

## Lab 1

### ● 要点：

1. 配置使用代码编辑器 EditPlus。
2. 编写第一个 java 程序。
3. 两种方式编译运行程序。
4. 熟悉 JOptionPane 的几个简单运用。(推荐，但不要求)

### ● 具体内容

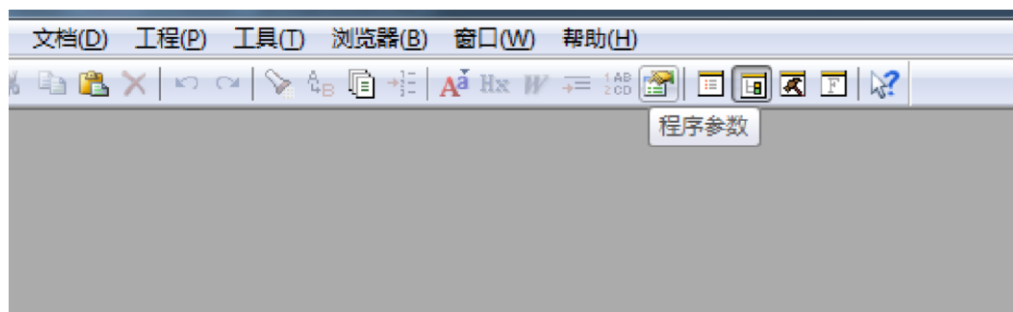
#### 1.安装 EditPlus

该软件已经放在 lab 文档同一目录下(Lab1->EditPlus 3，运行其中的 EditPlus.exe 文件，之后双击注册.reg 文件完成注册)。

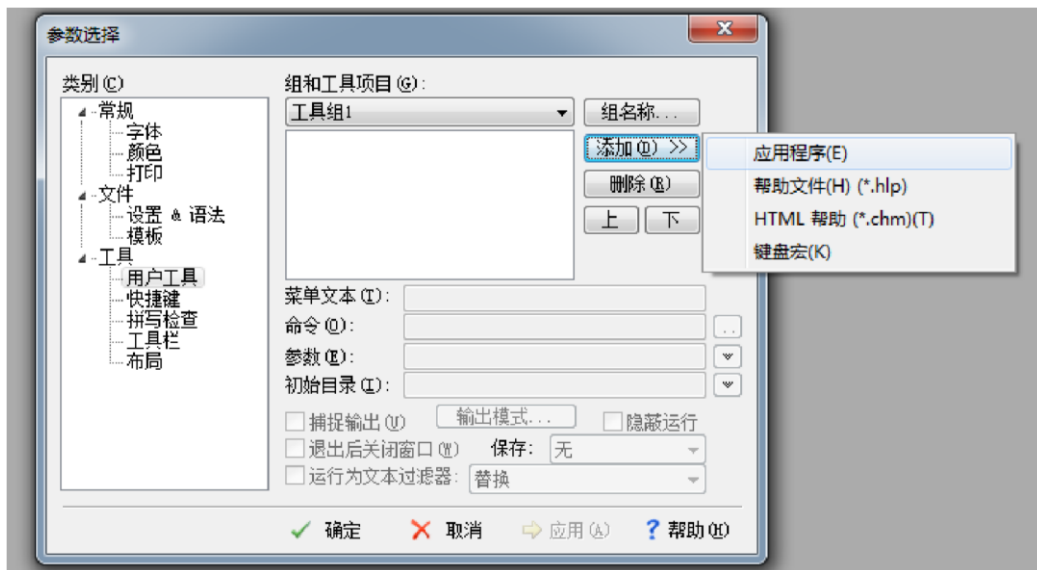
使用 EditPlus 编辑 Java 代码前需要配置 java 使用环境，具体操作流程参见目录下的视频。

