

关于junit实现过程详细介绍 PDF转换可能丢失图片或格式，
建议阅读原文

https://www.100test.com/kao_ti2020/475/2021_2022__E5_85_B3_E4_BA_8Ejuni_c67_475821.htm 测试分类：白箱测试、黑箱测试

、单元测试、集成测试、功能测试.....。白箱测试是指在知道被测试的软件如何（How）完成功能和完成什么样（What）的功能的条件下所作的测试，一般是由开发人员完成，单元测试是一种白箱测试，因为开发人员最了解自己编写的软件。

JUnit是由 Erich Gamma 和 Kent Beck 编写的一个回归测试框架，回归测试就是你不断地对所编写的代码进行测试（如单元测试）：编写一些，测试一些，调试一些，然后循环这一过程，你会不断地重复先前的测试，哪怕你正编写其他的类。

第一步：去JUnit主页（<http://www.junit.org>）下载最新版本3.8.1程序包junit-3.8.1.zip.解开压缩包到c:\junit（可自定义）。

第二步：假如目录是c:\junit那么，在classpath中加入：“c:\junit\;c:\junit\junit.jar:”定义类路径。在命令提示符下运行：java junit.swingui.TestRunner，如果一切正确，就会打开应用程序。在下拉菜单中寻找程序自带的例子，比如

junit.samples.AllTests，点击“Run”观察结果。第三步：实现自己的TEST计划，目前有一个叫MyBean的数据库操作类需要测试，如下：

```
package junit.samples.import java.sql.*.import java.io.*.public class MyBean{Statement stmt=null.ResultSet rs=null.Connection conn=null.String result=null.public String con(){ //初始化数据库 try{
```

```
Class.forName("org.gjt.mm.mysql.Driver").newInstance(). String url
```

```

="jdbc:mysql://192.168.0.88/weboa?user=root&password=".co
nn= DriverManager.getConnection(url).return "Connection
Success!".}catch(Exception e){System.out.println(e).return
"Connection Error!".}}public String gogo(String lmdm){ //查询数
据库try{ stmt=conn.createStatement(). String sql="0select * from
TB_LM where N_LMDM=" lmdm "".rs=stmt.executeQuery(sql). //
执行查询while
(rs.next()){result=rs.getString("N_SJID").}}catch(Exception
e){result=e.toString().}finally { //关闭JDBC资源 if(rs != null) try {
rs.close(). } catch(SQLException ex) { ex.printStackTrace().
}if(conn != null) try { conn.close(). } catch(SQLException ex) {
ex.printStackTrace(). }}return result.}} 接着，创建一个测试类
：TestMyBean，如下：package junit.samples.import
junit.samples.MyBean.import junit.framework.*.public class
TestMyBean extends TestCase { //TestCase的子类private MyBean
aName. //构造被测类的对象public TestMyBean(String name) {
super(name). }protected void setUp() { //进行初始化的任
务aName= new MyBean().}public static Test suite() { //进行测
试return new TestSuite(TestMyBean.class). }public void testCon() {
//对预期的值和con方法比
较Assert.assertTrue(!aName.equals(null)). //断
言Assert.assertEquals("Connection Success!",aName.con()).}public
void testGogo() { //对预期的值和gogo方法比
较aName.con().Assert.assertTrue(!aName.equals(null)). //断
言Assert.assertEquals("0",aName.gogo("1")).}}解释如下：首先要
引入待测试的类import junit.samples.MyBean；接着引入Junit框

```

架import junit.framework.* ;。 与一个Servlet类似，需要继承父类TestCase；在setUp（）方法中实例化一个MyBean，供后面的测试方法使用；suite（）是一个很特殊的静态方法，它会使用反射动态的创建一个包含所有的testXxxx方法的测试套件，确定有多少个测试可以执行；testCon（）方法对MyBean的Con方法进行测试，并断言（Assert）结果是"Connection Success！"，并在Assert.assertEquals（）方法中验证；testGogo（）方法和testCon（）方法类似。 第四步：把TestMyBean、MyBean类编译成*.class文件，在Junit的控制台上选择刚才定义的TestMyBean类，并运行。如果一切正确，就会显示绿条，证明测试正确。如果显示红色，在Results中会有相应显示，根据提示检查MyBean类中的错误。一般的，只要断言符合MyBean类的规范，TestMyBean类几乎不可能出错。 一些扩展：对于WEB应用程序，我们可以把Junit引入，只需适当配置环境。另外，可以把众多的测试类集成到一起，形成总测试类，并且只需要实现suite（）方法，例如：

```
public static Test suite（） {TestSuite suite= new TestSuite("All JUnit Tests").suite.addTest(VectorTest.suite()).suite.addTest(TestMyBean.suite()).return suite.}
```

 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