Exception Handling

Exception:

* Unwanted event occur during runtime which results in abrupt termination (execution) of our application or program.

Handling Exception:

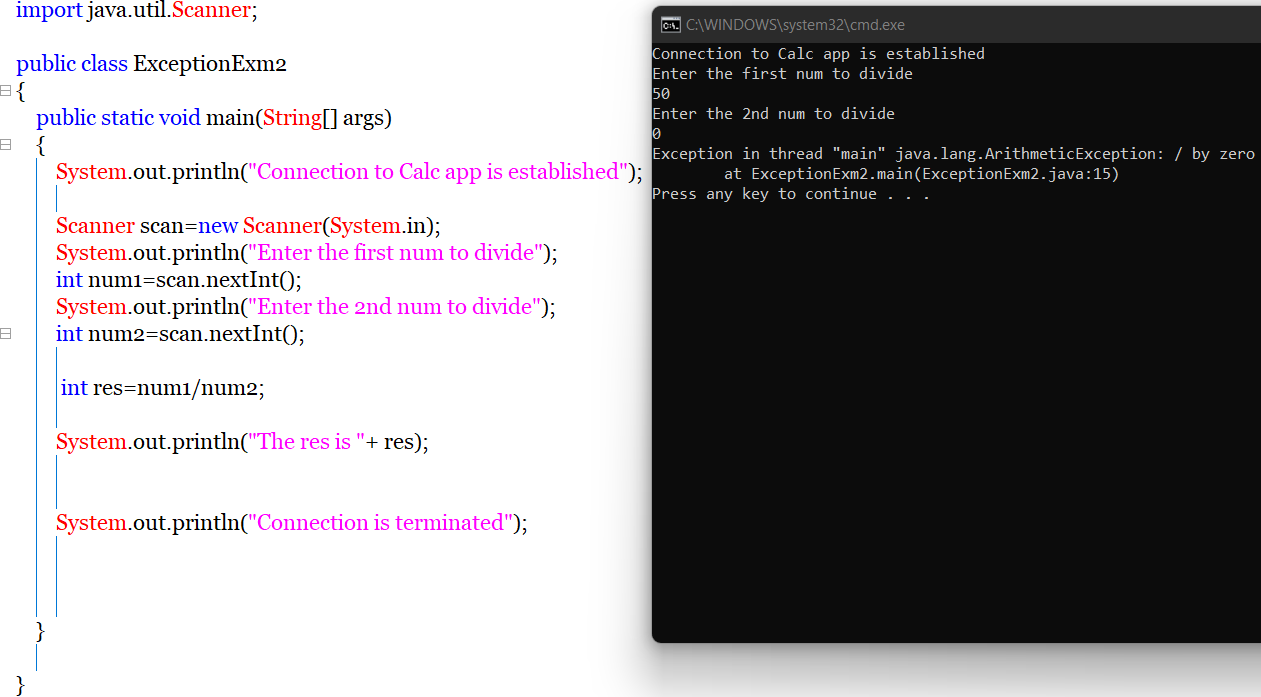
* Handling exception in such a way that application or program termination should not happen.

Keywords used in Exception Handling:

