**Java Exception**

An unexpected unwanted event that disturbs normal flow of program is called Exception.

Ex – FilenotFoundException, NullPointerException etc.

It is recommended to handle exception and the main objective is graceful termination of program.

Exception handling doesn’t mean repairing an exception. We have to provide alternate way to continue rest of the program normally.

**Runtime Stack Mechanism:**

For every thread, a runtime stack is created by JVM. All methods calls performed by that thread will be stored in the corresponding stack. Each entry in stack is called Stack frame or Activation call. After completing every method call, corresponding entry will be removed. After completing all method calls, the stack will become empty and that empty stack will be deleted by JVM before terminating thread.

Ex

Class Test{

P s v main(){

doStuff();

}

|  |
| --- |
|  |
|  |
|  |
|  |
|  |

doStuff(){

doMoreStuff();

}

doMoreStuff(){

s.o.p(“Hello”);

}

}

|  |
| --- |
|  |
|  |
|  |
|  |
|  |

|  |
| --- |
|  |
|  |
|  |
|  |
| main() |

|  |
| --- |
|  |
|  |
|  |
| doStuff() |
| main() |

|  |
| --- |
|  |
|  |
| doMoreStuf |
| doStuff() |
| main() |

|  |
| --- |
|  |
|  |
|  |
| doStuff() |
| main() |

|  |
| --- |
|  |
|  |
|  |
|  |
| main() |

Runtime Stack for main Thread for normal program flow.

**Default exception handling in Java:**

* Inside a method if any exception occurs the method in which it is raised is responsible to create exception object by including following information.

1. Name of exception
2. Exception Description
3. Location at which exception occurs (Stack trace)

* After creating Exception object, method handover the object to the JVM.
* JVM will check whether the method contains any exception handling code or not. If the method doesn’t have exception handling code, then JVM terminates the method abnormally and removes the corresponding entry from the Stack.
* Then JVM identifies caller method and checks whether caller method contains any handling code or not. If no handling code is found, then JVM terminates caller method also abnormally and removes the corresponding entry from the Stack.
* This process continues until the main method and if main method doesn’t have handling code, JVM terminates main abnormally and removes corresponding entry from the Stack.
* Then JVM handovers responsibility of Exception handling to DefaultException Handler, which is the part of JVM.
* DefaultExceptionHandler prints exception information in the following format and terminates program abnormally.
  + Exception in thread "xxx" “name of exception”:“Description” :Stack trace

public class Test{

public static void main(String []args){

doStuff();

}

public static void doStuff(){

doMoretuff();

}

public static void doMoretuff(){

System.out.println(10/ 0);

}

}

Exception in thread "main" java.lang.ArithmeticException: / by zero

at Test.doMoretuff(Test.java:12)

at Test.dostuff(Test.java:8)

at Test.main(Test.java:4)

**Note:**

In a program if one methods terminate abnormally, then program termination is abnormal termination.

If all methods terminate normally, then program termination is normal termination

**Exception Hierarchy:**

Throwable acts as root for Java exception hierarchy.

Throwable class defines 2 child classes:

1. Exception
2. Error

Exception:

Most if the times, exception are caused by our program and these are recoverable.

Ex: If our programming requirement to read data from remote file locate in remote storage at run time, if remote file is not available, then we get runtime exception – “FileNotFoundException”

If FileNotFoundException occurs we can provide local file and continue rest of the program normally.

try{

//read file at remote location

}catch (FileNotFoundException e){//use local file are continue normally)

Error:

Most if the times, errors are not caused by our program and this are due to lack of system resources.

Errors are non – recoverable.

Ex: If OutOfMemoryError occurs, the program terminates abnormally. System admin or server admin is responsible to increase heap memory.

C:\Users\sbanerje091417\Downloads\ExceptionHierarchy.png