| | [**Overview**](http://docs.google.com/overview-summary.html) | **Package** | Class | [**Use**](http://docs.google.com/package-use.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV PACKAGE**](http://docs.google.com/java/io/package-summary.html)   [**NEXT PACKAGE**](http://docs.google.com/java/lang/annotation/package-summary.html) | [**FRAMES**](http://docs.google.com/index.html?java/lang/package-summary.html)    [**NO FRAMES**](http://docs.google.com/package-summary.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

## Package java.lang

Provides classes that are fundamental to the design of the Java programming language.

**See:**

[**Description**](#3znysh7)

| **Interface Summary** | |
| --- | --- |
| [**Appendable**](http://docs.google.com/java/lang/Appendable.html) | An object to which char sequences and values can be appended. |
| [**CharSequence**](http://docs.google.com/java/lang/CharSequence.html) | A CharSequence is a readable sequence of char values. |
| [**Cloneable**](http://docs.google.com/java/lang/Cloneable.html) | A class implements the Cloneable interface to indicate to the [Object.clone()](http://docs.google.com/java/lang/Object.html#clone()) method that it is legal for that method to make a field-for-field copy of instances of that class. |
| [**Comparable<T>**](http://docs.google.com/java/lang/Comparable.html) | This interface imposes a total ordering on the objects of each class that implements it. |
| [**Iterable<T>**](http://docs.google.com/java/lang/Iterable.html) | Implementing this interface allows an object to be the target of the "foreach" statement. |
| [**Readable**](http://docs.google.com/java/lang/Readable.html) | A Readable is a source of characters. |
| [**Runnable**](http://docs.google.com/java/lang/Runnable.html) | The Runnable interface should be implemented by any class whose instances are intended to be executed by a thread. |
| [**Thread.UncaughtExceptionHandler**](http://docs.google.com/java/lang/Thread.UncaughtExceptionHandler.html) | Interface for handlers invoked when a Thread abruptly terminates due to an uncaught exception. |

| **Class Summary** | |
| --- | --- |
| [**Boolean**](http://docs.google.com/java/lang/Boolean.html) | The Boolean class wraps a value of the primitive type boolean in an object. |
| [**Byte**](http://docs.google.com/java/lang/Byte.html) | The Byte class wraps a value of primitive type byte in an object. |
| [**Character**](http://docs.google.com/java/lang/Character.html) | The Character class wraps a value of the primitive type char in an object. |
| [**Character.Subset**](http://docs.google.com/java/lang/Character.Subset.html) | Instances of this class represent particular subsets of the Unicode character set. |
| [**Character.UnicodeBlock**](http://docs.google.com/java/lang/Character.UnicodeBlock.html) | A family of character subsets representing the character blocks in the Unicode specification. |
| [**Class<T>**](http://docs.google.com/java/lang/Class.html) | Instances of the class Class represent classes and interfaces in a running Java application. |
| [**ClassLoader**](http://docs.google.com/java/lang/ClassLoader.html) | A class loader is an object that is responsible for loading classes. |
| [**Compiler**](http://docs.google.com/java/lang/Compiler.html) | The Compiler class is provided to support Java-to-native-code compilers and related services. |
| [**Double**](http://docs.google.com/java/lang/Double.html) | The Double class wraps a value of the primitive type double in an object. |
| [**Enum<E extends Enum<E>>**](http://docs.google.com/java/lang/Enum.html) | This is the common base class of all Java language enumeration types. |
| [**Float**](http://docs.google.com/java/lang/Float.html) | The Float class wraps a value of primitive type float in an object. |
| [**InheritableThreadLocal<T>**](http://docs.google.com/java/lang/InheritableThreadLocal.html) | This class extends ThreadLocal to provide inheritance of values from parent thread to child thread: when a child thread is created, the child receives initial values for all inheritable thread-local variables for which the parent has values. |
