|  |  |
| --- | --- |
| **成绩** |  |

**《程序设计4（Java平台）》**

**实验报告**

**班级： 13901801**

**姓名： 何庚坤**

**学号： 2018214139**

**指导老师： 马创**

**实验名称： java平台实验一**

**实验地点：**

1. **实验目的**

1、熟悉 Java 基础语法，包括运算符、表达式、基本数据类型和程序结构等;

2、掌握编写和调试简单 Java 应用程序的方法。

1. **实验内容**

1、掌握 Java 基本语法

包括:包括运算符、表达式、基本数据类型和程序结构等，具体见课件与课上实例。

2、上机题目

教材 P60:  
(1)练习题，第二大题的 2 题  
(2)练习题，第二大题的 5 题  
(3)练习题，第三大题。

1. **实验结果与分析**

成功完成实验，代码运行正确，且结果与预期相同

1. **核心代码清单**
2. import java.util.Scanner;  
     
   public class HelloWorld {  
    public static void main(String[] args) {  
    Scanner scan = new Scanner(System.*in*);  
    // 判断是否还有输入  
    if (scan.hasNextInt()) {  
    int int1 = scan.nextInt();  
    int int2 = scan.nextInt();  
    int int3 = scan.nextInt();  
     
    if (int1 > int2) {  
    if (int2 > int3) {  
    System.*out*.println(int3 + ", " + int2 + ", " + int1);  
    } else if (int1 > int3) {  
    System.*out*.println(int2 + ", " + int3 + ", " + int1);  
    } else {  
    System.*out*.println(int2 + ", " + int1 + ", " + int3);  
    }  
    } else {  
    if (int2 < int3) {  
    System.*out*.println(int1 + ", " + int2 + ", " + int3);  
    } else if (int1 < int3) {  
    System.*out*.println(int1 + ", " + int3 + ", " + int2);  
    } else {  
    System.*out*.println(int3 + ", " + int1 + ", " + int2);  
    }  
    }  
    }  
    scan.close();  
    }  
   }  
     
     
   import java.util.Scanner;  
     
   public class HelloWorld {  
    public static void main(String[] args) {  
    Scanner scan = new Scanner(System.*in*);  
    int max = Integer.*MIN\_VALUE*, min = Integer.*MAX\_VALUE*;  
    float average = 0;  
    while (scan.hasNextInt()) {  
    int i = scan.nextInt();  
    if (i == 0) break;  
    if (i > max) max = i;  
    if (i < min) min = i;  
    average = (average + i) / 2;  
    }  
    scan.close();  
    System.*out*.println("max: " + max + " min: " + min + " average: " + average);  
    }  
   }  
     
     
   import java.util.Scanner;  
     
   public class HelloWorld {  
    public static void main(String[] args) {  
    System.*out*.println(" ｜ 星期一 ｜ 星期二 ｜ 星期三 ｜ 星期四 ｜ 星期五 ");  
    System.*out*.println("1 ｜ 软件前沿技术 ｜ Java ｜ ｜ ｜ Java ");  
    System.*out*.println("2 ｜ ｜ Java ｜ ｜ 计算机组成原理 ｜ Java ");  
    System.*out*.println("3 ｜ 软件分析与设计 ｜ 计算机组成原理 ｜ ｜ ｜ ");  
    System.*out*.println("4 ｜ ｜ ｜ ｜ 体育俱乐部 ｜ ");  
    System.*out*.println("5 ｜ 中国近代史 ｜ ｜ ｜ 健康教育 ｜ ");  
    }  
   }

|  |  |
| --- | --- |
| **成绩** |  |

**《程序设计4（Java平台）》**

**实验报告**

**班级： 13901801**

**姓名： 何庚坤**

**学号： 2018214139**

**指导老师： 马创**

**实验名称： Java平台实验二**

**实验地点：**

1. **实验目的**

1、熟悉 Java 面向对象中类的概念;

2、熟悉和掌握类的定义、实例化、继承与多态等使用方法。

1. **实验内容**

1、掌握类的定义与实例化方法

类是某种类型的概念，类封装对象的属性和方法，是同一类对象的“模板”。 (1)类的声明

[类修饰符] class 类名 {

类主体

} (2)成员变量

又称之为“属性”。格式如下:

[修饰符] 类型 变量名; (3)方法

方法给出对象的行为，声明如 c 语言的函数定义。格式一般如下: [修饰符] 返回值 方法名(参数列表)  
{

方法体

} (4)声明对象

类名 对象名; (5)创建对象

new 构造函数([参数表]) 声明和创建可以一起进行:  
类名 对象名 = new 构造函数([参数表]);

1. **实验结果与分析**

test course and books

Book@74a14482 is for Course@3b9a45b3

Book@677327b6 is for Course@3b9a45b3

Book@1540e19d is for Course@3b9a45b3

第二个题：

table

\*---\*---\*---\*---\*

\* \* \* \* \*

\*---\*---\*---\*---\*

\* \* \* \* \*

\*---\*---\*---\*---\*

\* \* \* \* \*

\*---\*---\*---\*---\*

\* \* \* \* \*

\*---\*---\*---\*---\*

\* \* \* \* \*

\*---\*---\*---\*---\*

\* \* \* \* \*

\*---\*---\*---\*---\*

\* \* \* \* \*

\*---\*---\*---\*---\*

1. **核心代码清单**

import java.util.HashSet;

import java.util.Set;

public class HelloWorld {

public static void main(String[] args) {

Course course = new Course("Java course");

Book book1 = new Book("java book1");

Book book2 = new Book("java book2");

Book book3 = new Book("java book3");

Set<Book> books = new HashSet<Book>();

books.add(book1);

books.add(book2);

books.add(book3);

Test test = new Test("test course and books", course, books);

}

}

import java.util.HashSet;

import java.util.Set;

public class BookForCourse {

public Course course;

public Set<Book> books;

public BookForCourse(Course course) {

this.course = course;

this.books = new HashSet<Book>();

}

public void addBook(Book book) {

this.books.add(book);

}

public Set<Book> getBooks() {

return books;

}

}

public class Course {

public String name;

public Course(String courseName) {

this.name = courseName;

}

}

public class Book {

private String name;

public Book(String bookName) {

this.name = bookName;

}

}

import java.util.Set;

public class Test {

public Test(String description, Course course, Set<Book> books) {

System.out.println(description);

BookForCourse bfc = new BookForCourse(course);

books.forEach(bfc::addBook);

bfc.getBooks().forEach(book -> System.out.println(book + " is for " + bfc.course));

}

}

第二个题：

public class Main {

public static void main(String[] args) {

Table t = new Table("table", 4, 7, '\*');

t.draw();

}

}

public class Table {

private String name;

private int rawNumber;

private int colNumber;

private char lineType;

public Table(String name, int rawNumber, int colNumber, char lineType) {

this.name = name;

this.rawNumber = rawNumber;

this.colNumber = colNumber;

this.lineType = lineType;

}

public void draw() {

System.out.println(name);

for (int i = 0; i < 2 \* this.colNumber + 1; i++) {

for (int v = 0; v < 2 \* this.rawNumber + 1; v++) {

if (v % 2 == 0) {

System.out.print(lineType);

} else if (i % 2 == 0) {

System.out.print("---");

} else {

System.out.print(" ");

}

}

System.out.println("");

}

}

}

|  |  |
| --- | --- |
| **成绩** |  |

**《程序设计4（Java平台）》**

**实验报告**

**班级： 13901801**

**姓名： 何庚坤**

**学号： 2018214139**

**指导老师： 马创**

**实验名称： java平台实验三**

**实验地点：**

1. **实验目的**

1、熟悉 Java 节点流与处理流相关类与编程技巧;

2、熟悉和掌握 Java 字节流与字符流的概念与编程技巧。

1. **实验内容**

1、Java I/O 相关内容

(1)熟悉 Java 节点流与处理流相关类与编程技巧。

 节点流:该类流一般用于直接从指定位置进行读/写操作，如磁盘文件、内存区域、

网络连接等，其中一般只提供一些基本的读写操作方法，功能比较单一。

 处理流:该类流往往是用于对其他输入/输出流进行封装，对内容进行过滤处理，

一般提供了一些功能比较强大的读写方法。

实际开发中，二者常常结合起来使用。节点流直接与指定的源或目标相连，例如某 个文件或网络等。而处理流则对节点流或其他处理流进一步进行封装，提供更丰富的输 入/输出功能，如缓冲、按字符串读写等。

(2)熟悉和掌握 Java 字节流与字符流的概念与编程技巧。

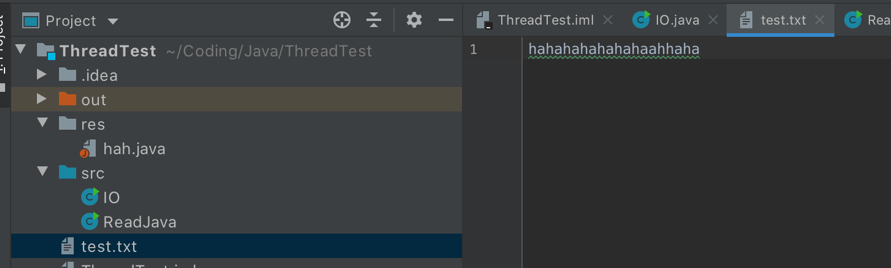
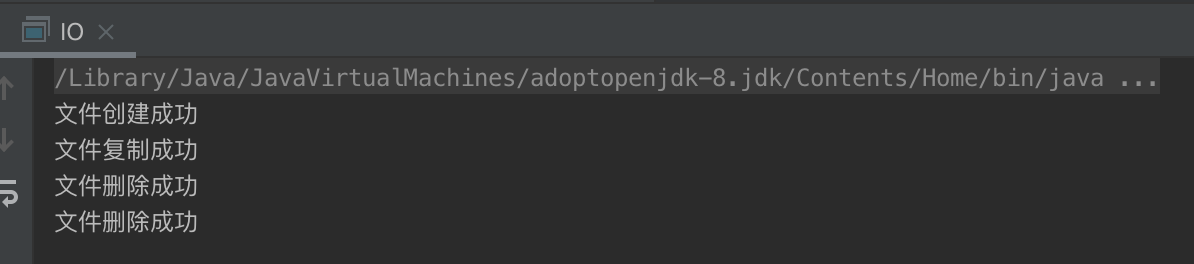
 字节流:字节流以字节为基本单位来处理数据，一般用于对二进制数据的读写，如 声音、图像等数据。

 字符流:字符流以字符为基本单位来处理数据，一般用于对文本类型数据的读写， 如文本文件，网络中发送的文本信息等，比字节流处理文本数据更高效。 字节流的两个抽象基类为:InputStream 和 OutputStream 类。用以处理文件的子类:

具体读写功能由 FileInputSteam 和 FileOutputSteam 类提供支持，它们是前者的对应 子类。

字符流的两个抽象基类为 Reader 和 Writer 类。用以处理文件的子类:具体读写功 能由 FileReader 和 FileWriter 类提供支持，它们是前者的对应子类。

