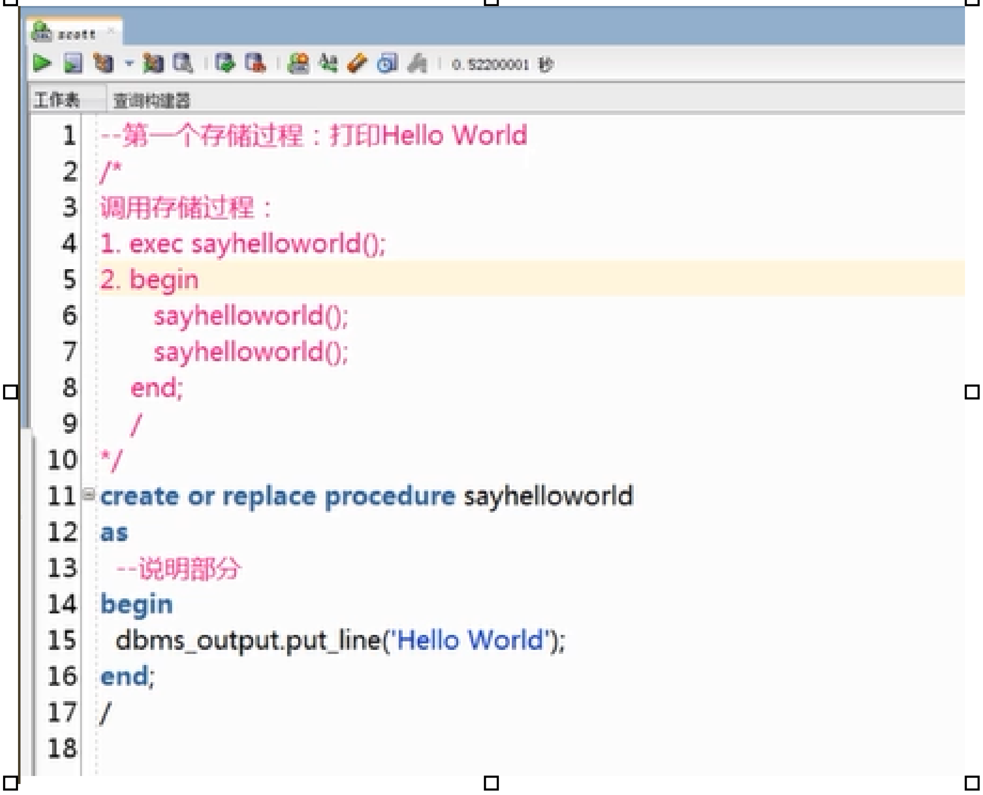
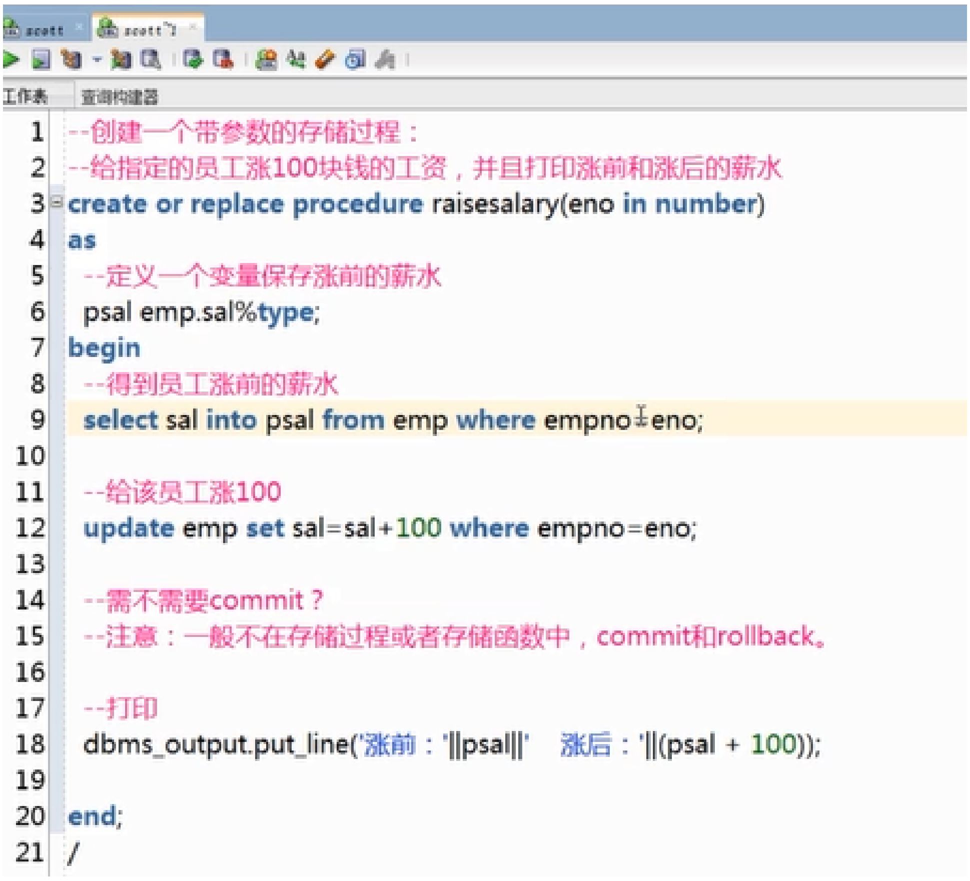
第一个存储过程