| [**Integer**](http://docs.google.com/java/lang/Integer.html) | The Integer class wraps a value of the primitive type int in an object. |
| [**Long**](http://docs.google.com/java/lang/Long.html) | The Long class wraps a value of the primitive type long in an object. |
| [**Math**](http://docs.google.com/java/lang/Math.html) | The class Math contains methods for performing basic numeric operations such as the elementary exponential, logarithm, square root, and trigonometric functions. |
| [**Number**](http://docs.google.com/java/lang/Number.html) | The abstract class Number is the superclass of classes BigDecimal, BigInteger, Byte, Double, Float, Integer, Long, and Short. |
| [**Object**](http://docs.google.com/java/lang/Object.html) | Class Object is the root of the class hierarchy. |
| [**Package**](http://docs.google.com/java/lang/Package.html) | Package objects contain version information about the implementation and specification of a Java package. |
| [**Process**](http://docs.google.com/java/lang/Process.html) | The [ProcessBuilder.start()](http://docs.google.com/java/lang/ProcessBuilder.html#start()) and [Runtime.exec](http://docs.google.com/java/lang/Runtime.html#exec(java.lang.String%5B%5D,%20java.lang.String%5B%5D,%20java.io.File)) methods create a native process and return an instance of a subclass of Process that can be used to control the process and obtain information about it. |
| [**ProcessBuilder**](http://docs.google.com/java/lang/ProcessBuilder.html) | This class is used to create operating system processes. |
| [**Runtime**](http://docs.google.com/java/lang/Runtime.html) | Every Java application has a single instance of class Runtime that allows the application to interface with the environment in which the application is running. |
| [**RuntimePermission**](http://docs.google.com/java/lang/RuntimePermission.html) | This class is for runtime permissions. |
| [**SecurityManager**](http://docs.google.com/java/lang/SecurityManager.html) | The security manager is a class that allows applications to implement a security policy. |
| [**Short**](http://docs.google.com/java/lang/Short.html) | The Short class wraps a value of primitive type short in an object. |
| [**StackTraceElement**](http://docs.google.com/java/lang/StackTraceElement.html) | An element in a stack trace, as returned by [Throwable.getStackTrace()](http://docs.google.com/java/lang/Throwable.html#getStackTrace()). |
| [**StrictMath**](http://docs.google.com/java/lang/StrictMath.html) | The class StrictMath contains methods for performing basic numeric operations such as the elementary exponential, logarithm, square root, and trigonometric functions. |
| [**String**](http://docs.google.com/java/lang/String.html) | The String class represents character strings. |
| [**StringBuffer**](http://docs.google.com/java/lang/StringBuffer.html) | A thread-safe, mutable sequence of characters. |
| [**StringBuilder**](http://docs.google.com/java/lang/StringBuilder.html) | A mutable sequence of characters. |
| [**System**](http://docs.google.com/java/lang/System.html) | The System class contains several useful class fields and methods. |
| [**Thread**](http://docs.google.com/java/lang/Thread.html) | A *thread* is a thread of execution in a program. |
| [**ThreadGroup**](http://docs.google.com/java/lang/ThreadGroup.html) | A thread group represents a set of threads. |
| [**ThreadLocal<T>**](http://docs.google.com/java/lang/ThreadLocal.html) | This class provides thread-local variables. |
| [**Throwable**](http://docs.google.com/java/lang/Throwable.html) | The Throwable class is the superclass of all errors and exceptions in the Java language. |
| [**Void**](http://docs.google.com/java/lang/Void.html) | The Void class is an uninstantiable placeholder class to hold a reference to the Class object representing the Java keyword void. |

