**集美大学计算机工程学院实验报告**

|  |  |  |
| --- | --- | --- |
| **课程名称**：Java程序设计 | **班级**：软件1912 | **实验成绩**： |
| **指导教师**：李　旺 | **姓名**：游嘉升 |  |
| **实验项目名称：Java IDE开发环境安装与配置** | **学号**：201921122050 | **上机实践日期**：9月18日 |
| **实验项目编号**：实验1 | **组号**： | **上机实践时间**： 2 学时 |

**一、目的**

掌握JDK安装与配置，掌握Java IDE开发环境配置与基本使用。

**二、实验内容与设计思想**

1. 熟悉控制台下常用命令

使用cmd命令进入控制台（参考附件“DOS常用命令.doc”文档）：

(1)dir 显示当前目录下所有内容。如：dir \*.java 显示当前目录下所有扩展名为java的文件。

(2)切换磁盘分区。如：c:\temp\java>**d:** 或者 **e:**，切换到d盘或者e盘。

(3)cd 进入某个目录。如：c:\temp\java>**cd abc** 指的是进入c:\temp\java下的abc目录；c:\temp\java>**cd ..** 进入java的父目录，即c:\temp；c:\temp\java>**cd \** 进入根目录，在这里即c盘根目录。

(4)关于java的命令。java -version 查看当前系统使用的jdk/jre版本；javac HelloWorld.java 对HelloWorld.java文件进行编译；java HelloWorld 将HelloWorld类载入虚拟机（即运行HelloWorld）。

2. JDK的配置与HelloWorld

(1)学会设置环境变量JAVA\_HOME、classpath与path(参考课件)；

(2)学会使用java –version查看版本号，确认jre是否装好；

(3)学会使用记事本编写HelloWorld.java；

(4)使用命令行(cmd)行进行编译(javac)、运行(java)。

注意：编译或者运行时碰到错误，请参见附件的相关提示，也可以在google或baidu中搜索相应解决方案。

3. eclipse/intellij idea安装与使用

(1)安装eclipse/intellij idea(如果已经安装了，实验报告中可不必添加安装过程的截图)；

(2)新建一个Java Project；

(3)编写HelloWorld.java；

(4)运行HelloWorld，在eclipse或intellij idea控制台中查看结果。

注意：观察eclipse或intellij idea开发环境中对编译错误的提示，并适当修改。

4. 筛法求素数

编写一个程序，从键盘读取正整数n，使用筛法求不大于n的所有素数（或称质数，Prime Number），并逐个打印出来。输出结果为：

n=10

The prime numbers are 2,3,5,7

附注：筛法是一种简单检索素数的算法。据说是古希腊的埃拉托斯特尼（Eratosthenes，约公元前274～194年）发明的，又称**埃拉托斯特尼筛法**（sieve of Eratosthenes）。具体做法是：给出要筛数值的范围n，先用2去筛，即把2留下，把2的倍数剔除掉；再用下一个素数，也就是3筛，把3留下，把3的倍数剔除掉；接下去用下一个素数5筛，把5留下，把5的倍数剔除掉；不断重复下去……

注意：处理控制台输入，请参考课件中相关内容（使用Scanner的nextLine()或nextInt()）；可以学习查询jdk文档，Integer类的parseInt方法，将字符串转换成int型。

(1)编写函数

编写public static void printPrimeNumbers(int[] n)方法，将不大于n的每个素数输出；

编写public static int[] getPrimeNumbers(int n)方法，将每个不大于n的素数放入数组，并返回。方法原型定义参考如下（逻辑代码需要自行填充！方法签名不能修改，但代码不限定）：

**public static int[] getPrimeNumbers(int n){**

**int[] result = new int[n];**

**//一系列处理代码**

**return result;**

**}**

(2)编写测试代码

在main方法中输入n，调用getPrimeNumbers方法获取素数数组，再调用printPrimeNumbers方法输出这些素数。

\*(3)思考

当n比较大时（如100,000,000），分析算法的瓶颈，并给出改进的策略！

5. 最大公约数

分别使用递归和非递归的算法，求解两个数的最大公约数。

说明：辗转相除法，又名欧几里德算法（Euclidean Algorithm），是求最大公约数的一种方法。具体做法是：用较小数除较大数，再用出现的余数（第一余数）去除除数，再用出现的余数（第二余数）去除第一余数，如此反复，直到最后余数是0为止。如果是求两个数的最大公约数，那么最后的除数就是这两个数的最大公约数。

