

One solution to this problem is to save a list of all exceptions generated. Then throw an exception that holds a reference to this list. In this way, the receiver of the new exception has information about all exceptions and critical error information is not lost.

The previous code is modified as follows to contain a new class to contain all of the exceptions that are generated from the `readFile` method. The `readFile` method adds all generated exceptions to a `Vector` and stores a reference to this `Vector` in a `ReadFileExceptions` object, which is thrown to the calling method. The modified code looks like this:

```
import java.io.*;
import java.util.Vector;

class ReadFileExceptions extends IOException
{
    private Vector excVector;
    public ReadFileExceptions(Vector v)
    {
        excVector = v;
    }
    public Vector exceptionVector()
    {
        return excVector;
    }
    //...
}

class NotHidden
{
    public static void main(String args[])
    {
        NotHidden nh = new NotHidden();
        try {
            nh.readFile();
        }
        catch (ReadFileExceptions rfe) {
            //...
        }
    }

    public void readFile() throws ReadFileExceptions
    {
        BufferedReader br1 = null;
        BufferedReader br2 = null;
        FileReader fr = null;
        Vector excVec = new Vector(2); //Vector to store exceptions
    }
}
```