



武汉大学
WUHAN UNIVERSITY

武汉大学计算机学院 面向对象程序设计实验

java 开发环境配置 与简单 Java 程序设计

专 业 名 称 : 特软

课 程 名 称 : 面向对象程序设计

指 导 教 师 : 丁建利

学 生 学 号 : 2023302071062

学 生 姓 名 : 刘博瑞

二〇二四年 三 月

目录

实验要求.....	3
1. 【实验目的】	3
2. 【实验环境】	3
3. 【实验内容】	3
实验实现.....	4
1. 【设计思路】	4
2. 【项目结构图和实验源代码】	4
3. 【实验结果运行结果】	7
实验小结.....	10
1. 【实验结果分析】	10
2. 【实验心得体会】	10

实验要求

1. 【实验目的】

1. 熟悉 JDK 开发环境安装与配置；
2. 熟练掌握简单 Java 程序设计；
3. 掌握在命令行控制台窗口，进行 Java 程序的编写、编译（javac.exe）、运行（java.exe）等过程；
4. 熟悉 IDEA 或 Eclipse 开发环境的基本使用方法；
5. 掌握 IDE 开发工具中的项目结构：project-module-package-class。

2. 【实验环境】

- 1.操作系统:
- 2.环境配置:
- 3.开发工具:

3. 【实验内容】

1. 安装 JDK 以及配置 JAVA 开发环境，在命令行控制台窗口输入命令查看 JAVA 版本信息。
2. 通过记事本编写 Java 程序实现以下功能：
 输入“学号 姓名”，输出字符串“Hello 学号 姓名”
 在命令行控制台窗口，进行 Java 程序的编译（javac.exe）和运行（java.exe）。
3. 安装一个 JAVA IDE 开发工具，对其进行配置（支持中文的正确输出），在 IDE 开发工具中创建“Test”项目，并在 src 下建立“cs.你的姓名拼音.exam1”（比如“曹操”同学的包名为：cs.caocao.exam1）的包名结构，类名根据题目自己命名。
4. 编写 Java 程序输出“武汉大学”这四个汉字分别在 Unicode 表中的顺序位置。
提示：观察一个字符在 Unicode 表中的顺序位置，需使用 int 类型转换，如(int)a。不可以使用 short 转换。要得到一个 0~65535 之间的数所代表的 Unicode 表中相应位置上的字符，需使用 char 型转换。
5. 打印输出所有的“水仙花数”，所谓“水仙花数”是指一个 3 位数，其中各位数字立方和等于该数本身。例如，153 是一个“水仙花数”。
6. 编程求 $1! + 2! + 3! + \dots + 20!$ 。
7. 编写程序，分别使用冒泡排序算法和类库完成对以下一组数排序，并输出排序结果。
 一组随机数如：0.89103033, 0.60550521, 0.13856488, 0.57468244, 0.370697, 0.31823162, 0.58358377, 0.97177935, 0.76400592, 0.11269547

实验实现

1. 【设计思路】

这几个题比较简单，没有设计类图或类关系图

T2 用 scanner 输入

T4 强制类型转换

T6 递归

T7 冒泡排序或者调用库排序

2. 【项目结构图和实验源代码】

```
T2.java x
1 package CS.LiuBoRui.exam1;
2
3 import java.util.Scanner;
4
5 public class T2
6 {
7     public static void main(String[] args)
8     {
9         Scanner input=new Scanner(System.in);
10        long num= input.nextLong();
11        String name=input.next();
12        System.out.println("Hello"+" "+num+" "+name);
13    }
14 }
15
```

```
1 package CS.LiuBoRui.exam1;
2
3 public class T4
4 {
5     public static void main(String[] args)
6     {
7         char a='武';
8         char b='汉';
9         char c='大';
10        char d='学';
11        int a1=(int)a;
12        int b1=(int)b;
13        int c1=(int)c;
14        int d1=(int)d;
15        System.out.println(a1+" "+b1+" "+c1+" "+d1);
16    }
17 }
18
```

```

1 package CS.LiuBoRui.exam1;
2
3 public class T5
4 {
5     public static void main(String[] args)
6     {
7         for(int i=100;i<=999;i++)
8         {
9             int a=i/100;
10            int b=(i%100)/10;
11            int c=i%10;
12            if(a*a+a+b*b+b+c*c*c==i) System.out.println(i);
13        }
14    }
15 }
16

```

```

1 package CS.LiuBoRui.exam1;
2
3 public class T6
4 {
5     1 usage
6     long times(int m)
7     {
8         if (m == 0 || m == 1) return 1;
9         else
10        {
11            long ans = 1;
12            for (int i = m; i >= 1; i--) ans *= i;
13            return ans;
14        }
15
16    2 usages
17    long f(int m)
18    {
19        if(m==1) return 1;
20        else return times(m)+f(m-1);
21    }
22    public static void main(String[] args)
23    {
24        T6 a=new T6();
25        System.out.println(a.f(m: 20));
26    }
27 }
28

