Exception

Difference between error and exception

Errors indicate that something severe enough has gone wrong, the application should crash rather than try to handle the error.
Exceptions are events that occurs in the code. A programmer can handle such conditions and take necessary corrective actions. Few examples:

- **NullPointerException** – When you try to use a reference that points to null.
- **ArithmeticException** – When bad data is provided by user, for example, when you try to divide a number by zero this exception occurs because dividing a number by zero is undefined.
- **ArrayIndexOutOfBoundsException** – When you try to access the elements of an array out of its bounds, for example array size is 5 and you are trying to access the 10th element.
- **IOException**- Catches errors associated with input and output

Types of exceptions

There are two types of exceptions in Java:

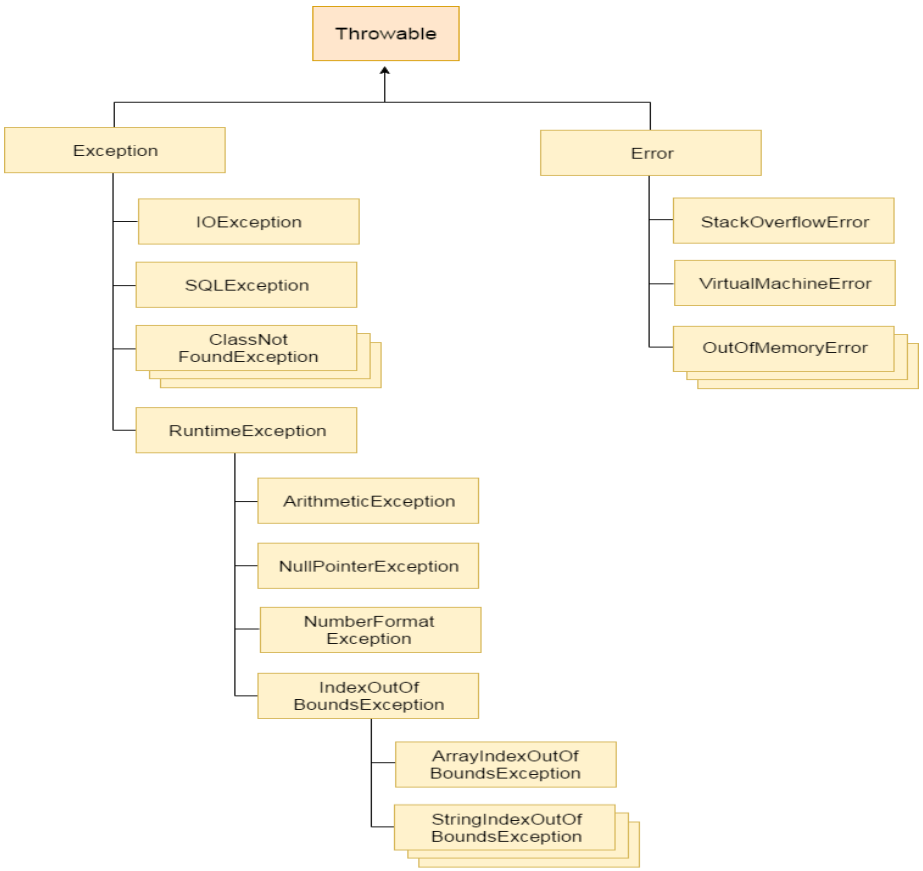
- **Checked exceptions**
  All exceptions other than Runtime Exceptions are known as Checked exceptions as the compiler checks them during compilation to see whether the programmer has handled them or not. If these exceptions are not handled/declared in the program, you will get compilation error. For example, SQLException, IOException, ClassNotFoundException etc.
- **Unchecked exceptions**
  Runtime Exceptions are also known as Unchecked Exceptions. These exceptions are not checked at compile-time so compiler does not check whether the programmer has handled them or not but it's the responsibility of the programmer to handle these exceptions and provide a safe exit. For example, ArithmeticException, NullPointerException, ArrayIndexOutOfBoundsException etc.

Compiler will never force you to catch such exception or force you to declare it in the method using throws keyword.

Java Exception Keywords

There are 5 keywords which are used in handling exceptions in Java.

Keyword	Description
try	The "try" keyword is used to specify a block where we should place exception code. The try block must be followed by either catch or finally. It means, we can't use try block alone.
catch	The "catch" block is used to handle the exception. It must be preceded by try block which means we can't use catch block alone. It can be followed by finally block later.
finally	The "finally" block is used to execute the important code of the program. It is executed whether an exception is handled or not.
throw	The "throw" keyword is used to throw an exception.
throws	The "throws" keyword is used to declare exceptions. It doesn't throw an exception. It specifies that there may occur an exception in the method. It is always used with method signature.



Example:

```
public class JavaExceptionExample{
    public static void main(String args[]){
        int  n ;
        int  data;

        n = KB.nextInt();
        if (n != 0)
            data =100/n;
        else{
            print("Division by 0, data is set to 0."
            data = 0;
        }
        //rest code of the program
        System.out.println(data);
    }
}
```

```
public class JavaExceptionExample{
    public static void main(String args[]){
        try{
            //code that may raise exception
            int data=100/10;
        }catch(ArithmeticException      e    ){
            System.out.println(e);
```

```
            System.out.println(“Division by 0, check denominator. “);
        }finally{     //Optional
            System.out.println(“Code associated with try is completed.\n:”);
        }

        //rest code of the program
        System.out.println("rest of the code...");
    }
}
```

Arrays

list of items sequenced one after one

objects

```
int [ ]    intList ;                    AB453A → NULL
intList =  new int [4];                 AB453A → BBB110 →0)    1)    2)    3)
int        num = 7;                     AB4532 →  7
float  [] floatList = new float [10];
String []  names;
int size ;
print(“Enter size of the array: “);
size = KB.nextInt();
names = new String[size] ;
```