1. **实验结果与分析**





1. **核心代码清单**

import java.io.\*;

public class IO {

public static void main(String[] args) throws IOException {

IO io = new IO();

io.createFile("test.txt", "hahahahahahahaahhaha");

io.copyFile("test.txt", "dest.txt");

io.deleteFile("dest.txt");

// io.deleteFile("test.txt");

}

public void createFile(String src, String content) throws IOException {

BufferedWriter out = new BufferedWriter(new FileWriter(src));

out.write(content);

out.close();

System.out.println("文件创建成功");

}

public void deleteFile(String src) {

File file = new File(src);

if (file.delete()) {

System.out.println("文件删除成功");

} else {

System.out.println("文件删除失败");

}

}

public void copyFile(String src, String dest) throws IOException {

FileInputStream in = new FileInputStream(src);

File file = new File(dest);

if(!file.exists()) {

file.createNewFile();

}

FileOutputStream out = new FileOutputStream(file);

int c;

byte buffer[] = new byte[1024];

while((c = in.read(buffer)) != -1) {

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

out.write(buffer[i]);

}

System.out.println("文件复制成功");

in.close();

out.close();

}

}

import java.io.\*;

public class ReadJava {

private static int normalLines = 0;//有效代码行数

public static void main(String[] args) throws IOException {

File folder = new File("res");//要统计的源目录路径

recur(folder);//递归该目录，统计其中代码行数

System.out.println("有效 Lines: " + normalLines);

}

/\*\*

\* 递归方法 若源路径为目录，则递归，直至不包含子目录

\* @param file 源文件或目录

\*/

public static void recur(File file) throws IOException {

File[] files = file.listFiles();

assert files != null;

for (File value : files) {

//若源文件是目录，则递归

if (value.isDirectory()) {

recur(value);

}

//若源文件是普通文件且为java源文件，则逐行分析之 匹配文件名时用到了正则表达式

else if ((value.isFile()) && value.getName().matches(".\*\\.java$")) {

parse(value);

}

}

}

/\*\*

\* 将java源文件逐行解析，统计每种代码数目 用到了正则表达式

\* @param file java源文件

\*/

public static void parse(File file) throws IOException {

BufferedReader br = null;

try {

br = new BufferedReader(new FileReader(file));

String line;

boolean isComment = false;

while((line = br.readLine()) != null) {

line = line.trim();

// 空行，/\* \*/ 注释，// 注释 的情况

if (!line.matches("^[\\s&&[^\\n]]\*") && !(line.startsWith("/\*") && line.endsWith("\*/")) && !line.startsWith("//")) {

if (line.startsWith("/\*") && !line.endsWith("\*/")) { //用来标记那些/\*...\*/注释不在同一行的情况

isComment = true;

} else if (isComment) {

if (line.endsWith("\*/")) {

isComment = false;

}

} else {

normalLines++;

}

}

}

} catch (FileNotFoundException e) {

e.printStackTrace();

} finally {

if(br != null)

{

try {

br.close();

} catch (IOException e) {

e.printStackTrace();

}

}

}

}

}

|  |  |
| --- | --- |
| **成绩** |  |

**《程序设计4（Java平台）》**

**实验报告**

**班级： 13901801**

**姓名： 何庚坤**

**学号： 2018214139**

**指导老师： 马创**

**实验名称： java平台实验四**

**实验地点：**

1. **实验目的**

1、熟悉 Java 多线程程序的编写方式;

2、熟悉和掌握 Thread 类的使用方法;

3、熟悉和掌握 Runnable 接口的使用方法。

1. **实验内容**

1、熟悉和掌握 Thread 类的使用方法

Java 程序实现多线程应用可以有两种途径:继承 Thread 类和实现 Runnable 接口。无论 使用哪种方式，都需要 java.lang.Thread 类及其方法的支持。

建立线程的步骤非常简单:  
(1)编写线程类的 run()方法;  
(2)创建线程对象;  
(3)调用线程对象的 start()方法。  
Thread 类在 java.lang 包下，因此不需要 import。常用构造函数:

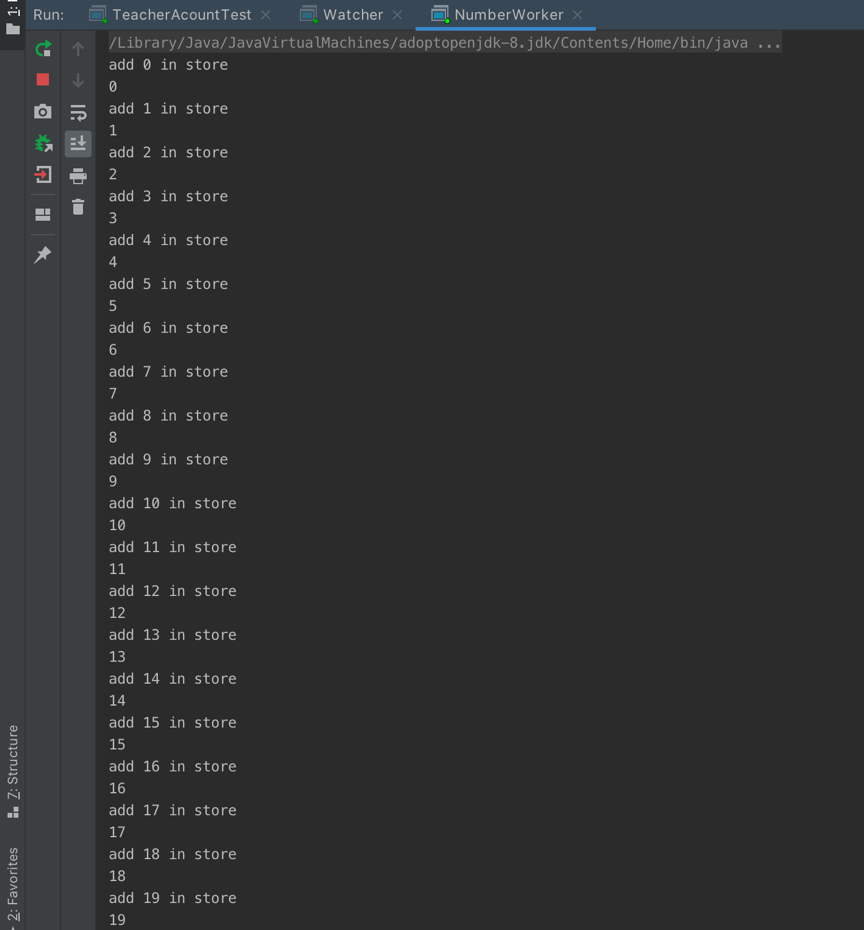
⚫Thread()

⚫Thread(Runnable) 2、熟悉和掌握 Runnable 接口的使用方法

java.lang.Runnable 接口只有 run()方法需要实现。一个实现了 Runnable 接口的类实际上 就拥有了新线程的功能。使用步骤为:

(1)定义实现 Runnable 接口的类，并实现其 run()方法;  
(2)通过 Thread t = new Thread(Runnable 实现类对象)创建线程对象; (3)通过 t.start()方法启动线程。

1. **实验结果与分析**



1. **核心代码清单**

**实验1**

class Sum implements Runnable {

private int num = 0;

private Store store;

Sum(Store s) {

store = s;

}

public void run() {

for (int i = 0; i < 20; i++) {

store.add(num++);

try {

Thread.sleep(10);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}

}

class Store implements Runnable {

private int[] nums = new int[10];

private int index = 0;

public synchronized void add(int num) {

System.out.println("add " + num + " in store");

nums[index] = num;

index += 1;

this.notify();

}

public synchronized int dispatch() {

while (index == 0) {

try {

wait();

} catch (InterruptedException e) {

e.printStackTrace();

}

}

index -= 1;

int num = nums[index];

return num;

}

@Override

public void run() {

}

}

class Print implements Runnable {

private Store store;

Print(Store s) {

store = s;

}

@Override

public void run() {

while (true) {

int num = store.dispatch();

System.out.println(num);

try {

Thread.sleep(10);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}

}

public class NumberWorker {

public static void main(String[] args) {

Store store = new Store();

Sum sum = new Sum(store);

Print print = new Print(store);

new Thread(store).start();

new Thread(sum).start();

new Thread(print).start();

}

}

实验二

public class Produce implements Runnable {

private SyncStack syncStack;

private Controller controller;

public Produce(SyncStack syncStack, Controller controller) {

this.syncStack = syncStack;

this.controller = controller;

}

@Override

public void run() {

char ca;

for(int i=0;i<30;i++){

ca=(char)(Math.random()\*26+'a');

syncStack.push(ca);

System.out.println("生产者生产："+ca+" ");

controller.counter();

try {

Thread.sleep((int)(Math.random()\*200));

}

catch (InterruptedException e){

e.printStackTrace();

}

}

·控制输出格式：

public class Controller {

//控制输出格式

public static int count;

void counter(){

count++;

if(count==5){

System.out.println();

count=0;

}

}

}

·SyncStack类：

import java.util.Vector;

public class SyncStack {

private Vector buffer = new Vector(400, 200);

public synchronized char pop() {

while (buffer.size() == 0) {

try {

this.wait();

} catch (InterruptedException e) {

e.printStackTrace();

}

}

return (Character)buffer.remove(buffer.size()-1);

}

public synchronized void push(char ca){

this.notify();

Character character=new Character(ca);

buffer.addElement(character);

}

}

·消费者类：

public class Consumer implements Runnable {

private SyncStack syncStack;

private Controller controller;

public Consumer(SyncStack syncStack, Controller controller) {

this.syncStack = syncStack;

this.controller = controller;

}

@Override

public void run() {

for(int i=0;i<30;i++){

System.out.println("消费者消费："+syncStack.pop()+" ");

controller.counter();

try {

Thread.sleep((int)(Math.random()\*2000));

}catch (InterruptedException e){

e.printStackTrace();

}

}

}

}

|  |  |
| --- | --- |
| **成绩** |  |

**《程序设计4（Java平台）》**

**实验报告**

**班级： 13901801**

**姓名： 何庚坤**

**学号： 2018214139**

**指导老师： 马创**

**实验名称： java平台实验五**

**实验地点：**

1. **实验目的**

1、熟悉 Java 流式 Socket 通信的程序结构与方法;

