JAVA.LANG.EXCEPTIONS

Introduction

The **java.lang.Exceptions** provides for different exceptions thrown under java lang package.

Exception Summary

S.N.	Exception & Description
1	**ArithmeticException** This is Thrown when an exceptional arithmetic condition has occurred.
2	**ArrayIndexOutOfBoundsException** This is Thrown to indicate that an array has been accessed with an illegal index.
3	**ArrayStoreException** This is Thrown to indicate that an attempt has been made to store the wrong type of object into an array of objects.
4	**ClassCastException** This is Thrown to indicate that the code has attempted to cast an object to a subclass of which it is not an instance.
5	**ClassNotFoundException** This is Thrown when an application tries to load in a class through its string name using: The forName method in class Class.
6	**CloneNotSupportedException** This is 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.
7	**EnumConstantNotPresentException** This is Thrown when an application tries to access an enum constant by name and the enum type contains no constant with the specified name.
8	**Exception** This is the class Exception and its subclasses are a form of Throwable that indicates conditions that a reasonable application might want to catch.
9	**IllegalAccessException** This is an IllegalAccessException is thrown when an application tries to reflectively create an instance *otherthananarray*, 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.

10 **IllegalArgumentException**

This is Thrown to indicate that a method has been passed an illegal or inappropriate argument.

11 **IllegalMonitorStateException**

This is 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.

12 **IllegalStateException**

The Signals that a method has been invoked at an illegal or inappropriate time.

13 **IllegalThreadStateException**

This is Thrown to indicate that a thread is not in an appropriate state for the requested operation.

14 **IndexOutOfBoundsException**

This is Thrown to indicate that an index of some sort *suchastoanarray*, *toastring*, *ortoavector* is out of range.

15 **InstantiationException**

This is 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.

16 **InterruptedException**

This is Thrown when a thread is waiting, sleeping, or otherwise occupied, and the thread is interrupted, either before or during the activity.

17 **NegativeArraySizeException**

This is Thrown if an application tries to create an array with negative size.

18 **NoSuchFieldException**

The Signals that the class doesn't have a field of a specified name.

19 **NoSuchMethodException**

This is Thrown when a particular method cannot be found.

20 **NullPointerException**

This is Thrown when an application attempts to use null in a case where an object is required.

21 **NumberFormatException**

This is 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.

22 **RuntimeException**

This is RuntimeException is the superclass of those exceptions that can be thrown during the normal operation of the Java Virtual Machine.

23 **SecurityException**

This is Thrown by the security manager to indicate a security violation.

24 **StringIndexOutOfBoundsException**

This is Thrown by String methods to indicate that an index is either negative or greater than the size of the string.

25 **TypeNotPresentException**

This is 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.

26 **UnsupportedOperationException**

This is Thrown to indicate that the requested operation is not supported.

Loading [MathJax]/jax/output/HTML-CSS/jax.js