```

```

1 package CS.LiuBoRui.exam1;
2 import java.util.Arrays;
3
4 public class T7
5 {
6     public static void main(String[] args)
7     {
8         double[] nums={0.89103033,0.60550521,0.13856488,0.57468244,0.370697,0.31823162, 0.58358377
9         Arrays.sort(nums);
10        for(double a:nums)
11        {
12            System.out.println(a);
13        }
14    }
15 }
16
17

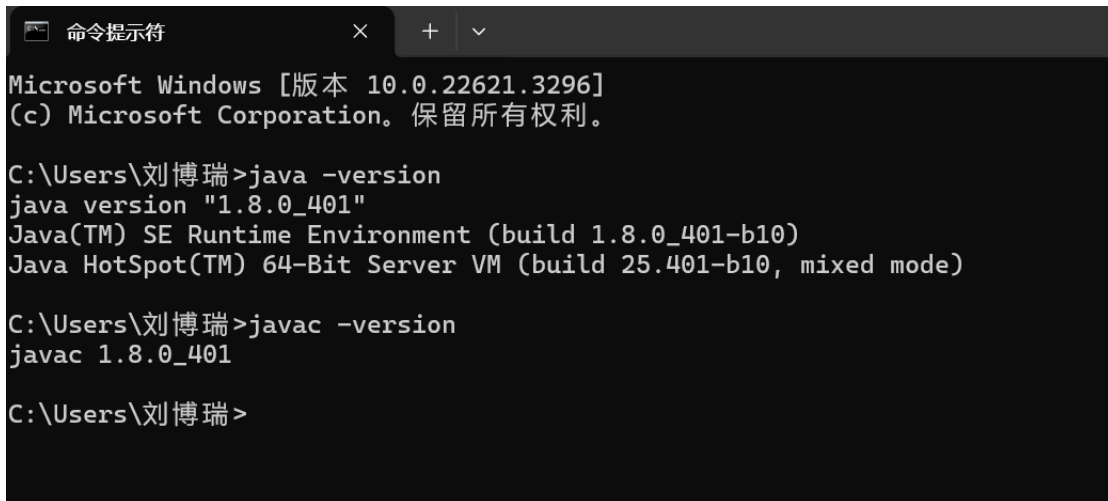
```

```

T2.java  T4.java  T5.java  T6.java  Fraction.java  T7.java  T7_.java x
1 package CS.LiuBoRui.exam1;
2
3 public class T7_
4 {
5     public static void main(String[] args)
6     {
7         double[] nums={0.89103033,0.60550521,0.13856488,0.57468244,0.370697,0.31823162, 0.583
8         for(int i=0;i<nums.length;i++)
9             for(int j=0;j< nums.length-i-1;j++)
10            {
11                if(nums[j]>nums[j+1])
12                {
13                    double t=nums[j];
14                    nums[j]=nums[j+1];
15                    nums[j+1]=t;
16                }
17            }
18
19
20        for(double n:nums)
21        {
22            System.out.println(n);
23        }
24    }
25 }

