# 《Java ProgramingⅡ》实验报告

**班级：1622107022**

**学号：162210702201**

**姓名：邓石铜**

**实验一 java 语言基本技能**

1. **实验目的**

掌握Java的一些基本技能。

1. **实验内容**

（1）static域和方法的理解和使用 ,

（2）this理解和使用

（3）string不可变类的理解和使用

（4）stringBuilder可变类的理解和使用

1. **实验过程和结果分析**

1、源代码

**package** shiyan1;

**class** Stu{

**private** String name;

**private** **int** age;

**public** **void** Stu(String name,**int** age){

**this**.name=name;

**this**.age=age;

}

**public** String getName(){

**return** name;

}

**public** **int** getAge(){

**return** age;

}

}

**public** **class** Student{

**public** **static** **void** main(String args[]){

Stu s1=**new** Stu();

s1.Stu("BOB", 18);

System.***out***.println(s1.getName());

System.***out***.println(s1.getAge());

StringBuilder s2=**new** StringBuilder();

s2.append("Mike");

System.***out***.println(s2);

s2.replace(4, 5, "19");

System.***out***.println(s2);

}

}

2、运行结果

C:\Users\lenovo\Documents\Tencent Files\1602104418\Image\C2C\5G_KS@V[V6]`1}UE09{UHL2.png

1. **总结**
2. **实验内容总结**

通过本次实验，我用了一个简单的例子，理解了static域，如果加上静态关键字static，可以构成静态全局变量，静态局部变量，静态函数。它们共同的特点就是不能被其他文件所用。可以在函数中使用this指针调用当前对象。在Java中string和stringBuilder有所区别。String是不可变对象，每次对string类型的改变等同于生成了一个新的string对象，所以遇到经常改变内容的字符串最好不用string，内存中引用的对象多了之后，速度会变慢，对系统的性能产生影响。而stringBuilder能改变字符串的长度，可以利用各种方法对字符串进行增删和修改。

1. **心得体会**

Java中有许多的关键字值得我们去深入研究，例如本次实验中的static关键字，用于定义静态变量和静态函数，this指针可以用来调用当前对象，Java中string和stringBuilder，一个是不要可变字符串，一个是可变字符串，String是不可变对象，每次对string类型的改变等同于生成了一个新的string对象，所以遇到经常改变内容的字符串最好不用string，内存中引用的对象多了之后，速度会变慢，对系统的性能产生影响。而stringBuilder能改变字符串的长度，可以利用各种方法对字符串进行增删和修改。所以，在以后的编程过程中，不能仅仅追求于程序运行出来了就结束，还要学会怎样去优化一个程序，减少占用的内存，缩短运行时间，增加运行的效率，和程序的可靠性和安全性。

**实验二** JDBC数据库编程

* 1. **实验目的**

学会使用JDBC存取典型关系数据库，比如MySql数据，对数据进行适当的处理。

* 1. **实验内容**
     1. 驱动程序装载
     2. 数据库连接
     3. 数据库数据的存取
     4. 数据库数据的处理
  2. **实验过程和结果分析**
     1. 思路

首先安装好MySQL并建立数据库，其次在myeclipse里进行驱动程序的装载，数据库的链接，数据库的存取，插入，删除，修改等处理。

* + 1. 源代码

**package** mysql;

**import** java.sql.\*;

**public** **class** Main {

**static** Connection *con*;

**static** PreparedStatement *sql*;

**static** ResultSet *re*;

**public** **static** **void** main(String []args) **throws** SQLException, ClassNotFoundException {

Class.*forName*("com.mysql.jdbc.Driver");//加载驱动器

*con*=DriverManager.*getConnection*("jdbc:mysql://localhost:3306/student", "root", "123456");//获得数据库连接

//对sql进行预处理预处理

//查询数据库

*sql*=*con*.prepareStatement("select \* from test");

*re*=*sql*.executeQuery();//执行sql语句

**while**(*re*.next()) {//遍历re集合

String id=*re*.getString("id");

String name=*re*.getString("name");

String sex=*re*.getString("sex");

System.***out***.println(id+" "+name+" "+sex+" ");

}

System.***out***.println();

//插入数据

*sql*=*con*.prepareStatement("insert into test values('05','BOBy','M')");

*sql*.executeUpdate();

//修改数据

*sql*=*con*.prepareStatement("update test set id='02' where name='David'");

*sql*.executeUpdate();

//删除数据

*sql*=*con*.prepareStatement("delete from test where id=03 ");

*sql*.executeUpdate();

*sql*=*con*.prepareStatement("select \* from test");

*re*=*sql*.executeQuery();//执行sql语句

**while**(*re*.next()) {//遍历re集合

String id=*re*.getString("id");

String name=*re*.getString("name");

String sex=*re*.getString("sex");

System.***out***.println(id+" "+name+" "+sex+" ");

