

零.今天的主要目的是

用Java语言写第一个程序HelloWorld

## 一.计算机基础知识(计算机概述)(了解)

### 1.硬件:

\* 冯.诺依曼体系结构 计算机的硬件分成5大组成部件：运算器、控制器、存储器、输入设备和输出设备。

\* 运算器和控制器是计算机的核心，合称中央处理单元（Central Processing Unit，CPU）或处理器。

### 2.软件:

软件定义:

\* 按照特定顺序组织的计算机数据和指令的集合

软件分类:

\* 系统软件

\* 应用软件

软件开发:

\* 借助**开发工具**与**计算机语言**制作软件

|-机器语言

|-汇编语言

|-高级语言

人机交互:

a:命令行方式

b:图形化界面方式

## 3. **键盘功能键和快捷键(能掌握多少掌握多少,越多越好)**

\* A:键盘功能键

\* a:Tab

\* b:Shift

\* c:Ctrl

\* d:Alt

\* e:空格

\* f:Enter

\* g:Window

\* h:上下左右键

\* i:PrtSc(PrintScreen)屏幕截图

\*F5 刷新

\*F2 重命名

\*Home 行头

\*End 行尾

## \*B:键盘快捷键

- \* Ctrl+A 全选
- \* Ctrl+C 复制
- \* Ctrl+V 粘贴
- \* Ctrl+X 剪切
- \* Ctrl+Z 撤销
- \* Ctrl+Y 后悔
- \* Ctrl+S 保存
- \* Ctrl+N 打开相同目录的我的电脑等
- \* Alt+Tab 切屏
- \* Win+E 打开我的电脑
- \* Win+R 打开运行窗
- \* Win+D 桌面切换

## 4.打开DOS控制台

windows+r 输入CMD 回车

windows 输入CMD 回车

## 5.常见的DOS命令

\* d: 回车 盘符切换

\* dir(directory):列出当前目录下的文件以及文件夹

\* cd (change directory)改变指定目录(进入指定目录)

\* cd.. : 退回到上一级目录

\* cd\ : 退回到根目录

\* cls : (clear screen)清屏

\* exit : 退出dos命令行(分割线上的需要掌握,下的了解)

\* ping ip/主机

\* config -all 查看IP信息

\* /==删==除==要==注==意==这==里==不==走==回==收==站==不==要==乱==删==除  
=====

\* start :打开一个新的命令行

\* md (make directory) : 创建目录

\* rd (remove directory): 删除目录

\* del (delete): 删除文件,删除一堆后缀名一样的文件\*.txt

\* notepad 创建文件

\* 删除带内容的文件夹

\* rd + /s 文件夹名称(询问是否删除)

\* rd + /q + /s 文件夹名称(直接删除)

## 二.Java语言基础

### 1.Java语言平台

\* J2SE(Java 2 Platform Standard Edition) 标准版

\* 是为开发普通桌面和商务应用程序提供的解决方案,该技术体系是其他两者的基础,

可以完成一些桌面应用程序 的开发

\* J2ME(Java 2 Platform Micro Edition)小型版

\* 是为开发电子消费产品和嵌入式设备提供的解决方案

\* J2EE(Java 2 Platform Enterprise Edition)企业版

\* 是为开发企业环境下的应用程序提供的一套解决方案,该技术体系中包含的技术如 Servlet、Jsp等,主要针对于 Web应用程序开发

## 2:Java语言特点

- \* 简单性
- \* 解释性
- \* 面向对象
- \* 高性能
- \* 分布式处理
- \* 多线程
- \* 健壮性
- \* 动态
- \* 结构中立
- \* 安全性
- \* 开源
- \* 跨平台

## 3.Java语言跨平台原理(重要)

\* 只要在需要运行java应用程序的操作系统上,先安装一个Java虚拟机(JVM Java Virtual Machine)即可。由JVM来负责Java程序在该系统中的运行。write once ,run anywhere!(一处编译,到处运行)

## 4.JRE和JDK和JVM(重要)

JRE:包括Java虚拟机(JVM Java Virtual Machine)和Java程序所需的核心类库等

如果想要运行一个开发好的Java程序,计算机中只需要安装JRE即可。

JDK:JDK是提供给Java开发人员使用的,其中包含了java的开发工具,也包括了JRE。

所以安装了JDK,就不用单独安装JRE了。

其中的开发工具:编译工具(javac.exe) 打包工具(jar.exe)等

可以简单的理解为:

JDK=开发工具+JRE

JRE=核心类库+JVM

JVM虚拟机,来实跨平台

## 5.JDK的下载和安装(官网 <http://www.oracle.com>)

## 6.JDK目录的了解

bin:放一些可执行的程序

db:java自带的一个数据库(JDK6.0开始)

jre:java运行环境

src:java特点开源包

7.Java开发工具介绍.....现在我们用notepad...txt文本和Editplus,以后我们用eclipse和其它

### 三.HelloWorld案例(重要)

- \* A:定义类
- \* B:写main方法
- \* C:写输出语句
- \* D:Java程序开发运行与工作原理
- \* E:编译和运行程序

```
1. class HelloWorld {
2.     public static void main(String[] args) {
3.         System.out.println("HelloWorld");
4.     }
5. }
```

#### 1.HelloWorld的常见错误

文件名写错了/抄的时候抄错了/括号不配对/中文标点

#### 2.Java语言的书写格式(约定俗成,约定由于编码) (掌握)

- \* 1,大括号要对齐,并且成对写
- \* 2,左大括号前面有空格
- \* 3,遇到左大括号要缩进
- \* 4,方法和程序块之间加空行让程序看起来清晰
- \* 5,并排语句之间加空格,例如for语句
- \* 6,运算符两侧加空格

#### 3.path环境变量的作用及配置方式(重要)

\* 右键点击桌面计算机→选择属性→选择高级系统设置→选择高级选项卡→点击环境变量→添加JAVA\_HOME变量,值为JDK目录→确定→系统变量中查找path→双击path→将%JAVA\_HOME%\bin目录添加到最左边并添加分号。