2、熟悉 Java 数据报 Socket 通信的程序结构与方法。

1. **实验内容**

1、熟悉 Java 流式 Socket 通信的程序结构与方法

端口号和 IP 地址的组合称为网络套接字(Socket)。

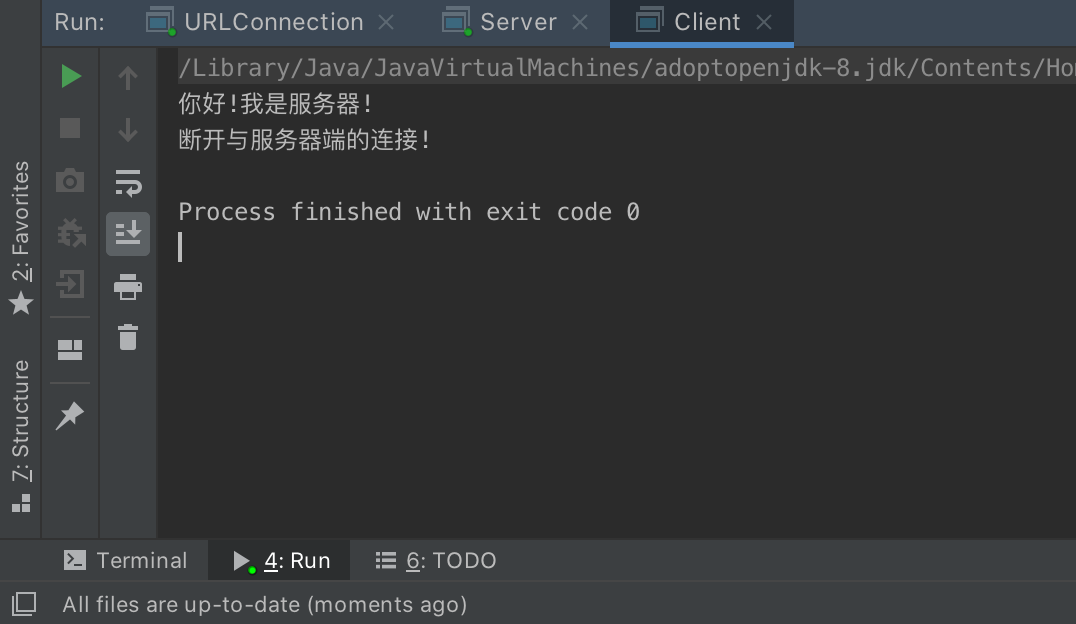
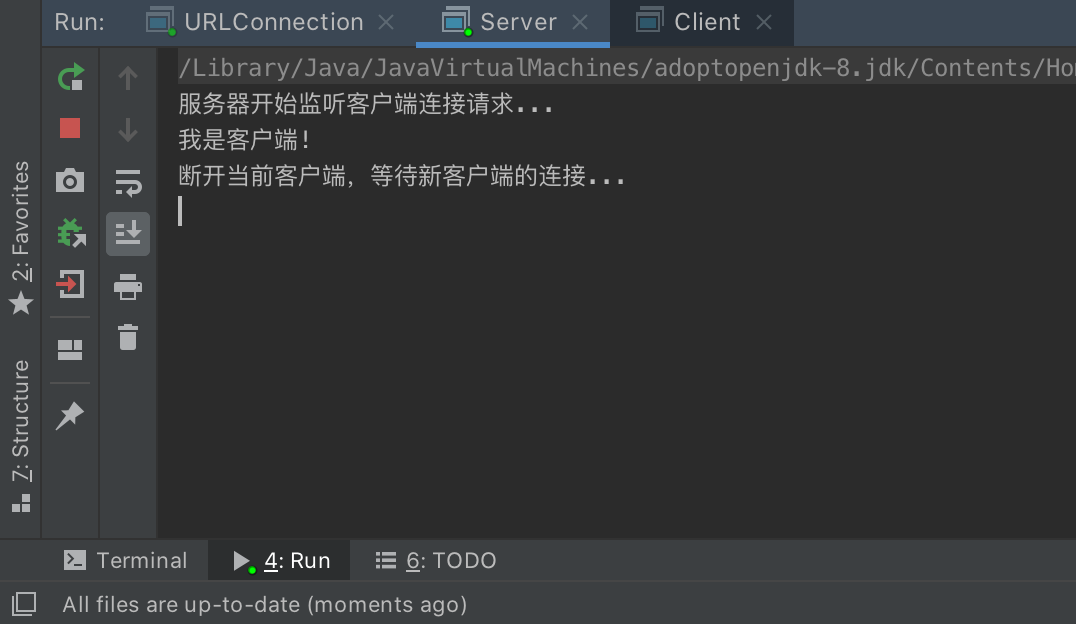
流式 socket:为有连接通信，即 TCP 协议。每次通信前建立连接，结束后断开连接。 可以保证传输的正确性、可靠性。其主要使用的两个类是:

⚫ServerSocket

⚫Socket  
流式 socket 在客户端和服务器间建立连接通信的通道。每个 socket 可以进行读和写两

种操作。对于任一端，与双方的通信会话过程如下: 建立socket连接，获得输入输出流对象，读写数据，通信完成后关闭socke(t 拆除连接)。

1. **实验结果与分析**



1. **核心代码清单**

**import** javax.swing.\*;  
**import** java.awt.\*;  
**import** java.awt.event.ActionEvent;  
**import** java.awt.event.WindowEvent;  
**import** java.io.IOException;  
**import** java.net.DatagramPacket;  
**import** java.net.DatagramSocket;  
**import** java.net.InetAddress;  
**import** java.net.SocketException;  
  
**public class** CommunicationClient **extends** JFrame **implements** Runnable {  
 JPanel **contentPane**;  
 JLabel **jLabel1** = **new** JLabel();  
 TextArea **jTextArea1** = **new** TextArea(**"你好! "**, 100, 250);  
 JLabel **jLabel2** = **new** JLabel();  
 JTextField **jTextField1** = **new** JTextField();  
 Thread **c**;  
 **private** DatagramSocket **sendSocket**, **receiveSocket**;  
 **private** DatagramPacket **sendPacket**, **receivePacket**;  
 **public** CommunicationClient() {  
 **try** {  
 jbInit();  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 **private void** jbInit() **throws** Exception {  
 **contentPane** = (JPanel) **this**.getContentPane();  
 **contentPane**.setLayout(**null**);  
 **this**.setSize(**new** Dimension(400, 200));  
 **this**.setTitle(**"UDPCLient"**);  
 **jLabel1**.setText(**"通信记录:"**);  
 **jLabel1**.setBounds(**new** Rectangle(16, 9, 68, 27));  
 **contentPane**.setLayout(**null**);  
 **jTextArea1**.setBounds(**new** Rectangle(15, 28, 349, 89));  
 **jTextArea1**.setEditable(**false**);  
 **jLabel2**.setText(**"输入通信内容:"**);  
 **jLabel2**.setBounds(**new** Rectangle(17, 125, 92, 37)); *//创建内容区域* **jTextField1**.setText(**"通信内容"**);  
 **jTextField1**.setBounds(**new** Rectangle(127, 129, 244, 31));  
 **jTextField1**.setEditable(**true**);  
 **jTextField1**.addActionListener(**new** java.awt.event.ActionListener() {  
 **public void** actionPerformed(ActionEvent e) {  
 jTextField1\_actionPerformed(e);  
 }  
 });  
 **contentPane**.add(**jLabel1**, **null**);  
 **contentPane**.add(**jTextArea1**, **null**);  
 **contentPane**.add(**jTextField1**, **null**);  
 **contentPane**.add(**jLabel2**, **null**);  
 **try** {  
 **sendSocket** = **new** DatagramSocket();  
 **receiveSocket** = **new** DatagramSocket(8000);  
 } **catch** (SocketException e) {  
 **jTextArea1**.append(**"不能打开数据报Socket,或者数据报Socket无法与指定端口连接！"**);  
 }  
 **c** = **new** Thread(**this**); *//创建一个线程* **c**.start();  
 }  
 **public void** run() {  
 **while** (**true**) {  
 **try** {  
 **byte** buf[] = **new byte**[100];  
 **receivePacket** = **new** DatagramPacket(buf, buf.**length**);  
 **receiveSocket**.receive(**receivePacket**);  
 **jTextArea1**.append(**"\n服务器"**);  
 **byte**[] data = **receivePacket**.getData();  
 String receiveString = **new** String(data);  
 **jTextArea1**.append(receiveString);  
 } **catch** (IOException e) {  
 **jTextArea1**.append(**"网络通信出现错误,问题在于"** + e.toString());  
 }  
 }  
 }  
 **protected void** processWindowEvent(WindowEvent e) {  
 **super**.processWindowEvent(e);  
 **if** (e.getID() == WindowEvent.***WINDOW\_CLOSING***) {  
 System.*exit*(0);  
 }  
 }  
 **void** jTextField1\_actionPerformed(ActionEvent e) {  
 **try** {  
 **jTextArea1**.append(**"\n客户端:"**);  
 String string1 = **jTextField1**.getText().trim();  
 **jTextArea1**.append(string1);  
 **byte**[] databyte = **new byte**[100];  
 string1.getBytes(0, string1.length(), databyte, 0);  
 DatagramPacket sendPacket = **new** DatagramPacket(databyte, string1.length(), InetAddress.*getLocalHost*(), 6666);  
 **sendSocket**.send(sendPacket);  
 } **catch** (IOException ioe) {  
 **jTextArea1**.append(**"网络通信出现错误,问题在于"** + e.toString());  
 }  
 }  
}

**import** java.awt.\*;  
**import** java.awt.event.\*;  
**import** javax.swing.\*;  
**import** java.net.\*;  
**import** java.io.\*;  
**class** CommunicationServer **extends** JFrame **implements** Runnable {  
 JPanel **contentPane**;  
 JLabel **jLabel1** = **new** JLabel(); *//创建图形用户界面* JLabel **jLabel2** = **new** JLabel();  
 **public** JTextField **jTextField1** = **new** JTextField();  
 **public** JTextArea **jTextArea1** = **new** JTextArea(**"你好！ "**, 100, 50);  
 Thread **s**;  
 **private** DatagramSocket **sendSocket**, **receiveSocket**;  
 **private** DatagramPacket **sendPacket**, **receivePacket**;  
 **private** String **name**;  
 **public** CommunicationServer() {  
 enableEvents(AWTEvent.***WINDOW\_EVENT\_MASK***);  
 **try** {  
 jbInit();  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 **private void** jbInit() **throws** Exception {  
 **jLabel1**.setText(**"通信记录:"**);  
 **jLabel1**.setBounds(**new** Rectangle(4, 4, 67, 29));  
 **contentPane** = (JPanel) **this**.getContentPane();  
 **contentPane**.setLayout(**null**);  
 **this**.setSize(**new** Dimension(400, 200));  
 **this**.setTitle(**"UDPServer"**);  
 **jLabel2**.setText(**"输入通信内容:"**);  
 **jLabel2**.setBounds(**new** Rectangle(11, 124, 93, 32));  
 **jTextField1**.setText(**"通信内容"**);  
 **jTextField1**.setBounds(**new** Rectangle(118, 125, 269, 30));  
 **jTextField1**.addActionListener(**new** java.awt.event.ActionListener() {  
 **public void** actionPerformed(ActionEvent e) {  
 jTextField1\_actionPerformed(e);  
 }  
 });  
 **jTextArea1**.setBounds(**new** Rectangle(97, 29, 380, 90));  
 **jTextArea1**.setEditable(**false**);  
 **jTextField1**.setEditable(**true**);  
 **contentPane**.add(**jLabel1**, **null**);  
 **contentPane**.add(**jTextArea1**, **null**);  
 **contentPane**.add(**jTextField1**, **null**);  
 **contentPane**.add(**jLabel2**, **null**);  
 **try** {  
 **sendSocket** = **new** DatagramSocket(); *//创建接受用数据报* **receiveSocket** = **new** DatagramSocket(6666);  
 } **catch** (SocketException e) {  
 e.printStackTrace();  
 System.*exit*(1);  
 }  
 **s** = **new** Thread(**this**); *//创建线程* **s**.start();  
 }  
  
