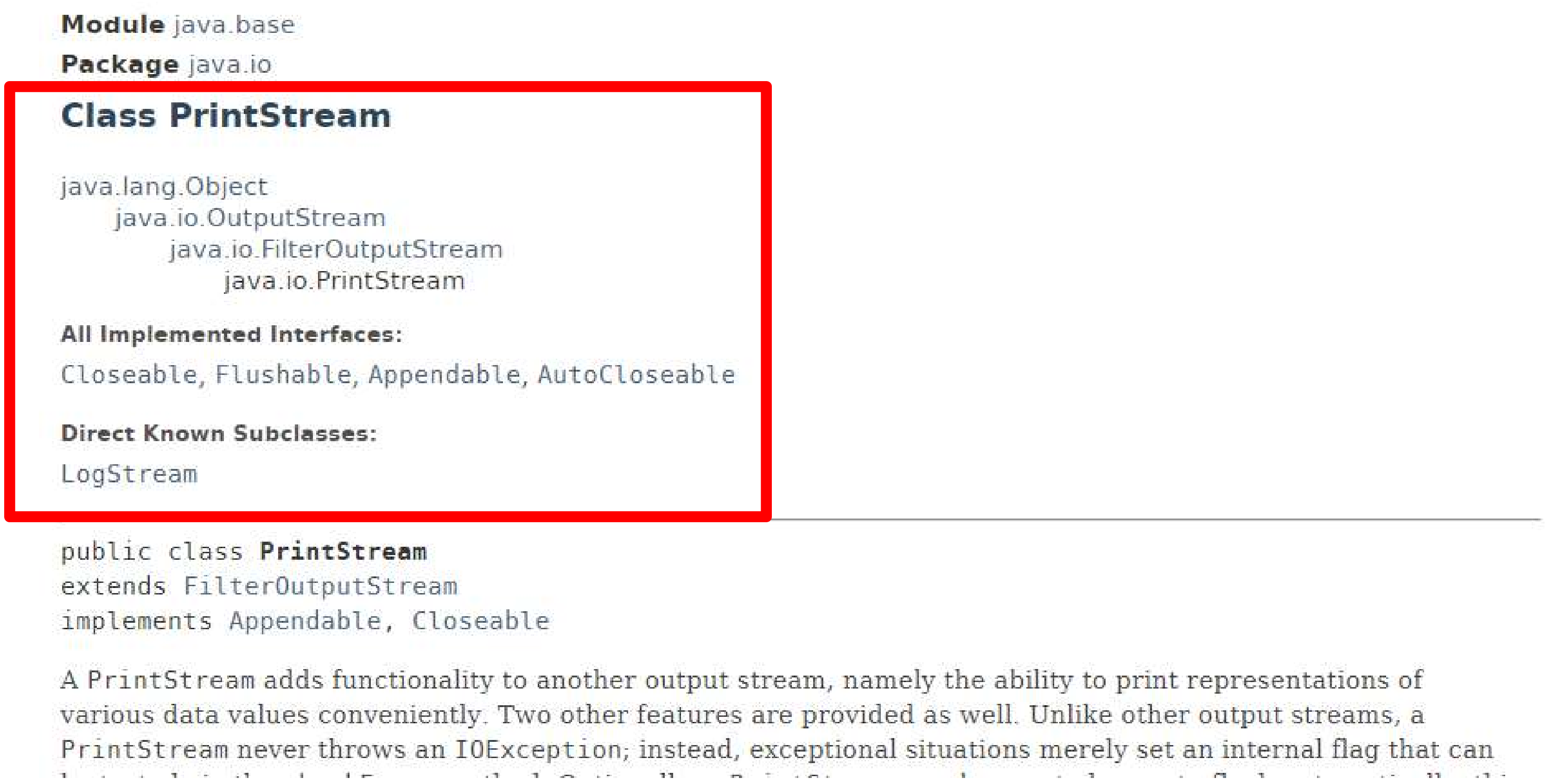
Modifier and Type	Method	Description
static void	arraycopy(Object src, int srcPos, Object dest, int destPos, int length)	Copies an array from the specified source array, beginning at the specified position, to the specified position of the destination array.
static String	clearProperty (String key)	Removes the system property indicated by the specified key.
static Console	console()	Returns the unique Console object associated with the current Java virtual machine, if any.
static long	currentTimeMillis()	Returns the current time in milliseconds.
static void	exit(int status)	Terminates the currently running Java Virtual Machine.
static void	gc()	Runs the garbage collector.
static Map<String, String>	getenv()	Returns an unmodifiable string map view of the current system environment.