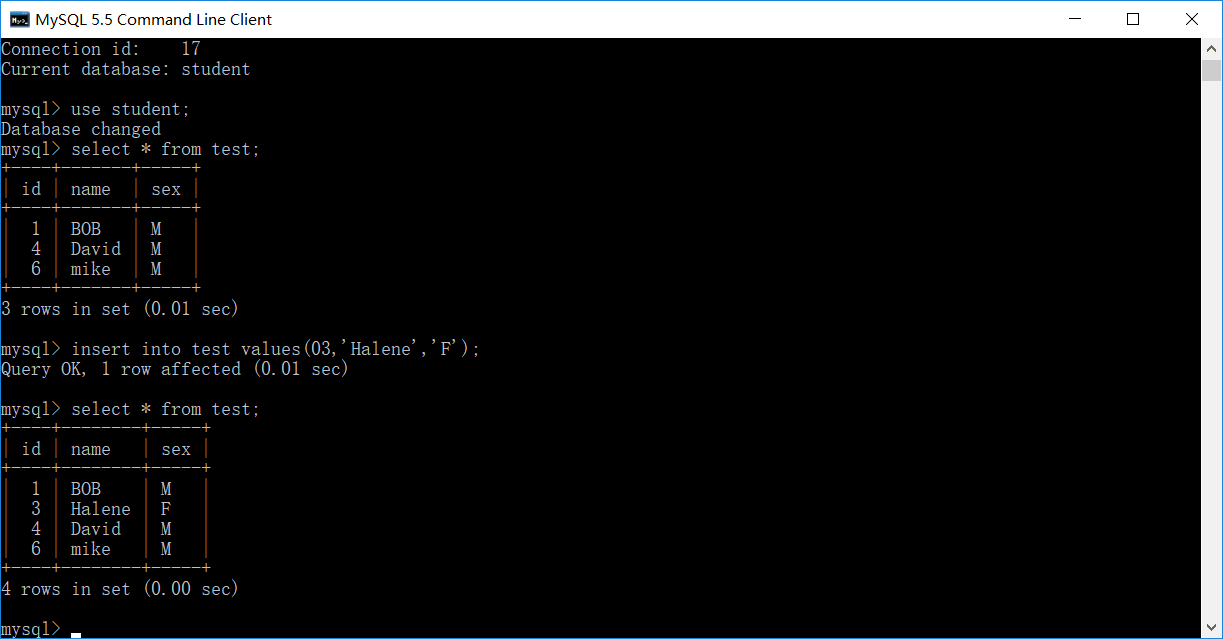
}

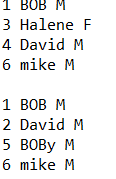
System.***out***.println();

}

}

* + 1. 实验结果





* 1. **总结**
     1. **实验内容总结**

在本次实验中，我了解到可以通过Java代码连接数据库，从而对数据库进行插入，删除，修改等操作。在本次实验中出现了几点错误，首先，在驱动程序装载的时候，我没有导入mysql的jar包，使得我运行结果错误，没有正确连接上mysql数据库，其次，连接上数据库之后，发现插入的操作有问题，我原本是几条记录一起插入，导致插入不成功，插入单条记录就能运行成功。最后，每执行一次程序代码，都能改变mysql里面的数据，如果重复执行，会出现主键不唯一等错误，导致某些操作不能执行。

* + 1. **心得体会**

通过本次实验，我学习到了Java不仅仅可以用来写一个单纯的程序，还能够连接MySQL数据库或其他数据库存取典型数据库，并对数据库进行一些基本的操作，如插入，修改，删除。通过导入Jar包，可以对数据库进行连接，首先要找到properties，选择add external jars，选择jar包路径下的jar包，只有驱动程序装载完成才能成功的连接数据库，在以后的学习道路上，我应该深入学习连接数据库，利用JDBC存取典型数据库。

## 实验三、四 Java web application

## 一、实验目的

了解和认识Java web application的涉及的核心技术、组成和开发过程

### 二、实验内容

实现一个简单的学生信息管理：学生信息（学号、姓名、性别和年龄等）输入；学生信息的查询；

* 客户端设计与实现
  + HTML
  + CSS
  + Javascript
* 服务器端设计与实现
  + Servlet
  + Java
  + JDBC

三、实验过程

1、源代码

**package** bean;

**import** java.io.IOException;

**import** java.io.PrintWriter;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** javax.servlet.ServletException;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**public** **class** Db **extends** HttpServlet {

String driver="com.mysql.jdbc.Driver";

String url="jdbc:mysql://localhost:3306/student";

String user="root";

String password="123456";

**public** **void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

Connection conn;

PreparedStatement sql;

ResultSet res;

String id=request.getParameter("id");

String id\_in=request.getParameter("id\_insert");