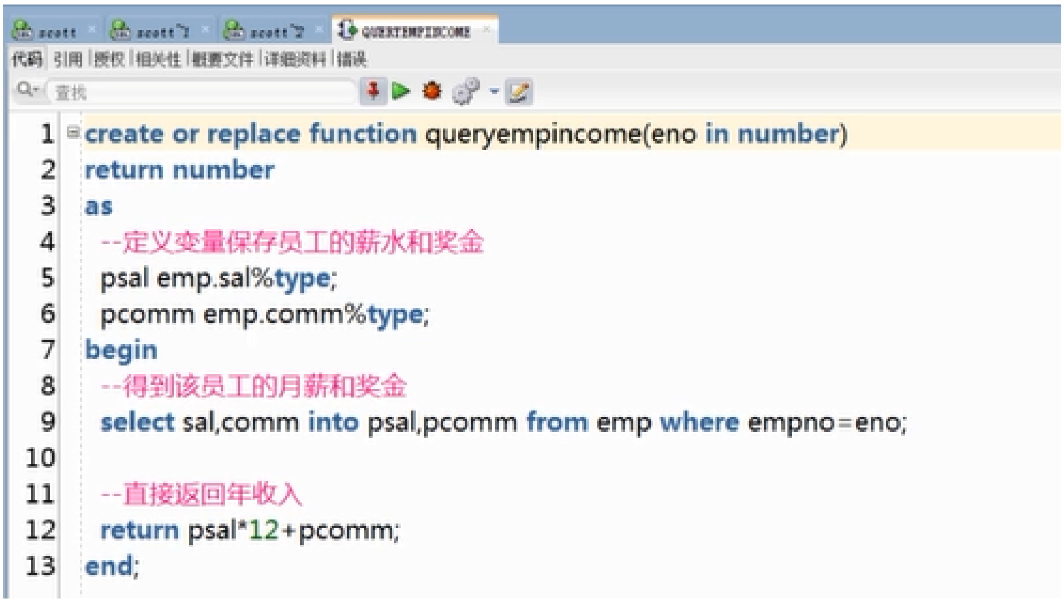
```
--第一个存储过程：打印Hello World
/*
调用存储过程：
1. exec sayhelloworld();
2. begin
       sayhelloworld();
       sayhelloworld();
   end;
   /
*/
create or replace procedure sayhelloworld
as
  --说明部分
begin
  dbms_output.put_line('Hello World');
end;
/
```

带参数的存储过程



```
--创建一个带参数的存储过程：
--给指定的员工涨100块钱的工资，并且打印涨前和涨后的薪水
create or replace procedure raisesalary(eno in number)
as
  --定义一个变量保存涨前的薪水
  psal emp.sal%type;
begin
  --得到员工涨前的薪水
  select sal into psal from emp where empno=eno;

  --给该员工涨100
  update emp set sal=sal+100 where empno=eno;

  --需不需要commit？
  --注意：一般不在存储过程或者存储函数中，commit和rollback。

  --打印
  dbms_output.put_line('涨前：'||psal||'   涨后：'||(psal + 100));

end;
/
```

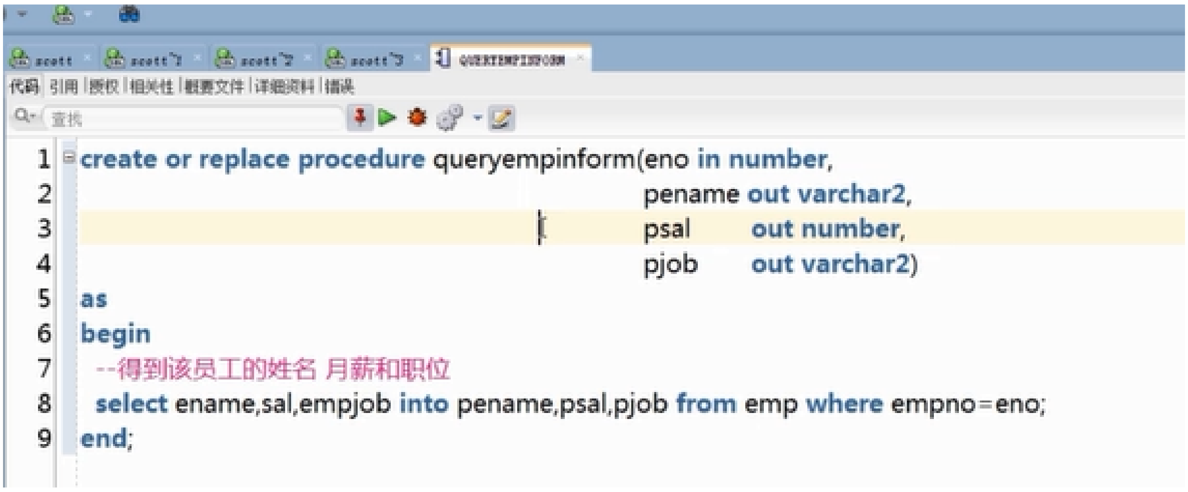
存储函数



```
create or replace function queryempincome(eno in number)
return number
as
  --定义变量保存员工的薪水和奖金
  psal emp.sal%type;
  pcomm emp.comm%type;
begin
  --得到该员工的月薪和奖金
  select sal,comm into psal,pcomm from emp where empno=eno;

  --直接返回年收入
  return psal*12+pcomm;
end;
```