| **Enum Summary** | |
| --- | --- |
| [**Thread.State**](http://docs.google.com/java/lang/Thread.State.html) | A thread state. |

| **Exception Summary** | |
| --- | --- |
| [**ArithmeticException**](http://docs.google.com/java/lang/ArithmeticException.html) | Thrown when an exceptional arithmetic condition has occurred. |
| [**ArrayIndexOutOfBoundsException**](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) | Thrown to indicate that an array has been accessed with an illegal index. |
| [**ArrayStoreException**](http://docs.google.com/java/lang/ArrayStoreException.html) | Thrown to indicate that an attempt has been made to store the wrong type of object into an array of objects. |
| [**ClassCastException**](http://docs.google.com/java/lang/ClassCastException.html) | Thrown to indicate that the code has attempted to cast an object to a subclass of which it is not an instance. |
| [**ClassNotFoundException**](http://docs.google.com/java/lang/ClassNotFoundException.html) | Thrown when an application tries to load in a class through its string name using: The forName method in class Class. |
| [**CloneNotSupportedException**](http://docs.google.com/java/lang/CloneNotSupportedException.html) | Thrown to indicate that the clone method in class Object has been called to clone an object, but that the object's class does not implement the Cloneable interface. |
| [**EnumConstantNotPresentException**](http://docs.google.com/java/lang/EnumConstantNotPresentException.html) | Thrown when an application tries to access an enum constant by name and the enum type contains no constant with the specified name. |
| [**Exception**](http://docs.google.com/java/lang/Exception.html) | The class Exception and its subclasses are a form of Throwable that indicates conditions that a reasonable application might want to catch. |
| [**IllegalAccessException**](http://docs.google.com/java/lang/IllegalAccessException.html) | An IllegalAccessException is thrown when an application tries to reflectively create an instance (other than an array), set or get a field, or invoke a method, but the currently executing method does not have access to the definition of the specified class, field, method or constructor. |
| [**IllegalArgumentException**](http://docs.google.com/java/lang/IllegalArgumentException.html) | Thrown to indicate that a method has been passed an illegal or inappropriate argument. |
| [**IllegalMonitorStateException**](http://docs.google.com/java/lang/IllegalMonitorStateException.html) | Thrown to indicate that a thread has attempted to wait on an object's monitor or to notify other threads waiting on an object's monitor without owning the specified monitor. |
| [**IllegalStateException**](http://docs.google.com/java/lang/IllegalStateException.html) | Signals that a method has been invoked at an illegal or inappropriate time. |
| [**IllegalThreadStateException**](http://docs.google.com/java/lang/IllegalThreadStateException.html) | Thrown to indicate that a thread is not in an appropriate state for the requested operation. |
| [**IndexOutOfBoundsException**](http://docs.google.com/java/lang/IndexOutOfBoundsException.html) | Thrown to indicate that an index of some sort (such as to an array, to a string, or to a vector) is out of range. |
| [**InstantiationException**](http://docs.google.com/java/lang/InstantiationException.html) | Thrown when an application tries to create an instance of a class using the newInstance method in class Class, but the specified class object cannot be instantiated. |
| [**InterruptedException**](http://docs.google.com/java/lang/InterruptedException.html) | Thrown when a thread is waiting, sleeping, or otherwise occupied, and the thread is interrupted, either before or during the activity. |
| [**NegativeArraySizeException**](http://docs.google.com/java/lang/NegativeArraySizeException.html) | Thrown if an application tries to create an array with negative size. |
| [**NoSuchFieldException**](http://docs.google.com/java/lang/NoSuchFieldException.html) | Signals that the class doesn't have a field of a specified name. |
| [**NoSuchMethodException**](http://docs.google.com/java/lang/NoSuchMethodException.html) | Thrown when a particular method cannot be found. |
| [**NullPointerException**](http://docs.google.com/java/lang/NullPointerException.html) | Thrown when an application attempts to use null in a case where an object is required. |
| [**NumberFormatException**](http://docs.google.com/java/lang/NumberFormatException.html) | Thrown to indicate that the application has attempted to convert a string to one of the numeric types, but that the string does not have the appropriate format. |
| [**RuntimeException**](http://docs.google.com/java/lang/RuntimeException.html) | RuntimeException is the superclass of those exceptions that can be thrown during the normal operation of the Java Virtual Machine. |
| [**SecurityException**](http://docs.google.com/java/lang/SecurityException.html) | Thrown by the security manager to indicate a security violation. |
| [**StringIndexOutOfBoundsException**](http://docs.google.com/java/lang/StringIndexOutOfBoundsException.html) | Thrown by String methods to indicate that an index is either negative or greater than the size of the string. |
| [**TypeNotPresentException**](http://docs.google.com/java/lang/TypeNotPresentException.html) | Thrown when an application tries to access a type using a string representing the type's name, but no definition for the type with the specified name can be found. |
| [**UnsupportedOperationException**](http://docs.google.com/java/lang/UnsupportedOperationException.html) | Thrown to indicate that the requested operation is not supported. |

