



扫雷英雄榜——Java常见错误及解决方案

1. 类定义未找到错误 `java.lang.NoClassDefFoundError`。

▶▶ 原因分析

- (1) 由于该程序调用到的 `JAVA` 类文件没有正确上传。
- (2) `JAVA` 类文件已经上传，但应用服务器并未检测到，建议将 `JSP` 页面重新更新。

▶▶ 解决方案

- (1) 将 `JAVA` 类文件重新上传。
- (2) 将 `JSP` 页面更新并上传；或将应用程序服务器进行重新启动。

2. 解决 `java.lang.StackOverflowError` 错误。

▶▶ 错误信息

严重:Exceptioninitializingpagecontext
`java.lang.StackOverflowError`.....

▶▶ 原因分析

运行一个程序,JVM 会开辟一块内存空间去储存程序进行时的某些信息,当程序运行时需要储存的信息超过了分配的空间,就会出现那样的问题.比如死循环,递归次数过多,等等。

▶▶ 解决方案

首先从程序代码优化方面着手，检查是否有死循环、递归等程序，如果有，修正、优化相关代码。

3. 代码错误分析。

▶▶ 错误信息

```
interface Playable {  
    void play();  
}  
interface Bounceable {  
    void play();
```





```
}  
interface Rollable extends Playable, Bounceable {  
    Ball ball = new Ball("PingPang");  
}  
class Ball implements Rollable {  
    private String name;  
    public String getName() {  
        return name;  
    }  
    public Ball(String name) {  
        this.name = name;  
    }  
    public void play() {  
        ball = new Ball("Football");  
        System.out.println(ball.getName());  
    }  
}
```

» 解决方案

"interface Rollable extends Playable, Bounceable"没有问题。interface 可继承多个 interfaces, 所以这里没错。问题出在 interface Rollable 里的 "Ball ball = new Ball("PingPang");"。任何在 interface 里声明的 interface variable (接口变量, 也可称成员变量), 默认为 public static final。即 "Ball ball = new Ball("PingPang");" 实际上是 "public static final Ball ball = new Ball("PingPang");"。在 Ball 类的 Play()方法中, "ball = new Ball("Football");"改变了 ball 的 reference, 而这里的 ball 来自 Rollable interface, Rollable interface 里的 ball 是 public static final 的, final 的 object 是不能被改变 reference 的。因此编译器将在 "ball = new Ball("Football");" 这里显示有错。

4. 解决 MyEclipse 开发 SSH 整合时 java.lang.NoSuchMethodError 方案

» 环境显示

MyEclipse5.5
Struts1.2
Spring2.0
Hibernate3.1
Tomcat



» 错误提示

```
406ERROR[org.hibernate.proxy.BasicLazyInitializer]-CGLIBEnhancementfaile
d:dao.User
    java.lang.NoSuchMethodError:org.objectweb.asm.ClassVisitor.visit(ILjava/lan
g/String;Ljava/lang/String;[Ljava/lang/String;Ljava/lang/String;)V
    atnet.sf.cglib.core.ClassEmitter.begin_class(ClassEmitter.java:77)
```

» 原因分析

Spring 和 Hibernate 共用的一些 jar 文件发生了版本冲突。

» 解决方案

从出错的字面看,肯定与 org.objectweb.asm 包有关,检查 lib 下面关于 asm 的包,发现包含下面的包:

```
asm-2.2.3.jar
asm.jar
asm-attrs.jar
asm-commons-2.2.3.jar
asm-util-2.2.3.jar
```

可以从包名猜,asm-2.2.3.jar 和 asm.jar 可能是不同版本的,你可以查看包里的类是否差不多。这时,你可以尝试删除其中一个,重启 Tomcat,经过尝试,发现把 asm-2.2.3.jar 包删除后,重启后没出现错误提示。

5. Java 错误 java.lang.Error。

» 原因分析

(1) 对系统所访问外部资源,未执行关闭操作,导致外部资源大量浪费,最终可能导致系统无法正常运行;

(2) 对系统所访问的外部资源关闭次数太多,外部系统无法正常处理;

(3) 系统访问的外部资源出现异常情况。

» 解决方案

(1) 访问外部资源前,首先检查该资源(如数据库)是否可正常连接或操作。

(2) 访问外部资源时,如果进行了连接,一定进行关闭操作,并仅进行一次关闭操作。

(3) 尽量在同一操作中共享外部资源,以减少该操作对资源的消费,提高程序的执行效率。

6. Java 调用 DLL 错误: Exceptioninthread"main"java.lang.UnsatisfiedLinkError 解决。

▶▶ 原因分析

Exceptioninthread"main"java.lang.UnsatisfiedLinkError, 编译没有问题, 运行出现问题, 如果错误 Exceptioninthread"main"java.lang.UnsatisfiedLinkError 后面是某一个类名的话, 很可能是 java 程序调用了有关 dll 文件而没有找到这样的 dll 文件造成的。

