

用java封装产品异常的代码 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/144/2021_2022__E7_94_A8java_E5_B0_81_c104_144873.htm

```
抽象类 继承于 Exception public abstract class AbstractException extends Exception { private ErrorInfo info. public AbstractException(ErrorInfo message) { super(message.getErrorCode() message.getErrorMessage(). info = message. } public String getCode() { return info.getErrorCode(). } } 错误实体 public class ErrorInfo { private String ErrorCode. private String ErrorName. private String ErrorInfo. public ErrorInfo(String temp1,String temp2,String temp3) { this.ErrorCode = temp1. this.ErrorName = temp2. this.ErrorInfo = temp3. } public String getErrorCode() { return this.ErrorCode. } public String getErrorName() { return this.ErrorName. } public String getErrorInfo() { return this.ErrorInfo. } } 错误集合 public class ErrorPool { private java.util.HashMap errorMap = new java.util.HashMap(). public ErrorPool() { errorMap.put("Center1001",new ErrorInfo("Center1001","严重错误，适配器无效","因为适配器所在前置机网络异常，造成适配器无效")). } public ErrorInfo getErrorInfo(Object errorCode) { return (ErrorInfo)errorMap.get(errorCode). } } 异常实现 public class TestException extends AbstractException { private ErrorInfo info. public TestException(ErrorInfo message) { super(message). info = message. }. public String getCode() { return super.getCode(). } public void LogDebug() { System.out.println("debug info....."). } }
```

具体使用代码

```
public class Test { public Test() { } public void
kk(String username) throws TestException { if
(username.equals("fuck")) { } else { throw(new TestException((new
ErrorPool()).getErrorInfo("Center1001"))). } } public static void
main(String[] agrgs) { try { Test tt = new Test(). tt.kk("xxx"). }
catch(TestException e) { System.out.println(e.getCode()).
e.printStackTrace(). e.LogDebug(). } } }
```

100Test 下载频道开通，
各类考试题目直接下载。详细请访问 www.100test.com