

# 实验 1--Java 基础与简单程序编写

## 实验内容

- 建议学时：2\*45
- 控制台下常用命令
- JDK 的配置，java、javac 命令的使用
- 基本 java 程序与命令行参数
- 控制台输入
- 数组的使用
- Java 函数的编写
- 常用类的使用
- Jdk 文档的使用
- **必须完成的题目**：1、2、3、4
- 加分考核点说明：基本加分项(画正字的一划)与特别加分项(星号)，如无标识额外加分项均为基本加分项。

## 题目 1—控制台下常用命令(详细讲解)

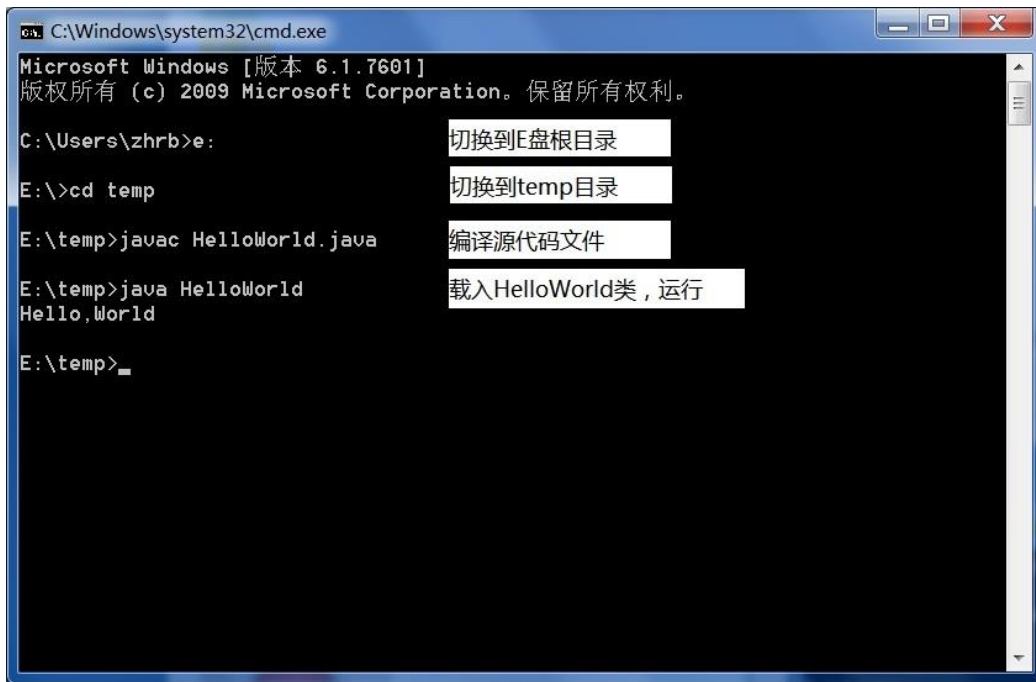
启动**控制台**：Win+R 键，然后输入 **cmd** 回车可以启动控制台。启动控制台后，会显示**命令行提示符**，如：“c:\temp>”，表示现在位于 c 盘的 temp 目录，其中 c:为**盘符**。

常用命令如下：

1. dir            显示当前目录下所有内容  
   dir \*.java    显示当前目录下所有扩展名为 java 的文件
2. c:\temp\java> d:    //从当前的 c 盘切换到 d 盘
3. cd            进入某个目录，如：  
   c:\temp\java>cd abc        //进入 temp\java 目录下的 abc 子目录  
   c:\temp\java>cd ..        //进入 temp\java 的父目录，即 temp 目录  
   c:\temp\java>cd \        //进入根目录，在这里代表进入 c 盘根目录
4. 关于 java 的命令  
   java -version            //查看当前系统使用的 jre 版本,javac -version 查看编译器版本  
   javac HelloWorld.java    //对 HelloWorld.java 源文件进行编译  
   java HelloWorld        //将 HelloWorld 类载入虚拟机（即运行 HelloWorld）

一个完整的命令执行序列如下图：

说明：“c:\Users\zhrb”为**命令行提示符**，表示现在位于 c:\Users\zhrb 目录下，不需要输入。



```
C:\Windows\system32\cmd.exe
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\zhrb>:
E:\>cd temp
E:\temp>javac HelloWorld.java
E:\temp>java HelloWorld
Hello,World
E:\temp>_
```

