STACK TRACE ELEMENT

The **StackTraceElement class element represents a single stack frame which is a stack trace when an exception occurs**. Extracting stack trace from an exception could provide useful information such as class name, method name, file name, and the source-code line number. The getStackTrace() method of the Throwable class returns an array of StackTraceEle- ments.

stackTraceElement class constructor

StackTraceElement(String declaringClass, String methodName, String fileName, int lineNumber)

This creates a stack trace element representing the specified execution point.

Stack Trace Element class methods

Method	Description
boolean equals(Object obj)	Returns true if the invoking StackTraceElement is the same as the one passed in obj. Otherwise, it returns false.
String getClassName()	Returns the class name of the execution point
String getFileName( )	Returns the filename of the execution point
int getLineNumber( )	Returns the source-code line number of the execution point
String getMethodName( )	Returns the method name of the execution point
String toString( )	Returns the String equivalent of the invoking sequence

Example:

```
public class StackTraceEx{
    public static void main(String[] args) {
     try{
            throw new RuntimeException("go"); //raising an runtime exception
     }
     catch(Exception e){
            System.out.println("Printing stack trace:");
//create array of stack trace elements
final StackTraceElement[] stackTrace = e.getStackTrace();
 for (StackTraceElement s : stackTrace) {
     System.out.println("\tat " + s.getClassName() + "." + s.getMethodName()
            + "(" + s.getFileName() + ":" + s.getLineNumber() + ")");
             }
    }
    }
}
```

Sample Output:

Printing stack trace:at StackTraceEx.main(StackTraceEx.java:5)