▶▶ 解决方案

把需要的 dll 文件放到和该 java 类相同的目录下即可。

7. java.lang.OutOfMemoryError 异常处理总结。

▶▶ 原因分析

- (1) 内存中加载的数据量过于庞大, 如一次从数据库取出过多数据。
- (2) 集合类中有对对象的引用, 使用完后未清空, 使得 JVM 不能回收。
- (3) 代码中存在死循环或循环产生过多重复的对象实体。
- (4) 使用的第三方软件中的 BUG。
- (5) 启动参数内存值设定的过小。

8. java.lang.NoSuchMethodError 异常处理。

▶▶ 原因分析

Java 程序运行过程中, 需要调用某个类的某个方法, 但却发现这个方法并不存在, 找不到所以会提示这个错误。

▶▶ 解决方案

- (1) 检查调用的方法名称是否正确, 要检查拼写和大小写。
- (2) 检查调用的类里是否已经有该方法。
- (3) 检查调用的方法的作用域。
- (4) 若是调用别人的包里的方法, 首先要确认包是可以找到的 (如开发工具中导入包、web 应用中有该包等), 再按上面 1、2、3 点检查。
- (5) 某些时候, 特别是 SSH 整合开发, 有可能会出现这个异常, 一般是包的版本冲突问题, 这时候可以尝试更换一下版本。

9. 类路径没有找到。

▶▶ 原因分析

可能: `ClassNotFoundException: oracle.jdbc.driver.OracleDriver`, 一般是指包名写错, 或者没有 `import` 包, 或者没有在类路径中找到 `jar` 文件。

» 解决方案

没有加载 Oracle 驱动 `jar`, 在 `.bash_profile` 中把 `ojdbc14.jar` 加进来即可。

10. 找出以下程序代码中的错误。

```
class Something {  
    final int i;  
    public void doSomething() {  
        System.out.println("i = " + i);  
    }  
}
```

» 解决方案

`final int i` 是个 `final` 的 `instant variable` (实例变量, 或叫成员变量)。`final` 的 `instant variable` 没有 `default value`, 必须在 `constructor` (构造器) 结束之前被赋予一个明确的值。可以修改为 `"final int i = 0;"` 或将 `final` 去掉也可以, `int i` 属于 `instant variable` (实例变量, 或叫成员变量)。`instant variable` 有默认值, 也就是初始化值会为 0。

11. 如果 JSP 页面跳转时出现下边的错误信息:

```
The request sent by the client was syntactically incorrect (Invalid path  
/web/shoppingCart was requested).
```

» 原因分析

`struts-config` 中的 `action = "x"` 写成了 `action = "x.do"`。

» 解决方案

将 `struts-config` 中的 `action = "x.do"` 改成 `action = "x"` 即可。

12. 常见错误提示。

```
tomcat:java.lang.OutOfMemoryError:PermGenspace  
tomcat:java.lang.OutOfMemoryError:Javaheap space  
weblogic:RootcauseofServletExceptionjava.lang.OutOfMemoryError  
resin:java.lang.OutOfMemoryError  
java:java.lang.OutOfMemoryError
```

» 原因分析

- (1) 应用服务器提示错误。
- (2) Java 代码导致错误。

» 解决方案

- (1) 把启动参数内存值设置足够大。
- (2) 重点排查以下几点：
 - 1) 检查代码中是否有死循环或递归调用。
 - 2) 检查是否有大循环重复产生新对象实体。
 - 3) 检查对数据库查询中，是否有一次获得全部数据的查询。一般来说，如果一次取十万条记录到内存，就可能引起内存溢出。这个问题比较隐蔽，在上线前，数据库中数据较少，不容易出问题，上线后，数据库中数据多了，一次查询就有可能引起内存溢出。因此对于数据库查询尽量采用分页的方式查询。
 - 4) 检查 List、MAP 等集合对象是否有使用完后，未清除的问题。List、MAP 等集合对象会始终存有对对象的引用，使得这些对象不能被 GC 回收。

13. 错误代码分析

```
public class Something {  
    public int addOne(final int x) {  
        return ++x;  
    }  
}
```

» 原因分析

int x 被修饰成 final，致使 x 不能在 addOne method 中被修改。

14. java 运行时出现错误。

```
Exception in thread "main" java.lang.NoClassDefFoundError: MyJava\Hello(wrong  
name:Hello)  
    at java.lang.ClassLoader.defineClass1(Native Method)
```

» 解决方案

- (1) 检查类名拼写是否正确。
- (2) 类名正确，确认 jdk 下的 tools.jar 有没有导进去。
- (3) jdk 下的 tools.jar 有没有导进去了，确认导入的 jdk 是不是有重复。

15. 在部署 Struts 时，出现如下错误信息：

```
HTTP Status 404 - Servlet action is not available

type Status report
message Servlet action is not available
description The requested resource (Servlet action is not available) is not
available.
```

▶▶ 原因分析

- (1) web.xml 文件中未配置 ActionServlet。
- (2) struts-config.xml 文件未配置你要访问的 Action。
- (3) 你的 jsp 文件 form 标记中 action 属性的路径名称错误。
- (4) 非以上三种情况。