提醒：本学期 lab 和 project 基本都可以使用 Editplus 完成，请大家务必熟悉该软件的使用。

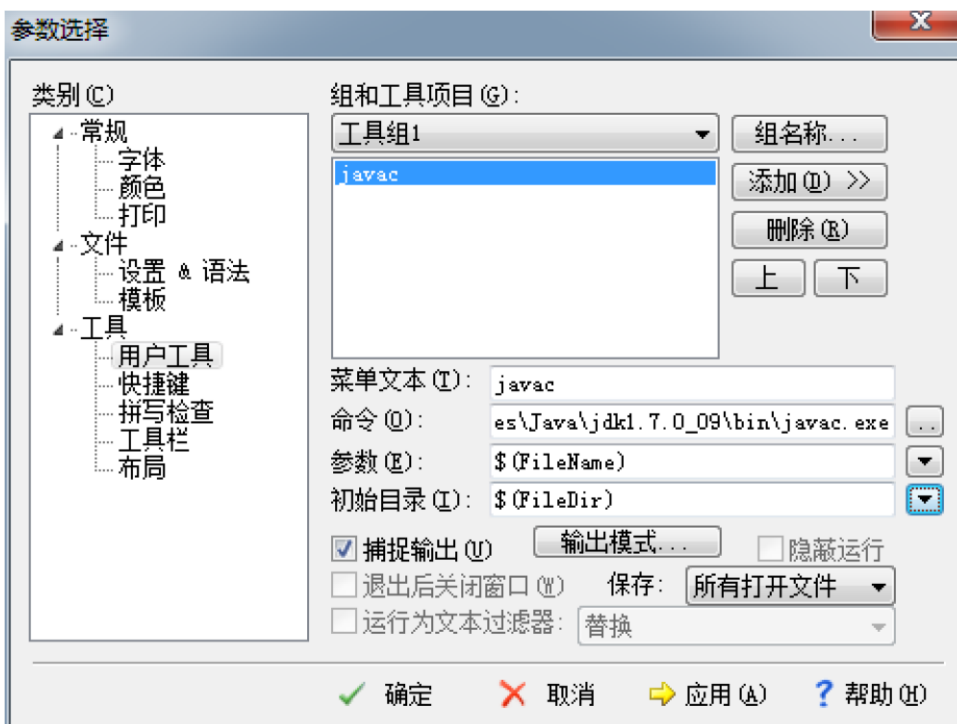
- 1.1 点击工具栏中的“程序参数”按钮（配置前请确认电脑中的 Java 环境已经安装完毕，否则请参照 Lab0）



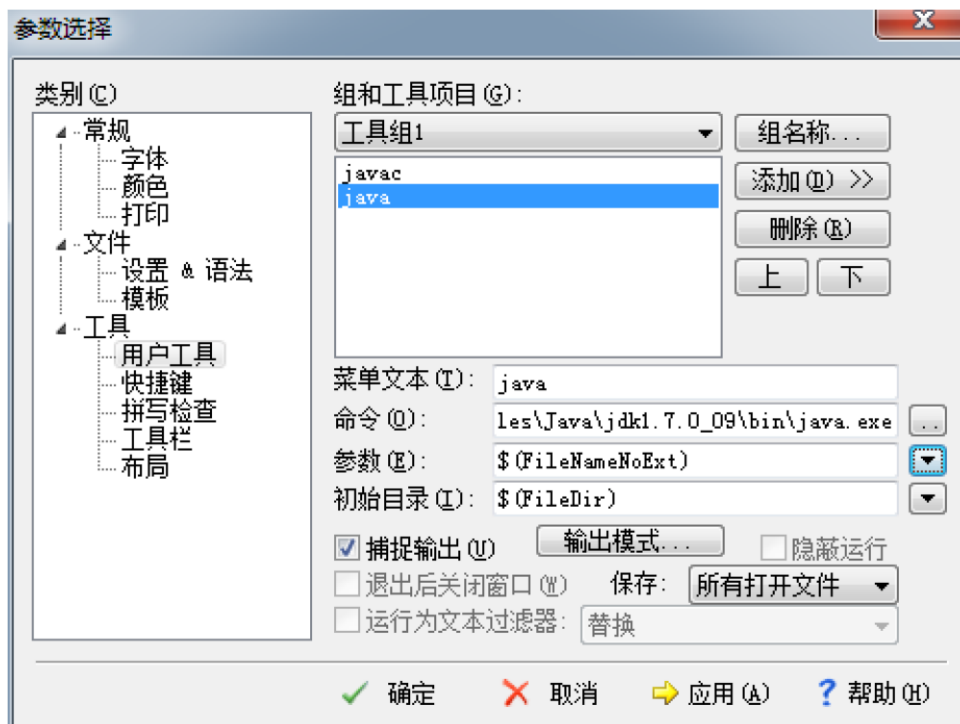
- 1.2 在弹出的窗口中的左侧类别中选择“用户工具”，然后在右侧点击“添加”按钮并在弹出的菜单中选择“应用程序”



