Object Oriented Programming JAVA

Dr. Prafulla Kalapatapu

Computer Science Engineering

Mahindra Ecole Centrale

prafulla.kalapatapu@mechyd.ac.in

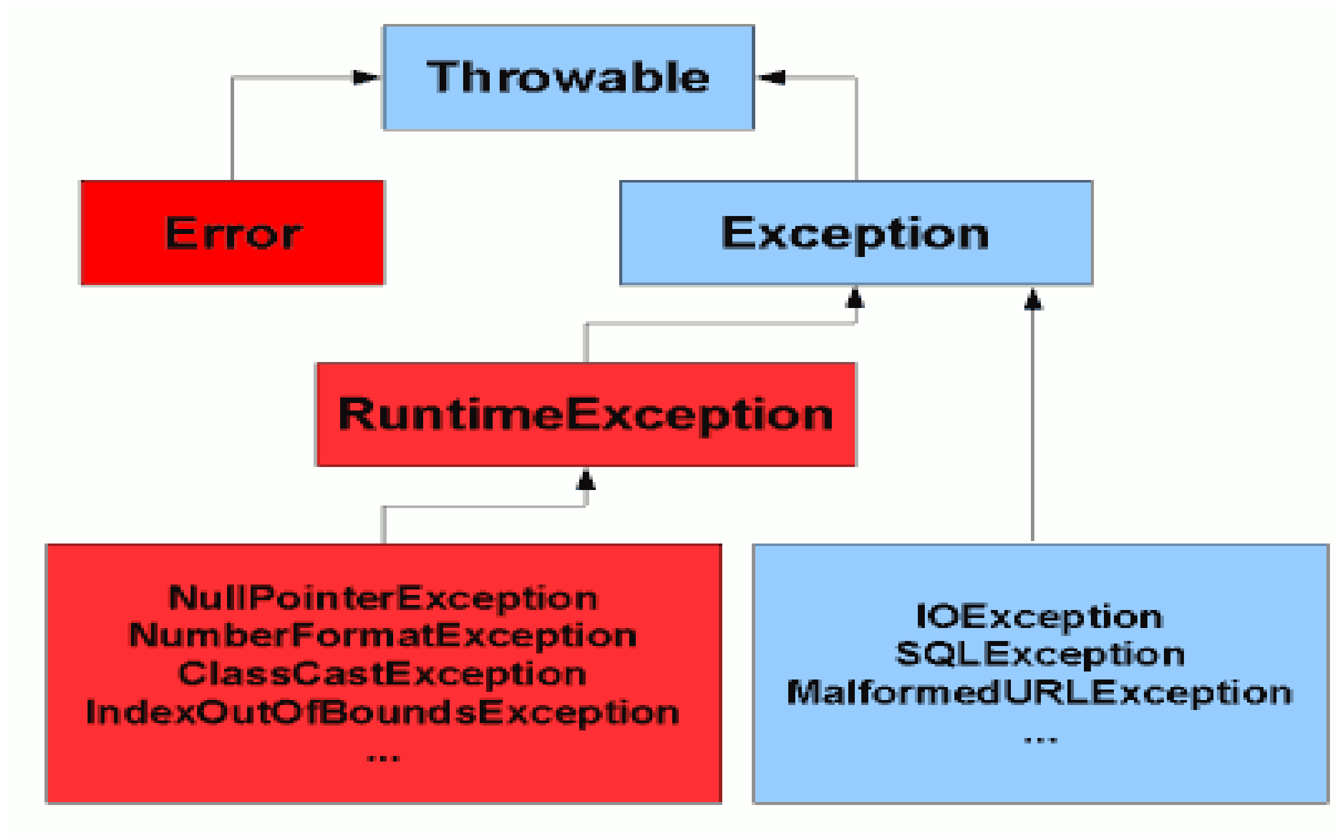
Mahindra
École Centrale
COLLEGE OF ENGINEERING

Exception Handling

LEADER ■ ENTREPRENEUR ■ INNOVATOR

Java Exception Hierarchy

Mahindra École Centrale
COLLEGE OF ENGINEERING



classes in Red and their sub classes are Unchecked Exceptions and all other are Checked Exceptions

Try with multiple catch blocks

- The way of handling an exception is varied from exception to exception.
- Hence for every exception type we have to maintain separate catch block that is try with multiple catch blocks is possible and recommended to use.
- Note: if try with multiple catch block present then the order of catch block is always important. It should be from child to parent. Otherwise we will get compile time error.
- Compile time error is

  Exception xxxx has already been caught

Examples

Mahindra École Centrale
COLLEGE OF ENGINEERING

a.

```
try
{
int a=10/0;
}
catch(ArithmeticException ae)
{
}
catch(Exception e)
{
}
```

b.

```
try
{
int a=10/0;
}
catch(Exception e)
{
}
catch(ArithmeticException ae)
{
}
```

Valid

Invalid

CE: Exception ArithmeticException has already been caught

Mahindra École Centrale
COLLEGE OF ENGINEERING

Very Important :

- If there is no chance of raising an exception in try block, then we are not allowed to define catch block.
- Otherwise compile time error
- But this rule applicable only for fully checked exception.

