电子科技大学信息与软件工程学院

**实 验 报 告**

学 号 2017221302021

姓 名 杨榆丰

（实验） 课程名称 面向对象程序设计JAVA

理论教师 周 帆

实验教师 周 帆

**电 子 科 技 大 学**

**实 验 报 告**

**学生姓名：杨榆丰 学号：2017221302021 指导教师：周帆**

**实验地点：信软楼西304 实验时间： 2018-12-09**

1. **实验名称：JDK使用及JAVA基础**
2. **实验学时：2**
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。

1. **实验原理：**

运用java语言以面向对象的程序设计方法设计程序

1. Java语言的特点：

Java具有简单性、面向对象、分布式、健壮性、安全性、平台独立与可移植性、多线程、动态性等特点

1. Java包含的三个版本及各版本功能和应用领域：
2. Java SE 包含了支持 Java Web 服务开发的类，并为 Java Platform，Enterprise Edition（Java EE）提供基础。它允许开发和部署在桌面、服务器、嵌入式环境和实时环境中使用的 Java 应用程序。
3. Java EE 是在 Java SE 的基础上构建的，它提供 Web 服务、组件模型、管理和通信 API，可以用来实现企业级的面向服务体系结构（service-oriented architecture，SOA）和 Web2.0应用程序。
4. Java ME 包括灵活的用户界面、健壮的安全模型、许多内置的网络协议以及对可以动态下载的连网和离线应用程序的丰富支持。Java ME 为在移动设备和嵌入式设备（比如手机、PDA、电视机顶盒和打印机）上运行的应用程序提供一个健壮且灵活的环境。
5. Java语言使用的字符集及其存储空间：

16位Unicode编码

1. Java中的数据类型分为的两大类及其类型范围：
2. 简单数据类型

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| boolean | byte | char | short | int | float | double | long |

1. 引用数据类型

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 类 | 接口类型 | 数组类型 | 枚举类型 | 注解类型 |

1. 简单数据类型的存储开销以及值的范围

|  |  |  |
| --- | --- | --- |
| 类型 | 存储 | 范围 |
| byte | 8bit | [-27,27-1] |
| short | 16bit | [-215，215-1] |
| int | 32bit | [-231, 231-1] |
| Long | 64bit | [-263，263-1] |
| float | 32bit | [-231, 231-1] |
| double | 64bit | [-263，263-1] |
| char | 16bit | [\u0000,\uffff] |
| boolean | 1bit | 默认false |

1. Java语言的逻辑运算优化：

用^实现异或运算

1. 数据类型转换的含义及其种类：

a.含义：将一种类型转换为其他类型

b.种类：简单数据类型之间的转换

字符串与其他数据类型间的转换

其他使用数据类型之间的转换

1. 变量的作用域种类及其分别的特点：

文件作用域：类似于C语言中全局变量

函数作用域 ：函数内的局部变量

代码块作用域 ：用{ }包围的变量只在大括号内有作用

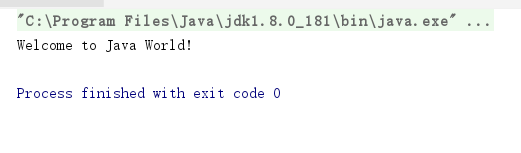
函数原型作用域 ：函数声明时定义的变量作用于整个函数中

1. **实验内容：**
2. 完成第一章习题5，6编程。
3. 完成第二章习题7编程。
4. 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。
5. 打印输出斐波拉契数列。
6. **实验器材（设备、元器件）：**

PC一台

1. **实验步骤：**
2. 需求分析
3. 算法结构设计
4. 实验编码
5. 测试功能
6. 填写实验报告
7. **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）**

|  |
| --- |
| **第一章习题5** |
| 程序源码：  **public class** 第一章5 {  **public static void** main(String[] agrs){  System.***out***.println(**"Welcome to Java World!"**);  } } |
| 测试结果： |



|  |
| --- |
| **第一章习题6** |
| **程序源码：**  **import** java.applet.\*; **import** java.awt.\*;  **public class** chapter1\_6 **extends** Applet{  **public void** paint(Graphics g){  g.drawString(**"Welcome to Java Applet World!"**,50,25);  } } |
| **测试结果：** |

|  |
| --- |
| **第二章习题7** |
| **程序源码：**  **public class** multiplication\_table {  **public static void** main(String args[])  {  **int** a ,b,c;  **for**(a=1;a<=9;a++)  {  **for**(b=1;b<=a;b++)  {  System.***out***.printf(**"%dx%d=%d\t"**,a,b,a\*b);  }  System.***out***.print(**'\n'**);  }  } } |
| **测试结果：** |

|  |
| --- |
| **整数的各位数字之和** |
| **程序源码：**  **import** java.util.\*; **public class** Super\_add {  **public static void** main(String[] args){  **int** s;  Number a=**new** Number();  a.s\_add();  a.output();  } }  **class** Number{  **private int num**;  **private int n**;  **private int s**;  **void** s\_add() {  **s**=0;  System.***out***.printf(**"Please input a number:"**);  Scanner input = **new** Scanner(System.***in***);  **num** = input.nextInt();  **n**=**num**%10;  **while**(**n**>0) {  **s** = **s** + **n**;  **num** = **num** / 10;  **n** = **num** % 10;  }  }   **void** output(){  System.***out***.println(**"\n各位数字之和为"**+**this**.**s**);  } } |
| **测试结果：** |

|  |
| --- |
| **打印输出斐波拉契数列** |
| **程序源码：**  **import** java.util.\*; **public class** Fibonacci {  **public static void** main(String[] args){  Fibonacci1 a=**new** Fibonacci1();  a.get\_number();  a.out\_put();  } }  **class** Fibonacci1{  **private long pre**=1,**next**=1,**n**;  **void** get\_number(){  System.***out***.printf(**"请输入你想要得到的Fibonacci数的个数(不超过80）："**);  Scanner input=**new** Scanner(System.***in***);  **n**=input.nextInt();  }   **void** out\_put(){  **long** i,j;  System.***out***.printf(**"\n你的Fibonacci数列来啦：\n%d %d"**,**this**.**pre**,**this**.**next**);  **for** (i=0;i<**this**.**n**-2;i+=1){  j=**pre**;  **pre**=**next**;  **next**=j+**next**;   System.***out***.printf(**" %d"**,**this**.**next**);  }  } } |
| **测试结果：** |

1. **总结及心得体会：**

面向对象的程序设计方法相对于面向过程的设计方法而言，能够将功能模块化，程序更加清晰易懂，程序编写也更加简便。

1. **对本实验过程及方法、手段的改进建议：**

在明确后设计程序之前，可以在脑袋中构想一个个对象，明确属性和方法以及对象之间的逻辑联系，做到心中有数，往往能够得到事半功倍的效果。

**报告评分：**

**指导教师签字：**