# 《Java ProgramingⅡ》实验报告

**班级：1622107022**

**学号：15**

**姓名：胡明明**

**实验一 java 语言基本技能**

1. **实验目的**

掌握java 语言的一些基本技能。

1. **实验内容**

* static域和方法的理解和使用
* this理解和使用
* string不可变类的理解和使用
* stringBuilder可变类的理解和使用

1. **实验过程和结果分析**

1）static域和方法的理解和使用

**package** aboutstatic;

**import** java.util.Scanner;

**public** **class** AboutStatic {

**private** Scanner scan;

//静态变量

**public** **static** **int** *a* = 1;

**public** **int** m = 1;

**public** **static** **void** showa(){

System.***out***.println("静态类无法使用this");

}

**public** **void** getValue(){

scan = **new** Scanner(System.***in***);

**int** a = scan.nextInt();

System.***out***.print(a);

}

}

**package** aboutstatic;

**public** **class** As {

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

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

//System.out.println(AboutStatic.m); m不是静态变量所以无法打印出来

AboutStatic A= **new** AboutStatic();

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

AboutStatic.*showa*();

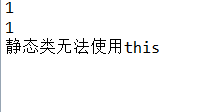
A.getValue();

// getValue()不是静态函数所以无法在不实例化的情况下进行调用

}

}

**运行结果：**



1. this理解和使用

**package** abouthis;

**public** **class** GetValue {

**int** a;

GetValue(){

a = 0;

}

**public** GetValue selfAdd(){

a++;

**return** **this**;

}

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

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

}

}

**package** abouthis;

**public** **class** Test {

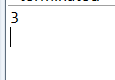
**public** **static** **void** main(String[] args) {

GetValue m = **new** GetValue();

m.selfAdd().selfAdd().selfAdd().print();

}

}



3）string不可变类的理解和使用

**package** aboustring;

**public** **class** AboutString{

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

**char**[] hA = { 'a', 'b', 'c', 'd'};

String hstring= **new** String(hA);

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

}

}

**运行结果：**



1. stringBuilder可变类的理解和使用

**package** stringandstringbuilder;

**public** **class** T {

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

String string1 = "a";

System.***out***.println("1、" + string1 );

StringBuilder sBuilder = **new** StringBuilder("a");

sBuilder.append("b");

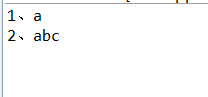
sBuilder.append("c");

System.***out***.println("2、" +sBuilder);

}

}

**运行结果：**



1. **总结**
2. **实验内容总结**

**1）**static 关键字用来声明独立于对象的静态方法。静态方法不能使用类的非静态变量。静态方法从参数列表得到数据，然后计算这些数据。

**2）Java中，一般来说this指针指的是当前正在访问的这段代码的对象，但是如果在内部类中需要使用外部类中的对象，这时就需要使用外部类的类名进行限定。**

**java中关键字this指针只能用于方法内，当一个对象被创建后，JVM就会给这个对象分配一个引用自身的指针，这个指针就是this。this只能在类中的非静态方法中使用，静态方法和静态代码块中不能出现this。this只和特定对象关联，不个类关联，所以同一个类的不同对象有不同的this。**

**3）**String 类是不可改变的，所以你一旦创建了 String 对象，那它的值就无法改变。

**4)同3）**一旦创建了 String 对象，那它的值就无法改变如果需要对字符串做很多修改，那么应该选择使用 [StringBuffer & StringBuilder 类](http://www.runoob.com/java/java-stringbuffer.html" \o "StringBuffer & StringBuilder 类" \t "http://www.runoob.com/java/_blank)。

1. **心得体会**

Java的基本语言技能是我们更深入学习Java语言的基础，只有学习好基础知识我们才能对以后学习的知识才能有更深刻的理解，学习基础语言需要我们不断练习、不断总结，从对错误我们可以更深入的了解这些基础，才能掌握所谓的“道”。

**实验二**  **JDBC数据库编程**

**一、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**

* 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**

**package** jdbc;

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

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

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

String driverName="com.microsoft.sqlserver.jdbc.SQLServerDriver";

String dbURL="jdbc:sqlserver://localhost:1433;DatabaseName=test";

String userName="hm";

String userPassword="123";

**try**{

Class.*forName*(driverName);

System.***out***.println("加载驱动成功！");

}**catch**(Exception e){

e.printStackTrace();

System.***out***.println("加载驱动失败！");

}

**try**{

Class.*forName*(driverName);

Connection dbConn=DriverManager.*getConnection*(dbURL,userName,userPwd);

System.***out***.println("连接数据库成功！");

String sql = "SELECT \* FROM stu";

Statement stmt = dbConn.createStatement();

ResultSet revise = stmt.executeQuery(sql);

**while**(rs.next()) {

System.***out***.print(revise.getInt("ID")+"\t");

System.***out***.print(revise.getString("name")+"\t");

System.***out***.println(revise.getString("score"));

}

String ins = "INSERT INTO stu VALUES ()";

PreparedStatement ad = dbConn.prepareStatement(ins);

ad.setString(1, "6");

ad.setString(2, "F");

ad.setString(3, "81");

ad.execute();

sql = "SELECT \* FROM stu";

rs = stmt.executeQuery(sql);

**while**(rs.next()) {

System.***out***.println(revise.getInt("ID")+"\t");

