电子科技大学信息与软件工程学院

**实 验 报 告**

学 号 2017221301001

姓 名 黄毅

（实验） 课程名称 JAVA程序与设计

理论教师 周帆

实验教师 何中海

**电 子 科 技 大 学**

**实 验 报 告**

**学生姓名：黄毅 学号：2017221301001 指导教师：何中海**

**实验地点：信软楼304 实验时间：2018.12.09**

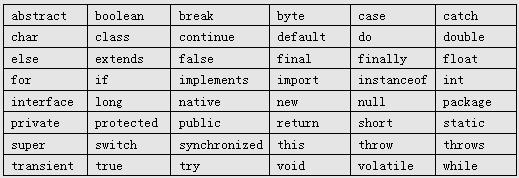
**一、实验名称：有关安装并配置Java的开发工具包JDK的实验**

**二、实验学时：2**

**三、实验目的：熟悉JDK安装和参数方法；熟悉Java的基本结构，变量，数组，数据类型。**

**四、实验原理：**

通过JDK下载与书本上的流程实现Java的jdk,jre的安装，设置。通过参数设置实现运行，并运用本身自带的ide实现编程。

通过第二章课本上的知识实现对于Java基本知识的了解，Java程序基础结构，包，import，类和接口。Java符号集：

**Java注释：**单行注释，多行注释，文章注释。

**Java数据简单类型：**1）四种整数类型(byte、short、int、long)： byte：8 位，用于表示最小数据单位，如文件中数据，-128~127 short：16 位，很少用，-32768 ~ 32767 int：32 位、最常用，-2^31-1~2^31 （21 亿） long：64 位、次常用 注意事项： int i=5; // 5 叫直接量（或字面量），即 直接写出的常数。 整数字面量默认都为 int 类型，所以在定义的 long 型数据后面加 L或 l。 小于 32 位数的变量，都按 int 结果计算。 强转符比数学运算符优先级高。见常量与变量中的例子。

2）两种浮点数类型(float、double)： float：32 位，后缀 F 或 f，1 位符号位，8 位指数，23 位有效尾数。 double：64 位，最常用，后缀 D 或 d，1 位符号位，11 位指数，52 位有效尾 注意事项： 二 进 制 浮 点 数 ： 1010100010=101010001.0\*2=10101000.10\*2^10（2次方)=1010100.010\*2^11(3次方)= . 1010100010\*2^1010(10次方) 尾数： . 1010100010 指数：1010 基数：2 浮点数字面量默认都为 double 类型，所以在定义的 float 型数据后面加F 或 f；double 类型可不写后缀，但在小数计算中一定要写 D 或 X.X float 的精度没有 long 高，有效位数（尾数）短。 float 的范围大于 long 指数可以很大。 浮点数是不精确的，不能对浮点数进行精确比较。

3）一种字符类型(char)： char：16 位，是整数类型，用单引号括起来的 1 个字符（可以是一个中文字符），使用 Unicode 码代表字符，0~2^16-1（65535） 。 注意事项： 不能为 0个字符。 转义字符：\n 换行 \r 回车 \t Tab 字符 \" 双引号 \\ 表示一个\ 两字符 char 中间用“+”连接，内部先把字符转成 int 类型，再进行加法运算，char 本质就是个数！二进制的，显示的时候，经过“处理”显示为字符。

4）一种布尔类型(boolean)：true 真 和 false 假。

5）类型转换： char--> 自动转换：byte-->short-->int-->long-->float-->double 强制转换：①会损失精度，产生误差，小数点以后的数字全部舍弃。②容易超过取值范围。

6）记忆：8位：Byte（字节型） 16位：short（短整型）、char（字符型） 32位：int（整型）、float（单精度型/浮点型） 64位：long（长整型）、double（双精度型） 最后一个：boolean(布尔类型

**Java变量和常量：**

在程序中存在大量的数据来代表程序的状态，其中有些数据在程序的运行过程中值会发生改变，有些数据在程序运行过程中值不能发生改变，这些数据在程序中分别被叫做变量和常量。

在实际的程序中，可以根据数据在程序运行中是否发生改变，来选择应该是使用变量代表还是常量代表。

变量

变量代表程序的状态。程序通过改变变量的值来改变整个程序的状态，或者说得更大一些，也就是实现程序的功能逻辑。

为了方便的引用变量的值，在程序中需要为变量设定一个名称，这就是变量名。例如在2D游戏程序中，需要代表人物的位置，则需要2个变量，一个是x坐标，一个是y坐标，在程序运行过程中，这两个变量的值会发生改变。