\* path配置的是可执行的文件.exe,配置后可以在不同的盘符下访问path路径下的可执行文件

#### 4.classpath环境变量的作用及其配置(了解)

\* path配置的是可执行的文件.exe,配置后可以在不同的盘符下访问path路径下的可执行文件

\* classpath配置的java的类文件,就是.class文件

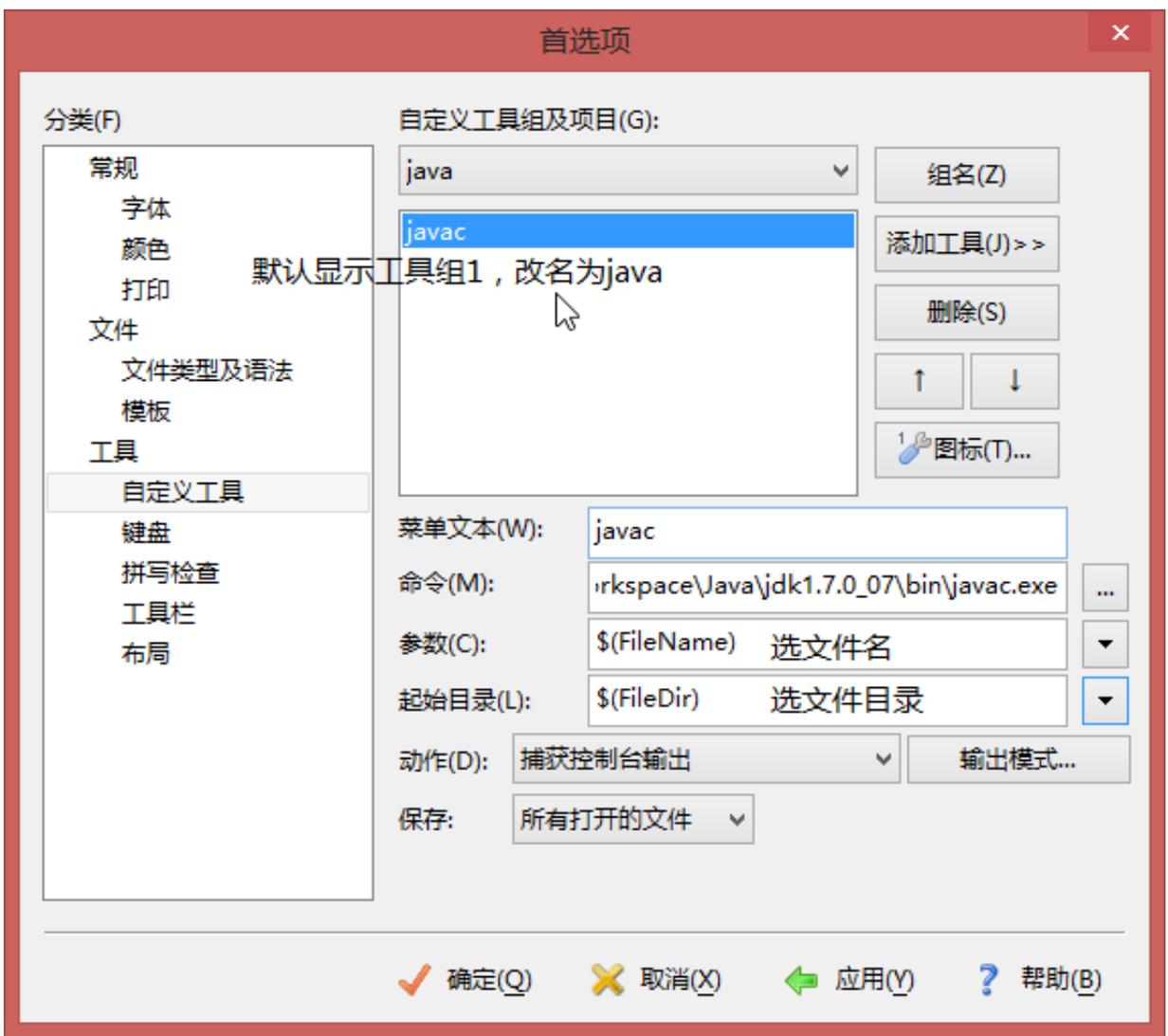
\*classpath加上分号和不加分号的区别

\*如果不加分号只去classpath路径找

\*如果加分号,先去classpath,然后再在当前路径找

#### 5.Editplus开发程序并编译运行(了解)

\* A:配置快捷键编译运行



具

首选项



分类(F)

- 常规
  - 字体
  - 颜色
  - 打印
- 文件
  - 文件类型及语法
  - 模板
- 工具
  - 自定义工具
  - 键盘
  - 拼写检查
  - 工具栏
  - 布局

自定义工具组及项目(G):

工具组 1

- javac
- java

组名(Z)

添加工具(I) >>

删除(S)

↑ ↓

图标(T)...

菜单文本(W): java

命令(M): orkspace\Java\jdk1.7.0\_07\bin\java.exe ...

参数(C): \$(FileNameNoExt) ▼

起始目录(L): \$(FileDir) ▼

动作(D): 无 英 ▼ 输出模式...

保存: 所有打开的文件 ▼

✓ 确定(Q)   ✕ 取消(X)   ← 应用(Y)   ? 帮助(B)

# 首选项



## 分类(F)

- 常规
  - 字体
  - 颜色
  - 打印
- 文件
  - 文件类型及语法
  - 模板
- 工具
  - 自定义工具
  - 键盘
  - 拼写检查
  - 工具栏
  - 布局

自动保存文件时间(每0-100分钟):

0.0

自动保存为 \*.tmp 文件

新文件默认类型:

PC

默认文本编码:

系统默认

把这里的对号去掉就可以

在保存文件时创建备份(F)

备份选项...

在保存时检查文件是否被变动

从其它程序切换到 EditPlus 时检查文件是否变动

在检查变动时锁定文件

自动重新加载变动的文件(Z)

检测文件是否被删除

在 ASP 网页中默认使用 JScript 语言

阻止编辑只读的文件(D)