▶▶ 解决方案

- (1) 在 web.xml 文件中加上 ActionServlet 的配置信息

```
?????? /WEB-INF/struts-config.xml?
```

- (2) 在 struts-config.xml 文件检查你要访问的 Action 配置文件。
- (3) 检查 jsp 文件 form 标记中 action 属性的路径名称是否与 struts-config.xml 文件中 action 标记的 path 属性的路径名称一致。
- (4) 非以上情况的解决办法就是检查 web 容器的 log 日志，如果是 tomcat 则检查下 logs 目录下的 localhost_log 文件，看里边是否记录有错误信息，然后根据错误信息提示将其纠正。

16. 代码错误分析。

```
public class PerttyTest {
    public static void main(String[] args){
        String i="123",s;
        s=getStr(i);

        System.out.println(s);
    }

    public String getStr(String s){
        StringBuffer sb=new StringBuffer();
        for(int i=s.length()-1;i>=0;i--){
            sb.append(s.charAt(i));
        }
    }
}
```



```
    }  
  
    return sb.toString();  
}  
  
}
```

» 原因分析

定义的 getStr 方法不是静态方法,需要实例一个对象才能调用。

» 解决方案

在 main 下 加一行 PerttyTest a=new PerttyTest();
s=getStr(i) 改成 s=a.getStr(i);
正确的代码如下所示。

```
public class PerttyTest {  
    public static void main(String[] args){  
        PerttyTest a=new PerttyTest();  
        String i="123",s;  
        s=a.getStr(i);  
  
        System.out.println(s);  
    }  
  
    public String getStr(String s){  
        StringBuffer sb=new StringBuffer();  
        for(int i=s.length()-1;i>=0;i--){  
            sb.append(s.charAt(i));  
        }  
  
        return sb.toString();  
    }  
  
}
```

17. could not initialize a collection: [hibernate.entity.Role.modules#32768] Syntax error: Encountered "-" at line 1, column 132.

» 原因分析

<set name="modules" table="module-role" lazy="false">红色字部分中“-”为非法字符。

» 解决方案

将非法字符“-”替换为 module_role。

18. 错误代码分析。

```
public class Something {  
    void doSomething () {  
        private String s = "";  
        int l = s.length();  
    }  
}
```

» 解决方案

局部变量前不能放置任何访问修饰符 (private, public, 和 protected)。final 可以用来修饰局部变量 (final 如同 abstract 和 strictfp, 都是非访问修饰符, strictfp 只能修饰 class 和 method 而非 variable)。

19. MyEclipse 开发 SSH 整合时 java.lang.NoSuchMethodError 解决方案。

» 环境显示

MyEclipse5.5
Struts1.2
Spring2.0
Hibernate3.1
Tomcat

» 错误信息

```
406ERROR[org.hibernate.proxy.BasicLazyInitializer]-CGLIBEnhancementfaile  
d:dao.User  
    java.lang.NoSuchMethodError:org.objectweb.asm.ClassVisitor.visit(IILjava/lan  
g/String;Ljava/lang/String;Ljava/lang/String;Ljava/lang/String;)V  
    atnet.sf.cglib.core.ClassEmitter.begin_class(ClassEmitter.java:77)
```

» 原因分析

Spring 和 Hibernate 共用的一些 jar 文件发生了版本冲突。



» 解决方案

从出错的字面看，肯定与 org.objectweb.asm 包有关，检查 lib 下面关于 asm 的包，发现包含下面的包：

```
asm-2.2.3.jar
asm.jar
asm-attrs.jar
asm-commons-2.2.3.jar
asm-util-2.2.3.jar
```

可以从包名猜，asm-2.2.3.jar 和 asm.jar 可能是不同版本的，你可以查看包里的类是否差不多。这时，你可以尝试删除其中一个，重启 Tomcat，经过尝试，发现把 asm-2.2.3.jar 包删除后，重启后没出现错误提示。

20. 字符串越界错误 java.lang.StringIndexOutOfBoundsException 等字符串类相关错误

» 错误信息

```
Strings_all_power="1010011";
Strings_access_power=s_all_power.substring(3,4);
```

» 原因分析

以上代码段功能的功能是获取字符串 s_all_power 中的第 4 个字符。

一般情况下，程序不会有问题，如果由于某种原因，s_all_power 长度变短，程序就会抛出字符串错误。

» 解决方案

对字符串进行截取(substring,charAt)、转换为字节数组(getBytes)，字符数组转换为字符串(valueOf)操作时，先对操作字符串对象的存在性(是否为空)及长度进行检查后，再进行操作。

改写后的正确代码如下。

```
Strings_all_power="1010011";
if(s_all_power.length>4)
    Strings_access_power=s_all_power.substring(3,4);
```

21. 数据插入异常，GenericJDBCException: could not insert

» 原因分析

(1) 没有建立表或者表中没有任何数据。





- (2) 插入数据后没有执行提交语句：`commit`。