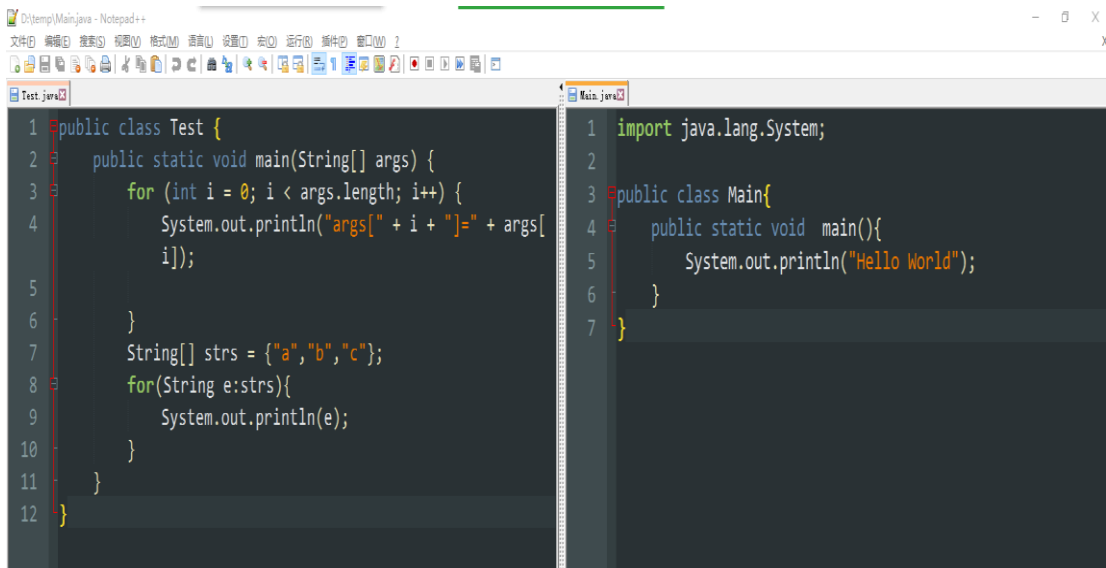
5.控制台几个常用的操作快捷键:

“上”键, 重复输入以前输入的命令。

“tab”键, 自动补全命令或者文件名, 可以多按几次在可选的文件名进行切换。

## 学会使用 Notepad++编写 HelloWorld.java

1. 字体尺寸更改: `ctrl + 鼠标滚轮`
2. 主题: 设置-语言格式设置, 推荐 Obsidian 主题, 推荐 Consolas 11 号字体, 勾选全局字体。
3. 打开当前文件所在目录(命令行): 右键单击标签页, 打开所在文件夹(命令行)。
4. 切换标签页: `ctrl + tab`
5. 双视图查看文件: 打开两个文件, 右键单击标签页, 复制到另一个视图。效果如下图:



```
1 public class Test {
2     public static void main(String[] args) {
3         for (int i = 0; i < args.length; i++) {
4             System.out.println("args[" + i + "]=" + args[
5                 i]);
6         }
7         String[] strs = {"a", "b", "c"};
8         for (String e: strs) {
9             System.out.println(e);
10        }
11    }
12 }
```

```
1 import java.lang.System;
2
3 public class Main{
4     public static void main(){
5         System.out.println("Hello World");
6     }
7 }
```

6. 菜单-视图-函数列表（可以快速定位函数）

7. 快捷键:

删除行 **Ctrl+L**, 复制行 **Ctrl+D**

注释/取消注释 **Ctrl+Q**,

列编辑 **Alt+** 鼠标左键, 列模式 **Alt+C**

查找下一个 **F3**, 查找上一个 **Shift+F3**

定位匹配括号 **Ctrl+B**

代码缩进: **Tab** 往右缩进, 往左缩进 **Shift+Tab**。

快速选择文本: 1.按住 **shift**, 然后移动光标或按 **Home**, **End** 等键选取一行。2. 双击某个单词, 可以快速选择该单词。

#### Java 命名规范:

1.类名: 类的首字母要大写, 如果包含多个单词, 每个单词的首字母要大写, 如 **HelloWorld** 而不是 **helloworld**。

2.包名: 要小写, 比如 `package test;`或者 `package lib`。

3.方法名: 首字母要小写, 后面的单词要大写, 比如应该为 `printName`, 而不是 `printname` 或者 `PrintName`

详见: `Java 代码规范.rar`

## 题目 2—JDK 的配置与 HelloWorld

1. 安装与配置 JDK

设置 `path` 变量

使用 `java -verson` 查看版本号, 确认 `jre` 是否装好

使用 `where` 命令查看 `java`、`javac` 等是否已经可执行。

在命令行下对源文件进行编译(`javac`)、运行字节码(`java`)

