Java Exception

bsse1004

February 2019

1 Example of Checked Exception

- IOException : It is thrown when an input-output operation failed or interrupted .IOExceptions are thrown when there is any input / output file operation issues while application performing certain tasks accessing the files. IOException is a checked exception.

```
    static ArrayList<BufferedImage> getFrames(File gif) throws
         IOException {
    ArrayList<BufferedImage> frames = new
         ArrayList<BufferedImage>();
    ImageReader ir = new GIFImageReader(new
         GIFImageReaderSpi());
    ir.setInput(ImageIO.createImageInputStream(gif));
    for (int i = 0; i < ir.getNumImages(true); i++) {
        frames.add(ir.read(i));
    }
    // Release resources for Garbage Collection
    ir.dispose();
    return frames;
}
```

- FileNotFoundException : This Exception is raised when a file is not accessible or does not open.
  - example :

```
   //Java program to demonstrate FileNotFoundException
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
 class File_notFound_Demo {

    public static void main(String args[]) {
        try {
            // Following file does not exist
            File file = new File("E://file.txt");

            FileReader fr = new FileReader(file);
        } catch (FileNotFoundException e) {
           System.out.println("File does not exist");
        }
    }
}
```

- NoSuchFieldException : It is thrown when a class does not contain the field (or variable) specified.
- ClassNotFoundException :This Exception is raised when we try to access a class whose definition is not found
- NoSearchMethodException : It is thrown when accessing a method which is not found.

2 Example of Unchecked Exception

- ArrayIndexOutOfBoundsException : It is thrown to indicate that an array has been accessed with an illegal index. The index is either negative or greater than or equal to the size of the array.
  - Example :

```
    // Java program to demonstrate
        ArrayIndexOutOfBoundException
class ArrayIndexOutOfBound_Demo
{
    public static void main(String args[])
    {
        try{
            int a[] = new int[5];
            a[6] = 9; // accessing 7th element in an array of
                    // size 5
        }
        catch(ArrayIndexOutOfBoundsException e){
            System.out.println ("Array Index is Out Of Bounds");
        }
    }
}
```

- Arithmetic Exception : It is thrown when an exceptional condition has occurred in an arithmetic operation.
  - Example

```
        // Java program to demonstrate ArithmeticException
class ArithmeticException_Demo
{
    public static void main(String args[])
    {
        try {
            int a = 30, b = 0;
            int c = a/b; // cannot divide by zero
            System.out.println ("Result = " + c);
        }
        catch(ArithmeticException e) {
            System.out.println ("Can't divide a number by 0");
        }
    }
}
```

- IllegalArgumentException : IllegalArgumentException is intended to be used anytime a method is called with any argument(s) that is improper, for whatever reason.
- StringIndexOutOfBoundsException : It is thrown by String class methods to indicate that an index is either negative than the size of the string
  - Example:

```


// Java program to demonstrate StringIndexOutOfBoundsException
class StringIndexOutOfBound_Demo
{
    public static void main(String args[])
    {
        try {
            String a = "This is like chipping "; // length is 22
            char c = a.charAt(24); // accessing 25th element
            System.out.println(c);
        }
        catch(StringIndexOutOfBoundsException e) {
            System.out.println("StringIndexOutOfBoundsException");
        }
    }
}
```

- NullPointerException : This exception is raised when referring to the members of a null object. Null represents nothing
  - Example

```java
    //Java program to demonstrate NullPointerException
class NullPointer_Demo
{
    public static void main(String args[])
    {
        try {
            String a = null; //null value
            System.out.println(a.charAt(0));
        } catch(NullPointerException e) {
            System.out.println("NullPointerException..");
        }
    }
}
```

- NumberFormatException : This exception is raised when a method could not convert a string into a numeric format.
  - Example

```java
    // Java program to demonstrate NumberFormatException
class  NumberFormat_Demo
{
    public static void main(String args[])
    {
        try {
            // "akki" is not a number
            int num = Integer.parseInt ("akki") ;

            System.out.println(num);
        } catch(NumberFormatException e) {
            System.out.println("Number format exception");
        }
    }
}
```

- AssertionError : the AssertionError in Java is thrown when an assert statement fails (i.e. the result is false).
- ExceptionInInitializerError :ExceptionInInitializerError, which is thrown when an error occurs within the static initializer of a class or object. Since an ExceptionInInitializerError isn't ever the cause of a thrown error, catching such an exception provides an underlying causal exception that indicates what the actual source of the issue was.
- StackOverflowError : The StackOverflowError in Java occurs when the application performs excessively deep recursion. However, what exactly qualifies as "excessively deep" depends on many factors.

- NoClassDefFoundError : In most cases, a java.lang.NoClassDefFoundError is thrown when a class reference was available during compile time, but has since disappeared (for whatever reason) during execution.