在加载文件时检查错误的回车换行符和空字符

不记录最近编辑的文件

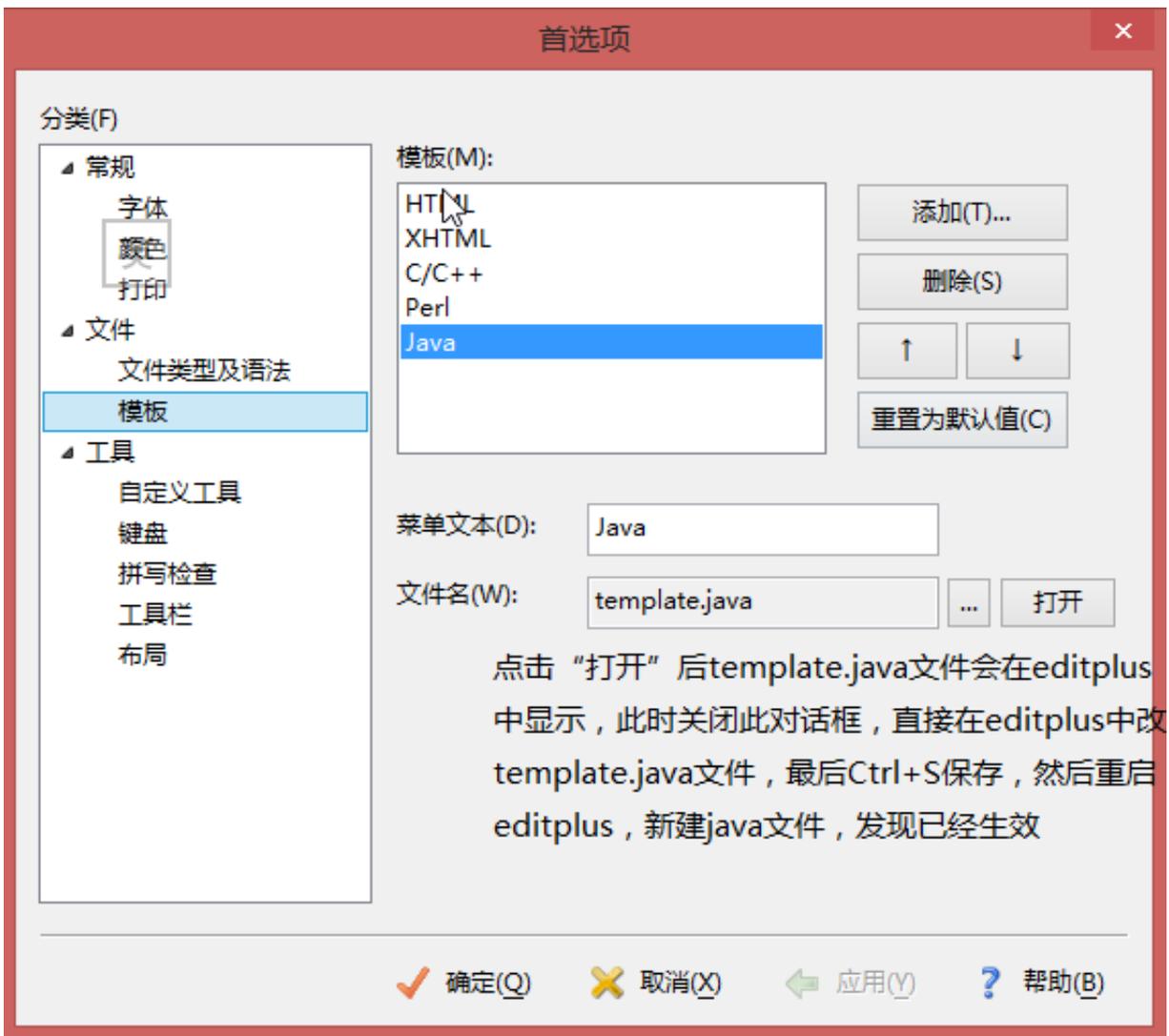
文件未被修改时禁用“保存”按钮

✓ 确定(Q)

