# 《Java ProgramingⅡ》实验报告

**班级：1622107022**

**学号：162210702218**

**姓名：李健**

## 实验一 java 语言基本技能

### 一．实验目的

掌握java 语言的一些基本技能。

### 二．实验内容和过程

* 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

三．实验过程和结果分析

static域和方法的理解和使用

public class Test5 {

private static int a;

private int b;

static{

Test5.a=3;

System.out.println(a);

Test5 t=new Test5();

t.f();

t.b=1000;

System.out.println(t.b);

}

static{

Test5.a=4;

System.out.println(a);

}

public static void main(String[] args) {

// TODO 自动生成方法存根

}

static{

Test5.a=5;

System.out.println(a);

}

public void f(){

System.out.println("hhahhahah");

}

}

运行结果分析：

3

hhahhahah

1000

4

5

This理解和使用

public class ReferenceVariable {

private int a;

public ReferenceVariable(int a){

this.a = a;

}

public int getA(){

return a;

}

public void setA(int a){

this.a = a;

}

}

public class ReferenceConstructor {

int a;

public ReferenceConstructor(){

this(0);

}

public ReferenceConstructor(int a){

this.a = a;

}

}

public class ReferenceObject {

ReferenceObject instance;

public ReferenceObject(){

instance = this;

}

public void test(){

System.out.println(this);

}

}

**string不可变类的理解和使用**

package string;

public class TestString {

public static void main(String[] args) {

String s1 = "hello";

String s2 = "hello";

System.out.println(s1 == s2);

s1 = new String("hello");

s2 = new String("hello");

System.out.println(s1 == s2);

System.out.println(s1.equals(s2));

char c[] = {'s','u','n',' ','j','a','v','a'};

String s3 = new String(c);

String s4 = new String(c,4,4);

System.out.println(s3);

System.out.println(s4);

String s5 = "sun java";

String s6 = "sun Java";

System.out.println(s5.charAt(1));

System.out.println(s6.length());

System.out.println(s5.indexOf("java"));

System.out.println(s5.indexOf("Java"));

System.out.println(s5.equals(s6));

System.out.println(s5.equalsIgnoreCase(s6));

System.out.println(s5.contains("sun"));

System.out.println(s5.replace("sun", "oracle"));

String s9 = "Welcome to Java World!";

String s10 = " sun java ";

System.out.println(s9.startsWith("Welcome"));

System.out.println(s9.endsWith("World"));

System.out.println(s9.toLowerCase());

System.out.println(s9.toUpperCase());

System.out.println(s9.substring(11));

System.out.println(s10.trim());

int num = 12345;

String strNum = String.valueOf(num);

System.out.println(strNum);

String str = "9,11,2012";

String[] strArray = str.split(",");

for(int i=0; i<strArray.length; i++){

System.out.println(strArray[i]);

}

}

}

**stringBuilder可变类的理解和使用**

public class AppendStringTest

{

public static void main(String[] args)

{

String text = "" ;

long beginTime = System.currentTimeMillis();

for ( int i= 0 ;i< 10000 ;i++)

text = text + i;

long endTime = System.currentTimeMillis();

System.out.println("执行时间：" +(endTime-beginTime));

StringBuilder sb = new StringBuilder ( "" );

beginTime = System.currentTimeMillis();

for ( int i= 0 ;i< 10000 ;i++)

sb.append(String.valueOf(i));

endTime = System.currentTimeMillis();

System.out.println("执行时间：" +(endTime-beginTime));

}

}

public class AppendStringTest

{

public static void main(String[] args)

{

String text = "";

long beginTime = System.currentTimeMillis();

for(int i=0;i<10000;i++)

text = text + i;

long endTime = System.currentTimeMillis();

System.out.println("执行时间："+(endTime-beginTime));

StringBuilder sb = new StringBuilder ("");

beginTime = System.currentTimeMillis();

for(int i=0;i<10000;i++)

sb.append(String.valueOf(i));

endTime = System.currentTimeMillis();

System.out.println("执行时间："+(endTime-beginTime));

}

}

**四、总结**

**1．实验内容总结**

Static静态函数只能调用静态变量

Static静态函数不能调用this指针

Static往往表示这一个共享区域，为这个类所有成员所拥有

This指针代表着这个类自身

String在附加另一个字符串时往往不会改变这个变量自身，而是重新创建一个变量将附加的字符串和原来的拼接起来重新存入一个新的变量。

StringBuilder类所使用的Append（）函数却不用再次经历这一个过程。因此在效率上StringBuilder会比string类更加有效。

**2．心得体会**

通过这次实验一，我对Java里的基础知识有了一定的了解，原来我对于static静态方法，this指针，string不可变类以及stringBuilder可变类并没有很清晰的认识，现在我却更加了解了。