System.***out***.println(revise.getString("name")+"\t");

System.***out***.println(revise.getString("score"));

}

revise.close();

stmt.close();

dbConn.close();

}**catch**(Exception e){

e.printStackTrace();

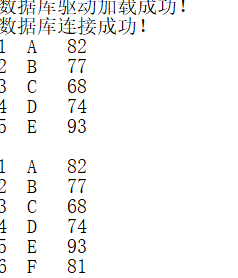
System.***out***.print("sql连接失败！");

}

}

}

**运行结果：**



**四、总结**

**1.实验内容总结**

**要使用Java进行数据库编程，需要一个数据库驱动包， forName()方法的参数用于指定要加载的数据库驱动，DriverManager类是用来管理数据库中所有的驱动程序，是JDBC的管理层，作用用户和驱动程序之间，并在数据库的驱动之间建立连接。DriverManager类中的方法都是静态方法，所以用的时候不需要实例化，直接调用类名就可以。**

**2.心得体会**

**在 jdbc 的学习过程中，可能会遇到各种问题感觉无从下手，应多去专业网站了解相关的知识，书本上的知识有限。要会从网上搜索有用的信息 加以整理，促进学习的深入和知识水平的提高。**

**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、实验过程和结果分析**

**网页代码：**

<!DOCTYPE html>

<html>

<head>

<title>M</title>

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

<meta name=*"viewport"* content=*"width=device-width, initial-scale=1"*>

<title>T</title>

<meta name=*"description"* content=*""*>

<meta name=*"keywords"* content=*""*>

<meta name=*"author"* content=*""*>

</head>

<body>

<form action=*"Test"* method=*"POST"*>

Name:<input type=*"text"* name=*"Nmae"*><br>

<input type=*"submit"* value=*"search"*>

</form>

<form action=*"Insert"*>

</form>

</body>

</html>

**package** sev\_DB;

**import** java.io.IOException;

**import** java.io.PrintWriter;

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

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebInitParam;

**import** javax.servlet.annotation.WebServlet;

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

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

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

@WebServlet(

urlPatterns = { "/Insert" },

initParams = {

@WebInitParam(name = "Insert", value = "/TomcatTest/Insert")

})

**public** **class** Insert **extends** HttpServlet {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

//数据库服务器端

**static** **final** String ***ODBC\_DRIVER*** = "com.microsoft.sqlserver.jdbc.SQLServerDriver";

**static** **final** String ***DB\_URL*** = "jdbc:sqlserver://localhost:1433;DatabaseName=test";

//用户登录名和登录密码

**static** **final** String ***USER*** = "hm";

**static** **final** String ***PASS*** = "123";

**static** PreparedStatement *ps*;

/\*\*

\* Constructor of the object.

\*/

**public** Insert() {

**super**();

}

/\*\*

\* Destruction of the servlet. <br>

\*/

**public** **void** destroy() {

**super**.destroy(); // Just puts "destroy" string in log

// Put your code here

}

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

Connection conn = **null**;

Statement stmt = **null**;

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

request.setCharacterEncoding("UTF-8");//防止乱码

PrintWriter out = response.getWriter();

String title = "插入学生信息";

String docType = "<!DOCTYPE html>\n";

out.println(docType +

"<html>\n" +

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

"<body bgcolor=\"#f0f0f0\">\n" +

"<h1 align=\"center\">" + title + "</h1>\n");

**try**{

//加载数据库驱动

Class.*forName*("com.microsoft.sqlserver.jdbc.SQLServerDriver");

//连接数据库

conn = DriverManager.*getConnection*(***DB\_URL***,***USER***,***PASS***);

String na = request.getParameter("Nmae");

stmt = conn.createStatement();

String ins = "INSERT INTO stu VALUES (?,?,?)";

*ps* = conn.prepareStatement(ins);

*ps*.setString(1, request.getParameter("ID"));

*ps*.setString(2,request.getParameter("name"));

*ps*.setString(3, request.getParameter("score"));

*ps*.execute();

String sql = "SELECT \* FROM stu";

ResultSet rs = stmt.executeQuery(sql);

//显示查询结果

**while**(rs.next()){

**int** id = rs.getInt("id");

String name = rs.getString("name");

String score = rs.getString("score");

out.println("ID: " + id);

out.println(" 姓名: " + name);

out.println(" 得分: " + score);

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

}

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

rs.close();

stmt.close();

conn.close();

} **catch**(SQLException se) {

se.printStackTrace();

} **catch**(Exception e) {

e.printStackTrace();

}**finally**{

**try**{

**if**(stmt!=**null**)

stmt.close();

}**catch**(SQLException se2){

}

**try**{

**if**(conn!=**null**)

conn.close();

}**catch**(SQLException se){

se.printStackTrace();

}

}

}

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

**this**.doGet(request, response);

}

**public** **void** init() **throws** ServletException {

}

}

**四、总结**

**1.实验内容总结**

首先需要加载数据库驱动；

通过DriverManager获取数据库连接；

创建Statement对象

完成数据库的连接，并且查询表中的所有数据；

处理数据

当访问完毕后需要关闭连接；

**2.心得体会**

本次试验注重让我们掌握了如何使用的开发工具来制作简单的应用,在没有类似开发网页应用经验的情况下，这更考验我们的自学能力，遇到难以解决的问题，我们需要高效快捷找出解决问题的方法。