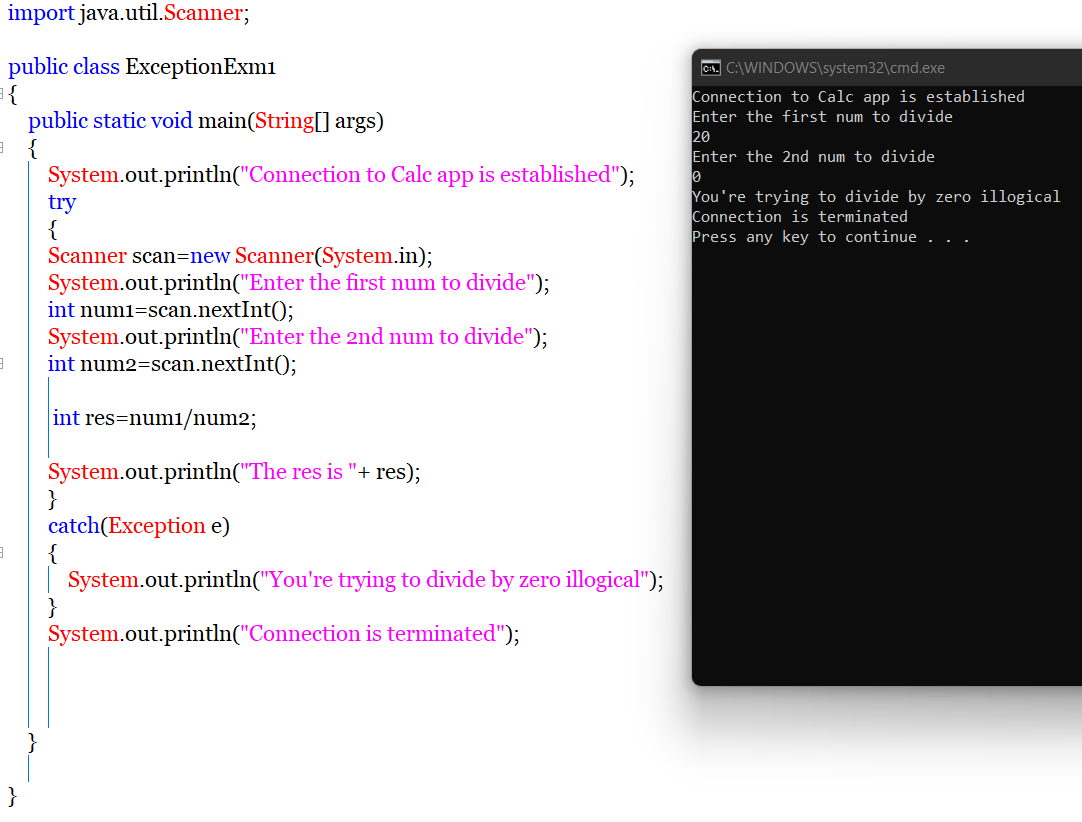
* try-catch, throws, throw, finally--------

Runtime Stack:

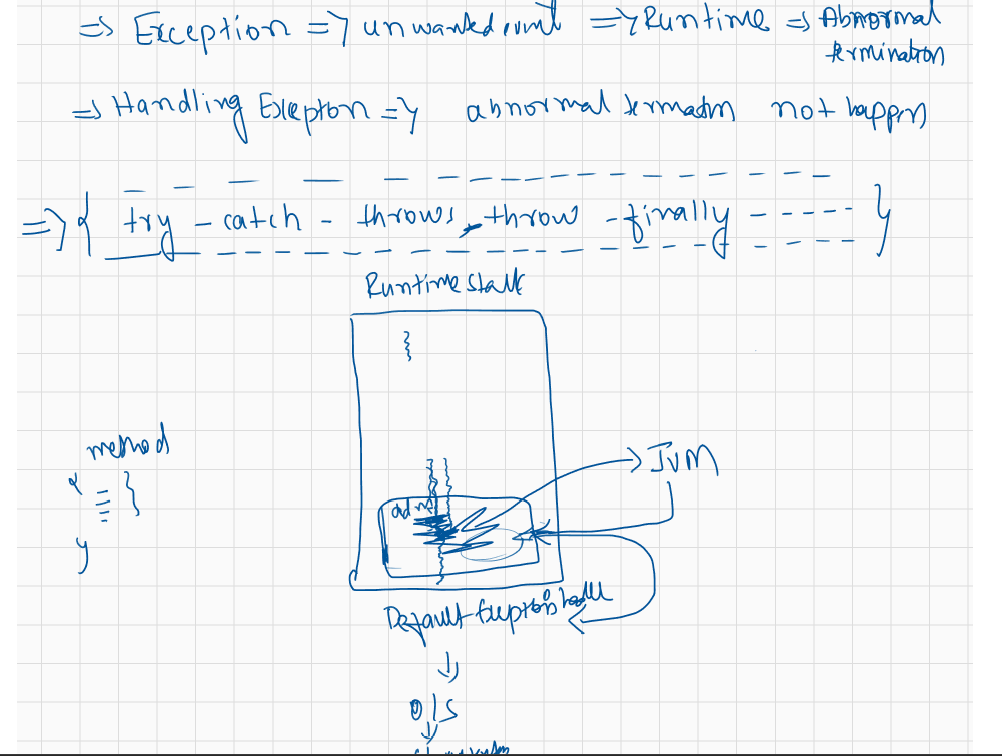
* Anything or everything in our program, if it has to execute (like logic or activity inside a method), it has to come to area called “Runtime Stack Area”. Here execution will occur.
* In whichever method logic stack frame gets exception that method stack generates Exception Object.
* If it occurs in main method-----main method stack creates exception object, if it is add method -----add method stack creates exception object like this----
* That exception object will be thrown to JVM first.
* Inside JVM there is a feature called Run Time System(RTS). JVM internally using that RTS it will do the later activities.
* JVM goes to the same stack frame and checks in that method whether developer handles that exception or not.



