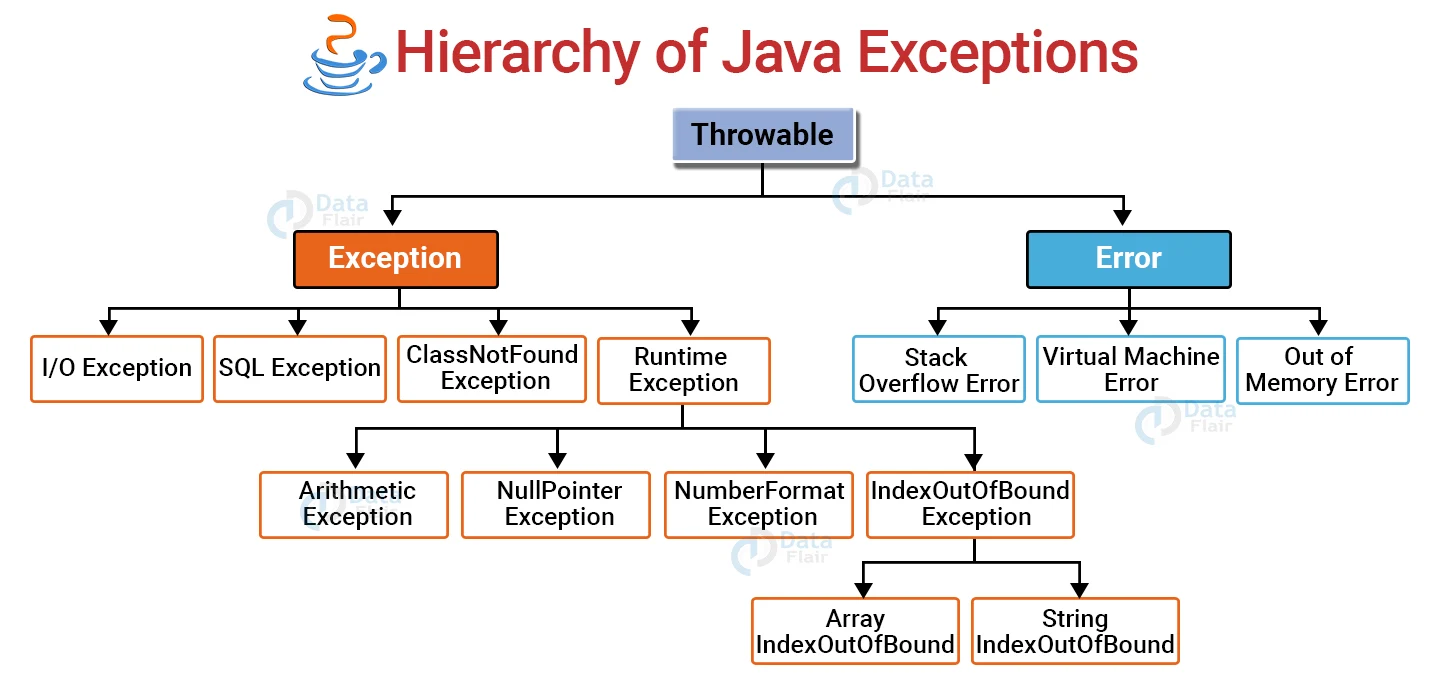
**Exception Handling**

1. **What is Exception Handling in Java?**

* Exception means Abnormal Condition.
* Exception Handling in Java is a powerful mechanism to handle abnormal condition which will occurs while running Java Program so that the normal flow of the application can maintained.
* Exception Handling is a mechanism to handle runtime errors such as ClassNotFoundException, IOException, SQLException, RemoteException, etc.



1. **What is explicit and implicit in java?**

* **Explicit** means something is done by the programmer **Implicit** means that it’s done by the complier.

1. **What is the difference between Checked and Unchecked Exception?**

* Checked Exception occurs while Compiling Time like I/OException, SQLException, ClassNotFound.
* Unchecked Exception occurs Runtime like NullPointerException, NumberFormatException.

1. **What is Try Catch Block in Java?**

* Try-Catch Block is use for Handling Exception
* Try block is use for enclose the data that might be throw an Exception.
* Catch block is use to catch those exception thrown by try block.
* Try-Catch Block is must be in within the Method.
* Try block must having catch or finally Block.

The JVM firstly checks whether the exception is handled or not. If exception is not handled, JVM provides a default exception handler that performs the following tasks:

* Prints out exception description.
* Prints the stack trace (Hierarchy of methods where the exception occurred).
* Causes the program to terminate.

1. **Can we Use Multiple Catch Block and Why?**

* Yes we can Use multiple Catch Block.
* Each Block contain Different Exception Handler.
* If there is possibility to have multiple exception will occurs then we will use multiple Catch Block.

1. **What is Nested Try-Catch Block in Java and why we use and how it will Work?**

* As name suggest Nested Try-Catch Block mean’s Try-Catch block in Try Block.
* If we want use particular Exception handle in some block of code in Try Block then we use Nested try-catch.
* If inner try catch block dose not handle exception the outer will handle it.
* If any inner catch block will not handle then main try catch block will handle.
* If none of the catch block will unable to handle then Java runtime Exception handle the Exception.

1. **What is Finally in Java?**

* Finally is a block which is use to execute important code like closing of connection.
* Finally block is always execute whether exception is handle or not.

1. **Is there any possibility when finally block is not executed?**

* Yes, If before calling finally block there is System.exit(0); Statement is present.

1. **Difference between Final, Finally and Finalize**

|  |  |  |
| --- | --- | --- |
| **Final** | **Finally** | **Finalize** |
| Final keyword is use to restrict the class, method, or variables. | Finally is a block which will execute whether the exception occurs or not. | Finalize is the method of Java which will perform to clean up process just before Garbage collected. |
| Final method will execute when we call. | Finally block will always execute if before this System.exit() is not present. | Finalize method is execute just before the object is destroyed. |
| (1) Once declared, final variable becomes constant and cannot be modified.  (2) final method cannot be overridden by sub class.  (3) final class cannot be inherited. | (1) finally block runs the important code even if exception occurs or not.  (2) finally block cleans up all the resources used in try block | finalize method performs the cleaning activities with respect to the object before its destruction. |
| final int age = 18; | // executes regardless of exception occurred or not  finally {  System.out.println("finally block is always executed");  } | // defining the finalize method  protected void finalize() {  System.out.println("Called the finalize() method");  } |

1. **What is the difference between the throw and throws keyword?**

|  |  |
| --- | --- |
| **Throw** | **Throws** |
| Throw is keyword used to explicitly throw the exception within a method or block of code. | Throws is used in Method signature to declare the type of exception that a method may throw |
| public void someMethod() {  if (condition) {  throw new SomeException("Some error occurred");  }  } | public void someMethod() throws SomeException {  // method implementation  } |

1. **What is Java Custom Exception and How and Why we use?**

* We can create our own Exception that is derived class of the Exception class is called as Custom Exception.
* We can create this for to catch and provide specific treatment for application.
* Also to understand if any specific exception will happen in Business Logic so we can catch this exact problem in application and help to understand a developer.

public class WrongFileNameException extends Exception {

public WrongFileNameException(String errorMessage) {

super(errorMessage);

}

}

**Exception handling with Method Overriding**

* If Superclass Method dose not declare an Exception then Subclass overridden method cannot declare the checked exception but it can declare unchecked exception.
* If superclass method declare an exception the subclass overridden method can declare same exception, subclass exception or no exception but cannot declare parent exception.