电子科技大学信息与软件工程学院

**实 验 报 告**

学 号 2017221105003

姓 名 黄文杰

（实验） 课程名称 面向对象程序设计Java

理论教师 江春华

实验教师 郝晓青

**电 子 科 技 大 学**

**实 验 报 告**

**学生姓名：**黄文杰 **学号：**2017221105003 **指导教师：**江春华

**实验地点：**信软学院楼西306 **实验时间：**2018.09.19

1. **实验名称：**JDK使用及Java基础
2. **实验学时：**2学时
3. **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。
4. **实验原理：**

1. Java源程序机构组成

|  |
| --- |
| ClassName.java |
| package nameOfPackage; //包名  import example.OtherClassName; //引入类  class ClassName{ //声明创建类  int x, y; //声明成员变量  public static void main(String args[]){//声明创建成员方法main()  methoad(); //调用方法  }  } //创建类结束 |

2.Java符号集：

采用Unicode编码，使用16位存储空间，能够支持多种语言。

Java符号分为五种类型：

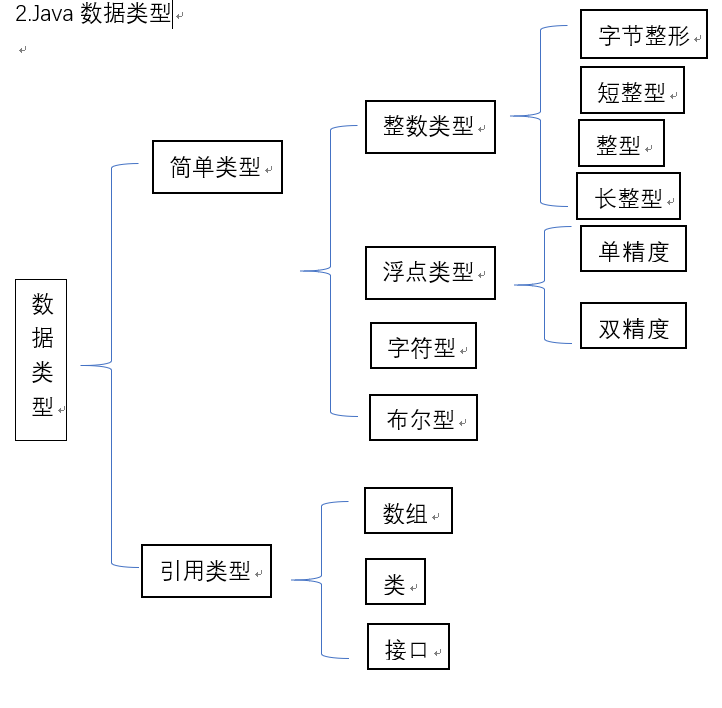
①关键字

②标识符

③常量

④运算符

⑤分隔符

3.Java数据类型

3.Java变量的作用域

变量的作用域是指程序执行中能够对变量访问的范围。根据作用域，变量可分为四类：

①成员变量：作用域在类中是全局的，可以被该类的方法访问；

②方法的局部变量：作用域自方法的声明处开始，到本段代码块结束为止可以访问；

③语句块的局部变量：这是在语句块中声明创建的变量，作用域从声明开始，到本段代码块结束为止可以访问；

④异常处理的局部变量：这是在异常处理代码段中声明创建的变量。

4.运算符及表达式

①单目运算符：有一个操作数，如 ++, --, !, -等；

②双目运算符：有两个操作数，如 +，&&，||等；

③三目运算符：有三个操作数，如 : ?;

5.Java数据类型转换

①拓宽类型：指把值范围小类型的数据转换成值范围大类型的数据；

②缩窄类型：指把指范围大类型的数据转换成值范围小类型的数据；

③自动类型转换：是指Java虚拟机自动地将数据类型实施转换。由于拓宽类型转换不造成数据访问不安全，所以自动类型转换是拓宽类型的数据转换。

④强制类型转换：是指通过显式地将数据类型实施转换。缩窄类型转换会造成数据访问的不安全，所以必须通过强制转换实现数据的类型转换。

强制类型转换的形式为：(type)identifier

6.Java流控制语句

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 分支语句 | if-else | break | switch | return |
| 循环语句 | while | do-while | for | continue |
| 异常处理语句 | try | catch | finally | throw |