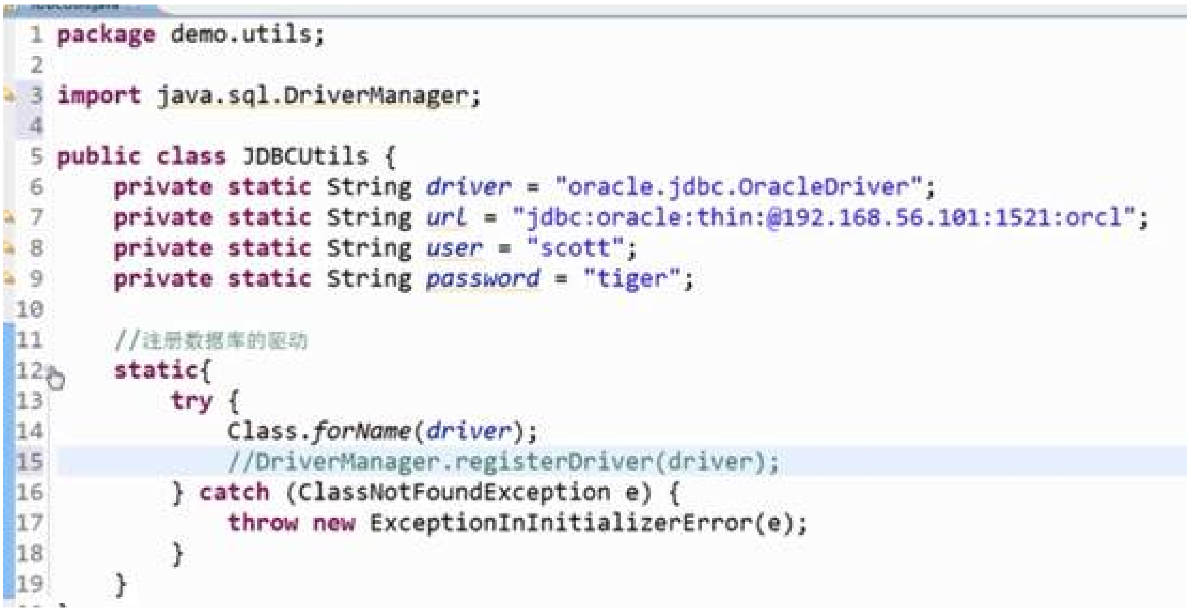
注意：return 带有null的表达式，结果为null。用nvl(param,0)函数，把null当做0处理。

输出参数

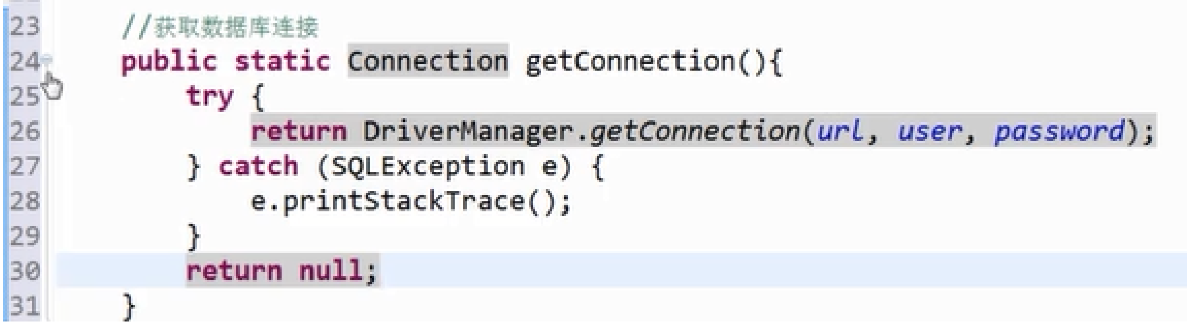


```
create or replace procedure queryempinform(eno in number,
                                           pename out varchar2,
                                           psal   out number,
                                           pjob   out varchar2)
as
begin
  --得到该员工的姓名 月薪和职位
  select ename,sal,empjob into pename,psal,pjob from emp where empno=eno;
end;
```

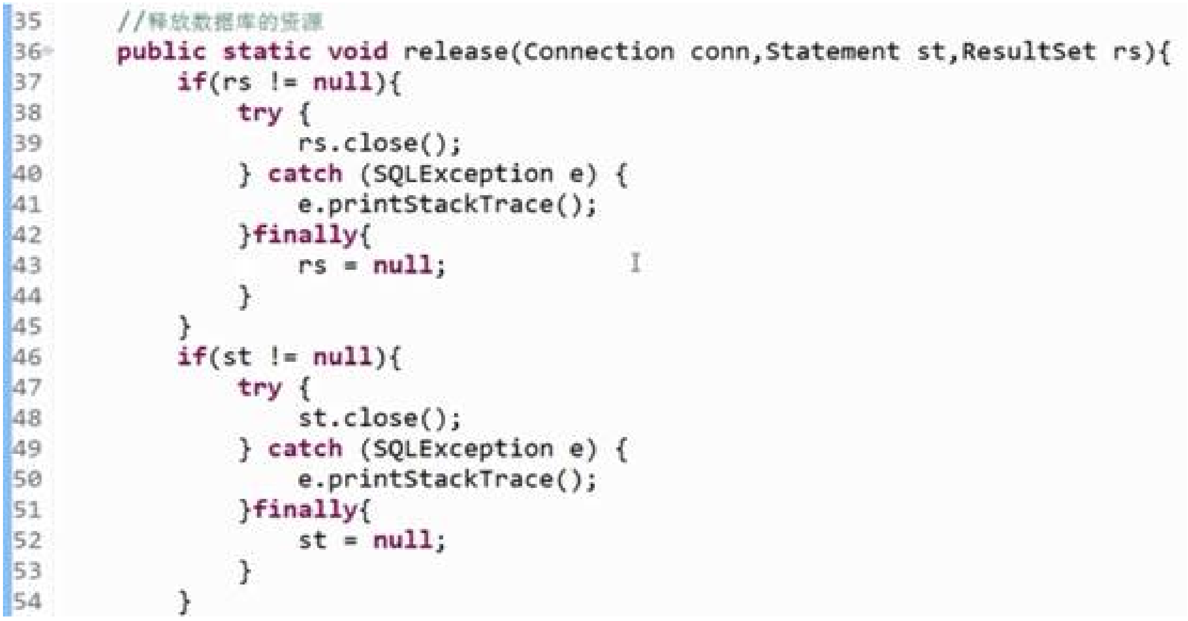
在java中访问存储过程



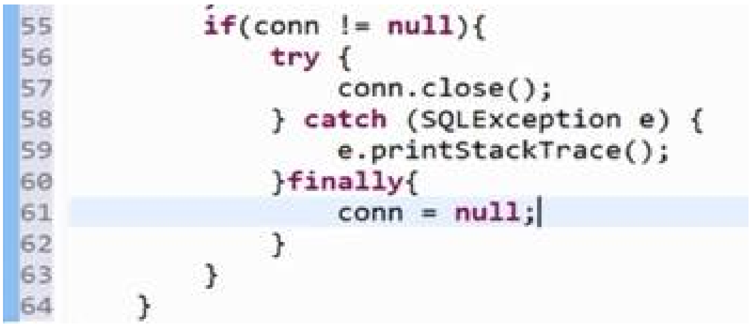
```java
package demo.utils;

import java.sql.DriverManager;

public class JDBCUtils {
    private static String driver = "oracle.jdbc.OracleDriver";
    private static String url = "jdbc:oracle:thin:@192.168.56.101:1521:orcl";
    private static String user = "scott";
    private static String password = "tiger";

    //注册数据库的驱动
    static{
        try {
            Class.forName(driver);
            //DriverManager.registerDriver(driver);
        } catch (ClassNotFoundException e) {
            throw new ExceptionInInitializerError(e);
        }
    }
```



```java
    //获取数据库连接
    public static Connection getConnection(){
        try {
            return DriverManager.getConnection(url, user, password);
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return null;
    }
```

