# Shell 流程控制

## 重写fillInStackTrace

异常性能差是因为fillInStackTrace方法，该方法是带锁的并且需要填充线程异常栈信息。

而我们的**业务类**异常是不需要记录异常栈的，可以考虑覆写fillInStackTrace方法减小性能开支（据说覆写该方法能提高10倍性能）。

* 示例

public class NoSpiMatchException extends Exception {  
  
 public NoSpiMatchException() {  
 }  
  
 public NoSpiMatchException(String message) {  
 super(message);  
 }  
  
 public NoSpiMatchException(Throwable cause) {  
 super(cause);  
 }  
 //重写父类方法  
 **@Override  
 public synchronized Throwable fillInStackTrace() {  
 return this;  
 }**  
}

## 增加构造函数，用参数决定

* 示例

public class NoSpiMatchException extends Exception {  
  
 public NoSpiMatchException() {  
 }  
  
 public NoSpiMatchException(String message) {  
 super(message);  
 }  
  
 public NoSpiMatchException(Throwable cause) {  
 super(cause);  
 }

/\*\*  
 \* 业务异常类增加构造函数，用参数决定是否需要异常栈  
 \* @param message  
 \* @param cause  
 \* @param enableSuppression  
 \* @param writableStackTrace 决定需不需要执行fillInStackTrace  
 \*/  
 **public NoSpiMatchException(String message, Throwable cause, boolean enableSuppression, boolean writableStackTrace) {  
 super(message, cause, enableSuppression, writableStackTrace);  
 }**  
}

# 常用异常类

## java.lang.UnsupportedOperationException

不支持的操作,

比如说抽象类的的某些派生类实现里，或者接口的某个实现类里面，某个方法可能是无意义的或者未实现，调用该方法会抛出一个UnspportedOperationException异常

## java.lang IllegalArgumentException

非法参数

比如方法入参不是期望的,可以抛出异常

**if** (initialCapacity < 0)  
 **throw new** IllegalArgumentException(**"Illegal initial capacity: "** +initialCapacity);

## java.lang .IllegalStateException

无效状态

比如 连接池对象已经关闭,试图调用 驱逐 空闲时间超过阈值的 连接对象时，会抛出此异常

**final void** assertOpen() **throws** IllegalStateException {  
 **if** (isClosed()) {  
 **throw new** IllegalStateException(**"Pool not open"**);  
 }  
}

## java.lang.AbstractMethodError

抽象方法错误。当试图调用抽象方法时抛出

一般是 子类 没有覆盖 父类方法，父类是要求每个子类都要重写此方法，当子类没有此方法，向上一级调用父类的这个方法时，抛出此错误

如:

java.util.concurrent.TimeUnit(枚举)

//父类定义的方法

**public long** toMillis(**long** duration) {  
 **throw new** AbstractMethodError();  
}

//枚举元素 SECONDS中重写方法

java.util.concurrent.TimeUnit#SECONDS

**public long** toMillis(**long** d) { **return** *x*(d, ***C3***/***C2***, ***MAX***/(***C3***/***C2***)); }