1. **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。

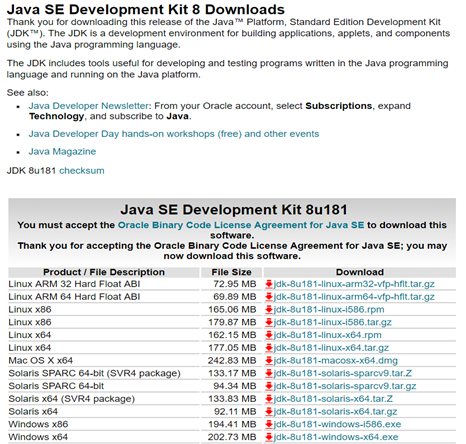
3. 编写程序，计算一个整数的各位数字之和，例如，整数20170925，则计算并显示2+0+1+7+0+9+2+5的值。

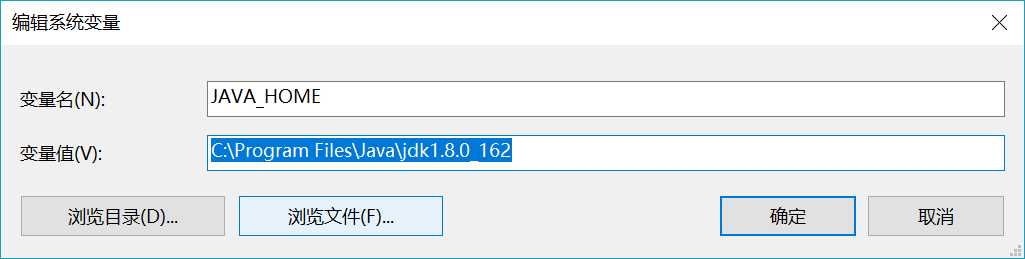
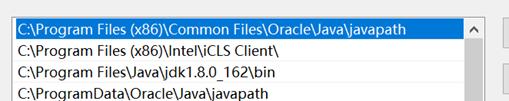
4. 打印输出斐波拉契数列。