If handled in that method itself…



* If not handled JVM handed over that exception to Default Exception Handler. If that is also not happens then O/S will come into the picture and that also fails then System crash comes into picture. Mostly it won’t happen. Default handler will takes care.



* Exception hierarchy—Inbuilt classes using which we can handle exceptions(We can know who is parent, which is child etc and usage to handle exception)
* To avoid Exception===we have to follow Handling using like try-catch block, with in that block, we will use inbuilt classes.
* Checked or unchecked exception===both occur during run time only, it’s not like compile time exception and run time exception

|  |  |
| --- | --- |
| Checked Exception | Unchechecked Exception |
|  |  |

How to handle Exception?

* Using try and catch block…for every try block we have to provide catch block also

try {

// *Block of code to try(risky statements in developer’s view)*

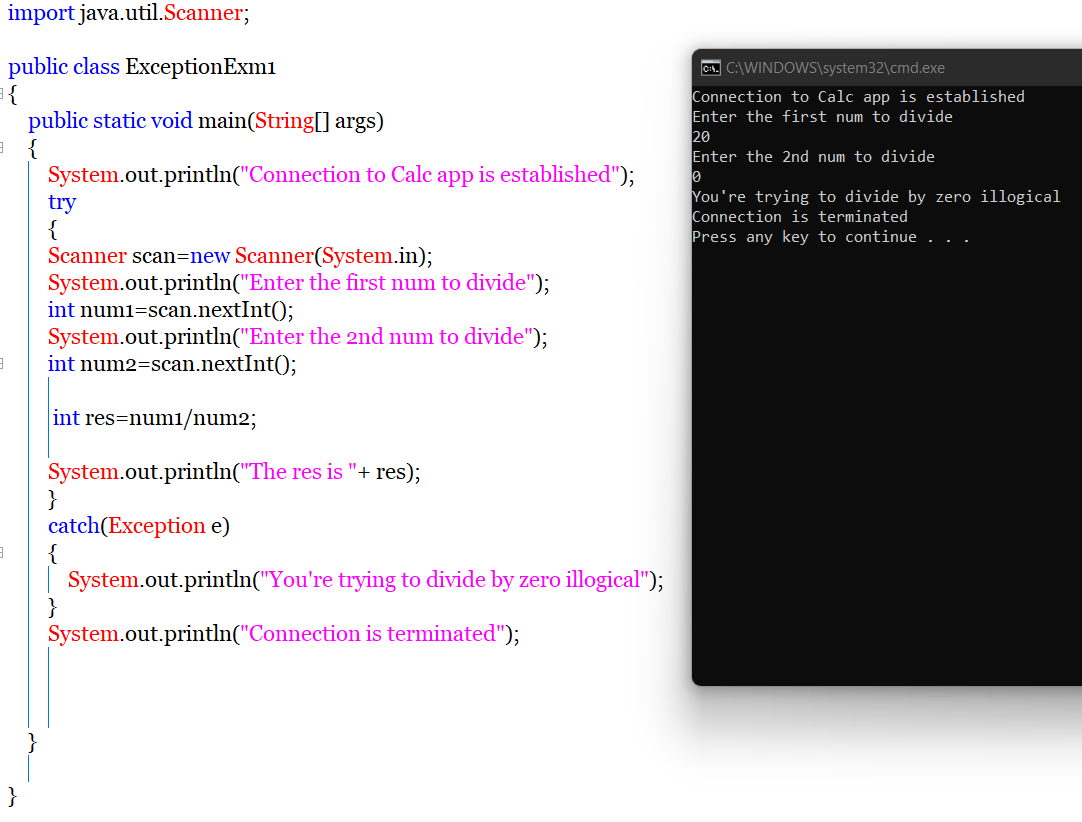
}

catch(inbuilt classes/utility classes) {

// *Block of code to handle errors*

}

* Whenever exception occur in try block since code won’t execute, we need to provide related/suitable info in catch block.In the below picture we can see catch block execution.



* In the catch block in the parenthesis (Exception name …) we have to mention what is the type of related Exception. Usually Exception is the parent class and can handle any type of exception but we write like AthematicException,ArrayIndexOutofBoundsExpn etc…..
* Catch block will execute when exception occurs in try block otherwise it won’t.

The statements from exception occurring point to till catch block(middle lines) will never execute.

* Single try block can have multiple catch blocks.
* After all specific catch blocks it’s highly recommended to provide generic catch block(Exception e) at the end which is parent of all exceptions and can handle all exceptions.