```

3. 【实验运行结果】

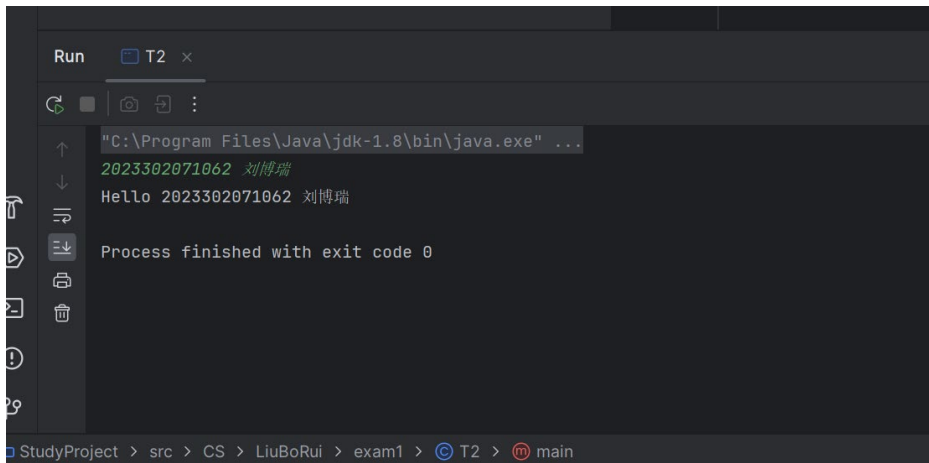


```
命令提示符
Microsoft Windows [版本 10.0.22621.3296]
(c) Microsoft Corporation。保留所有权利。

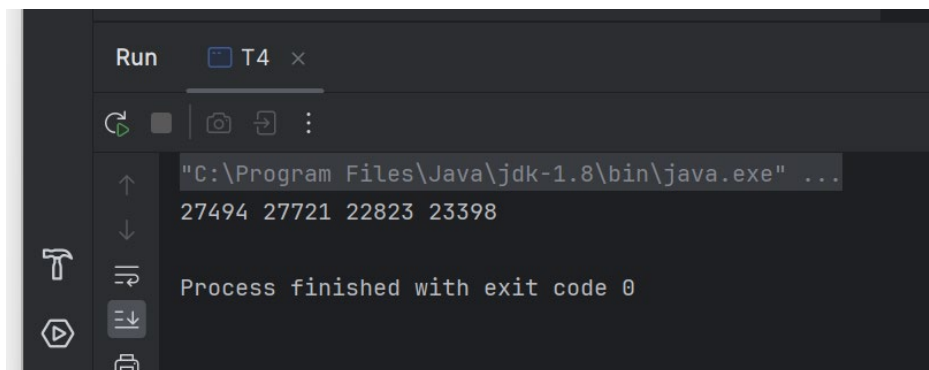
C:\Users\刘博瑞>java -version
java version "1.8.0_401"
Java(TM) SE Runtime Environment (build 1.8.0_401-b10)
Java HotSpot(TM) 64-Bit Server VM (build 25.401-b10, mixed mode)

C:\Users\刘博瑞>javac -version
javac 1.8.0_401

C:\Users\刘博瑞>
```



```
Run T2 x
"C:\Program Files\Java\jdk-1.8\bin\java.exe" ...
2023302071062 刘博瑞
Hello 2023302071062 刘博瑞
Process finished with exit code 0
StudyProject > src > CS > LiuBoRui > exam1 > T2 > main
```



```
Run T4 x
"C:\Program Files\Java\jdk-1.8\bin\java.exe" ...
27494 27721 22823 23398
Process finished with exit code 0
```


Build

Build Output x

StudyPr 2 sec, 785 ms

Checking sources
Parsing java... [StudyProject]
Writing classes... [StudyProject]
Checking dependencies... [StudyProject]
Dependency analysis found 0 affected files
Updating dependency information... [StudyProject]
Adding nullability assertions... [StudyProject]
Adding pattern assertions... [StudyProject]
Running 'after' tasks
javac 8 was used to compile java sources
Finished, saving caches...
Executing post-compile tasks...
Synchronizing output directories...
2024/3/18 16:16 - Build completed successfully

Run T7 x

0.13856488
0.31823162
0.370697
0.57468244
0.58358377
0.60550521
0.76400592
0.89103033
0.97177935

Process finished with exit code 0

Build Output x

StudyPr 2 sec, 678 ms

Running 'before' tasks
Checking sources
Parsing java... [StudyProject]
Writing classes... [StudyProject]
Checking dependencies... [StudyProject]
Dependency analysis found 0 affected files
Updating dependency information... [StudyProject]
Adding nullability assertions... [StudyProject]
Adding pattern assertions... [StudyProject]
Running 'after' tasks
javac 8 was used to compile java sources
Finished, saving caches...
Executing post-compile tasks...
Synchronizing output directories...
2024/3/18 21:14 - Build completed successfully

Run T7_ x

0.11269547
0.13856488
0.31823162
0.370697
0.57468244
0.58358377
0.60550521
0.76400592
0.89103033
0.97177935

Process finished with exit code 0

实验小结

1. 【实验结果分析】

第六题：

20! 求和一直到 1!，这个数字远大于 int 的范围，因而我们使用 int 定义函数无法得到正确结果，使用 double 则会默认为科学计数法，但是 long 就能得到正确答案。

另外，由于 java 中所有的代码都放在 public class 的内部，导致在调用函数前必须先创建对象或者使用静态 static 来定义函数成全局函数。这是 java 中正确调用函数的两个方法

采用了递归的方法，但是课上提到递归的效率比较低，但是不使用递归的方法貌似比较复杂，目前没有想到什么简单的思路

第二题

注意 System.out.println()，括号内如果有多个内容，之间用“+”连接，而不是用“，”

第五题：注意 sout 会默认换行，于是不用再加上换行符

第七题：import 库的时候，import 语句要放在 public class 之外，同时采用了 foreach 循环。库中 sort 的时间复杂度是 $O(n\log n)$ ，效率比一般的冒泡排序，选择排序...都要高

2. 【实验心得体会】

学会了安装 idea 和 jdk，配置环境变量，并且在 vscode 上装了 java 插件，两边都成功实现了 java 程序的运行。但是 cmd 的运行方法还不太会。

一些需要注意的点：和 c++ 不同，java 中的 string 是大写而非小写

由于 java 中所有的代码都放在 public class 的内部，导致在调用函数前必须先创建对象或者使用静态 static 来定义函数成全局函数。这是 java 中正确调用函数的两个方法