- 1.3 然后在下方依次进行如下填写：“菜单文本”填写“javac”；“命令”通过右侧的目录选择按钮找到你的机器上 JDK 的安装目录下 bin 文件夹中的 javac.exe（目录一般为 C:\Program Files\Java\jdk1.7.0\_09\bin\javac.exe 具体因机器而异）；“参数”一栏点击右侧的下拉按钮选择“文件名”；“初始目录”一栏通过右侧下拉列表选择“文件目录”；最后点选下方的“捕捉输出”。



- 1.4 再次点击 添加 → 应用程序：“菜单文本”填写“java”；“命令”选择相同目录下的 java.exe；“参数”一栏选择“文件名（不含扩展名）”；“初始目录”一栏选择“文件目录”；同样点选下方的“捕捉输出”。最后点击“确定”



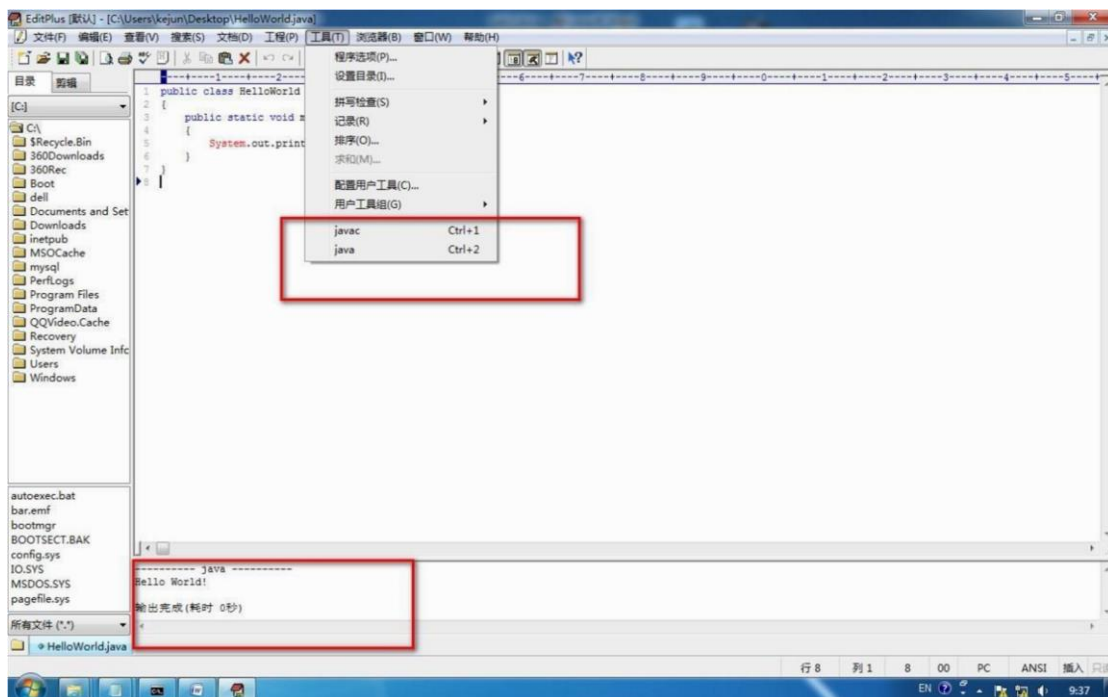
## 2.编写第一个 java 程序 HelloWorld :

```
1 public class HelloWorld
2 {
3     public static void main(String[] args)
4     {
5         System.out.println("Hello World!");
6     }
7 }
8 |
```