String name\_in=request.getParameter("name\_insert");

String sex\_in=request.getParameter("sex\_insert");

response.setContentType("text/html;charset=utf-8");

PrintWriter out=response.getWriter();

**if**(id!=**null**){

String title="check :";

out.println("<html>\n" +

"<head><title>" + title + "</title></head>\n" +

"<body>\n" +

"<strong>"+title+"</strong>\n");

**try**{

Class.*forName*(driver);

conn=DriverManager.*getConnection*(url, user, password);

sql=conn.prepareStatement("select \*from test where id=?");

sql.setString(1,id);

res=sql.executeQuery();

**while**(res.next()){

String id1=res.getString("id");

String name1=res.getString("name");

String sex=res.getString("sex");

out.println(id1+" "+name1+" "+sex);

out.println("<br/>");

}

out.println("</body></html>");

}

**catch**(Exception e){

e.printStackTrace();

}

}

**if**(id\_in!=**null**){

String title="insert";

out.println("<html>\n" +

"<head><title>" + title + "</title></head>\n" +

"<body>\n" );

**try**{

Class.*forName*(driver);

conn=DriverManager.*getConnection*(url, user, password);

sql=conn.prepareStatement("insert into test values(?,?,?)");

sql.setString(1,id\_in);

sql.setString(2,name\_in);

sql.setString(3,sex\_in);

sql.executeUpdate();

out.println("<strong>"+"success"+"</strong>\n"+"</body></html>");

}

**catch**(Exception e){

e.printStackTrace();

}

}

}

**public** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

doGet(request,response);

}

}

<!DOCTYPE html>

<html>

<head>

<meta charset=*"utf-8"*>

<title>insert</title>

<meta name=*"keywords"* content=*"keyword1,keyword2,keyword3"*>

<meta name=*"description"* content=*"this is my page"*>

<meta name=*"content-type"* content=*"text/html; charset=UTF-8"*>

<!--<link rel="stylesheet" type="text/css" href="./styles.css">-->

</head>

<body>

<form action=*"Db"* method=*"post"*>

<strong>insert:</strong><br><br>

id:<input type=*"text"* name=*"id\_insert"* style="width:*174px*;"><br><br>

name:<input type=*"text"* name=*"name\_insert"*><br><br>

sex:<input type=*"radio"* name=*"sex"* value=*"M"* checked=*"checked"*>M

<input type=*"radio"* name=*"sex"* value=*"F"* checked=*"checked"*>F<br/>

<input type=*"submit"* value=*"提交"* style="color:*Bule*;width:*46px*;height:*30px*;">

</form>

</body>

</html>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"utf-8"*>

<title>search</title>

<meta name=*"keywords"* content=*"keyword1,keyword2,keyword3"*>

<meta name=*"description"* content=*"this is my page"*>

<meta name=*"content-type"* content=*"text/html; charset=UTF-8"*>

<!--<link rel="stylesheet" type="text/css" href="./styles.css">-->

</head>

<body>

<strong>select:</strong><br>

<form action=*"Db"* method=*"post"*>

id:<input type=*"text"* name=*"id"*>

<input type=*"submit"* value=*"提交"* style="color:*Bule*;width:*46px*;height:*31px*">

<br><br>

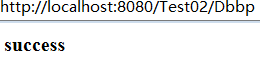
</form>

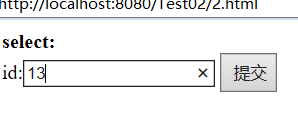
</body>

</html>

* + 1. 实验结果









四、总结

1、实验内容总结

首先，这次的web应用开发，实现了一个小功能，即通过连接学生数据库，实现网页输入一个学生的学号，输出该学生的姓名和性别。在这次实验中，不仅要实现客户端，也要实现服务端。在这个过程中我遭遇了许多困难，也请教了许多同学，百度了很多，学习到了很多，首先创建项目时，不应该直接点finish，应该选择建立web.xml的项目。其次再创建servlet，再建立html文件，最后进行相关的操作。在web.xml中设置的url-pattern应该与action里的名字一样，否则会出错。

2、心得体会

通过本次实验，我了解和认识Java web application的涉及的核心技术、组成和开发过程。虽然只是一个简单的应用实现了一个简单的学生信息管理：学生信息（学号、姓名、性别和年龄等）输入；学生信息的查询；但是我却受益匪浅。第一，我学会了如何去配置所需要的环境变量。第二，我学会了用html去实现客户端的设计与实现，了解了如何去用servlet，jdbc去进行客户端的设计与实现，总而言之，通过这次的实验我对Java中的jdbc，servlet有了更深刻的理解，也让我意识到，学习Java的道路还很长，需要我去花时间，去探索，去不断地学习，让自己更深入地了解Java中的道，理解道，正如老师所说，悟道才能升仙，才能不被这个社会所淘汰。