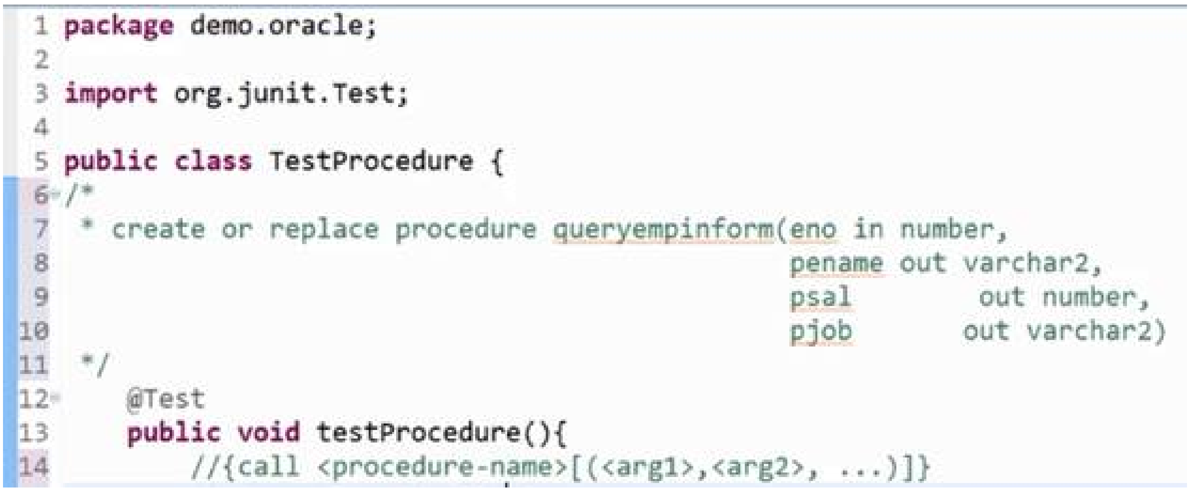


```java
    //释放数据库的资源
    public static void release(Connection conn,Statement st,ResultSet rs){
        if(rs != null){
            try {
                rs.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }finally{
                rs = null;
            }
        }
        if(st != null){
            try {
                st.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }finally{
                st = null;
            }
        }
```

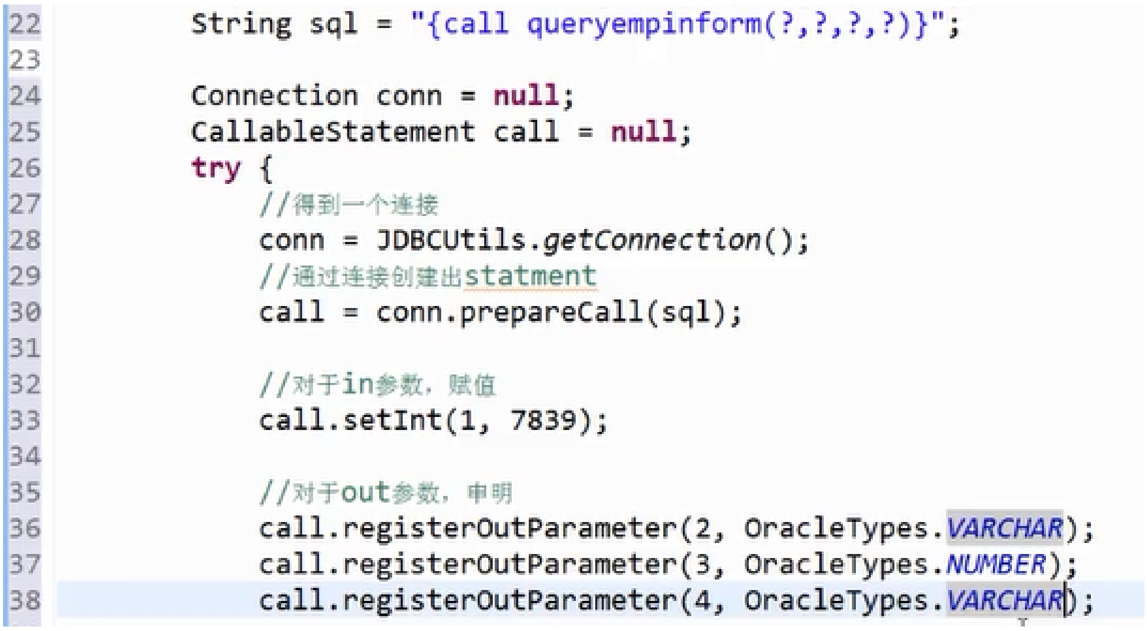


```java
        if(conn != null){
            try {
                conn.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }finally{
                conn = null;
            }
        }
    }
```