 **public void** run() {  
 **while** (**true**) {  
 **try** {  
 **byte** buf[] = **new byte**[100];  
 **receivePacket** = **new** DatagramPacket(buf, buf.**length**);  
 **receiveSocket**.receive(**receivePacket**);  
 **name** = **receivePacket**.getAddress().toString().trim();  
 **jTextArea1**.append(**"\n来自主机:"** + **name** + **"\n端口:"** + **receivePacket**.getPort());  
 **jTextArea1**.append(**"\n客户端:\t"**);  
 **byte**[] data = **receivePacket**.getData();  
 String receivedString = **new** String(data, 0);  
 **jTextArea1**.append(receivedString);  
 } **catch** (IOException e) {  
 **jTextArea1**.append(**"网络通信出现错误,问题在于"** + e.toString());  
 }  
 }  
 }  
 **protected void** processWindowEvent(WindowEvent e) {  
 **super**.processWindowEvent(e);  
 **if** (e.getID() == WindowEvent.***WINDOW\_CLOSING***) {  
 System.*exit*(0);  
 }  
 }  
 **void** jTextField1\_actionPerformed(ActionEvent e) {  
 **try** {  
 **jTextArea1**.append(**"\n服务器:"**);  
 String string = **jTextField1**.getText().trim();  
 **jTextArea1**.append(string);  
 **byte**[] databyte = **new byte**[100];  
 string.getBytes(0, string.length(), databyte, 0);  
 DatagramPacket sendPacket = **new** DatagramPacket(databyte, string.length(), InetAddress.*getLocalHost*(), 8000);  
 **sendSocket**.send(sendPacket);  
 } **catch** (IOException ioe) {  
 **jTextArea1**.append(**"网络通信出现错误,问题在于"** + e.toString());  
 }  
 }  
}  
 **class** UDPCommunicationServer {  
 **public static void** main(String[] args) {  
 CommunicationServer frame1 = **new** CommunicationServer();  
 frame1.setVisible(**true**);  
 }  
}

**public class** UDPClientTest{  
 **public static void** main(String[] args) {  
 CommunicationClient frame1 = **new** CommunicationClient();  
 frame1.setVisible(**true**);  
 }  
}

|  |  |
| --- | --- |
| **成绩** |  |

**《程序设计4（Java平台）》**

**实验报告**

**班级： 13901801**

**姓名： 何庚坤**

**学号： 2018214139**

**指导老师： 马创**

**实验名称： java平台实验六**

**实验地点：**

1. **实验目的**

1、熟悉 JDBC 相关类与接口的编程规则;

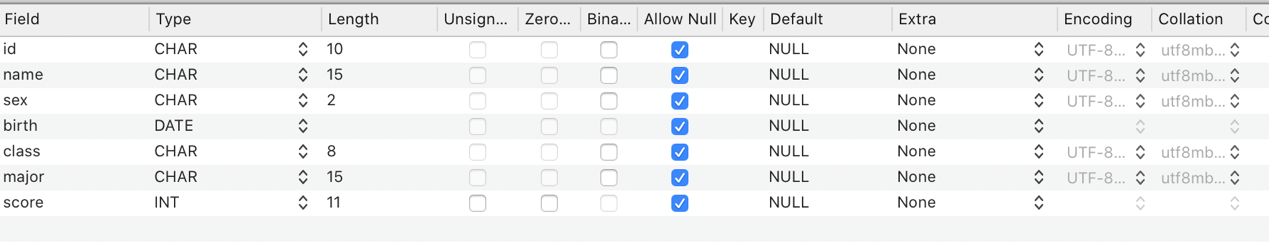
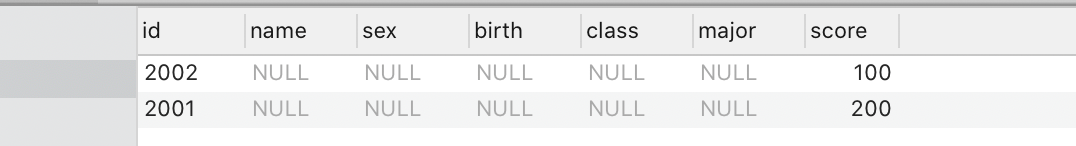
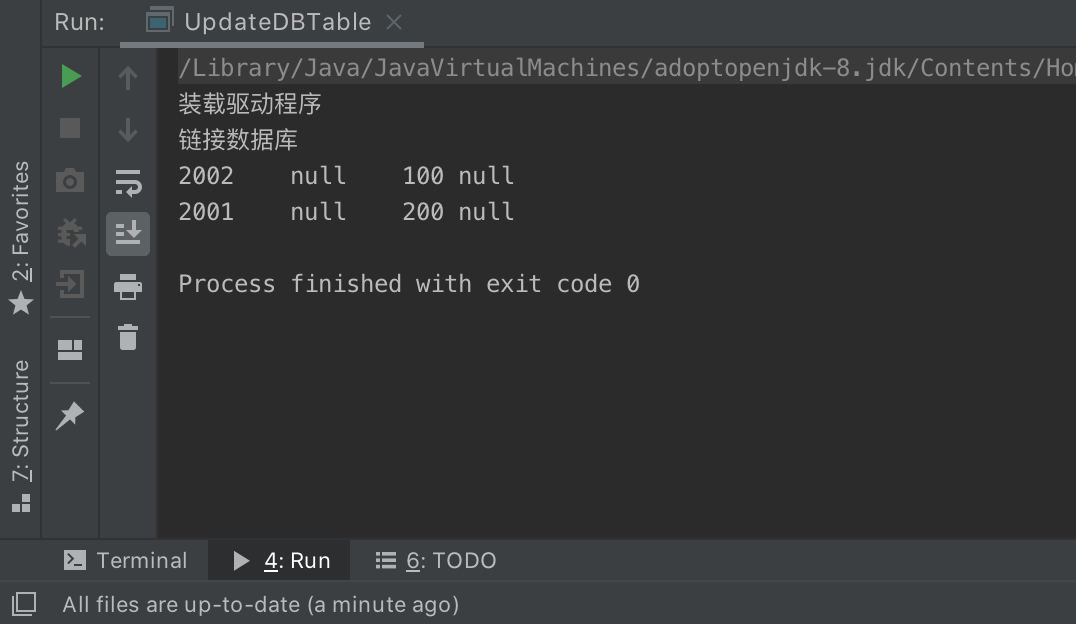
2、熟悉 mysql 数据库管理软件之下进行表设计;

3、熟悉与掌握 JDBC 进行数据库的增删改查操作步骤与接口规范。

1. **实验内容**

实现教材 P63 10.2、10.4 综合应用举例中的程序功能

1. **实验结果与分析**



1. **核心代码清单**

创建DB链接单例

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

public class DB {

private static Connection conn = null;

static {

String url = "jdbc:mysql://localhost:3306/java\_jdbc?useSSL=false";

try {

Class.forName("com.mysql.jdbc.Driver");

} catch (ClassNotFoundException e) {

e.printStackTrace();

}

System.out.println("装载驱动程序");

try {

conn = DriverManager.getConnection(url, "root", "password123");

} catch (SQLException e) {

e.printStackTrace();

}

System.out.println("链接数据库");

}

public static Connection getConnection(){

return conn;

}

}

建表

import java.sql.\*;

public class CreateDBTable {

public static void main(String[] args) throws Exception {

Connection conn = DB.getConnection();

Statement stat = conn.createStatement();

String query = "create table student(id char(10), name char(15), sex char(2), birth date, class char(8), major char(15), score integer)";

stat.executeUpdate(query);

ResultSet rs = stat.executeQuery("select \* from student");

System.out.println("查询结果为：");

while (rs.next()) {

System.out.println(rs.getString(1) + "\t" + rs.getString(2) + "\t" + rs.getString(3) + "\t" + rs.getString(4) + "\t" + rs.getString(5) + "\t" + rs.getString(6) + "\t" + rs.getString(7));

}

rs.close();

stat.close();

conn.close();

}

}

更新表

import java.sql.\*;

public class UpdateDBTable {

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

Connection conn = DB.getConnection();

PreparedStatement ps = conn.prepareStatement("update student set score=? where id=?");

String[] studentIds = {"2002", "2001"};

int[] scores = {100, 200};

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

ps.setInt(1, scores[i]);

ps.setString(2, studentIds[i]);

ps.execute();

}

ps.close();

Statement stat = conn.createStatement();

ResultSet rs = stat.executeQuery("select \* from student");

while (rs.next()) {

System.out.println(rs.getString("id") + "\t" + rs.getString("name") + "\t" + rs.getInt("score") + "\t" + rs.getDate("birth"));

}

stat.close();

conn.close();

}

}

|  |  |
| --- | --- |
| **成绩** |  |

**《程序设计4（Java平台）》**

**实验报告**

**班级： 13901801**

**姓名： 何庚坤**

**学号： 2018214139**

**指导老师： 马创**

**实验名称： java平台实验七**

**实验地点：**

1. **实验目的**

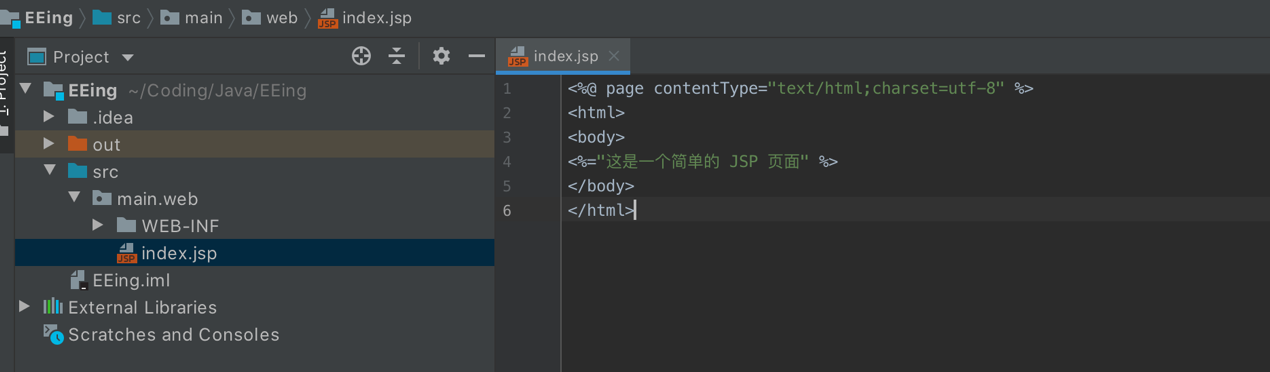
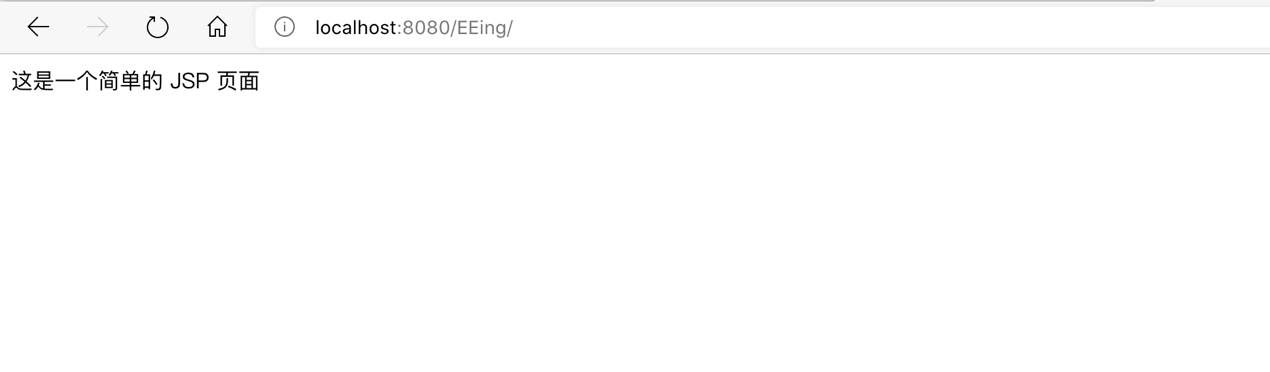
1、熟悉开发环境:安装 Tomcat，安装 Eclipse 下的 Tomcat 插件