**注意:** 编译或者运行时碰到错误, 请参见“`Java HelloWorld FAQ.txt`”。

## 题目 3-获取输入

先演示 InputTest.java, 然后讲解:

### 任务 1: 编写程序 1(控制台交互):

编写 GetInput.java, 需实现如下功能“从控制台输入两个浮点数, 将两个数值读取、转换然后相加, 最后输出运算。”

#### 参考资料:

阅读 InputTest.java, 注意如下要点:

Scanner 类中的 nextDouble()方法用于从命令行读取并转化为浮点数。

```
int a = Integer.parseInt("123"); //将字符串 123 转化成整型 123
```

扩展: Scanner 类的使用.txt(进阶), 掌握其中的 nextLine()方法、nextInt()方法、next()方法。

**小任务:** 从控台循环输入两个数然后两两相加输出, 具体见 PTA。

### 任务 2: 编写程序 2(从命令行获取参数):

1.从控制台获得参数参考源代码:

```
public class CommArg {
    public static void main(String args[]) {
        // 显示控制台传递进来的所有参数
        if (args.length > 0) { // 如果命令行有参数
            for (int i = 0; i < args.length; i++) {
                System.out.println(args[i] + " ");
            }
        } else { // 如果命令行没有参数
            System.out.println("No arguments!");
        }
    }
}
```

运行程序 (命令行参数获取):

```
java CommArg first second third
```

运行结果:

```
first second third
```

注意: 将字符串转化为 double 的方法, 查询 jdk 文档 Double 的 parseDouble 方法。

**2.加分进阶:** 考虑输入参数个数不足的情况, 当执行 java ReadFromConsole, 因为没有输入参数, 应该提示用户该命令的正确用法。

**小任务:** 编写 ReadFromConsole.java 从控制台获取两个参数, 将两个参数转化成 double 型并相加, 最后输出。

如: 命令行执行 java ReadFromConsole 123 456, 那么 123 与 321 即为程序的输入参数, 输出结果为 579

## 题目 4-取数字(PTA)

编写一个程序,从键盘或者命令行参数读取该数值,如-123(不局限于该数),然后-123 中的每一个数字都抽出来并打印出来,并将每个数字加总求和。

**参考资料:**

处理控制台输入,参考 InputTest.java

1.编写两个不同功能的函数分别实现本题目要求:

`public static void printDigit(int n)`方法,将 `n` 的每个数字输出,并将其和输出。

输入: -123 输出: 1 2 3 6

`public static int[] getDigits(int n)`方法,将 `n` 的每个数字抽取出来并放入 `int` 数组,并返回。

2.对传递进来的 10 进制整数,可以将其转换为 2 进制、8 进制、16 进制,并输出。(提示:查询 jdk 文档, Integer 的使用方法)

输入: 10000 输出: 10011100010000,23420,2710

参考: 格式输出使用 `System.out.printf`, 类似 c 语言中的 `printf`

3.**加分考核点:** 改为一通用的程序,对传递进来的任意位数的 10 进制数浮点数(要考虑传递进来的参数可能是负数),抽取出其中的数字,然后将这些数字相加输出结果。

输入: -123.02 输出: 8

提示: 查询 JDK 文档 Integer, Double, Scanner(nextInt, hasNextInt, nextDouble....),String, Character, System.out.printf。

提示:

将 `int` 型转换成 `String` 型, 查询 jdk 文档

方法 1: `""+1`

方法 2: 查询 jdk 文档 Integer 的 `toString` 方法或者 `String` 的 `valueOf` 方法。

`Integer` 类的 `parseInt()`方法,将字符串转换成 `int` 型。

`Scanner` 的 `nextInt()`方法,可以将字符串转化为 `int` 型。

## 选做: 题目 5-编写开根号程序(PTA 上已有)

1.使用逐步逼近法编写函数 `sqrt` 对传入的 `x` 进行开根号

在类 `MySqrt` 中编写

```
public static double sqrt(double x){}
```

注意要考虑以下几种情况:

1. 如何处理 `x` 为负数的情况
2. 如何处理 `x==0` 与 `x==1` 的情况
3. 如何处理  $0 < x < 1$  的情况

注意:

1.查看自己的输出结果与 `Math.sqrt` 的输出结果有何区别?

2. **特别加分:** 可以使用牛顿迭代法与二分逼近法解决该问题。考核时需要讲清楚算法实现

原理。