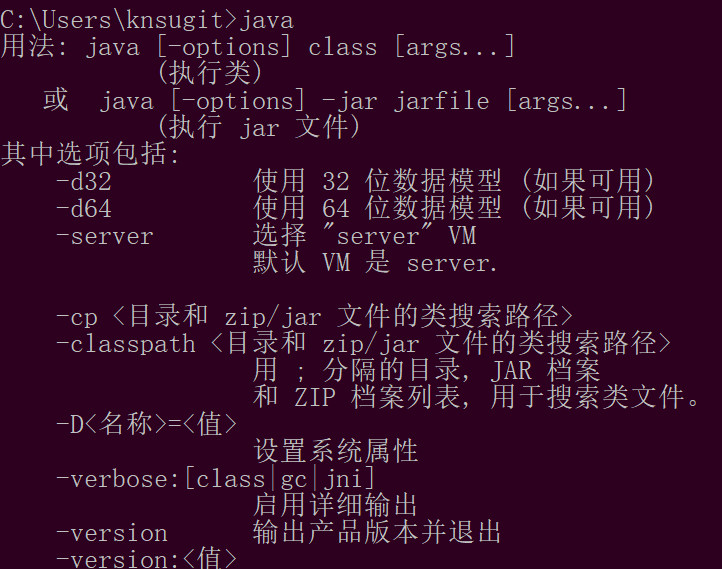
1. **实验器材（设备、元器件）：**

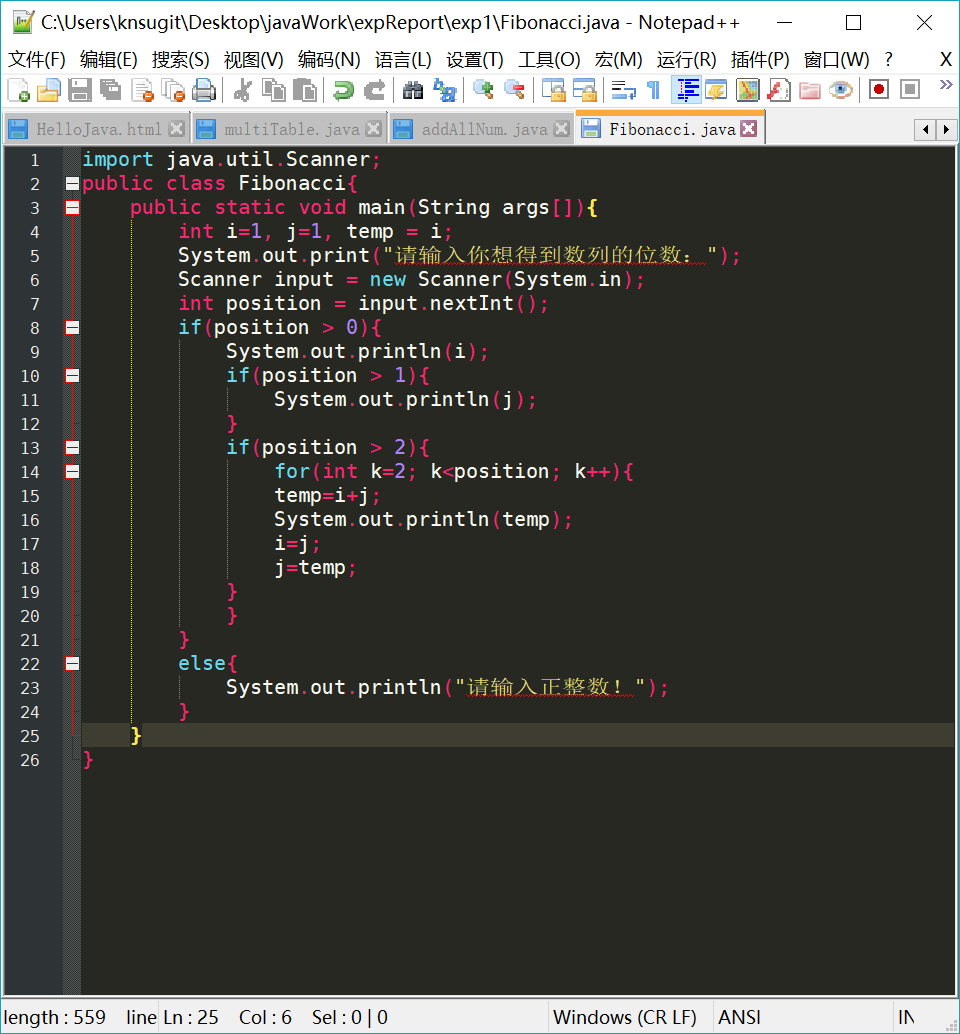
PC机一台

1. **实验步骤：**

1.下载安装Java开发工具JDK和Java运行环境JRE；

2.Windows操作系统下配置环境变量；

3.进入命令行，测试编译命令javac和运行命令java能否正常执行

4.若相应命令能正常执行，则使用文本编辑器进行Java源码编写工作。

5.Java源码编写完成后，在命令行中使用javac ClassName.java命令进行编译，如没有编译错误，则得到ClassName.class文件。

6.得到class文件后，继续在命令行中使用java ClassName命令，运行Java程序。

1. **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）**

1.第一章习题5，6章编程

|  |
| --- |
| Java Application(JavaWorld.java) |
| public class JavaWorld{  public static void main(String args[]){  System.out.print("Welcome to Java World!");  }  } |
| 运行结果:   |  | | --- | | PS C:\Users\knsugit\Desktop\javaWork\expReport\exp1> javac .\JavaWorld.java  PS C:\Users\knsugit\Desktop\javaWork\expReport\exp1> java JavaWorld  Welcome to Java World! | |
| 实验结论:测试正确 |

|  |
| --- |
| Java Applet(HelloJavaApp.java) |
| import java.applet.Applet;  import java.awt.Graphics;  public class HelloJavaApp extends Applet{  public void paint(Graphics g){  g.drawString("Welcome to Java applet World!", 100, 100);  }  } |
| HelloJava.html |
| <html>  <body>  <applet code="HelloJavaApp.class"WIDTH=1000 HEIGHT=500>  </applet>  </body>  </html> |
| 运行结果:   |  | | --- | |  | |
| 实验结论:测试正确 |