## 3.编译运行 java 程序 :

### 3.1 Editplus 配置好的功能模块运行 (推荐) :

先选择 javac, 等到下方界面中提示编译完成后(大概需要等待 30S 左右)再选择 java



### 3.2 CMD 界面下的编译运行：

本次 lab 需要用到的 DOS 命令简单介绍：

**cd**——改变当前目录，进入特定的目录，比如“cd..”表示退回到上级目录，“cd 子目录路径”表示进入该子目录，“D:”表示进入 D 盘根目录，如果是“G:”则表示进入 G 盘根目录，此处依据各人电脑而异，如下图：

```
C:\Users\ZJ>cd..

C:\Users>cd..

C:\>cd Users

C:\Users>D:

D:\>G:

G:\>
```

**dir**——列出当前目录中的文件，进入特定的目录后，使用 **dir** 命令，则可以得到该目录下所有文件

```

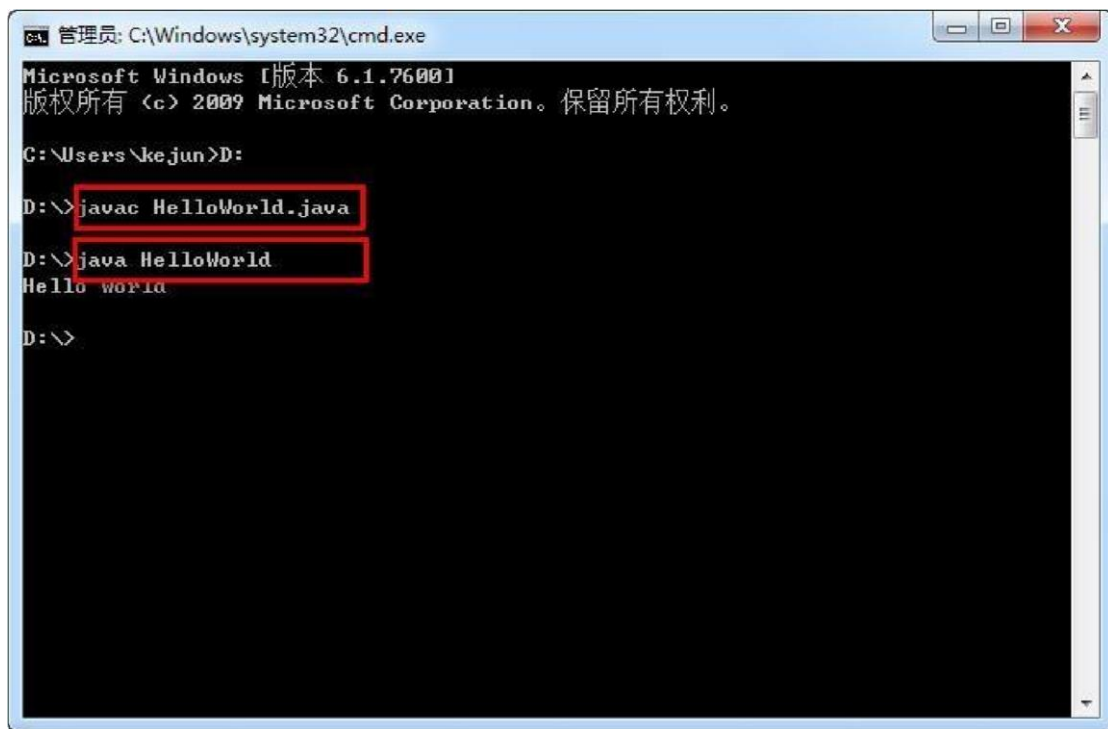
G:\>cd java助教
G:\java助教>cd labs
G:\java助教\labs>cd Lab1
G:\java助教\labs\Lab1>cd EditPlus 3
系统找不到指定的路径。
G:\java助教\labs\Lab1>cd EditPlus 3
G:\java助教\labs\Lab1\EditPlus 3>dir
驱动器 G 中的卷没有标签。
卷的序列号是 9866-08AE

G:\java助教\labs\Lab1\EditPlus 3 的目录

2013/09/15  16:12    <DIR>          .
2013/09/15  16:12    <DIR>          ..
2012/09/18  10:08             264 ?>卸载.reg
2012/09/18  10:08             802 ?>注册.reg
2012/09/18  10:08           3,148 ansi.ctl
2012/09/18  10:08           4,195 codepage.txt
2012/09/18  10:08             804 control.ctl
2012/09/18  10:08          21,491 correct.tlx
2012/09/18  10:08             407 cpp.acp