由于Java语言是一种强类型的语言，所以变量在使用以前必须首先声明，在程序中声明变量的语法格式如下：

数据类型 变量名称；

例如：int x；

在该语法格式中，数据类型可以是Java语言中任意的类型，包括前面介绍到的基本数据类型以及后续将要介绍的复合数据类型。变量名称是该变量的标识符，需要符合标识符的命名规则，在实际使用中，该名称一般和变量的用途对应，这样便于程序的阅读。数据类型和变量名称之间使用空格进行间隔，空格的个数不限，但是至少需要1个。语句使用“；”作为结束。

也可以在声明变量的同时，设定该变量的值，语法格式如下：

数据类型 变量名称 = 值；

例如：int x = 10；

在该语法格式中，前面的语法和上面介绍的内容一致，后续的“=”代表赋值，其中的“值”代表具体的数据，注意区别“==”代表为判断是否相等。在该语法格式中，要求值的类型需要和声明变量的数据类型一致。

在程序中，变量的值代表程序的状态，在程序中可以通过变量名称来引用变量中存储的值，也可以为变量重新赋值。例如：

int n = 5；

n = 10；

在实际开发过程中，需要声明什么类型的变量，需要声明多少个变量，需要为变量赋什么数值，都根据程序逻辑决定，这里列举的只是表达的格式而已。

常量

常量代表程序运行过程中不能改变的值。

常量在程序运行过程中主要有2个作用：

1. 代表常数，便于程序的修改（例如：圆周率的值）

2. 增强程序的可读性（例如：常量UP、DOWN、LEFT和RIGHT分辨代表上下左右，其数值分别是1、2、3和4）

常量的语法格式和变量类型，只需要在变量的语法格式前面添加关键字final即可。在Java编码规范中，要求常量名必须大写。

则常量的语法格式如下：

final 数据类型 常量名称 = 值；

final 数据类型 常量名称1 = 值1， 常量名称2 = 值2，……常量名称n = 值n；

例如：

final double PI = 3.14；

final char MALE=‘M’，FEMALE=‘F’；

在Java语法中，常量也可以首先声明，然后再进行赋值，但是只能赋值一次，示例代码如下：

final int UP；

UP = 1；

**Java流程控制：**

Java流程控制包括顺序控制、条件控制和循环控制。

顺序控制，就是从头到尾依次执行每条语句操作。条件控制，基于条件选择执行语句，比方说，如果条件成立，则执行操作A，或者如果条件成立，则执行操作A，反之则执行操作B。循环控制，又称为回路控制，根据循环初始条件和终结要求，执行循环体内的操作。

顺序结构只能顺序执行，不能进行判断和选择，因此需要分支结构。

Java有两种分支结构：

if语句

switch语句

Java中有三种主要的循环结构：

while循环

do…while循环

for循环

**Java数组：**Java 中定义数组的语法有两种：

type arrayName[];

type[] arrayName;

type 为Java中的任意数据类型，包括基本类型和组合类型，arrayName为数组名，必须是一个合法的标识符，[ ] 指明该变量是一个数组类型变量。例如：

int demoArray[];

int[] demoArray;

这两种形式没有区别，使用效果完全一样，读者可根据自己的编程习惯选择。

与C、C++不同，Java在定义数组时并不为数组元素分配内存，因此[ ]中无需指定数组元素的个数，即数组长度。而且对于如上定义的一个数组是不能访问它的任何元素的，我们必须要为它分配内存空间，这时要用到运算符new，其格式如下：

arrayName=new type[arraySize];

其中，arraySize 为数组的长度，type 为数组的类型。如：

demoArray=new int[3];

为一个整型数组分配3个int 型整数所占据的内存空间。

通常，你可以在定义的同时分配空间，语法为：

type arrayName[] = new type[arraySize];

例如：

int demoArray[] = new int[3];

数组的初始化

你可以在声明数组的同时进行初始化（静态初始化），也可以在声明以后进行初始化（动态初始化）。例如：

// 静态初始化

// 静态初始化的同时就为数组元素分配空间并赋值

int intArray[] = {1,2,3,4};

String stringArray[] = {"微学苑", "http://www.weixueyuan.net", "一切编程语言都是纸老虎"};

// 动态初始化

float floatArray[] = new float[3];

floatArray[0] = 1.0f;

floatArray[1] = 132.63f;

floatArray[2] = 100F;

数组引用

可以通过下标来引用数组：

arrayName[index];

与C、C++不同，Java对数组元素要进行越界检查以保证安全性。

每个数组都有一个length属性来指明它的长度，例如 intArray.length 指明数组 intArray 的长度。

【示例】写一段代码，要求输入任意5个整数，输出它们的和。

import java.util.\*;

public class Demo {

public static void main(String[] args){

int intArray[] = new int[5];

long total = 0;

int len = intArray.length;

// 给数组元素赋值

System.out.print("请输入" + len + "个整数，以空格为分隔：");

Scanner sc = new Scanner(System.in);

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

intArray[i] = sc.nextInt();

