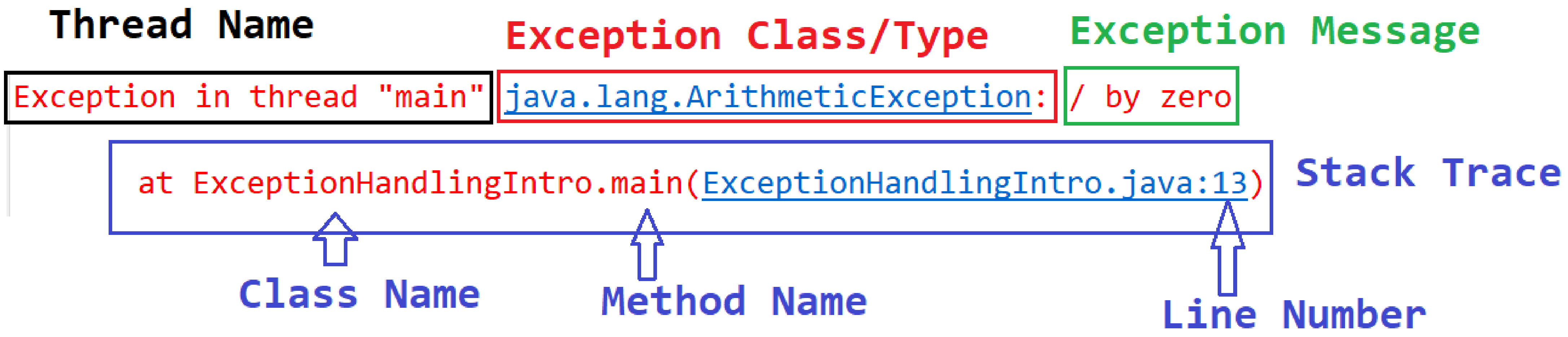
EXCEPTION HANDLING

JavaInBeats
https://javainbeats.com/

EXCEPTION AND EXCEPTION HANDLING

- **Exception:** is an unwanted or unexpected event, which occurs at the time of execution of the program. Due to this program may terminate abnormally or disrupts the normal flow of the program.
- **Exception Handling:** It is a process in which you can avoid the abnormal termination of the program by handling the exception. You can also provide an alternate execution, if the exception occurs in the normal flow of the program.
- Read and Understand the Exception

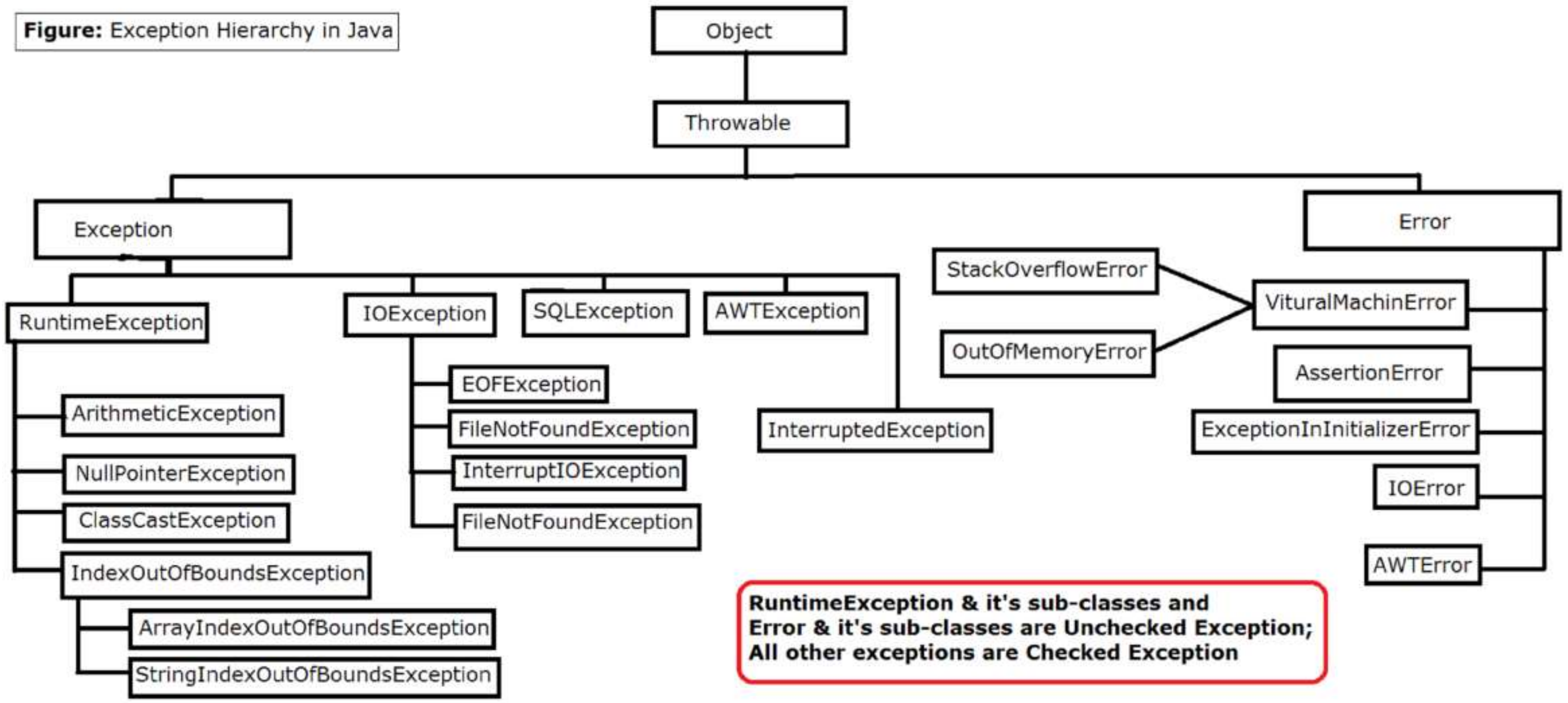


```
Exception in thread "main" java.lang.ArithmeticException: / by zero
	at ExceptionHandlingIntro.main(ExceptionHandlingIntro.java:13)
```