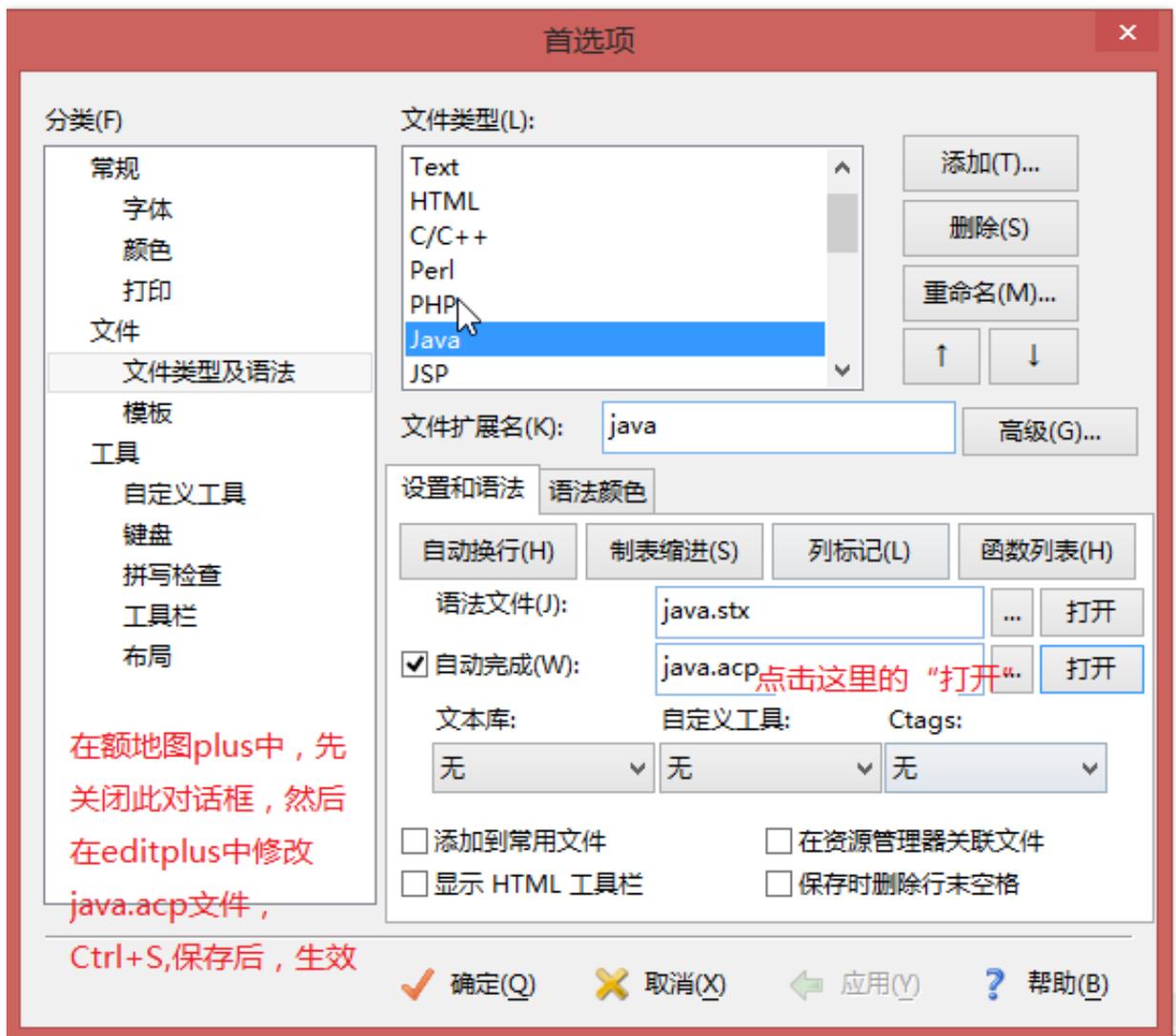
✗ 取消(X)

← 应用(Y)

? 帮助(B)



)



## 6. 注释概述及其分类(掌握)

- \* A: 什么是注释, 我们一般叫注释说明, 不影响我们程序的运行
- \* B: 注释的分类及讲解
  - \* 单行 //内容
  - \* 多行 /\* 内容 \*/
  - \* 文档 /\*\* 内容 \*/
- \* C: 注释的作用
  - \* A: 解释说明程序
  - \* B: 帮助我们调试错误

## 7. 关键字的概述和使用(掌握)

- \* A: 被Java语言赋予特定含义的单词
- \* B: 关键字的特点 : 组成关键字的字母全部小写
- \* C: 关键字的注意事项
  - \* 起名字的时候就不能用了.
  - \* goto和const作为保留字存在, 目前并不使用, 类似Editplus这样的高级记事本, 针对关键字有特殊的颜色标记, 非常直观

## 8. 标识符的概述和组成规则(掌握)

- \* A: 标识符: 就是给类, 接口, 方法, 变量等起名字时使用的字符序列

## 标识符的合法性

### \* B:标识符的组成规则

- \* 英文大小写字母

- \* 数字字符

- \* \$和\_

### \* C:标识符注意事项

- \* 1,不能使用关键字

- \* 2,不能数字开头