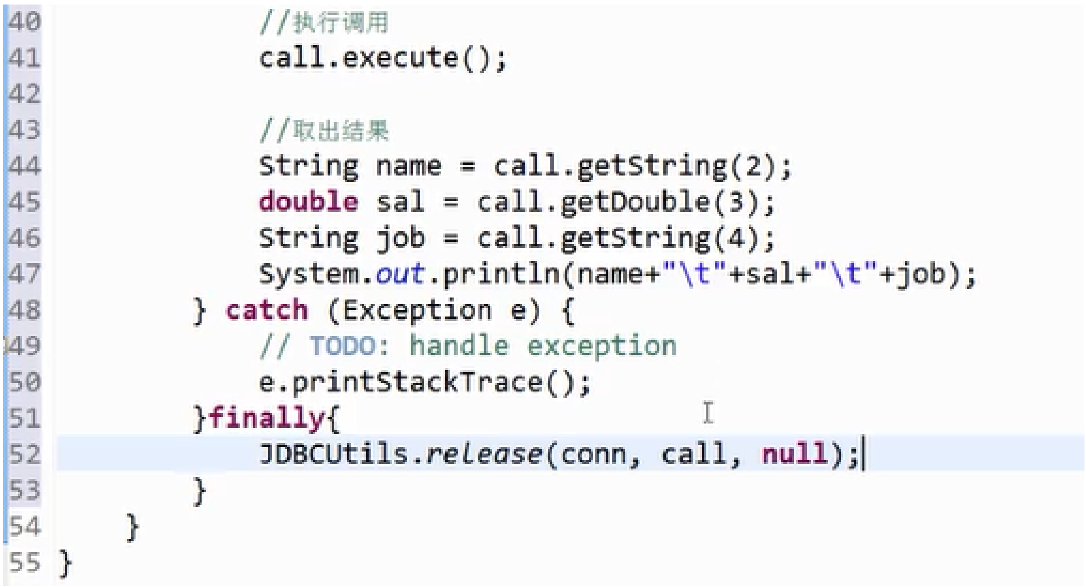
访问存储过程



```java
package demo.oracle;

import org.junit.Test;

public class TestProcedure {
/*
 * create or replace procedure queryempinform(eno in number,
                                               pename out varchar2,
                                               psal      out number,
                                               pjob      out varchar2)
 */
    @Test
    public void testProcedure(){
        //{call <procedure-name>[(<arg1>,<arg2>, ...)]}
```

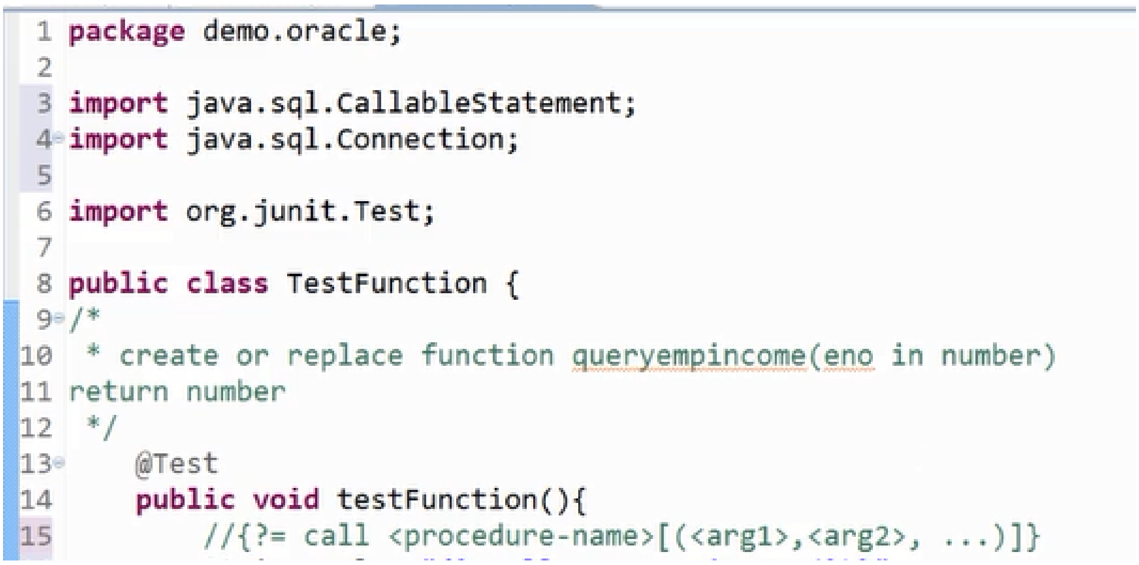


```java
        String sql = "{call queryempinform(?,?,?,?)}";

        Connection conn = null;
        CallableStatement call = null;
        try {
            //得到一个连接
            conn = JDBCUtils.getConnection();
            //通过连接创建出statment
            call = conn.prepareCall(sql);

            //对于in参数，赋值
            call.setInt(1, 7839);

            //对于out参数，申明
            call.registerOutParameter(2, OracleTypes.VARCHAR);
            call.registerOutParameter(3, OracleTypes.NUMBER);
            call.registerOutParameter(4, OracleTypes.VARCHAR);
```