提示：编写静态方法private static int gcd(int m, int n)用于实现非递归算法，编写private static int gcd\_recursive(int m, int n)用于实现递归算法。主程序提供输入，并分别调用这两个方法测试。

6. 求e的近似值

自然对数底数e是一个超越数，是一个无限不循环小数，它的值约为2.718281828459…。数学上，e定义为**(1+1/n)n**的极限(n趋于无穷大）。此外，e也可以定义为：**1/0!+1/1!+1/2!+…+1/n!+…**，更适合计算机迭代算法。

**要求：使用迭代法，且误差控制在10-6以内。**

**三、实验使用环境**

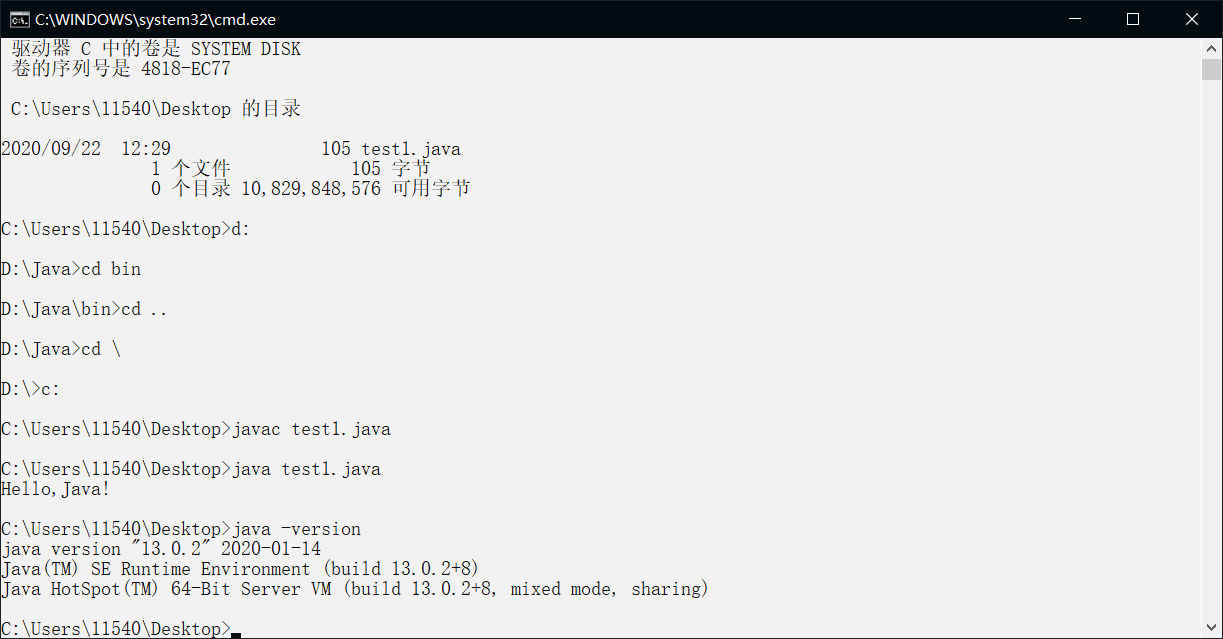
Java version "13.0.2"