2、调试一个 JSP helloworld 程序。

1. **实验内容**

配置tomcat服务器，编写第一个helloworld jsp 页面

1. **实验结果与分析**

****

1. **核心代码清单**

<%@ page contentType="text/html;charset=utf-8" %>  
<html>  
<body>  
<%="这是一个简单的 JSP 页面" %>  
</body>  
</html>

|  |  |
| --- | --- |
| **成绩** |  |

**《程序设计4（Java平台）》**

**实验报告**

**班级： 13901801**

**姓名： 何庚坤**

**学号： 2018214139**

**指导老师： 马创**

**实验名称： java平台实验八**

**实验地点：**

1. **实验目的**

1、熟悉 JSP 基础语法:java 程序片、表达式、jsp 动作标签等;

2、在 JSP 页面上输出当前时间信息;

3、通过 forward 动作传值到另一个页面。

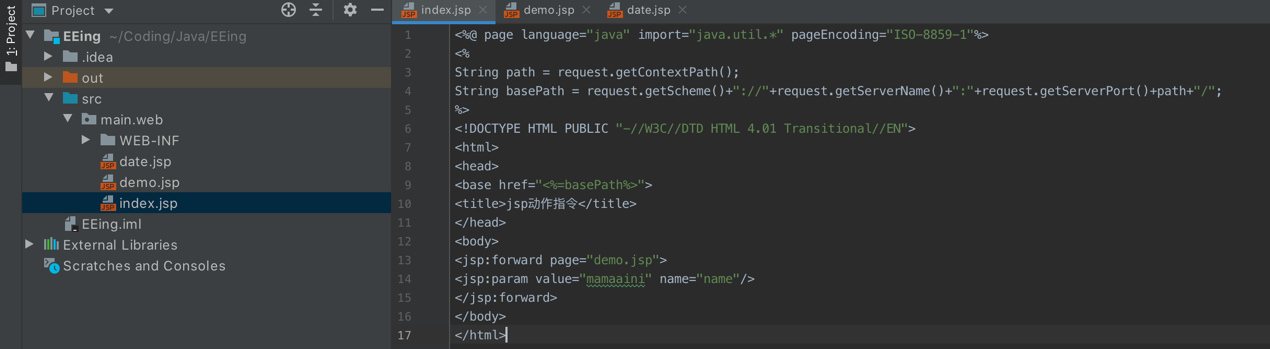
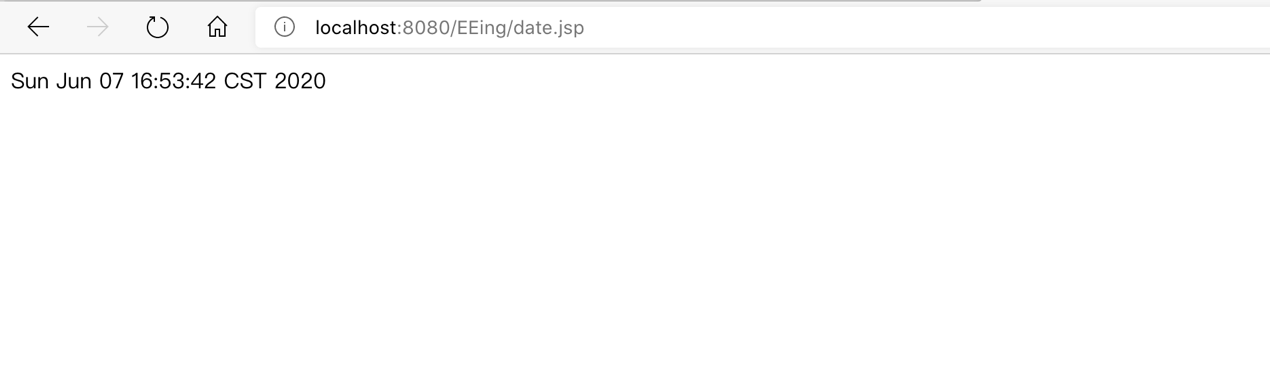
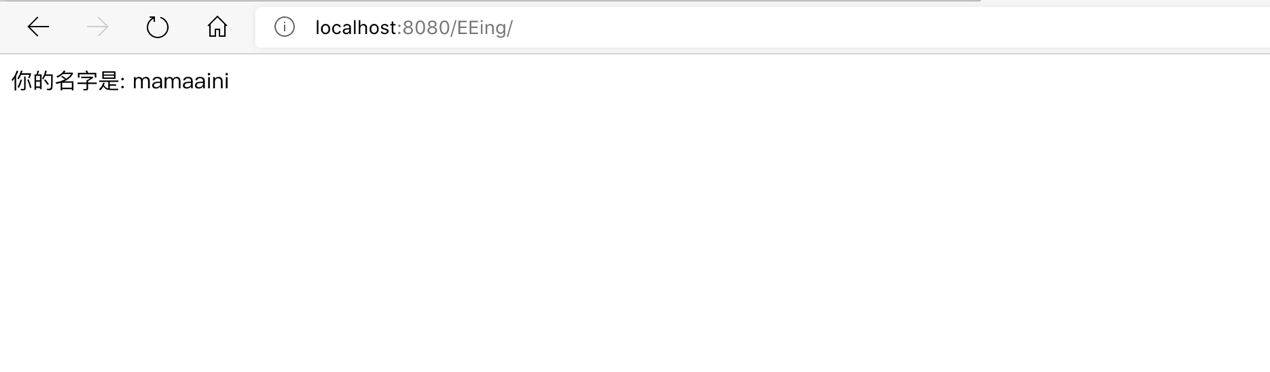
1. **实验内容**

1、显示当前时间的页面(date.jsp)

2、拥有跳转指令的页面(index.jsp)

3、跳转到的目的资源(demo.jsp)

1. **实验结果与分析**



1. **核心代码清单**

**Index.jsp**

<%@ page language="java" import="java.util.\*" pageEncoding="ISO-8859-1"%>  
<%  
String path = request.getContextPath();  
String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";  
%>  
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">  
<html>  
<head>  
<base href="<%=basePath%>">  
<title>jsp动作指令</title>  
</head>  
<body>  
<jsp:forward page="demo.jsp">  
<jsp:param value="mamaaini" name="name"/>  
</jsp:forward>  
</body>  
</html>

Demo.jsp

<%@ page language="java" import="java.util.\*" pageEncoding="UTF-8"%>  
<%  
String path = request.getContextPath();  
String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";  
%>  
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">  
<html>  
<head>  
<base href="<%=basePath%>">  
<title>目的地址</title>  
</head>  
<body>  
你的名字是:  
<%=request.getParameter("name") %>  
</body>  
</html>

Date.jsp

<%@ page language="java" import="java.util.\*" pageEncoding="ISO-8859-1"%>  
<%  
String path = request.getContextPath();  
String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";  
%>  
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">  
<html>  
<body>  
<%  
String date = new Date().toString();  
out.print(date);  
%>  
</body>  
</html>

|  |  |
| --- | --- |
| **成绩** |  |

**《程序设计4（Java平台）》**

**实验报告**

**班级： 13901801**

**姓名： 何庚坤**

**学号： 2018214139**

**指导老师： 马创**

**实验名称： java平台实验九**

**实验地点：**

1. **实验目的**

1、熟悉 JSP 常用内置对象 request 和 response 的意义与功能;  
2、掌握 request 和 response 的常用方法;  
3、实现登录功能，共有 login.htm、check.jsp、ok.jsp、error.jsp 页面，熟悉 request 和 response 对象的使用。  
4、观察 post 和 get 方式的不同、forward 动作和 response 重定向的不同。

1. **实验内容**

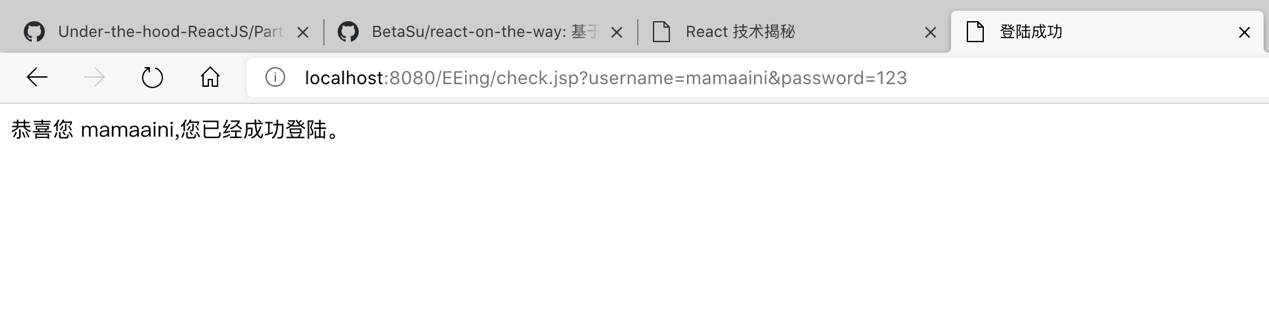
1、登陆页面(login.jsp)

2、验证页面(check.jsp)

3、登录成功页面(ok.jsp)

4、登录失败页面(error.jsp)

1. **实验结果与分析**



1. **核心代码清单**

Login.jsp

<%@ page language="java" import="java.util.\*" pageEncoding="UTF-8"%>  
<%  
String path = request.getContextPath();  
String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";  
%>  
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">  
<html>  
<head>  
<base href="<%=basePath%>">  
<title>登陆验证</title>  
</head>  
<body>  
<form action="check.jsp" method="get">  
请输入您的用户名:<input type="text" name="username"><br><br>  
请输入您的密码:<input type="password" name="password"><br>  
<input type="submit" value="提交">  
</form>  
</body>  
</html>

Check.jsp

<%@ page language="java" import="java.util.\*" pageEncoding="UTF-8"%>  
<%  
String path = request.getContextPath();  
String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";  
%>  
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">  
<html>  
<head>  
<base href="<%=basePath%>">  
<title>验证</title>  
</head>  
<body>  
<%  
String username = request.getParameter("username");  
String password = request.getParameter("password");  
String name="mamaaini";  
String pass="123";  
if(name.equals(username)&&pass.equals(password)){  
%>  
<jsp:forward page="ok.jsp">  
<jsp:param value="<%=username %>" name="name"/></jsp:forward>  
<%  
}  
else{  
%>  
<jsp:forward page="error.jsp"></jsp:forward>  
<% }%>  
</body>  
</html>