Field Summary

Fields

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



Module java.base

Package java.io

Class PrintStream

```
java.lang.Object
    java.io.OutputStream
        java.io.FilterOutputStream
            java.io.PrintStream
```

All Implemented Interfaces:

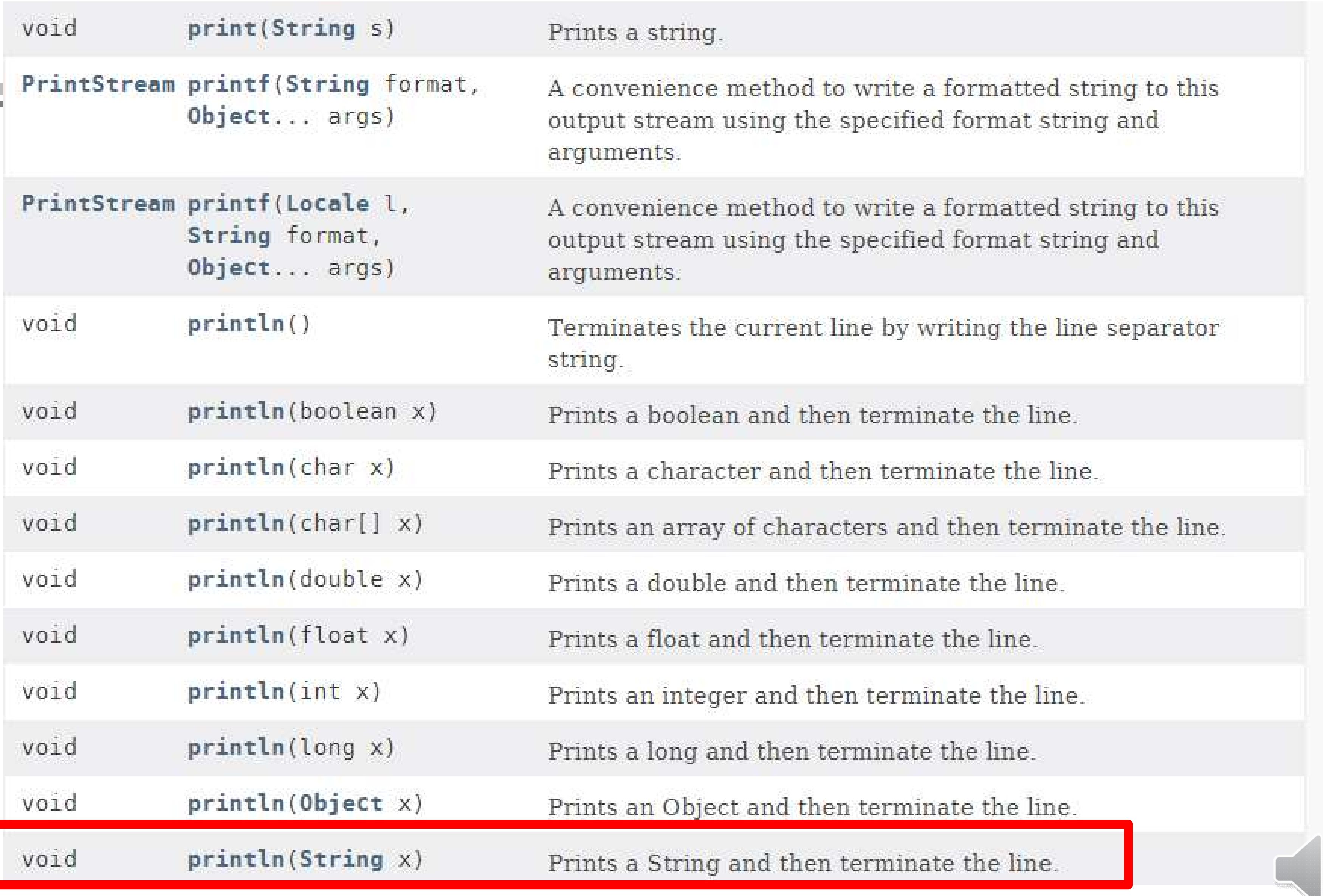
Closeable, Flushable, Appendable, AutoCloseable

