

Java.lang package in Java

Last Updated : 07 Aug, 2018

Java.lang package in Java

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.

Following are the Important Classes in Java.lang package :

1. **Boolean**: The Boolean class wraps a value of the primitive type boolean in an object.
2. **Byte**: The Byte class wraps a value of primitive type byte in an object.
3. **Character – Set 1, Set 2**: The Character class wraps a value of the primitive type char in an object.
4. **Character.Subset**: Instances of this class represent particular subsets of the Unicode character set.
5. **Character.UnicodeBlock**: A family of character subsets representing the character blocks in the Unicode specification.
6. **Class – Set 1, Set 2** : Instances of the class Class represent classes and interfaces in a running Java application.
7. **ClassLoader**: A class loader is an object that is responsible for loading classes.
8. **ClassValue**: Lazily associate a computed value with (potentially) every type.
9. **Compiler**: The Compiler class is provided to support Java-to-native-code compilers and related services.
10. **Double**: The Double class wraps a value of the primitive type double in an object.
11. **Enum**: This is the common base class of all Java language enumeration types.
12. **Float**: The Float class wraps a value of primitive type float in an object.
13. **InheritableThreadLocal**: 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.
14. **Integer** :The Integer class wraps a value of the primitive type int in an object.
15. **Long**: The Long class wraps a value of the primitive type long in an object.
16. **Math – Set 1, Set 2**: The class Math contains methods for performing basic numeric operations such as the elementary exponential, logarithm, square root, and trigonometric functions.
17. **Number**: The abstract class Number is the superclass of classes BigDecimal, BigInteger, Byte, Double, Float, Integer, Long, and Short.
18. **Object**: Class Object is the root of the class hierarchy.
19. **Package**: Package objects contain version information about the implementation and specification of a Java package.
20. **Process**: The ProcessBuilder.start() and Runtime.exec 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.

21. **ProcessBuilder**: This class is used to create operating system processes.
22. **ProcessBuilder.Redirect**: Represents a source of subprocess input or a destination of subprocess output.
23. **Runtime**: 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.
24. **RuntimePermission**: This class is for runtime permissions.
25. **SecurityManager**: The security manager is a class that allows applications to implement a security policy.
26. **Short**: The Short class wraps a value of primitive type short in an object.
27. **StackTraceElement**: An element in a stack trace, as returned by Throwable.getStackTrace().
28. **StrictMath- Set1, Set2**: The class StrictMath contains methods for performing basic numeric operations such as the elementary exponential, logarithm, square root, and trigonometric functions.
29. **String- Set1, Set2**: The String class represents character strings.
30. **StringBuffer**: A thread-safe, mutable sequence of characters.
31. **StringBuilder**: A mutable sequence of characters.
32. **System**: The System class contains several useful class fields and methods.
33. **Thread**: A thread is a thread of execution in a program.
34. **ThreadGroup**: A thread group represents a set of threads.
35. **ThreadLocal**: This class provides thread-local variables.
36. **Throwable**: The Throwable class is the superclass of all errors and exceptions in the Java language.
37. **Void**: The Void class is an uninstantiable placeholder class to hold a reference to the Class object representing the Java keyword void.

More articles:

- Flexible nature of java.lang.Object
- [Java.lang.system arraycopy method](#)
- Equals() and hashCode() methods in Java
- Why to Override equals(Object) and hashCode() method ?