Ok.jsp

<%@ page language="java" import="java.util.\*" pageEncoding="UTF-8"%>  
<%  
String path = request.getContextPath();  
String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";  
%>  
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">  
<html>  
<head>  
<base href="<%=basePath%>">  
<title>登陆成功</title>  
</head>  
<body>  
恭喜您  
<%out.print(request.getParameter("name"));%>,您已经成功登陆。  
</body>  
</html>

Error.jsp

<%@ page language="java" import="java.util.\*" pageEncoding="UTF-8"%>  
<%  
String path = request.getContextPath();  
String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";  
%>  
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">  
<html>  
<head>  
<base href="<%=basePath%>">  
<title>My JSP 'index.jsp' starting page</title>  
<!--  
<link rel="stylesheet" type="text/css" href="styles.css">  
-->  
</head>  
<body> 抱歉，您登陆失败。 </body>  
</html>

|  |  |
| --- | --- |
| **成绩** |  |

**《程序设计4（Java平台）》**

**实验报告**

**班级： 13901801**

**姓名： 何庚坤**

**学号： 2018214139**

**指导老师： 马创**

**实验名称： java平台实验十**

**实验地点：**

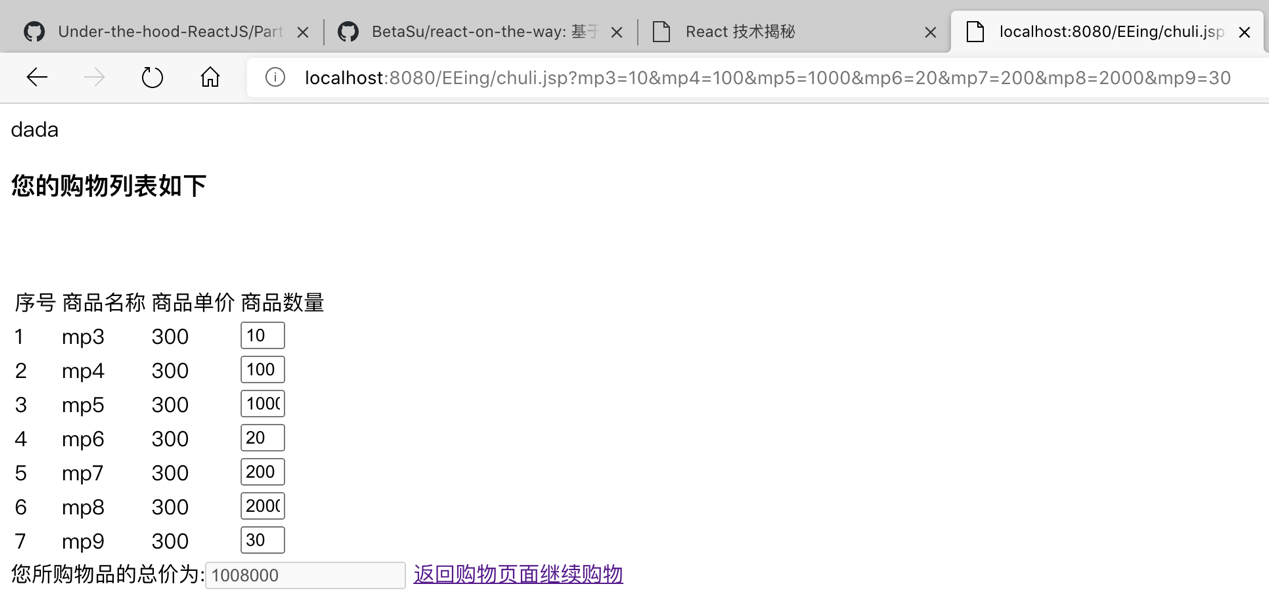
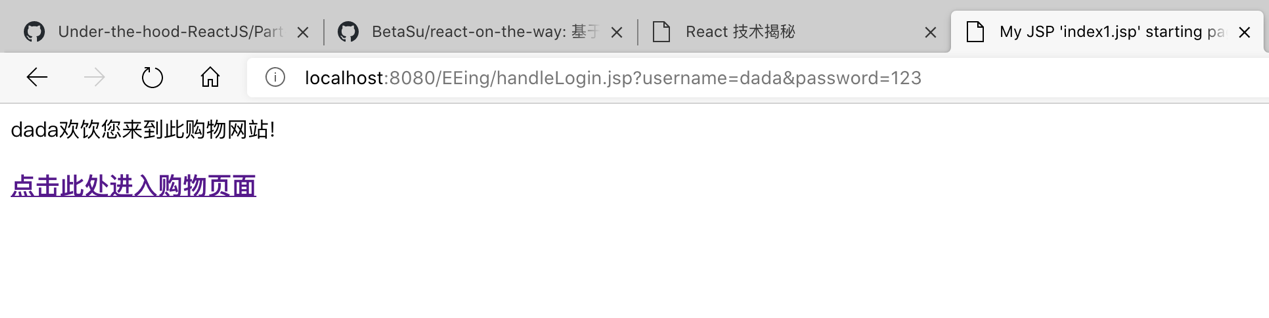
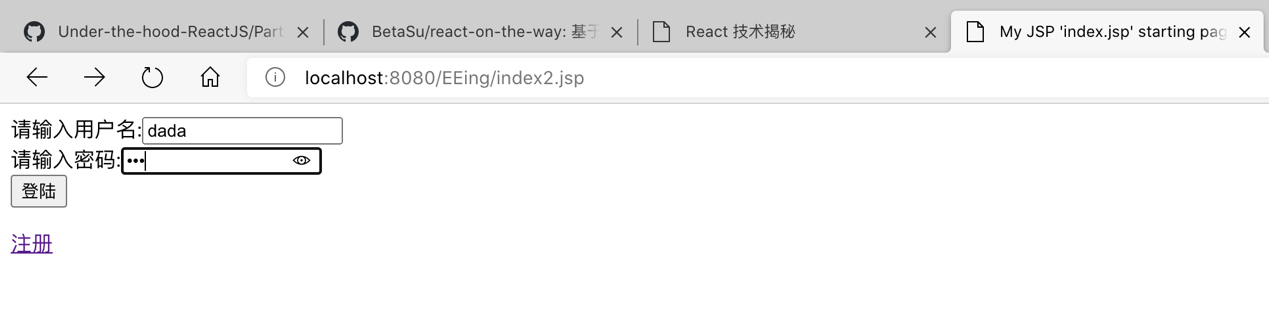
1. **实验目的**

1、熟悉 JSP 常用内置对象 session 与 application 的意义与功能;  
2、掌握 session 与 application 的常用方法;  
3、通过 session 与 application 对象模拟购物车功能。

1. **实验内容**

用户登录后，进入购物页面， 选择商品与数量，然后点击“放入购物车”按钮，所选商品存入购物车，页面还是在购物页 面。当用户点击其上超链接“结算中心”时，跳转到结算中心页面。该页面显示用户目前为 止所购买的商品名称、数量、总价等信息。

1. **实验结果与分析**



1. **核心代码清单**

Index2.jsp

<%@ page language="java" import="java.util.\*" pageEncoding="UTF-8"%>  
<%  
String path = request.getContextPath();  
String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";  
%>  
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">  
<html>  
<head>  
<base href="<%=basePath%>">  
<title>My JSP 'index.jsp' starting page</title>  
<!--  
<link rel="stylesheet" type="text/css" href="styles.css">  
-->  
</head>  
<body>  
<form action="handleLogin.jsp">  
请输入用户名:<input type="text" name="username"><br>  
请输入密码:<input type="password" name="password"><br>  
<input type="submit" value="登陆">  
</form>  
<a href="reg.jsp">注册</a>  
</body>  
</html>

Reg.jsp

<%@ page language="java" import="java.util.\*" pageEncoding="UTF-8"%>  
<%  
String path = request.getContextPath();  
String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";  
%>  
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">  
<html>  
<head>  
<base href="<%=basePath%>">  
<title>My JSP 'gouwu.jsp' starting page</title>  
<!--  
<link rel="stylesheet" type="text/css" href="styles.css">  
-->  
</head>  
<body>  
<form action="handleReg.jsp">  
用户名<input type="text" name="username"><br>  
密码<input type="password" name="password"><br>  
<input type="submit" value=”注册”>  
</form>  
</body>  
</html>

handleReg.jsp

<%@ page language="java" import="java.util.\*" pageEncoding="UTF-8"%>  
<%@ page language="java" import="com.People.People" %>  
<%  
String path = request.getContextPath();  
String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";  
%>  
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">  
<html>  
<head>  
<base href="<%=basePath%>">  
<title>My JSP 'gouwu.jsp' starting page</title>  
<!--  
<link rel="stylesheet" type="text/css" href="styles.css">  
-->  
</head>  
<body>  
<%!  
ArrayList arraylist=new ArrayList();  
%>  
<%  
String username=(String)request.getParameter("username");  
String password=(String)request.getAttribute("password");  
People people=new People();  
if (application.getAttribute("list")==null) {  
 people.setUsername(username);  
 people.setPassword(password);  
 arraylist.add(people);  
 application.setAttribute("list",arraylist);  
} else{  
 arraylist=(ArrayList)application.getAttribute("list");  
 Iterator itr=arraylist.iterator();  
 while (itr.hasNext()) {  
 People people1=(People)itr.next();  
 if (people1.getUsername().equals(username)) {  
%>  
 <jsp:forward page="reg.jsp"></jsp:forward>  
<%  
 }  
}  
People people2=new People();  
people2.setUsername(username);  
people2.setPassword(password);  
arraylist.add(people2);  
application.setAttribute("list",arraylist);  
}  
%>  
</body>  
</html>

handleLogin.jsp

<%@ page language="java" import="java.util.\*" pageEncoding="UTF-8"%>  
<%  
String path = request.getContextPath();  
String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";  
%>  
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"> <html>  
<head>  
<base href="<%=basePath%>">  
<title>My JSP 'index1.jsp' starting page</title>  
<!--  
<link rel="stylesheet" type="text/css" href="styles.css">  
-->  
<script type="text/javascript">  
function demo(){  
}  
</script>  
</head>  
<body>  
<%  
String username=request.getParameter("username");  
String password=request.getParameter("password");  
ArrayList arraylist=(ArrayList)application.getAttribute("list");  
session.setAttribute("username",username);  
out.print(username+"欢饮您来到此购物网站!");  
%>  
<a href="gouwu.jsp"><h3>点击此处进入购物页面<h3></a>  
<br><br>  
</body>  
</html>