EXCEPTION, AND ERROR

- Throwable is a super/parent class of Exception and Error.
- Exception is categories into 2 types.
  - **Checked Exception**
    - All the Sub classes of java.lang.Exception class are checked exception, excepts java.lang.RuntimeException and its sub classes.
    - Checked Exception has to handle else java compiler will not allows you to compile program.
    - Example, FileNotFoundException, SqlException, InterruptedException etc.
  - **Un-Checked Exception**
    - All the sub classes of java.lang.RuntimeException are unchecked exceptions.
    - Java Compiler will not make mandatory to handle un-checked exception, but if any un-checked exception occurs in the program then program will terminates abnormally.
    - Example, NullPointerException, IndexOutofBoundException, ClassCastException etc.
- An Error is a condition that can't be control by your code.
- An Error is a subclass of java.lang.Error.
- An Error can be caught by exception handler, but it shouldn't be.
- Examples of Errors are, StackOverflowError, NoClassDefFoundError etc.

EXCEPTION AND ERROR



RuntimeException & it's sub-classes and Error & it's sub-classes are Unchecked Exception; All other exceptions are Checked Exception

KEYWORD IN THE EXCEPTION HANDLING

- **try :** try is a block in which we can write a statements which may throws an exception.
- **catch :** catch block is use to catch the exception thrown by try block and also it is use to provide an alternative way after getting exception.
- **finally :** this block gets executed always irrespective of try and catch execution. So the statement which you want to execute always can be write into this block.
- **throw :** you can throw an exception manually by using throw keyword.
- **throws :** you can declare exception at method declaration level by using throws keyword.

USING TRY, CATCH AND FINALLY

- In Java try is a block which contains a statement(s) which may throws an exception.
- Try block can be use with catch, finally or with both the blocks.
- Syntax for try-catch

```
try {
    //statement(s) may exception
} catch(ExceptionClass reference) {
    //statement(s)
}
```

- Syntax for try-finally

```
try {
    //statement(s) may exception
} finally{
    //statement(s)
}
```

- Syntax for try-catch-finally

```
try {
    //statement(s) may exception
} catch(ExceptionClass reference) {
    //statement(s)
} finally{
    //statement(s)
}
```

USING THROW AND THROWS

- throw keyword in Exception handling
  - Throw is a keyword use in exception handling.
  - It is use to throw object of exception manually.
  - It must be use at a statement level inside method only.
  - The thrown exception has to handle by try-catch or declare to be thrown.

- throws keyword in Exception handling
  - Throws keyword is use to declare exception at method declaration level.
  - If exceptions declare at method level then calling method has to handle exception or declare to be thrown.