Mahindra École Centrale
COLLEGE OF ENGINEERING

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

Exception Handling

Mahindra École Centrale
COLLEGE OF ENGINEERING

- What is bug

  Errors are called bugs

- Debugging

  Process of removing errors/bugs.

- Errors in java program

  There are 3 types of errors in java program

1. Compile time error
2. Runtime error
3. Logical error

1. Compile time error :

- These are syntactical errors found in the code, due to which a program fails to compile .

  Ex: forgetting semicolon etc..
- Easily programmers can resolve the errors guided by the compiler in java.

2. Runtime error :

- These errors represent inefficiency of the computer system to execute a particular statement

  Ex: inability of microprocessor to execute some statement like

  public static void main()
- Runtime errors are not detected by the java compiler. They are detected by the JVM and only at runtime.

3. Logical error :

- These errors depict flaws in the logic of he program. The programmer might be using a wrong formula in the program.
- Logical errors are not detected by java compiler or JVM.
- The programmer is solely responsible for them.

Exception

Mahindra École Centrale
COLLEGE OF ENGINEERING

- What is an Exception
  - It is a runtime error
  - An unwanted or unexpected event that disturb normal flow of the program execution is called exceptions

- Can I call compile time error as an exception
  No

- Types of Exceptions
  - Checked exception
  - Unchecked exception

- Checked exception :
  The exceptions that are detected by java compiler

- Unchecked exceptions :
  The exceptions that are detected by JVM.

- Program termination – 2 types
  - Normal termination
  - Abnormal termination

Normal termination :

It executes all statements and it will exit from the program without any unwanted disturbances.

```
public static void main(String []a)
{
        s1;
        s2;
        s3;
        s4;
}
```

s1, s2, s3, s4 are statements

All will execute and after s4, it will come out of the scope.

Abnormal termination :

program will exit with some unwanted disturbances. It may happen at any statement and from there it will come out.

```
public static void main(String []a)
{
        s1;
        s2; // unwanted disturbance / unexpected event
        s3;
        s4;
}
```

s1, s2, s3, s4 are statements

It will come out of the program at s2 and s3,s4 wont be executed.

- Disadvantage of abnormal termination :
  - User doesn't know, what went wrong.
  - Loss of work
  - Resources taken from operating system not submitted gracefully

- What is the solution for abnormal termination.
  Exception handling

- What is exception handling

  - Handling an exception to avoid abnormal termination

  - Handling an exception doesn't mean repairing an exception, we are providing alternative way to continue rest of the program normally.

- Objective of exception handling

  The main objective of exception handling is graceful termination of the program

- How can we implement exception handling in java

  Using try and catch block

- What is try and catch block

```
try : syntax :          try
                        {
                        // doubtfull code (set of statements,
        may or may not have unwanted disturbances
                        }
catch : syntax :        catch(Exceptionname   refvariable)
                        {
                        // statements to handle exception
                        }
```

- Note:

  - try block should immediately followed by catch block

  - Not allowed to write any statements between try and catch. It leads to compile time error

- What is the responsibility of try block

  JVM executes statement by statement from the try block. If any exception occurs at any statements, JVM will throw that exception to the respective catch block.

- What is the responsibility of catch block

  - Catching exception thrown by JVM from try block

  - Handling that exception. That means printing proper message, which can be understand by the user.

- Write a java program, in which an exception raises, but it is not handled by using try and catch block.

Save as Demo.java

```
class Sample {
public static void main(String []a) {
System.out.println("s1");
System.out.println("s2");
int c=3/0;
System.out.println("s4");
System.out.println("s5");
System.out.println("s6");
} }
```

Exception is raised in this statement

Default exception format :
Fully qualified name of the exception : description
stack trace

o/p : s1
s2
Exception in thread "main" java.lang.ArithmeticException : / by zero
at Sample.main(Demo.java:5)

- How to handle exception in previous program example

Save as Demo.java

```
class Sample {
public static void main(String []a) {
System.out.println("s1");
System.out.println("s2");
try {
int c=3/0;        Doubtful code
}
catch(ArithmeticException ae) {
System.out.println("don't give zero as denominator");
System.out.println("s4");
System.out.println("s5");
} }
```

