```java
 *  Licensed to the Apache Software Foundation (ASF) under one or more
package com.google;

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.util.AbstractList;
import java.util.Arrays;
import java.util.Collections;
import java.util.List;

/**
 * Possible Errors.
 *
 * @author  lihg
 * @version 2013-6-11
 */
public class PossibleError {
    private List<String> names;

    /**
     * List is stored <font color="red"><b>without copying</b></font>.
     *
     * @param names
     */
    public void setNames(List<String> names) {
        this.names = names;
    }

    /**
     * Avoid <font color="red"><b>null</b></font> return values.
     *
     * @param l
     * @return
     */
    public List<String> nullReturn(List<String> l) {
        if (l.isEmpty()) {
            // avoid
//          return null;
            return Collections.emptyList();
        }
        return l;
    }

    /**
     * <pre>
     * 1, Check Type in <font color="red"><b>equals</b></font>.
     * 2, Overloaded equals.
     * </pre>
     */
    @Override
    public boolean equals(Object obj) {
        // 性能优化：比较操作有可能很昂贵
        if (this == obj)
            return true;

        if (! (obj instanceof PossibleError))
            return false;

        PossibleError other = (PossibleError) obj;
        return names == null ? other.names == null : names.equals(other.names);
    }

    /**
     * {@link AbstractList#hashCode()}已实现生成{@link List}的hash值功能.
     *
     * 参考{@link Arrays#hashCode(Object[])}实现.
     */
    @Override
    public int hashCode() {
        if (names == null)
            return 0;

        return names.hashCode();
    }

    /**
     * <pre>
     * 1, <font color="red"><b>Close</b></font> in <b>finally</b>.
     * 2, Throw in <b>finally</b>.
     * </pre>
     *
     * @param fileName
     * @return
     * @throws IOException
     */
    public static String bufferedRead(String fileName) throws IOException {
        BufferedReader in = null;

        try {
            in = new BufferedReader(new FileReader(fileName));

            StringBuilder content = new StringBuilder(1024); // 2KB
            String l = null;
            while ((l = in.readLine()) != null) {
                content.append(l);
            }
            return content.toString();
        } finally {
            if (in != null) {
                in.close();
            }
            // don't throw exception in finally!
        }
    }

    /**
     * <pre>
     * 1, <font color="red"><b>Empty</b></font> <b>catch</b> clause.
     * 2, Handle numeric parsing errors.
     * </pre>
     *
     * @param s
     * @return
     */
    public static long parseLong(String s) {
        try {
            long l = Long.parseLong(s, 10);
            return l;
        } catch (NumberFormatException nfe) {
            // empty catch
            return -1;
        }
    }

    private String content;

    /**
     * <pre>
     * 1, <b>Field</b> might have <font color="red"><b>null</b></font> value.
     * 2, Use <font color="red"><b>==</b></font> to compare with <b>null</b>.
     * </pre>
     *
     * @return
     */
    public String getContent() {
        if (content == null)
            content = "";

        return content;
    }

    /**
     * <font color="red"><b>Improper conversion</b></font> of <b>Array to String</b>.
     *
     * @param ids
     * @return
     */
    public static String arrayToString(long[] ids) {
//      return ids.toString();
        return Arrays.toString(ids);
    }
}

```