2.完成第二章习题7编程

|  |
| --- |
| multiTable.java |
| public class multiTable{  public static void main(String args[]){  for(int i=1; i<10; i++){  for(int j=1; j<=i; j++){  System.out.print(i + "\*" + j + "=" + i\*j +"\t");  }  System.out.println("");  }  }  } |
| 运行结果:   |  | | --- | | 1\*1=1  2\*1=2 2\*2=4  3\*1=3 3\*2=6 3\*3=9  4\*1=4 4\*2=8 4\*3=12 4\*4=16  5\*1=5 5\*2=10 5\*3=15 5\*4=20 5\*5=25  6\*1=6 6\*2=12 6\*3=18 6\*4=24 6\*5=30 6\*6=36  7\*1=7 7\*2=14 7\*3=21 7\*4=28 7\*5=35 7\*6=42 7\*7=49  8\*1=8 8\*2=16 8\*3=24 8\*4=32 8\*5=40 8\*6=48 8\*7=56 8\*8=64  9\*1=9 9\*2=18 9\*3=27 9\*4=36 9\*5=45 9\*6=54 9\*7=63 9\*8=72 9\*9=81 | |
| 实验结论:测试正确 |

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。

|  |
| --- |
| addAllNum.java |
| import java.util.Scanner;  public class addAllNum{  public static void main(String args[]){  int sum=0;  System.out.print("Please input a number:");    Scanner input = new Scanner(System.in);  int num = input.nextInt();    while(num > 0){  //System.out.println("num=" + num);  sum += num%10;  num = num/10;  }  System.out.println("sum=" + sum);  }  } |
| 运行结果:   |  | | --- | | PS C:\Users\knsugit\Desktop\javaWork\expReport\exp1> java addAllNum  Please input a number:23334  sum=15 | |
| 实验结论:测试正确 |

4. 打印输出斐波拉契数列。

|  |
| --- |
| Fibonacci.java |
| import java.util.Scanner;  public class Fibonacci{  public static void main(String args[]){  int i=1, j=1, temp = i;  System.out.print("请输入你想得到数列的位数：");  Scanner input = new Scanner(System.in);  int position = input.nextInt();  if(position > 0){  System.out.println(i);  if(position > 1){  System.out.println(j);  }  if(position > 2){  for(int k=2; k<position; k++){  temp=i+j;  System.out.println(temp);  i=j;  j=temp;  }  }  }  else{  System.out.println("请输入正整数！");  }  }  } |
| 运行结果:   |  | | --- | | PS C:\Users\knsugit\Desktop\javaWork\expReport\exp1> java Fibonacci  请输入你想得到数列的位数：10  1  1  2  3  5  8  13  21  34  55 | |
| 实验结论:测试正确 |

1. **总结及心得体会：**

1.与过去C语言的学习相比，Java最为突出的特点就是面向对象的思想。虽然在某种程度上语法更为复杂，代码量更大，比如输出时需要长长一段的System.out.println();而不是直接地printf();但是这种“一切皆对象”的思想更为贴近真实的情况，语言更为自然流畅。在解决某个问题上，我们能自然地想到应该指派某个对象；选定某个对象后，又自然而然地想起该对象的某种方法……这是我初学Java编程语言时感受到面向对象思想的巨大力量。

2.与C语言只需编译器相比，Java由于采用了先编译后解释的运行原理，它的开发环境搭建更为复杂，甚至要配置环境变量。因此，对系统的了解也是一项重要的技能，有助于对自身编程环境的了解与改良。

3.初学Java，容易忽视主类名与文件名要一致，导致编译出现问题；而且同一Java文件中有且仅有1个public修饰的类；另外Java的修饰符更多，使用顺序也相当复杂，容易写错等等，这些易错之处都应及时发现并纠正。

1. **对本实验过程及方法、手段的改进建议：**

1.在实验过程中发现，Java Applet是一项较为过时的技术，在Java10中，如果源码中引入了java.applet.Applet类，编译时编译器会提示相关API的过时性。另外越来越少的浏览器支持Applet，往往只能采用AppletViewer才能观察到小程序的运行效果。这些问题都造成了困扰和不便。

2.随着代码量的逐渐增大，程序结构的复杂化，我们除了在命令行下编译运行Java程序外，也应学会在Java相关IDE中，譬如eclipse， idea等，创建项目工程、编写、调试代码等。

**报告评分：**

**指导教师签字：**