Direct Known Subclasses:

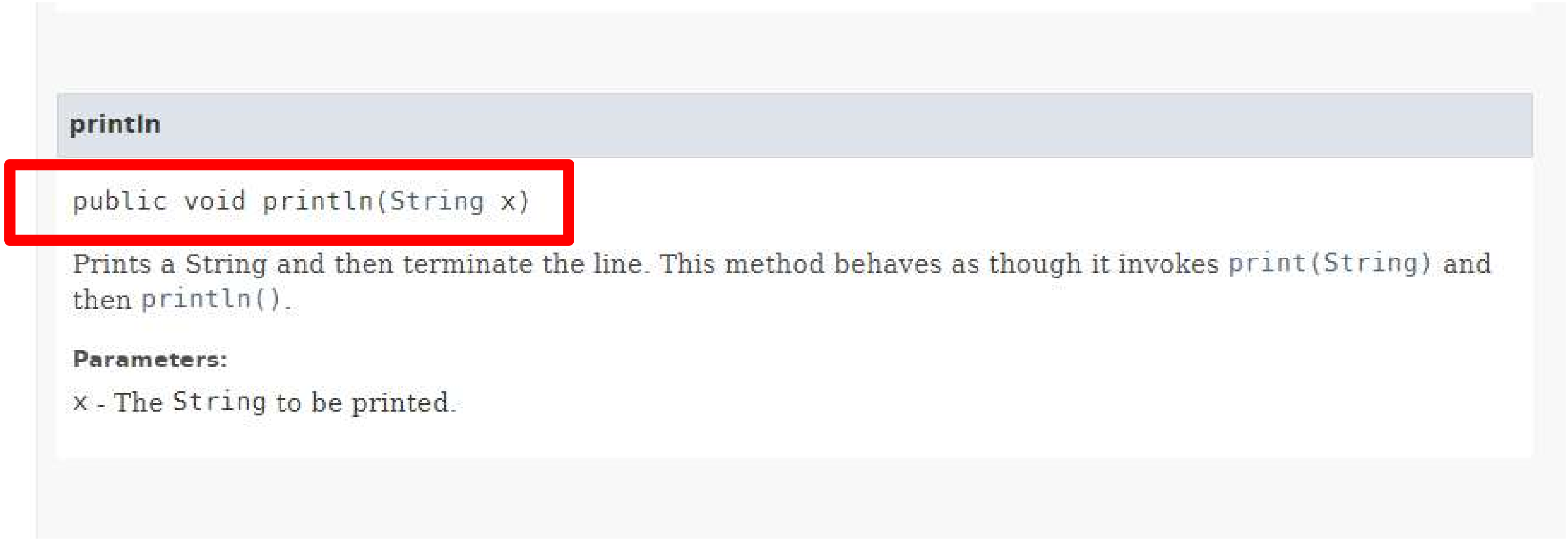
LogStream

```
public class PrintStream
extends FilterOutputStream
implements Appendable, Closeable
```

A PrintStream adds functionality to another output stream, namely the ability to print representations of various data values conveniently. Two other features are provided as well. Unlike other output streams, a PrintStream never throws an IOException; instead, exceptional situations merely set an internal flag that can



void	**print**(**String** s)	Prints a string.
PrintStream	**printf**(**String** format, **Object**... args)	A convenience method to write a formatted string to this output stream using the specified format string and arguments.
PrintStream	**printf**(**Locale** l, **String** format, **Object**... args)	A convenience method to write a formatted string to this output stream using the specified format string and arguments.
void	**println**()	Terminates the current line by writing the line separator string.
void	**println**(boolean x)	Prints a boolean and then terminate the line.
void	**println**(char x)	Prints a character and then terminate the line.
void	**println**(char[] x)	Prints an array of characters and then terminate the line.
void	**println**(double x)	Prints a double and then terminate the line.
void	**println**(float x)	Prints a float and then terminate the line.
void	**println**(int x)	Prints an integer and then terminate the line.
void	**println**(long x)	Prints a long and then terminate the line.
void	**println**(**Object** x)	Prints an Object and then terminate the line.
void	**println**(**String** x)	Prints a String and then terminate the line.



println

```
public void println(String x)
```

Prints a String and then terminate the line. This method behaves as though it invokes `print(String)` and then `println()`.

Parameters:

`x` - The `String` to be printed.

끝