}

// 计算数组元素的和

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

total += intArray[i];

}

System.out.println("所有数组元素的和为：" + total);

}

}

运行结果：

请输入5个整数，以空格为分隔：10 20 15 25 50

所有数组元素的和为：120

数组的遍历

实际开发中，经常需要遍历数组以获取数组中的每一个元素。最容易想到的方法是for循环，例如：

int arrayDemo[] = {1, 2, 4, 7, 9, 192, 100};

for(int i=0,len=arrayDemo.length; i<len; i++){

System.out.println(arrayDemo[i] + ", ");

}

输出结果：

1, 2, 4, 7, 9, 192, 100,

不过，Java提供了”增强版“的for循环，专门用来遍历数组，语法为：

for( arrayType varName: arrayName ){

// Some Code

}

arrayType 为数组类型（也是数组元素的类型）；varName 是用来保存当前元素的变量，每次循环它的值都会改变；arrayName 为数组名称。

每循环一次，就会获取数组中下一个元素的值，保存到 varName 变量，直到数组结束。即，第一次循环 varName 的值为第0个元素，第二次循环为第1个元素......例如：

int arrayDemo[] = {1, 2, 4, 7, 9, 192, 100};

for(int x: arrayDemo){

System.out.println(x + ", ");

}

输出结果与上面相同。

这种增强版的for循环也被称为”foreach循环“，它是普通for循环语句的特殊简化版。所有的foreach循环都可以被改写成for循环。

但是，如果你希望使用数组的索引，那么增强版的 for 循环无法做到。

二维数组

二维数组的声明、初始化和引用与一维数组相似：

int intArray[ ][ ] = { {1,2}, {2,3}, {4,5} };

int a[ ][ ] = new int[2][3];

a[0][0] = 12;

a[0][1] = 34;

// ......

a[1][2] = 93;

Java语言中，由于把二维数组看作是数组的数组，数组空间不是连续分配的，所以不要求二维数组每一维的大小相同。例如：

int intArray[ ][ ] = { {1,2}, {2,3}, {3,4,5} };

int a[ ][ ] = new int[2][ ];

a[0] = new int[3];

a[1] = new int[5];

**五，实验内容：**

**1.完成第一章习题5，6编程。**

**2.完成第二章习题7编程。**

**3.编写程序，计算一个整数的各位数字之和，例如，整数20170925，则计算并显示2+0+1+7+0+9+2+5的值。**

**4.打印输出斐波拉契数列。**

**六、实验器材（设备、元器件）：电脑**

**七，实验步骤：**

**（一），实现Java安装，并实现代码。**

**（二），对于运算符号实现乘法运用，并通过for循环逐步输出9\*9乘法表**

**（三）：通过循环中实现各个数的相加。**

**（四），通过斐波拉契数列的性质，通过数组实现。**

**八，实验结果与分析（含重要数据结果分析或核心代码流程分析）**

**一，完成第一章习题5，6编程。**

**代码：**

**5题：**

**class HelloWorldApp{**

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

**System.out.println("Hello,Java World!");**

**}**

****

**6题：**

**import java.applet.Applet;**

**import java.awt.Graphics;**

**public class HelloJavaApp extends Applet{**

**public void paint(Graphics g){**

**g.drawString("Hello, Java Applet World !",50,25);**

**}**

**}**

**<html>**

**<head>**

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

**<title>无标题文档</title>**

**</head>**

**<body>**

**<applet>**

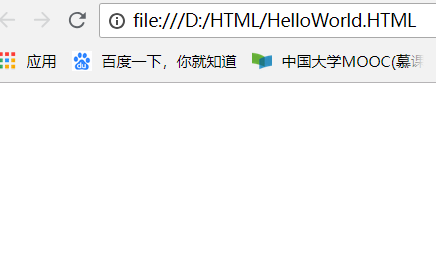
**<CODE="HelloJavaApp.class"WIDTH=500 HEIGHT=200>**