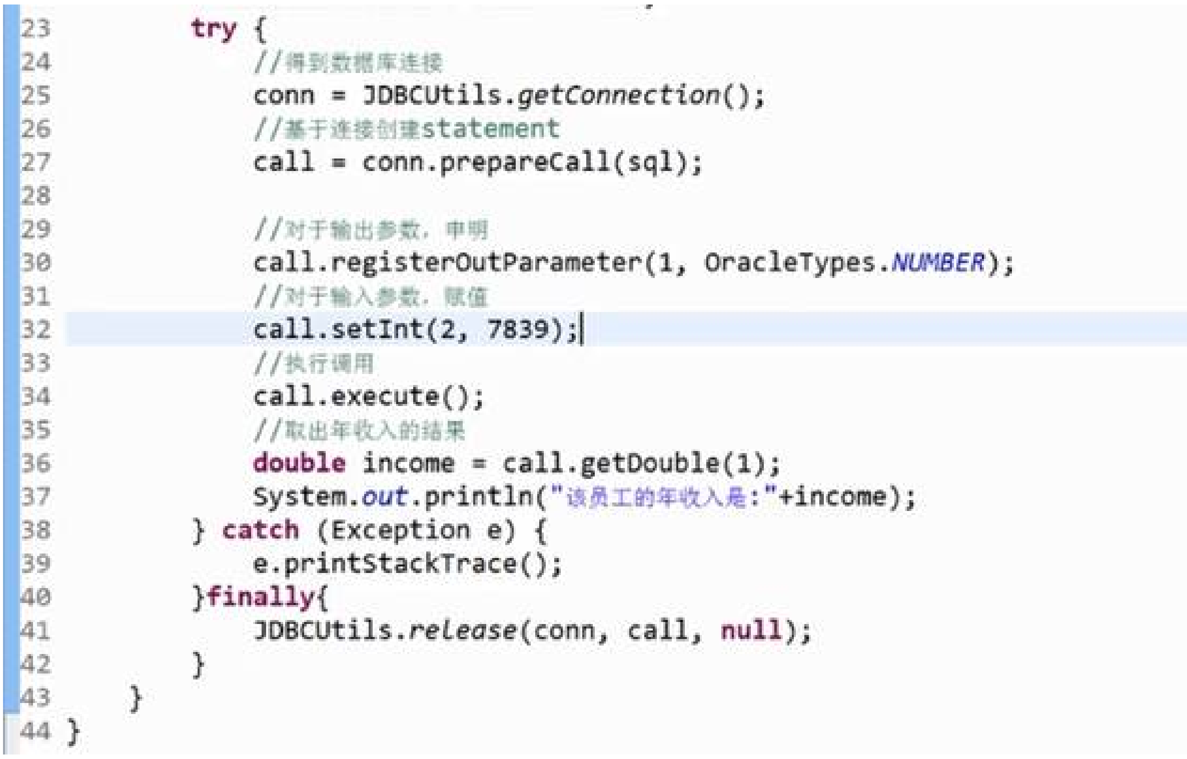


```java
            //执行调用
            call.execute();

            //取出结果
            String name = call.getString(2);
            double sal = call.getDouble(3);
            String job = call.getString(4);
            System.out.println(name+"\t"+sal+"\t"+job);
        } catch (Exception e) {
            // TODO: handle exception
            e.printStackTrace();
        }finally{
            JDBCUtils.release(conn, call, null);
        }
    }
}
```

访问存储函数

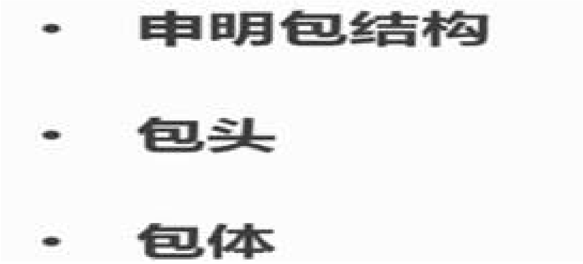


```
package demo.oracle;

import java.sql.CallableStatement;
import java.sql.Connection;

import org.junit.Test;

public class TestFunction {
/*
 * create or replace function queryempincome(eno in number)
return number
 */
    @Test
    public void testFunction(){
        //{?= call <procedure-name>[(<arg1>,<arg2>, ...)]}
```

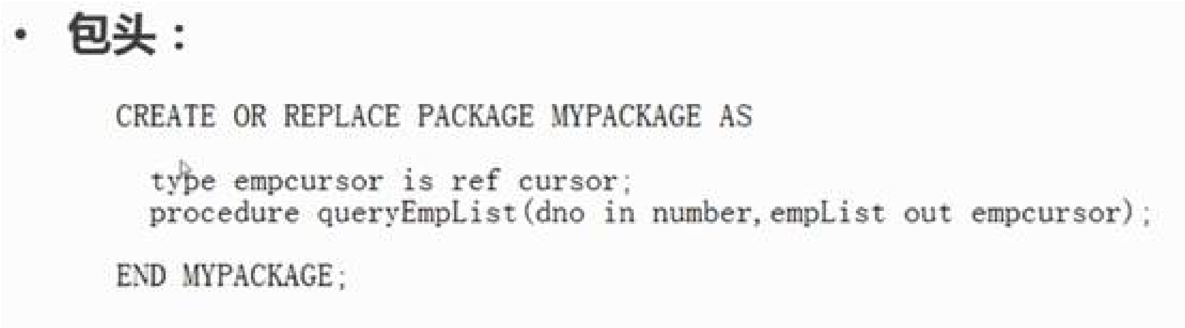


```
        try {
            //得到数据库连接
            conn = JDBCUtils.getConnection();
            //基于连接创建statement
            call = conn.prepareCall(sql);

            //对于输出参数，申明
            call.registerOutParameter(1, OracleTypes.NUMBER);
            //对于输入参数，赋值
            call.setInt(2, 7839);
            //执行调用
            call.execute();
            //取出年收入的结果
            double income = call.getDouble(1);
            System.out.println("该员工的年收入是:"+income);
        } catch (Exception e) {
            e.printStackTrace();
        }finally{
            JDBCUtils.release(conn, call, null);
        }
    }
}
```

