<https://docstore.mik.ua/orelly/java/langref/ch09_01.htm>

<https://www.cs.cmu.edu/~pattis/15-1XX/15-200/lectures/exceptions/lecture.html> //anotherone

9.4 The Exception Hierarchy:

The possible exceptions in a Java program are organized in a hierarchy of exception classes. The Throwable class, which is an immediate subclass of Object, is at the root of the exception hierarchy. Throwable has two immediate subclasses: Exception and Error. Figure 9.1 shows the standard exception classes defined in the java.lang package, while Figure 9.2 shows the standard error classes defined in java.lang.

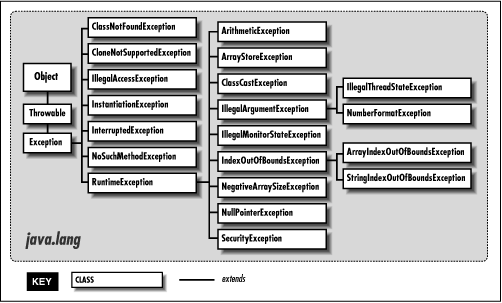


Figure 9.1: Standard Java exception classes

**Exceptions :**

All of the subclasses of Exception represent exceptional conditions that a normal Java program may want to handle. Many of the standard exceptions are also subclasses of RuntimeException. Runtime exceptions represent runtime conditions that can generally occur in any Java method, so a method is not required to declare that it throws any of the runtime exceptions. However, if a method can throw any of the other standard exceptions, it must declare them in its throws clause.

A Java program should try to handle all of the standard exception classes, since they represent routine abnormal conditions that should be anticipated and caught to prevent program termination.

***Runtime exceptions:***

The java.lang package defines the following standard runtime exception classes: ArithmeticException This exception is thrown to indicate an exceptional arithmetic condition, such as integer division by zero.

***ArrayIndexOutOfBoundsException:***

This exception is thrown when an out-of-range index is detected by an array object. An out-of-range index occurs when the index is less than zero or greater than or equal to the size of the array.

***ArrayStoreException:-***

This exception is thrown when there is an attempt to store a value in an array element that is incompatible with the type of the array.

***ClassCastException:-***

This exception is thrown when there is an attempt to cast a reference to an object to an inappropriate type.

***IllegalArgumentException:-***

This exception is thrown to indicate that an illegal argument has been passed to a method.

***IllegalMonitorStateException:-***

This exception is thrown when an object's wait(), notify(), or notifyAll() method is called from a thread that does not own the object's monitor.

***IllegalStateException:-*** This exception is thrown to indicate that a method has been invoked when the run-time environment is in an inappropriate state for the requested operation. This exception is new in Java 1.1.

***IllegalThreadStateException:-***This exception is thrown to indicate an attempt to perform an operation on a thread that is not legal for the thread's current state, such as attempting to resume a dead thread.

***IndexOutOfBoundsException:-*** The appropriate subclass of this exception (i.e., ArrayIndexOutOfBoundsException or StringIndexOutOfBoundsException) is thrown when an array or string index is out of bounds. NegativeArraySizeException This exception is thrown in response to an attempt to create an array with a negative size.

***NullPointerException:-*** This exception is thrown when there is an attempt to access an object through a null object reference. This can occur when there is an attempt to access an instance variable or call a method through a null object or when there is an attempt to subscript an array with a null object.

***NumberFormatException:-*** This exception is thrown to indicate that an attempt to parse numeric information in a string has failed.

***RuntimeException:-*** The appropriate subclass of this exception is thrown in response to a runtime error detected at the virtual machine level. Because these exceptions are so common, methods that can throw objects that are instances of RuntimeException or one of its subclasses are not required to declare that fact in their throws clauses.

***SecurityException:-*** This exception is thrown in response to an attempt to perform an operation that violates the security policy implemented by the installed SecurityManager object.

***StringIndexOutOfBoundsException***:- This exception is thrown when a String or StringBuffer object detects an out-of-range index. An out-of-range index occurs when the index is less than zero or greater than or equal to the length of the string. docstore.

**Other exceptions:**

The java.lang package defines the following standard exception classes that are not runtime exceptions:

***ClassNotFoundException:-*** This exception is thrown to indicate that a class that is to be loaded cannot be found.

***CloneNotSupportedException:-*** This exception is thrown when the clone() method has been called for an object that does not implement the Cloneable interface and thus cannot be cloned. ***Exception:-*** The appropriate subclass of this exception is thrown in response to an error detected at the virtual machine level. If a program defines its own exception classes, they should be subclasses of the Exception class.

***IllegalAccessException:-*** This exception is thrown when a program tries to dynamically load a class (i.e., uses the forName() method of the Class class, or the findSystemClass() or the loadClass() method of the ClassLoader class) and the currently executing method does not have access to the specified class because it is in another package and not public. This exception is also thrown when a program tries to create an instance of a class (i.e., uses the newInstance() method of the Class class) that does not have a zero-argument constructor accessible to the caller.

***Instantiation Exception:-*** This exception is thrown in response to an attempt to instantiate an abstract class or an interface using the new Instance() method of the Class. ***InterruptedException:-*** This exception is thrown to signal that a thread that is sleeping, waiting, or otherwise paused has been interrupted by another thread.

***NoSuchFieldException:-*** This exception is thrown when a specified variable cannot be found. This exception is new in Java 1.1.

***NoSuchMethodException:-*** This exception is thrown when a specified method cannot be found.