**</applet>**

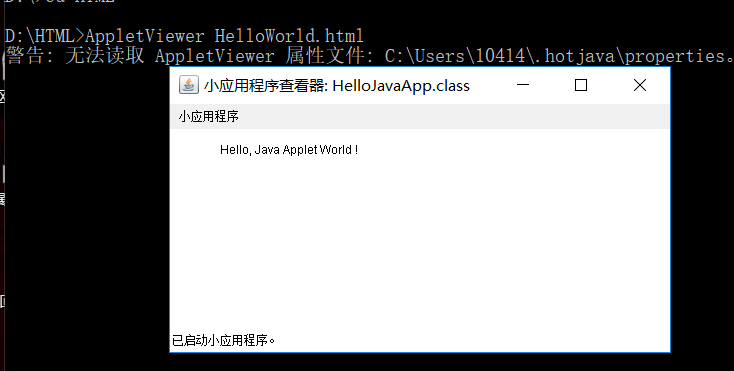
**</body>**

**</html>**

**无法实现：**

****

**用appletviewer工具实现：**

****

**二，完成第二章习题7编程。**

**public class chengfabiao {**

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

**for(int n=0;n<=9;n++) {**

**for(int m=0;m<=9;m++) {**

**System.out.print(n+"\*"+m+"="+(n\*m)+"\t");**

**}**

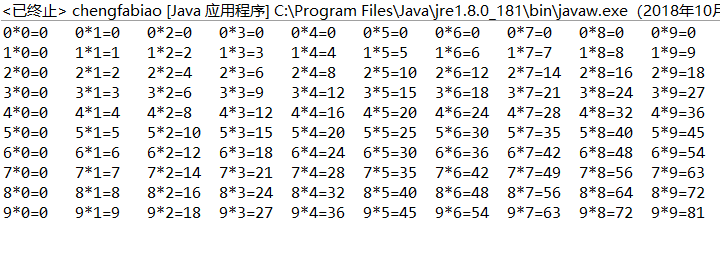
**System.out.println();**

**}**

**}**

**}**

**实现：**

****

**三，编写程序，计算一个整数的各位数字之和，例如，整数20170925，则计算并显示2+0+1+7+0+9+2+5的值。**

**代码：**

**public class 数位相加{**

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

**//调用求和方法**

**System.out.println(getSum(20170925));**

**}**

**/\*\* \* 求和方法 \* 接收一个int型整数 \*/**

**public static int getSum(int num) {**

**int n;**

**int sum = 0;**

**while( num != 0 ){ //未取尽**

**n = num%10; //取个位数**

**sum += n; //增加**

**num = (num-n)/10; //剩余数字**

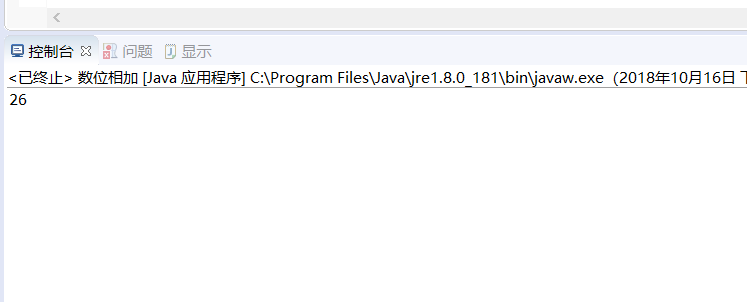
**}**

**return(sum);**

**}**

**}**

**运行结果：**

**运行成功**

**四，斐波拉契数列**

**代码：**

**public class 斐波拉契数列 {**

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

**int fib[]=new int[10];**

**fib[0]=1;**

**fib[1]=1;**

**for(int i=2;i<10;i++) {**

**fib[i]=fib[i-1]+fib[i-2];**

**}**

**for(int i=1;i<=10;i++) {**

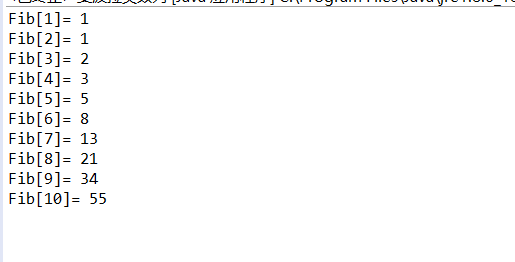
**System.out.println("Fib["+i+"]= "+fib[i-1]);**

**}**

**}**

**}**

**运行结果：**

****

**运行成功**

**九，总结及心得体会：**

**对于实验，开始安装Java的jre与jdk对于PATH,CLASSPATH设置是最头痛的事情，也加深我对于设置环境变量的印象，让我学会自主安装一门语言环境，对于通过控制命令实现代码也是我接触到了一种不同于C语言的编辑方法。**

**对于数组的实验，一开始还没有适应Java不同于C的数据方式，后来通过看书和查阅资料初步的了解，也加深对于知识的印象。**

**十，对本实验过程及方法、手段的改进建议：**

**对于第一个实验，书上的代码具有些问题没有实现，查找资料后，**

**代码：**

**import java.applet.\*;**

**import java.awt.\*;**

**public class HelloWorldApplet extends Applet**

**{**

**public void paint (Graphics g)**

**{**

**g.drawString ("Hello World", 25, 50);**

**}**

**}**

**HTML**

**<html>**

**<title>The Hello, World Applet</title>**

**<hr>**

**<applet code="HelloWorldApplet.class" width="320" height="120">**

**"Hello, World"**

**</applet>**

**<hr>**

**</html**

**运行结果：**

****

**运行成功**

**报告评分：**

**指导教师签字：**