## 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。

### 二．实验内容和过程

* 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

**三．实验过程和结果分析**

**实验过程**

**package** pkg;

**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="sa";

String userPwd="shixing19970805";

**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 rs = stmt.executeQuery(sql);

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

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

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

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

}

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

PreparedStatement ps = dbConn.prepareStatement(ins);

ps.setString(1, "17");

ps.setString(2, "老虎");

ps.setString(3, "71");

ps.execute();

sql = "SELECT \* FROM stu";

rs = stmt.executeQuery(sql);

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

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

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

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

}

rs.close();

stmt.close();

dbConn.close();

}**catch**(Exception e){

e.printStackTrace();

System.***out***.print("SQL Server连接失败！");

}

}

}

**结果分析**

加载驱动成功！

连接数据库成功！

1 wanga 47

2 laotian 77

3 xiaozi 98

16 dvd 14

76 huangmao 76

1 wanga 47

2 laotian 77

3 xiaozi 98

16 dvd 14

17 老虎 71

76 huangmao 76

**四、总结**

**1．实验内容总结**

数据库加载需要相应的包和对应服务器地址

服务器连接时需要对应和匹配的密码

Class.forName()函数用来加载相应的数据库驱动程序

DriverManager.getConnection()是一个用于连接数据库的函数，会返回一个Connection类型的变量，同时三个参数类型分别是服务器地址，服务器用户名，服务器密码。

Statement类是用于处理SQL语句的

**2．心得体会**

在这次实验二中，我学会使用JDBC存取典型关系数据库，对JDBC数据库有了一定的了解，以后会更加熟练应用JDBC数据库的。

## 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．实验内容和过程

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**

<!DOCTYPE html>

<html>

<head>

<title>MyHtml.html</title>

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

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

<meta name=*"content-type"*>

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

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

</head>

<body>

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

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

<input type=*"submit"* value=*"查询"*>

</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 = { "/Test" },

initParams = {

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

})

**public** **class** Test **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*** = "sa";

**static** **final** String ***PASS*** = "shixing19970805";

**static** PreparedStatement *ps*;

/\*\*

\* Constructor of the object.

\*/

**public** Test() {

**super**();

}

**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 = "查询学生信息";

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 sql = "SELECT \* FROM stu WHERE name = '"+ na + "'";

//执行数据库的语句

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 {

doGet(request,response);

}

/\*\*

\* Initialization of the servlet. <br>

\*

\* **@throws** ServletException if an error occurs

\*/

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

// Put your code here

}

}

<!DOCTYPE html>

<html>

<head>

<title>Insertl.html</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"*>

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

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

</head>

<body>

This is my HTML page. <br>

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

学号：<input type=*"text"* name=*"ID"*><br>

姓名：<input type=*"text"* name=*"name"*><br>

分数：<input type=*"text"* name=*"score"*><br>

<input type=*"submit"* value=*"插入"*>

</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*** = "sa";

**static** **final** String ***PASS*** = "shixing19970805";

**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 ocType = "<!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 {

// Put your code here

}

}

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN" "http://java.sun.com/dtd/web-app\_2\_3.dtd">

<web-app>

<servlet>

<servlet-name>Test</servlet-name>

<display-name>This is the display name of my J2EE component</display-name>

<description>This is the description of my J2EE component</description>

<servlet-class>sev\_DB.Test</servlet-class>

</servlet>

<servlet>

<servlet-name>Insert</servlet-name>

<display-name>This is the display name of my J2EE component</display-name>

<description>This is the description of my J2EE component</description>

<servlet-class>sev\_DB.Insert</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>Test</servlet-name>

<url-pattern>/TomcatTest/Test</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>Insert</servlet-name>

<url-pattern>/TomcatTest/Insert</url-pattern>

</servlet-mapping>

</web-app>

**四、总结**

**1．实验内容总结**

环境配置时要将驱动包放在lib文件内，否则会出现找不到驱动的异常情况，其次直接将jar文件直接拖入lib文件中，通过importPath操作方法无法使对应的数据库驱动文件放置在lib文件夹内。

HTML文件中<form>标签内action属性的值要对应相关的Severlet名称。Name属性是放置在<form>标签的子元素内，用于使Severlet获得相应表单的元素。

对于数据的查找可使用PreparementStatement类。

Request.getParameter用于获得页面中表单元素的信息。

**2．心得体会**

在这次的实现一个简单的学生信息管理系统中，我更加进一步的了解了Java的具体编程的乐趣，也对Java所做出来的成果有更加进一步了理解。相信在以后的Java编程中会走的更远。