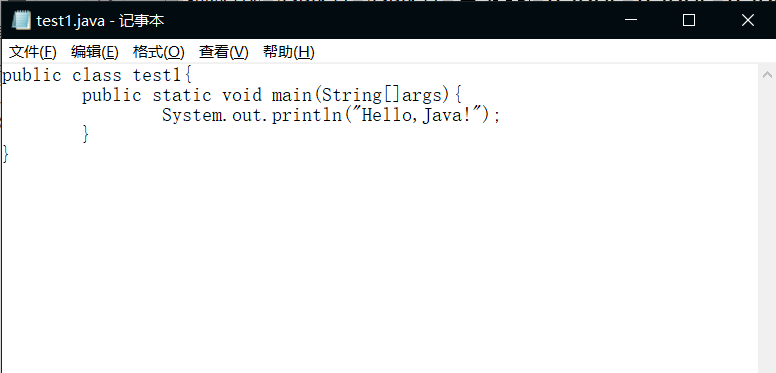
IntelliJ IDEA 2020.2.2 x64

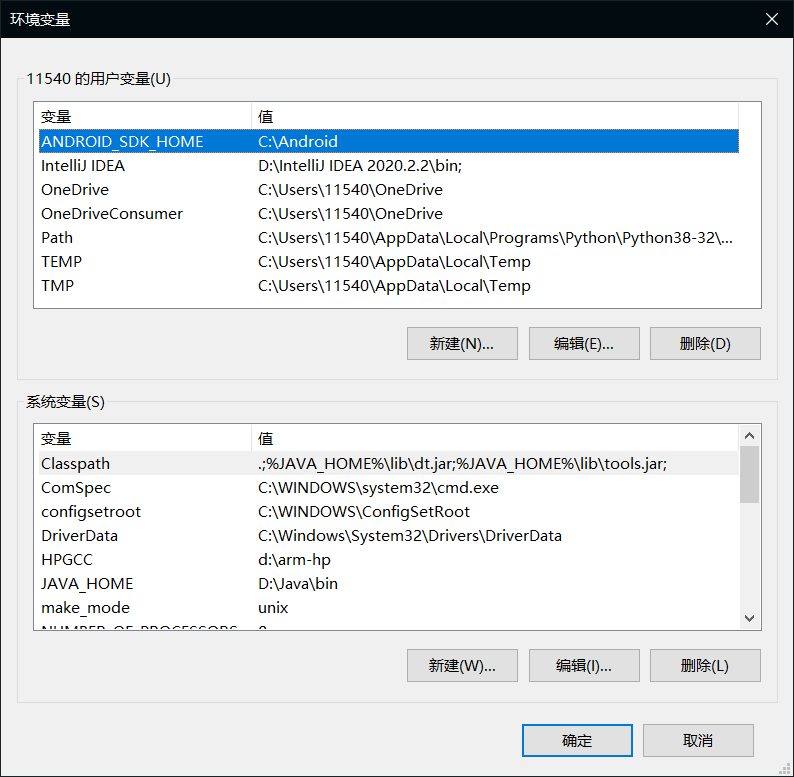
**四、实验步骤和调试过程**

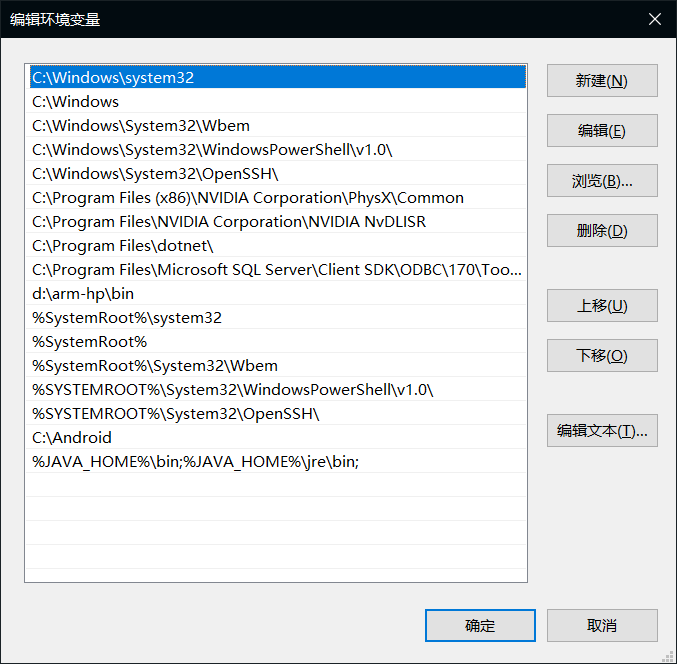
1. 熟悉控制台下常用命令

2. JDK的配置与HelloWorld

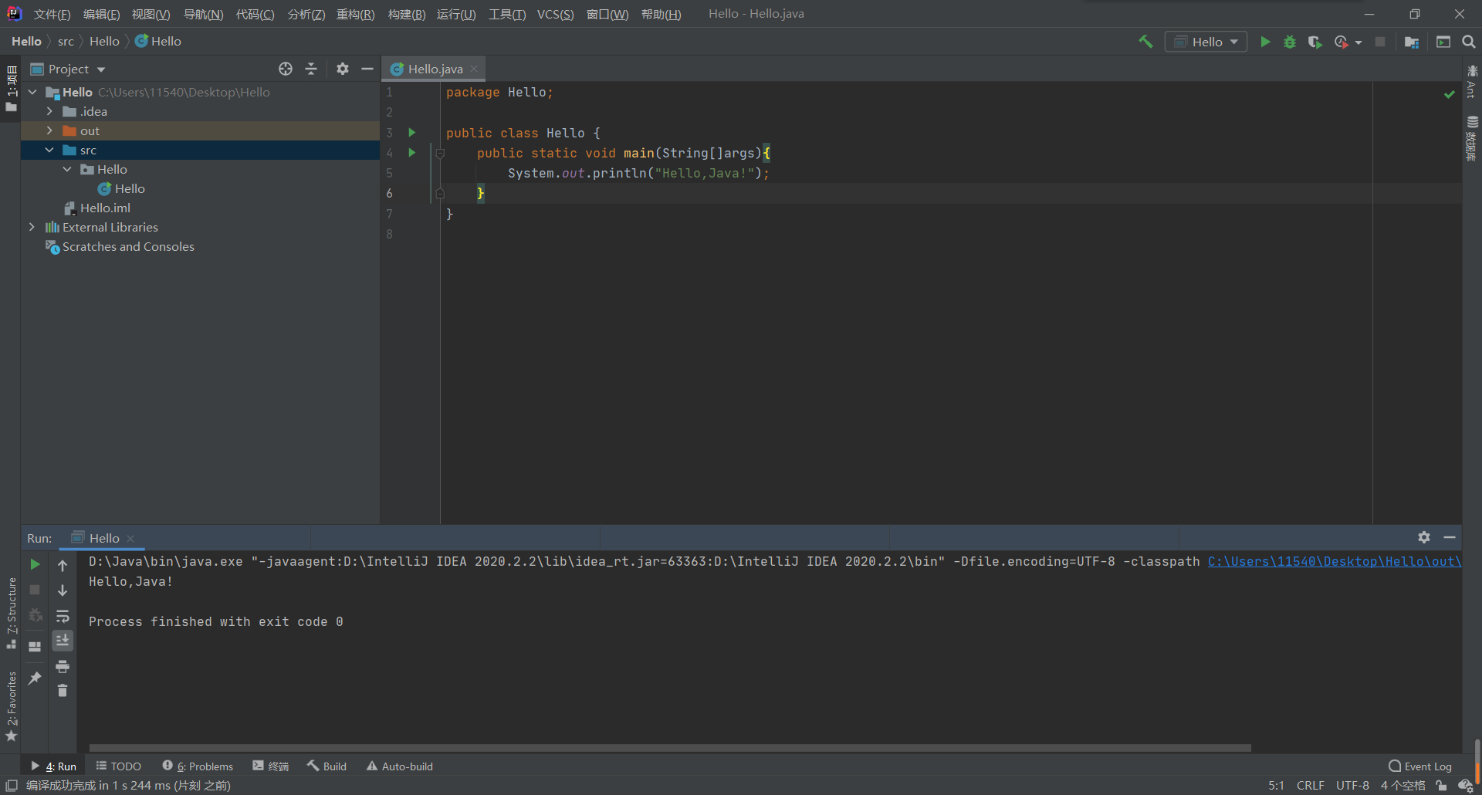








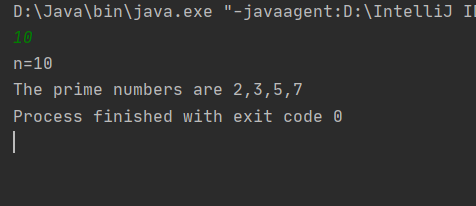
3. eclipse/intellij idea安装与使用

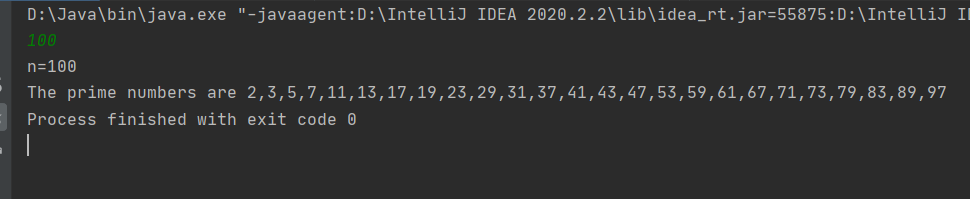