Gouwu.jsp

<%@ page language="java" import="java.util.\*" pageEncoding="UTF-8"%>  
<%  
String path = request.getContextPath();  
String basePath =  
request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";  
%>  
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">  
<html>  
<head>  
<base href="<%=basePath%>">  
<title>My JSP 'gouwu.jsp' starting page</title>  
<!--  
<link rel="stylesheet" type="text/css" href="styles.css">  
-->  
</head>  
<body>  
<%  
HashMap map=null;  
map=(HashMap)session.getAttribute("cart");  
if (map==null) {  
 map=new HashMap();  
 map.put("mp3",0);  
 map.put("mp4",0);  
 map.put("mp5",0);  
 map.put("mp6",0);  
 map.put("mp7",0);  
 map.put("mp8",0);  
 map.put("mp9",0);  
}  
%>  
<form action="chuli.jsp">  
<table>  
<tr>  
<td>序号</td>  
<td>商品名称</td>  
<td>商品单价</td>  
<td>商品数量</td>  
</tr> <tr><td>1</td><td>mp3</td><td>300</td><td><input  
value="<%=map.get("mp3")%>" size="3" type="text" name="mp3"></td></tr> <tr><td>2</td><td>mp4</td><td>300</td><td><input  
value="<%=map.get("mp4")%>" size="3" type="text" name="mp4"></td></tr> <tr><td>3</td><td>mp5</td><td>300</td><td><input  
value="<%=map.get("mp5")%>" size="3" type="text" name="mp5"></td></tr> <tr><td>4</td><td>mp6</td><td>300</td><td><input  
value="<%=map.get("mp6")%>" size="3" type="text" name="mp6"></td></tr> <tr><td>5</td><td>mp7</td><td>300</td><td><input  
value="<%=map.get("mp7")%>" size="3" type="text" name="mp7"></td></tr> <tr><td>6</td><td>mp8</td><td>300</td><td><input  
value="<%=map.get("mp8")%>" size="3" type="text" name="mp8"></td></tr> <tr><td>7</td><td>mp9</td><td>300</td><td><input  
value="<%=map.get("mp9")%>" size="3" type="text" name="mp9"></td></tr> </table>  
<input type="submit" value="提交">  
</form>  
<a href="list.jsp">查看商品列表</a>  
</body>  
</html>

Chuli.jsp

<%@ page language="java" import="java.util.\*" pageEncoding="UTF-8"%>  
<%  
String path = request.getContextPath();  
String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";  
%>  
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"> <html>  
<head>  
<base href="<%=basePath%>">  
<title>My JSP 'index.jsp' starting page</title>  
<meta http-equiv="pragma" content="no-cache">  
<meta http-equiv="cache-control" content="no-cache">  
<meta http-equiv="expires" content="0">  
<meta http-equiv="keywords" content="keyword1,keyword2,keyword3"> <meta http-equiv="description" content="This is my page">  
<!--  
<link rel="stylesheet" type="text/css" href="styles.css">  
-->  
</head>  
<body>  
<%!  
HashMap map=new HashMap(); int mp3=0;  
int mp4=0; int mp5=0; int mp6=0; int mp7=0; int mp8=0; int mp9=0;  
%> <%  
if(request.getParameter("mp3")==null){ mp3=0;  
}  
  
else{ mp3=Integer.parseInt(request.getParameter("mp3"));  
} map.put("mp3",mp3);  
if(request.getParameter("mp4")==null) {  
mp4=0; }  
else{ mp4=Integer.parseInt(request.getParameter("mp4"));  
}  
map.put("mp4",mp4); if(request.getParameter("mp5")==null){  
mp5=0; }  
else{ mp5=Integer.parseInt(request.getParameter("mp5")); }  
map.put("mp5",mp5); if(request.getParameter("mp6")==null){  
mp6=0; }  
else{ mp6=Integer.parseInt(request.getParameter("mp6"));  
} map.put("mp6",mp6);  
if(request.getParameter("mp7")==null){ mp7=0;  
}  
else{ mp7=Integer.parseInt(request.getParameter("mp7")); }  
map.put("mp7",mp7);  
  
if(request.getParameter("mp8")==null){ mp8=0;  
}  
else{ mp8=Integer.parseInt(request.getParameter("mp8")); }  
map.put("mp8",mp8); if(request.getParameter("mp9")==null){  
mp9=0; }  
else{ mp9=Integer.parseInt(request.getParameter("mp9")); }  
map.put("mp9",mp9);  
session.setAttribute("cart",map); %>  
<jsp:forward page="list.jsp"/> </body>  
</html>

List.jsp

<%@ page language="java" import="java.util.\*" pageEncoding="UTF-8"%> <%  
String path = request.getContextPath();  
String basePath =  
request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";  
%>  
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"> <html>  
<head>  
<base href="<%=basePath%>">  
 </head>  
 <body>  
<%  
HashMap map = (HashMap)session.getAttribute("cart");  
out.print(session.getAttribute("username")+"<h3>"+"您的购物列表如下 "+"</h3>"+"<br><br>");  
%> <table>  
<tr>  
<td>序号</td>  
<td>商品名称</td>  
<td>商品单价</td>  
<td>商品数量</td>  
</tr> <tr><td>1</td><td>mp3</td><td>300</td><td><input  
value="<%=map.get("mp3")%>" size="3" type="text" name="mp3"></td></tr> <tr><td>2</td><td>mp4</td><td>300</td><td><input  
value="<%=map.get("mp4")%>" size="3" type="text" name="mp4"></td></tr> <tr><td>3</td><td>mp5</td><td>300</td><td><input  
value="<%=map.get("mp5")%>" size="3" type="text" name="mp5"></td></tr> <tr><td>4</td><td>mp6</td><td>300</td><td><input  
value="<%=map.get("mp6")%>" size="3" type="text" name="mp6"></td></tr> <tr><td>5</td><td>mp7</td><td>300</td><td><input  
value="<%=map.get("mp7")%>" size="3" type="text" name="mp7"></td></tr> <tr><td>6</td><td>mp8</td><td>300</td><td><input  
value="<%=map.get("mp8")%>" size="3" type="text" name="mp8"></td></tr> <tr><td>7</td><td>mp9</td><td>300</td><td><input  
value="<%=map.get("mp9")%>" size="3" type="text" name="mp9"></td></tr> </table>  
<%  
int  
total=300\*((Integer)map.get("mp3")+(Integer)map.get("mp4")+(Integer)map.get("mp5")+(Integer)map.get("mp6")+(Integer)map.get("mp7")+(Integer)map.get("mp8")+(Integer)map.get("mp9"));  
%>  
您所购物品的总价为:<input type="text" disabled="disabled" value="<%=total %>">  
  
<a href="gouwu.jsp">返回购物页面继续购物</a> </body>  
</html>

|  |  |
| --- | --- |
| **成绩** |  |

**《程序设计4（Java平台）》**

**实验报告**

**班级： 13901801**

**姓名： 何庚坤**

**学号： 2018214139**

**指导老师： 马创**

**实验名称： java平台实验十一**

**实验地点：**

1. **实验目的**

1、熟悉和了解 Servlet 与 JSP 编程引用的相同与不同;

2、熟悉 Servlet 编程步骤与编程技巧，包括 Servlet 类定义、Servlet 映射、Servlet 应用等;

3、使用 servlet 实验登录的案例。要求:

1. **实验内容**

(1)jsp 提交 form 到 servlet。

(2)servlet 得到提交信息，检查登录是否成功。若成功，则使用 session 记录用户 名，并转向到 index.jsp，并在之上显示用户的欢迎信息;若失败，则转向到 error.jsp 页 面。

(3)用户不能在未登录的情况下，访问 index.jsp 页面，如果直接访问，则直接转向 到 error.jsp 上。

1. **实验结果与分析**

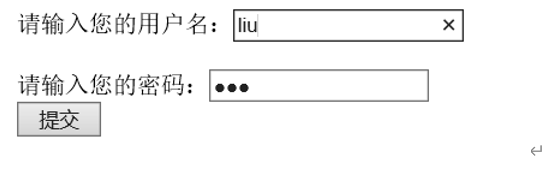
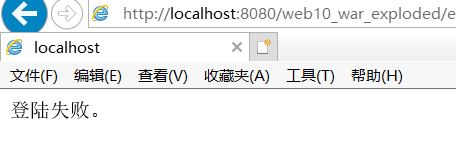
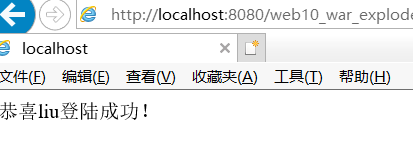


图11-1登录界面、



1. **核心代码清单**

<%@ **page language**= "**java**" **import**= "**java.util.\***" **pageEncoding**= "**UTF-8**" %>  
**<%** String path = request.getContextPath();  
 String basePath =  
 request.getScheme()+**"://"**+request.getServerName()+**":"**+request.getServerPort()  
 +path+**"/"**;  
**%>**<!DOCTYPE **HTML** PUBLIC **"-//W3C//DTD HTML 4.01 Transitional//EN"**>  
<**html**>  
<**head**>  
 <**base href="<%=**basePath**%>"**>  
 <**title**>登陆验证</**title**>  
 <**meta http-equiv= "pragma" content= "no - cache"** >  
 <**meta http-equiv= "cache - control" content= "no - cache"** >  
 <**meta http-equiv= "expires" content= "0"** >  
 <**meta http-equiv= "keywords" content= "keyword1,keyword2,keyword3"** >  
 <**meta http-equiv= "description" content= "This is my page"** >  
</**head**>  
<**body**>  
<**form action= " LoginServlet " method= "get"** >  
 请输入您的用户名：<**input type= "text" name= "username"** ><**br**><**br**>  
 请输入您的密码：<**input type= "password" name= "password"** ><**br**>  
 <**input type= "submit" value= " 提交 "** >  
</**form**>  
</**body**>  
</**html**>

<%@ **page language**= "**java**" **import**= "**java.util.\***" **pageEncoding**= "**UTF-8**" %>  
**<%** String path = request.getContextPath();  
 String basePath =  
 request.getScheme()+**"://"**+request.getServerName()+**":"**+request.getServerPort()  
 +path+**"/"**;  
**%>**<!DOCTYPE **HTML** PUBLIC **"-//W3C//DTD HTML 4.01 Transitional//EN"**>  
<**html**>  
<**head**>  
 <**base href="<%=**basePath**%>"**>  
</**head**>  
<**body**>  
登陆失败。  
</**body**>  
</**html**>

<%@ **page language**= "**java**" **import**= "**java.util.\***" **pageEncoding**= "**UTF-8**" %>  
**<%** String path = request.getContextPath();  
 String basePath = request.getScheme()+**"://"**+request.getServerName()+**":"**+request.getServerPort() +path+**"/"**;  
**%>**<!DOCTYPE **HTML** PUBLIC **"-//W3C//DTD HTML 4.01 Transitional//EN"**>  
<**html**>  
<**head**>  
 <**base href="<%=**basePath**%>"**>  
</**head**>  
<**body**>  
**<%** String username=(String)session.getAttribute(**"username"**);  
 **if**(username==**null**){  
**%>**<**jsp:forward page="error.jsp"**></**jsp:forward**>  
**<%** }**else**{  
**%>  
<%** out.print(**"恭喜"**+username+**"登陆成功！"**);  
 }  
**%>**</**body**>  
</**html**>