在out参数中使用光标

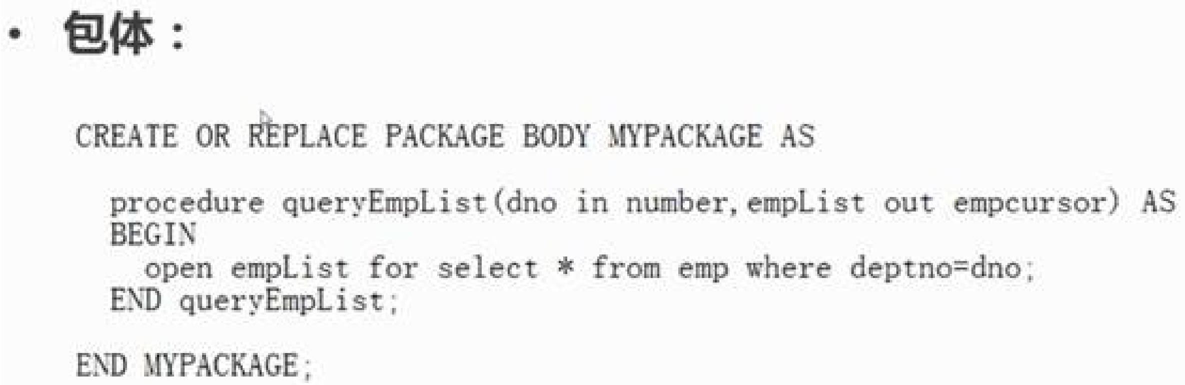


- 申明包结构
- 包头
- 包体



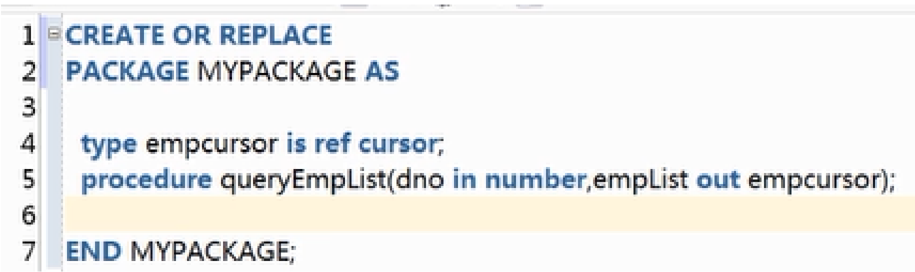
- 包头：

```
CREATE OR REPLACE PACKAGE MYPACKAGE AS

  type empcursor is ref cursor;
  procedure queryEmpList(dno in number,empList out empcursor);

END MYPACKAGE;
```

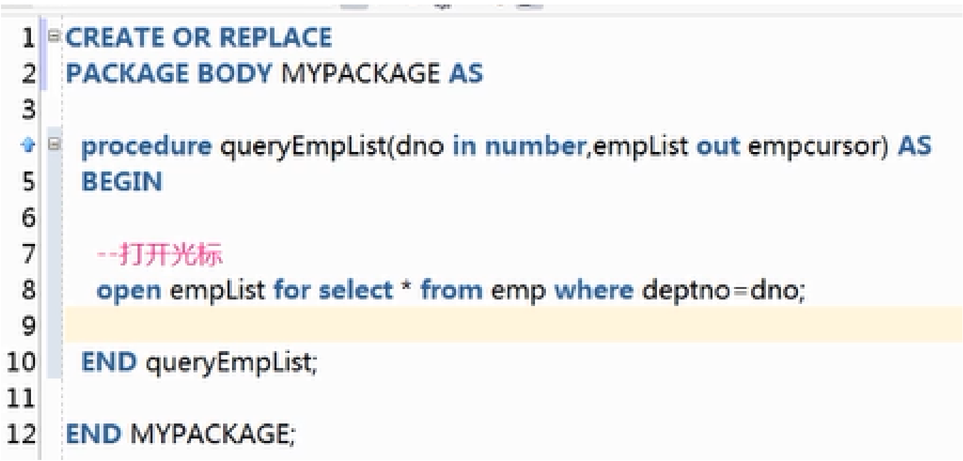


- 包体：

```
CREATE OR REPLACE PACKAGE BODY MYPACKAGE AS

  procedure queryEmpList(dno in number,empList out empcursor) AS
  BEGIN
    open empList for select * from emp where deptno=dno;
  END queryEmpList;

END MYPACKAGE;
```

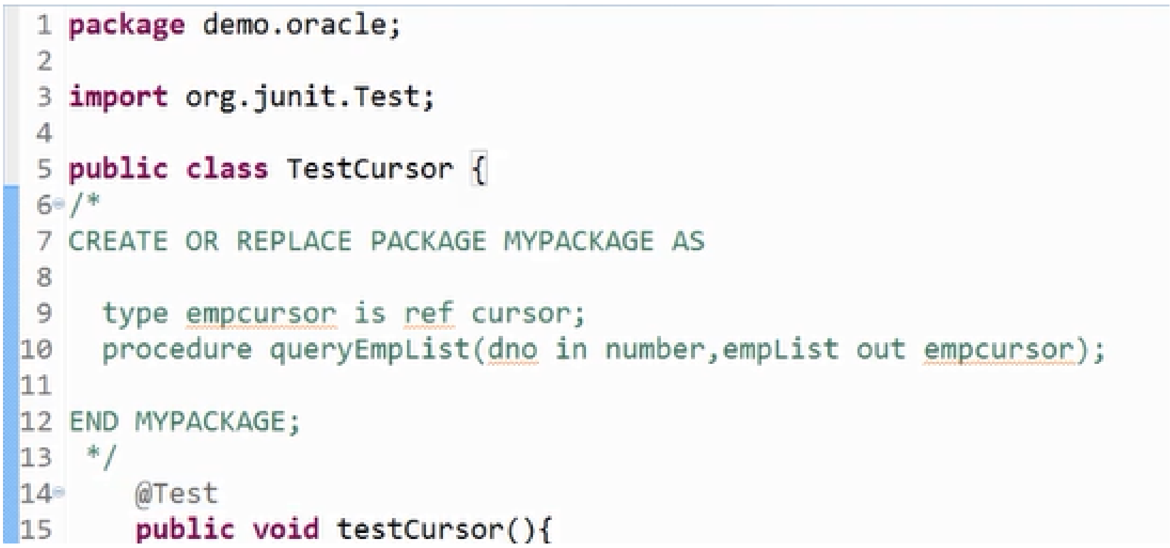


```
CREATE OR REPLACE
PACKAGE MYPACKAGE AS

  type empcursor is ref cursor;
  procedure queryEmpList(dno in number,empList out empcursor);

END MYPACKAGE;
```