4. 筛法求素数

代码见附件IsPrime.java

实验数据：

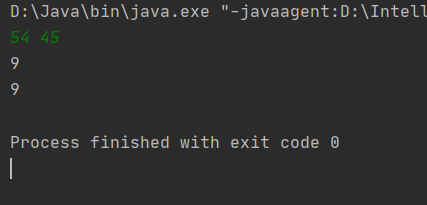


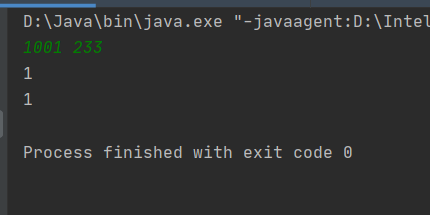


5. 最大公约数

代码见附件gcd.java

实验数据：

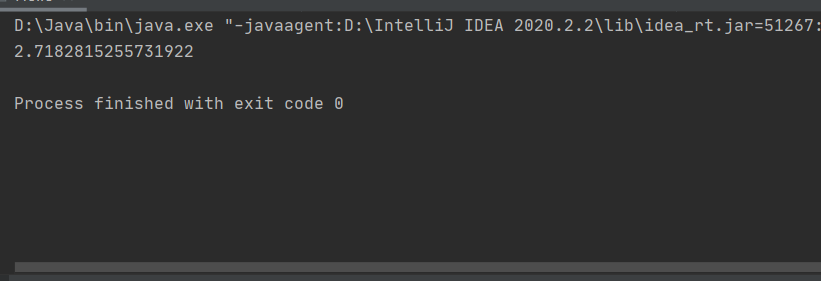




6. 求e的近似值

代码见附件e.java

实验数据：



**五、实验小结**

1．实验中遇到的问题及解决过程

①main函数的参数未填用javac命令编译报错。

解决过程：翻阅了一下书籍，百度了一下一些资料，最后发现java语言中main函数

需要有一个String[]的参数。

②使用Intellij Idel编译器一直无法运行程序，显示需要配置文件。

解决过程：百读了一些资料，对比了一下代码，发现需要建立一个package的目录才能进行正常编译运行，具体原因还需要再研究一下。

2．实验中产生的错误及原因分析

很多时候习惯于c\c++的语法格式之类的，惯性思维导致问题出现，比如上述的main函数的参数设定，package的建立等等等的问题，说到底就是对java代码规则语法什么的还不熟悉。

3．实验体会和收获。

虽然java和c\c++语言语法方面有一定的相似之处，但是实际上还是挺多区别的，但凭借建立在学习c\c++语言累计的一些经验上，应该能够很快适应java语言的学习。

**六、附录**

[1][美] 凯.S.霍斯特曼（Cay S. Horstmann）.Java核心技术·卷 I（原书第10版）[M].机械工业出版社