**import** java.io.IOException;  
**import** java.io.PrintWriter;  
**import** javax.servlet.ServletException;  
**import** javax.servlet.annotation.WebServlet;  
**import** javax.servlet.http.HttpServlet;  
**import** javax.servlet.http.HttpServletRequest;  
**import** javax.servlet.http.HttpServletResponse;  
**import** javax.servlet.http.HttpSession;  
@WebServlet(**"/LoginServlet"**)  
**public class** LoginServlet **extends** HttpServlet {  
 **public void** doGet(HttpServletRequest request, HttpServletResponse response)  
 **throws** ServletException, IOException {  
 String username=request.getParameter(**"username"**);  
 String password=request.getParameter(**"password"**);  
 **if**(username.equals(**"liu"**)&&password.equals(**"123"**)){  
 HttpSession session=request.getSession();  
 session.setAttribute(**"username"**,username);  
 session.setAttribute(**"password"**, password);  
  
 response.sendRedirect(request.getContextPath()+**"/index.jsp"**);  
 }  
 **else**{  
 response.sendRedirect(request.getContextPath()+**"/error.jsp"**);  
 }  
 }  
}

|  |  |
| --- | --- |
| **成绩** |  |

**《程序设计4（Java平台）》**

**实验报告**

**班级： 13901801**

**姓名： 何庚坤**

**学号： 2018214139**

**指导老师： 马创**

**实验名称： java平台实验十二**

**实验地点：**

1. **实验目的**

1、了解JavaBean的意义、编程规范；

2、熟悉和掌握一般JavaBean的编程技巧与应用；

3、熟悉javabean与jdbc的配合使用

1. **实验内容**

使用javabean操作数据库。要求：

（1）使用mysql，建立数据库表，表结构为：序号、姓名、学号、性别。

（2）在index.jsp中，实现两个功能：显示所有信息、按学号查询记录。其中，数据持久化操作使用javabean完成。

1. **实验结果与分析**

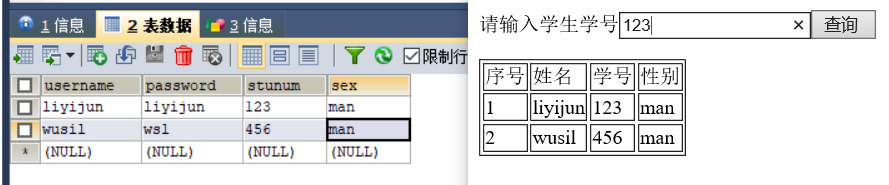


图12-1 表和首页

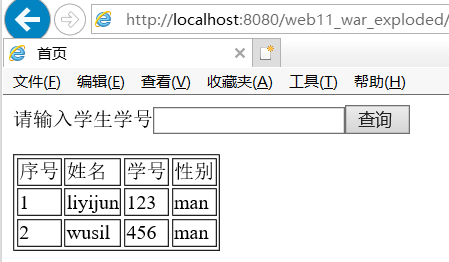
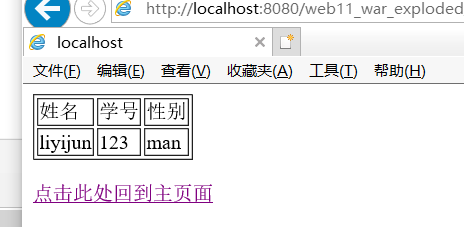


图12-2 首页



1. **核心代码清单**

<%@ **page language**= "**java**" **import**= "**java.util.\***" **pageEncoding**= "**UTF-8**" %>  
<%@ **page language**= "**java**" **import**= "**com.login.\***" %>  
<%@ **page language**= "**java**" **import**= "**com.demo.\***" %>  
**<%** String path = request.getContextPath();  
 String basePath =  
 request.getScheme()+**"://"**+request.getServerName()+**":"**+request.getServerPort()+path+**"/"**;  
**%>**<!DOCTYPE **HTML** PUBLIC **"-//W3C//DTD HTML 4.01 Transitional//EN"**>  
<**html**>  
<**head**>  
 <**base href="<%=**basePath**%>"**>  
 <**title**>首页</**title**>  
 <**meta http-equiv= "pragma" content= "no - cache"** >  
 <**meta http-equiv= "cache - control" content= "no - cache"** >  
 <**meta http-equiv= "expires" content= "0"** >  
 <**meta http-equiv= "keywords" content= "keyword1,keyword2,keyword3"** >  
 <**meta http-equiv= "description" content= "This is my page"** >  
</**head**>  
<**body**>  
<**form action= "findbyStunum"** >  
 请输入学生学号<**input type= "text" name= "stunum"** ><**input type= "submit" value= "  
查询 "** >  
</**form**>  
<**table border= "1"** >  
 <**tr**>  
 <**td**>序号</**td**>  
 <**td**>姓名</**td**>  
 <**td**>学号</**td**>  
 <**td**>性别</**td**>  
 </**tr**>  
 **<%** SqlBean sqlbean=**new** SqlBean();  
 **if**(sqlbean.getAll()!=**""**){  
 ArrayList<Student> stu=(ArrayList<Student>)sqlbean.getAll();  
 Iterator itr=stu.iterator();  
 Student stu1=**new** Student();  
 **int** i=0;  
 **while**(itr.hasNext()){  
 i++;  
 stu1=(Student)itr.next();  
 String name=stu1.getUsername();  
 String stunum=stu1.getStunum();  
 String sex=stu1.getSex();  
 **%>** <**tr**>  
 <**td**>**<%=**i **%>**</**td**>  
 <**td**>**<%=**name **%>**</**td**>  
 <**td**>**<%=**stunum **%>**</**td**>  
 <**td**>**<%=**sex **%>**</**td**>  
 </**tr**>  
 **<%** }  
 }  
 sqlbean.close();  
 **%>**</**table**>  
</**body**>  
</**html**>

**package** com.login;  
  
**import** java.io.IOException;  
**import** java.io.PrintWriter;  
**import** java.sql.SQLException;  
**import** javax.servlet.ServletException;  
**import** javax.servlet.annotation.WebServlet;  
**import** javax.servlet.http.HttpServlet;  
**import** javax.servlet.http.HttpServletRequest;  
**import** javax.servlet.http.HttpServletResponse;  
**import** com.demo.SqlBean;  
**import** com.demo.Student;  
@WebServlet(**"/findbyStunum"**)  
**public class** findbyStunum **extends** HttpServlet {  
 **public void** doGet(HttpServletRequest request, HttpServletResponse  
 response)  
 **throws** ServletException, IOException {  
 response.setContentType(**"text/html;charset=utf-8"**);  
 PrintWriter out = response.getWriter();  
 String stunum=request.getParameter(**"stunum"**);  
 **try** {  
 SqlBean bean=**new** SqlBean();  
 **if**(bean.getByStunum(stunum)!=**""**){  
 Student stu= (Student)bean.getByStunum(stunum);  
 out.print(**"<table border=\"1\">"** +**"<tr>"** +**"<td>姓名</td>"** +**"<td>学号</td>"** +**"<td>性别</td>"** +**"</tr>"**);  
 out.print(**"<tr>"**);  
 out.print(**"<td>"**+stu.getUsername()+**"</td>"**);  
 out.print(**"<td>"**+stu.getStunum()+**"</td>"**);  
 out.print(**"<td>"**+stu.getSex()+**"</td>"**);  
 out.print(**"</tr>"**);  
 out.print(**"</table>"**);  
 out.print(**"<br>"**);  
 out.print(**"<a href=\"index.jsp\">点击此处回到主页面</a>"**);  
 bean.close();  
 }  
 } **catch** (ClassNotFoundException e) {  
 *//* ***TODO Auto-generated catch block*** e.printStackTrace();  
 } **catch** (SQLException e) {  
 *//* ***TODO Auto-generated catch block*** e.printStackTrace();  
 }  
 }  
}

**package** com.demo;  
  
**public class** Student {  
 **private** String **username**;  
 **private** String **password**;  
 **private** String **stunum**;  
 **private** String **sex**;  
 **public** String getUsername() {  
 **return username**;  
 }  
 **public void** setUsername(String username) {  
 **this**.**username** = username;  
 }  
 **public** String getPassword() {  
 **return password**;  
 }  
 **public void** setPassword(String password) {  
 **this**.**password** = password;  
 }  
 **public** String getStunum() {  
 **return stunum**;  
 }  
 **public void** setStunum(String stunum) {  
 **this**.**stunum** = stunum;  
 }  
 **public** String getSex() {  
 **return sex**;  
 }  
 **public void** setSex(String sex) {  
 **this**.**sex** = sex;  
 }  
}

**package** com.demo;  
**import** java.sql.Connection;  
**import** java.sql.DriverManager;  
**import** java.sql.ResultSet;  
**import** java.sql.SQLException;  
**import** java.sql.Statement;  
**import** java.util.ArrayList;  
**public class** SqlBean {  
 **private** Connection **conn**;  
 **private** Statement **stm**;  
 **private** ResultSet **set**;  
 **private boolean sign**=**false**;  
 **public** SqlBean() **throws** ClassNotFoundException, SQLException{  
 Class.*forName*(**"com.mysql.jdbc.Driver"**);  
 String user=**"root"**;  
 String password=**"root"**;  
 String url=**"jdbc:mysql://localhost:3306/user"**;  
 **conn**=DriverManager.*getConnection*(url, user, password);  
 }  
 **public** Connection getConn(){  
 **return conn**;  
 }  
 *//按学号查询学生* **public** Object getByStunum(String stunum) **throws** SQLException{  
  
 **stm**=**conn**.createStatement();  
 **set**=**stm**.executeQuery(**"select** *\** **from user where stunum='"**+stunum+**"'"**);  
 **while**(**set**.next()){  
 String username2=**set**.getString(**"username"**);  
 String sex=**set**.getString(**"sex"**);  
 String password=**set**.getString(**"password"**);  
 Student stu=**new** Student();  
 stu.setSex(sex);  
 stu.setPassword(password);  
 stu.setUsername(username2);  
 stu.setStunum(stunum);  
 **set**.close();  
 **stm**.close();  
 **return** stu;  
 }  
 **return ""**;  
 }  
 *//获取全部学生信息* **public** Object getAll(){  
 **if**(**sign**==**true**){  
 **sign**=**false**;  
 }  
 ArrayList<Student> container=**new** ArrayList<Student>();  
 **try** {  
 **stm**=**conn**.createStatement();  
 **set**=**stm**.executeQuery(**"select** *\** **from user"**);  
 **while**(**set**.next()){  
 String username=**set**.getString(**"username"**);  
 String stunum=**set**.getString(**"stunum"**);  
 String sex=**set**.getString(**"sex"**);  
 String password=**set**.getString(**"password"**);  
 Student stu=**new** Student();  
 stu.setStunum(stunum);  
 stu.setUsername(username);  
  
 stu.setPassword(password);  
 stu.setSex(sex);  
 container.add(stu);  
 **sign**=**true**;  
 }  
 } **catch** (SQLException e) {  
 *//* ***TODO Auto-generated catch block*** e.printStackTrace();  
 }  
 **if**(**sign**==**true**){  
 **return** container;  
 }  
 **else**{  
 **return ""**;  
 }  
 }  
 *//关闭连接* **public void** close(){  
 **try** {  
 **conn**.close();  
 } **catch** (SQLException e) {  
 *//* ***TODO Auto-generated catch block*** e.printStackTrace();  
 }  
 }  
}