o/p:

```
s1
s2
don't give zero as denominator
s4
s5
```

Exception Hierarchy

Mahindra École Centrale
COLLEGE OF ENGINEERING

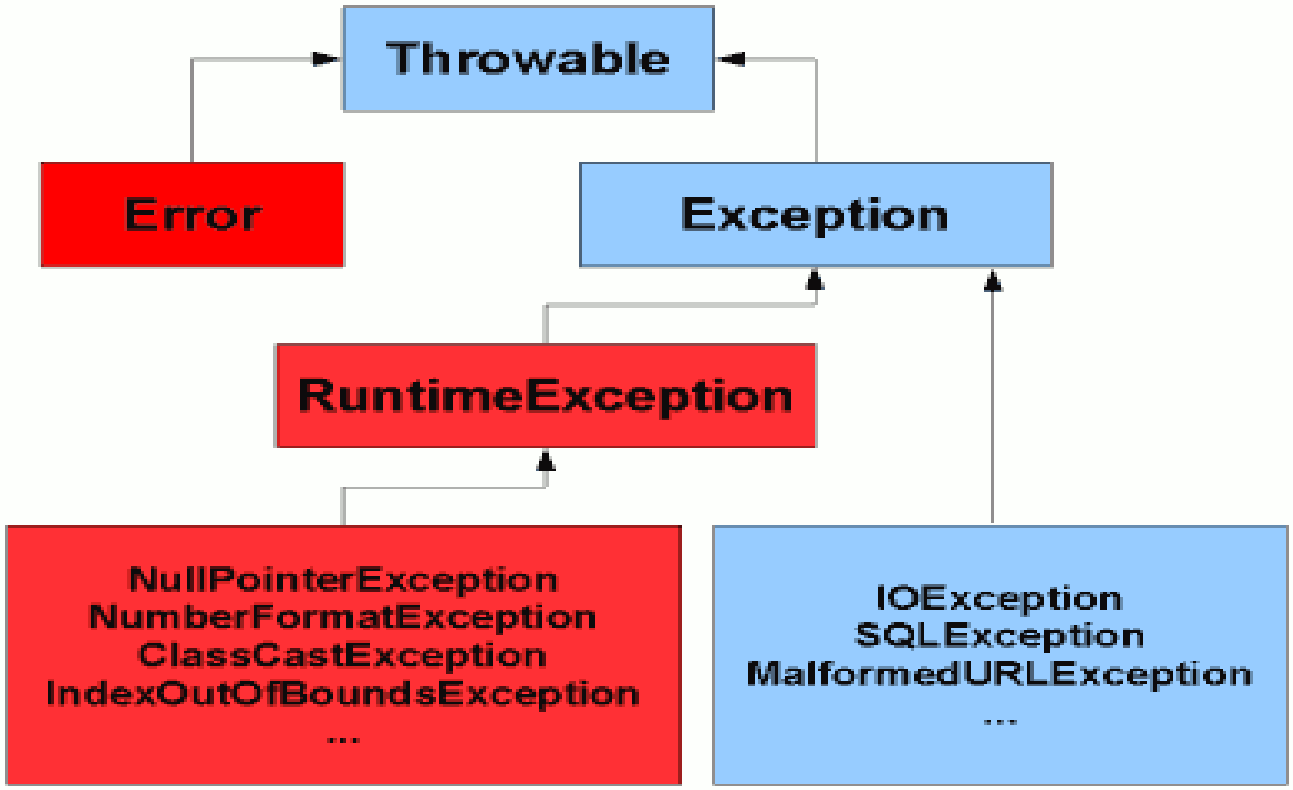
- All exceptions and errors in java are child classes of Throwable either directly or indirectly.
- Java Exception hierarchy starts from Throwable.
- Throwable contains two child classes, i.e. Exception and Error.

- Exception: These are mostly caused by our program and these are recoverable.

- Error: These are not caused by our program, these are due to lack of system resources. These are unrecoverable.

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

Checked Vs Unchecked Exceptions

Mahindra École Centrale
COLLEGE OF ENGINEERING

- The exceptions which are checked by compiler for smooth execution of program at runtime.

  Eg: FileNotFoundException, EOFException

- The exceptions which are not checked by compilers.

  Eg: NullPointerException, ClassCastException.

- RuntimeException and its child classes, error and its child classes are unchecked and all the remaining by default considered as checked Exceptions.

- Whether Exception is checked or unchecked it will occur at runtime, there is no chance of occurring at compile time.

Partially checked Vs fully checked

Mahindra École Centrale
COLLEGE OF ENGINEERING

- A checked exception should be fullychecked, if and only if every child is also checked.

  Eg: IOException, InterruptedException.

- A checked exception is to be partially checked, if and only if, its child classes need not be checked.

  Eg: Throwable, Exception

Methods to print Exception Information

Mahindra École Centrale
COLLEGE OF ENGINEERING

Throwable class defines the following three methods to display Exception Information.

1. printStackTrace() : It prints Exception information in the following format
   Name of Exception : description
   stackTrace.

2. toString() : It returnsException information in the following format
   Name of Exception : description

3. getMessage() : It returns just description of the exception.
   description

Note: Default exception handler always uses printStackTrace() method.

Eg:

```
class Test {
public static void main(String[] a) {
try {
        System.out.println(10/0);
    }
catch(ArithmeticException e) {
        e.printStackTrace();
        System.out.println(e);
        System.out.println(e.getMessage());
    }
}
}
```

AE: / by zero

AE: / by zero
at main

// System.out.println(e.toString());

/ by zero