| **Error Summary** | |
| --- | --- |
| [**AbstractMethodError**](http://docs.google.com/java/lang/AbstractMethodError.html) | Thrown when an application tries to call an abstract method. |
| [**AssertionError**](http://docs.google.com/java/lang/AssertionError.html) | Thrown to indicate that an assertion has failed. |
| [**ClassCircularityError**](http://docs.google.com/java/lang/ClassCircularityError.html) | Thrown when a circularity has been detected while initializing a class. |
| [**ClassFormatError**](http://docs.google.com/java/lang/ClassFormatError.html) | Thrown when the Java Virtual Machine attempts to read a class file and determines that the file is malformed or otherwise cannot be interpreted as a class file. |
| [**Error**](http://docs.google.com/java/lang/Error.html) | An Error is a subclass of Throwable that indicates serious problems that a reasonable application should not try to catch. |
| [**ExceptionInInitializerError**](http://docs.google.com/java/lang/ExceptionInInitializerError.html) | Signals that an unexpected exception has occurred in a static initializer. |
| [**IllegalAccessError**](http://docs.google.com/java/lang/IllegalAccessError.html) | Thrown if an application attempts to access or modify a field, or to call a method that it does not have access to. |
| [**IncompatibleClassChangeError**](http://docs.google.com/java/lang/IncompatibleClassChangeError.html) | Thrown when an incompatible class change has occurred to some class definition. |
| [**InstantiationError**](http://docs.google.com/java/lang/InstantiationError.html) | Thrown when an application tries to use the Java new construct to instantiate an abstract class or an interface. |
| [**InternalError**](http://docs.google.com/java/lang/InternalError.html) | Thrown to indicate some unexpected internal error has occurred in the Java Virtual Machine. |
| [**LinkageError**](http://docs.google.com/java/lang/LinkageError.html) | Subclasses of LinkageError indicate that a class has some dependency on another class; however, the latter class has incompatibly changed after the compilation of the former class. |
| [**NoClassDefFoundError**](http://docs.google.com/java/lang/NoClassDefFoundError.html) | Thrown if the Java Virtual Machine or a ClassLoader instance tries to load in the definition of a class (as part of a normal method call or as part of creating a new instance using the new expression) and no definition of the class could be found. |
| [**NoSuchFieldError**](http://docs.google.com/java/lang/NoSuchFieldError.html) | Thrown if an application tries to access or modify a specified field of an object, and that object no longer has that field. |
| [**NoSuchMethodError**](http://docs.google.com/java/lang/NoSuchMethodError.html) | Thrown if an application tries to call a specified method of a class (either static or instance), and that class no longer has a definition of that method. |
| [**OutOfMemoryError**](http://docs.google.com/java/lang/OutOfMemoryError.html) | Thrown when the Java Virtual Machine cannot allocate an object because it is out of memory, and no more memory could be made available by the garbage collector. |
| [**StackOverflowError**](http://docs.google.com/java/lang/StackOverflowError.html) | Thrown when a stack overflow occurs because an application recurses too deeply. |
| [**ThreadDeath**](http://docs.google.com/java/lang/ThreadDeath.html) | An instance of ThreadDeath is thrown in the victim thread when the stop method with zero arguments in class Thread is called. |
| [**UnknownError**](http://docs.google.com/java/lang/UnknownError.html) | Thrown when an unknown but serious exception has occurred in the Java Virtual Machine. |
| [**UnsatisfiedLinkError**](http://docs.google.com/java/lang/UnsatisfiedLinkError.html) | Thrown if the Java Virtual Machine cannot find an appropriate native-language definition of a method declared native. |
| [**UnsupportedClassVersionError**](http://docs.google.com/java/lang/UnsupportedClassVersionError.html) | Thrown when the Java Virtual Machine attempts to read a class file and determines that the major and minor version numbers in the file are not supported. |
| [**VerifyError**](http://docs.google.com/java/lang/VerifyError.html) | Thrown when the "verifier" detects that a class file, though well formed, contains some sort of internal inconsistency or security problem. |
| [**VirtualMachineError**](http://docs.google.com/java/lang/VirtualMachineError.html) | Thrown to indicate that the Java Virtual Machine is broken or has run out of resources necessary for it to continue operating. |

| **Annotation Types Summary** | |
| --- | --- |
| [**Deprecated**](http://docs.google.com/java/lang/Deprecated.html) | A program element annotated @Deprecated is one that programmers are discouraged from using, typically because it is dangerous, or because a better alternative exists. |
| [**Override**](http://docs.google.com/java/lang/Override.html) | Indicates that a method declaration is intended to override a method declaration in a superclass. |
| [**SuppressWarnings**](http://docs.google.com/java/lang/SuppressWarnings.html) | Indicates that the named compiler warnings should be suppressed in the annotated element (and in all program elements contained in the annotated element). |

## Package java.lang Description

Provides classes that are fundamental to the design of the Java programming language. The most important classes are Object, which is the root of the class hierarchy, and Class, instances of which represent classes at run time.

Frequently it is necessary to represent a value of primitive type as if it were an object. The wrapper classes Boolean, Character, Integer, Long, Float, and Double serve this purpose. An object of type Double, for example, contains a field whose type is double, representing that value in such a way that a reference to it can be stored in a variable of reference type. These classes also provide a number of methods for converting among primitive values, as well as supporting such standard methods as equals and hashCode. The Void class is a non-instantiable class that holds a reference to a Class object represening the primitive type void.

The class Math provides commonly used mathematical functions such as sine, cosine, and square root. The classes String and StringBuffer similarly provide commonly used operations on character strings.

Classes ClassLoader, Process, Runtime, SecurityManager, and System provide "system operations" that manage the dynamic loading of classes, creation of external processes, host environment inquiries such as the time of day, and enforcement of security policies.

Class Throwable encompasses objects that may be thrown by the throw statement (§14.16). Subclasses of Throwable represent errors and exceptions.

## Package Specification

### Character Encodings

The specification of the [java.nio.charset.Charset](http://docs.google.com/java/nio/charset/Charset.html) class describes the naming conventions for character encodings as well as the set of standard encodings that must be supported by every implementation of the Java platform.

**Since:** JDK1.0

| | [**Overview**](http://docs.google.com/overview-summary.html) | **Package** | Class | [**Use**](http://docs.google.com/package-use.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV PACKAGE**](http://docs.google.com/java/io/package-summary.html)   [**NEXT PACKAGE**](http://docs.google.com/java/lang/annotation/package-summary.html) | [**FRAMES**](http://docs.google.com/index.html?java/lang/package-summary.html)    [**NO FRAMES**](http://docs.google.com/package-summary.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).