Mahindra École Centrale
COLLEGE OF ENGINEERING

a.
```
try
{
System.out.println(“Hi”);
}
catch(ArithmeticException ae)
{
}
```
Unchecked Exception

Valid

b.
```
try
{
System.out.println(“Hi”);
}
catch(Exception e)
{
}
```
partially Exception

Valid

c.
```
try
{
System.out.println(“Hi”);
}
catch(IOException e)
{
}
```
Fully checked Exception

Invalid

Clean-up code

- What is clean-up code

  - File closing

  - DB connection closing

  - object reference assigned to null

- Can we write clean-up code in try block

  yes, but limitation is, if there is any exception in the try block, it wont execute the clean-up code.

- Can we write clean-up code in the catch block

  yes, but the limitation is, if there is respective exception only executes clean-up code

- Whether exception occurs or not, program’s clean-up code has to execute. Is there any block which does the above.

  finally

- Note:

  - We can write try, catch, finally together or try, finally.

  - We cant write finally block individually. It should be paired up with try.

- The main objective of finally block is to maintain clean-up code which should be executed always

a.
```
try
{
System.out.println(“try”);
}
catch(Exception e)
{
System.out.println(“catch”);
}
finally
{
System.out.println(“finally”);
}
```
o/p : try
finally

b.
```
try
{
System.out.println(10/0);
}
catch(ArithmeticException e)
{
System.out.println(“catch”);
}
finally
{
System.out.println(“finally”);
}
```
o/p : catch
finally

c.
```
try
{
System.out.println(10/0);
}
catch(NPE e)
{
System.out.println(“catch”)
}
finally
{
System.out.println(“finally”
}
```
o/p : finally
Abnormal termination

Mahindra École Centrale
COLLEGE OF ENGINEERING

- With in the try or in catch, if there is any return statement, it will be executed only after executing finally block

```
class  Test {
public static void main(String[] a) {
try
{
System.out.println(“try”);
return ;
}
catch(Exception e)
{
System.out.println(“catch”);
}
finally
{
System.out.println(“finally”);
}
}
}
```

o/p : try
finally

finally dominates return statement

- There is only one situation where the finally block wont execute when ever we use System.exit(0) then there is no chance of executing finally block

```
class  Test {
public static void main(String[] a) {
try
{
System.out.println("try");
System.exit(0);
}
catch(Exception e)
{
System.out.println("catch");
}
finally
{
System.out.println("finally");
}
}
}
```

o/p : try

System.exit() dominates finally block

Various possible combination of try, catch, finally

Mahindra École Centrale
COLLEGE OF ENGINEERING

a.
```
try {
}
catch(ArithmeticException ae) {
}
finally {
}
```
Valid

b.
```
try {
}
catch(ArithmeticException ae) {
}
```
Valid

c.
```
try {
}
finally {
}
```
Valid

d.
```
try {
}
```
CE: try without catch or finally

e.
```
try {
}
finally {
}
catch(ArithmeticException ae) {
}
```
CE: catch without try

f.
```
try {
}
System.out.println(“hi”);
catch(ArithmeticException ae) {
}
```
CE: try without catch or finally
catch without try

g.
```
try {
}
catch(ArithmeticException ae) {
}
System.out.println(“hi”);
finally {
}
```
CE: finally without try

h.
```
try {
}
catch(ArithmeticException ae) {
}
finally {
}
finally {
}
```
CE: finally without try

i.
```
finally {
}
```
CE: without try

j.
```
try {
}
System.out.println(“hi”);
finally {
}
```
CE: try without catch or finally
finally without try

throw

Mahindra École Centrale
COLLEGE OF ENGINEERING

- who throws an Exception if there is any Exception at any statement in the program.

  JVM

- JVM throws Exception to respective catch block otherwise it throws outof the program execution(it leads abnormal termination).

- Can Programmer explicitly throws an Exception.

  Yes

- How?

  Using throw keyword.

- Can Throw any Exception using throw keyword

  yes(Userdefined as well as predefined Exceptions)

- Syntax:

  throw object/refrence;

- Example:

  throw new ArithmeticException("Don't give 0 as Denom");

or

  ArithmeticException a=new ArithmeticException("Divide / Zero Error");

  throw a;

- Userdefined Exception:

  An Exception class defined by the programmer , those are called as user defined exceptions.

- why we need Userdefined Exceptions:

  programmer wants to handle customized exception(like negative salary, voteagecheck etc), at that instance, programmer has to write his own Exception class..

- Rules to write Userdefined Exceptions:

1) write a separate class to handle exception.
2) that class should extends from Exception class (directly or indirectly).
3) public constructor(optional)

- How can we use Userdefined Exceptions:

  1) create an object to Userdefined Exception class.

  2) throw that object explicitly using throw keyword.