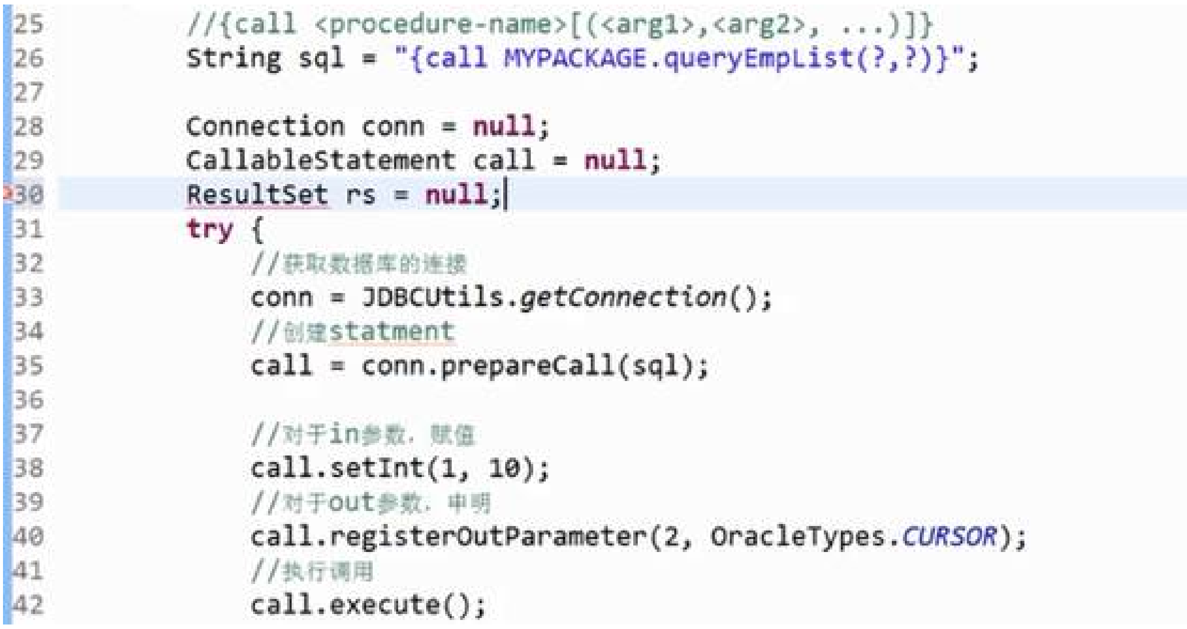


```
CREATE OR REPLACE
PACKAGE BODY MYPACKAGE AS

  procedure queryEmpList(dno in number,empList out empcursor) AS
  BEGIN

    --打开光标
    open empList for select * from emp where deptno=dno;

  END queryEmpList;

END MYPACKAGE;
```

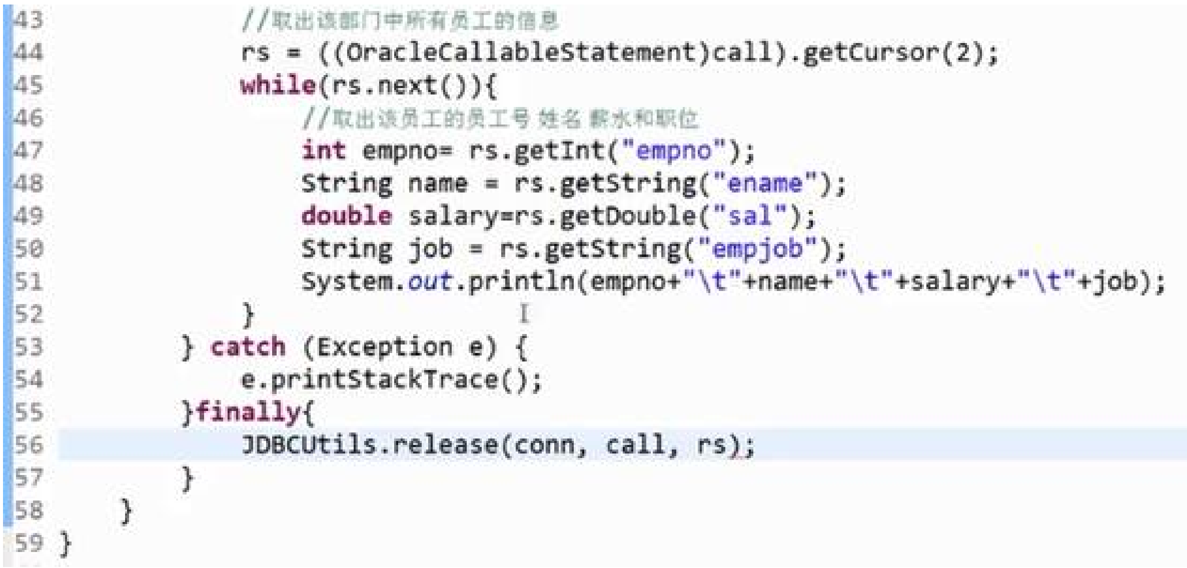
在java中访问包中的存储过程



```
package demo.oracle;

import org.junit.Test;

public class TestCursor {
/*
CREATE OR REPLACE PACKAGE MYPACKAGE AS

  type empcursor is ref cursor;
  procedure queryEmpList(dno in number,empList out empcursor);

END MYPACKAGE;
 */
    @Test
    public void testCursor(){
```



```
        //{call <procedure-name>[(<arg1>,<arg2>, ...)]}
        String sql = "{call MYPACKAGE.queryEmpList(?,?)}";

        Connection conn = null;
        CallableStatement call = null;
        ResultSet rs = null;
        try {
            //获取数据库的连接
            conn = JDBCUtils.getConnection();
            //创建statment
            call = conn.prepareCall(sql);

            //对于in参数，赋值
            call.setInt(1, 10);
            //对于out参数，申明
            call.registerOutParameter(2, OracleTypes.CURSOR);
            //执行调用
            call.execute();
```



```
            //取出该部门中所有员工的信息
            rs = ((OracleCallableStatement)call).getCursor(2);
            while(rs.next()){
                //取出该员工的员工号 姓名 薪水和职位
                int empno= rs.getInt("empno");
                String name = rs.getString("ename");
                double salary=rs.getDouble("sal");
                String job = rs.getString("empjob");
                System.out.println(empno+"\t"+name+"\t"+salary+"\t"+job);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }finally{
            JDBCUtils.release(conn, call, rs);
        }
    }
}
```

注意：记得带上包名。