```

- 1) 开始—>运行—>cmd
- 2) 首先通过 CD 命令进入存储程序的目录
- 3) 输入命令：javac HelloWorld.java 编译程序
- 4) 输入命令：java HelloWorld 装载运行程序



```
管理员: C:\Windows\system32\cmd.exe
Microsoft Windows [版本 6.1.7600]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\kejun>D:
D:\>javac HelloWorld.java
D:\>java HelloWorld
Hello world
D:\>
```

#### 4. JOptionPane 的简单应用 (推荐, 但不要求)

JOptionPane 是 Java 众多的系统类之一, 通过可以在对话框中显示、读取任何文本。课本上只介绍了显示文本框这个例子, 大家可以用 Java API ([JDK\\_API\\_1\\_6\\_zh\\_CN.chm](#)) 查看所有用法。

##### 4.1 打开 Java API 文档后, 选择左上角的“显示”按钮



4.2 然后会发现左侧出现索引栏，输入想要查阅的类的名字，例如 JOptionPane，然后在下方双击选择第一个。



4.3 在弹出的窗口中选择“显示”后会在右侧看到该类的基本信息

[java.lang.Object](#)

- [L java.awt.Component](#)
- [L java.awt.Container](#)
  - [L javax.swing.JComponent](#)
    - [L javax.swing.JOptionPane](#)

所有已实现的接口：  
[ImageObserver](#), [MenuContainer](#), [Serializable](#), [Accessible](#)

---

```
public class JOptionPane
extends JComponent
implements Accessible
```

JOptionPane 有助于方便地弹出要求用户提供值或向其发出通知的标准对话框。有关使用 JOptionPane 的信息，请参见 *The Java Tutorial* 中的 [How to Make Dialogs](#) 一节。

虽然由于方法数多使 JOptionPane 类可能显得复杂，但几乎所有此类的使用都是对下列静态 showXxxDialog 方法之一的单行调用：

方法名	描述
showConfirmDialog	询问一个确认问题，如 yes/no/cancel。
showInputDialog	提示要求某些输入。
showMessageDialog	告知用户某事已发生。
showOptionDialog	上述三项的大统一（Grand Unification）。

所有这些方法还可能以 showInternalXXX 风格出现，该风格使用内部窗体来保存对话框（请参见 [JInternalFrame](#)）。此外还定义了多种便捷方法，这些方法重载那些基本方法，使用不同的参数列表。

所有对话框都是有模式的。在用户交互完成之前，每个 showXxxDialog 方法都一直阻塞调用者。

以下是几个简单的示例：

1.







## ● 检查内容

1. 配置好 EditPlus 后，在 EditPlus 上编写、编译、运行 HelloWorld 程序，演示效果。  
( 6' )
2. 在 cmd 上编译、运行 HelloWorld 程序，演示效果。(4')
3. 任取 4 个本 lab 给出的 JOptionPane 示例完成代码，演示效果。

## ● 截止时间

检查内容中 1,2 要求在 lab 课内完成，当堂提交 TA 检查给分。

有兴趣的同学可以在课上或者课后完成检查内容中 3，将代码文件以“学号\_姓名”（例：13302010XXX\_张三）上传到 FTP 的目录

**“/classes/13/131 程序设计（戴开宇）/WORK\_UPLOAD/Lab1/”**