# 《Java ProgramingⅡ》实验报告

**班级：1622107022**

**学号：162210702232**

**姓名：袁野**

**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**

**实验过程：**

**1.**static域和方法的理解和使用

**public** **class** StaticDemo {

**private** **static** **int** a=5;

**private** **int** b=6;

//在类加载时执行一次，被类共享，可以优化程序性能

**static** {

System.out.println("static demo");

}

StaticDemo(){

System.out.println("construction demo");

}

**public** **static** **void** print1() {

System.out.println(a);

/\*System.out.println(b);\*///在静态方法中无法访问非静态成员

/\*print2();\*///不可以在没有对象被创建的情况下被调用

}

**public** **void** print2() {

System.out.println(a);//在非静态方法中可以访问静态方法

System.out.println(b);

print1();//可以在没有对象被创建的情况下被调用

}

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

StaticDemo demo=**new** StaticDemo();

demo.print1();

demo.print2();

}

}

1. this理解和使用

**public** **class** ThisDemo {

**private** String str;

**public** ThisDemo(String str) {

**this**.str=str;//区分

**this**.print();//调用print方法

}

**public** **void** print() {

System.***out***.println("调用print方法");

}

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

ThisDemo demo=**new** ThisDemo("字符串");

System.***out***.print(demo.str);

}

}

1. string不可变类的理解和使用

**public** **class** demo {

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

String str1="asdf";//创建一个对象,并使str1指向该对象

System.***out***.println("str1:"+str1);

str1="12345";//创建一个新的对象，并使str1指向该对象

System.***out***.println("str1:"+str1);

String str2=**new** String("12345");

System.***out***.println("str2:"+str2);

System.***out***.println(str1==str2);//说明不是同一个对象

}

}

1. stringBuilder可变类的理解和使用

**public** **class** demo {

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

StringBuilder sb=**new** StringBuilder("Hello");

StringBuilder sb1=sb;

sb=sb.append(" world");

System.***out***.println(sb==sb1);//返回true,说明指向同一个对象，说明StringBuilder类是可变的

}

}

1. **总结**

1. static域和方法的理解和使用

在没有创建类对象的前提下可以被调用。

static在类加载时使用。

类对象共享，在静态方法中不能调用非静态方法，反之可以。

2. this理解和使用

指向当前对象。

构造函数中，区分类成员变量和参数。

3. string不可变类的理解和使用

string类是不可变类，每次创建并初始化后，都会在内存中开辟一个空间。

4. stringBuilder可变类的理解和使用

stringBuilder 是可变类，与每次都会开辟一个空间，用以存储内容的string不同。

# 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，对数据进行处理。

1. **实验内容**

* 驱动程序装载
* 数据库连接
* 数据库数据的存取
* 数据库数据的处理

1. **实验过程和结果分析**

**实验过程：**

package top.cyan.dbhandle;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

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/firstdb", "cyan", "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");

String age=re.getString("age");

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

}

System.out.println();

//插入数据

//sql=con.prepareStatement("insert into test values('2232','袁野','男','20')");

// sql.executeUpdate();

//修改数据

sql=con.prepareStatement("update test set id='0002' where name='小王'");

sql.executeUpdate();

//删除数据

sql=con.prepareStatement("delete from test where id=2222 ");

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");

String age=re.getString("age");

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

}

System.out.println();

}

}

1. **总结**
2. jar包和jsoup的使用，确实不错。
3. **对Jdbc有了了解，即J**ava **D**ata**b**ase **C**onnectivity，是用于Java编程语言和数据库之间的数据库无关连接的标准Java API，换句话说：JDBC是用于在Java语言编程中与数据库连接的API。

## 实验三、四 Java web application

1. **实验目的**

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

1. **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、年龄等）输入；学生信息的查询；

1. **实验过程和结果分析**

**实验过程：**

客户端设计与实现

HTML

CSS

Javascript

服务器端设计与实现

Servlet

Java

JDBC

Servlet:

package net.cyan.test;

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 Demo extends HttpServlet {

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

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

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");

String age\_in=request.getParameter("age\_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");

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

out.println(id+" "+name1+" "+sex+" "+age);

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.setString(4,age\_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);

}

}

Select.html:

<!DOCTYPE html>

<html>

<head>

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

<title>select database</title>

</head>

<body>

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

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

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

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

<br><br>

</form>

</body>

</html>

Insert.html:

<!DOCTYPE html>

<html>

<head>

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

<title>insert</title>

</head>

<body>

<form action=*"Demo"* 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=*"text"* name=*"sex\_insert"*>

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

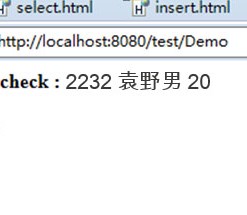
<input type=*"submit"* value=*"提交"* style="color: *Blue*; width: *46px*; height: *30px*;">

</form>

</body>

</html>

**运行结果：**







1. **总结**

Post方式的安全性比get方式要好，适合用Post数据提交方式

处理机密信息。