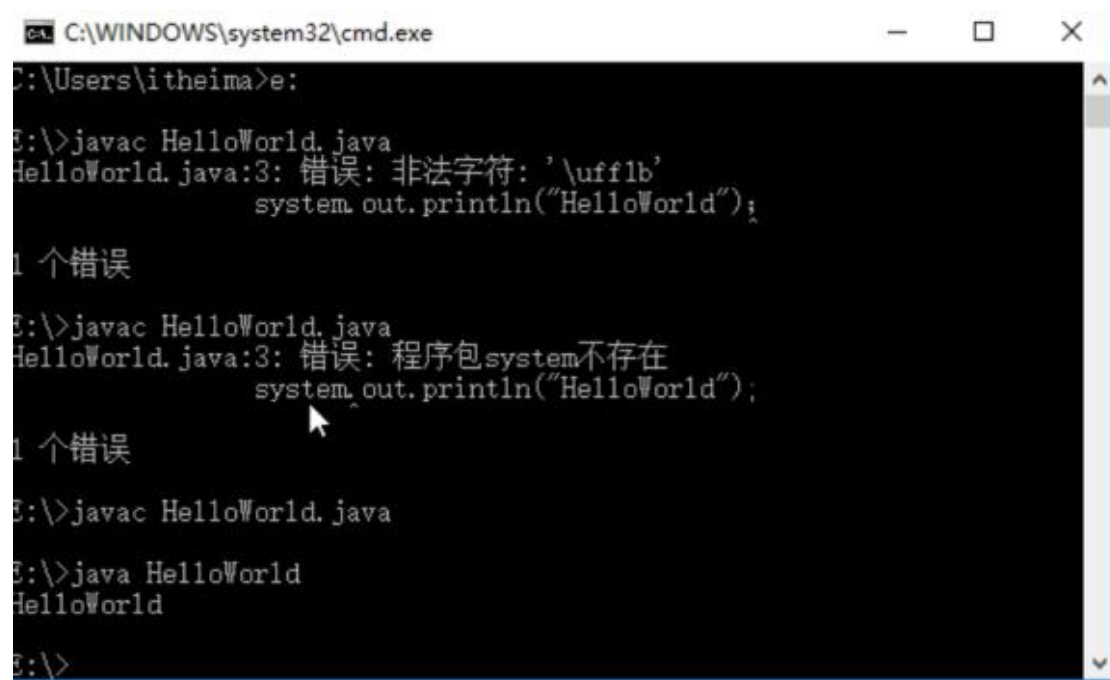


HelloWorld 错误案例



```
C:\WINDOWS\system32\cmd.exe
C:\Users\itheima>
E:\>javac HelloWorld.java
HelloWorld.java:3: 错误: 非法字符: '\ufffb'
    system.out.println("HelloWorld");
                        ^
1 个错误

E:\>javac HelloWorld.java
HelloWorld.java:3: 错误: 程序包system不存在
    system.out.println("HelloWorld");
    ^
1 个错误

E:\>javac HelloWorld.java

E:\>java HelloWorld
HelloWorld

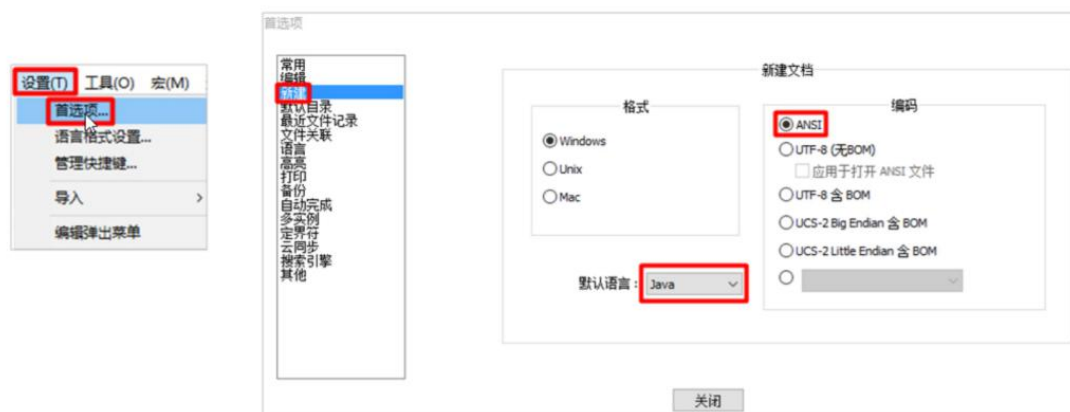
E:\>
```

“非法字符”：中英文符号错误

“程序包不存在” 英文拼错或者大小写出错

5.3 Notepad软件的配置

安装完毕之后，为了使用方便，做一个简单的配置：修改默认语言和编码



Notepad++软件配置设置（不配置可能会出现乱码现象）

1.2 注释分类

- 单行注释

格式：// 注释信息

- 多行注释

格式：/* 注释信息 */

- 文档注释

格式：/** 注释信息 */

文档注释目前用不上，暂不讲解。

```
D:\>javac Dy.java
Dy.java:8: 错误: 类重复: Dy
public class Dy{
      ^
1 个错误
D:\>
```

错误：类名重复

我们知道计算机是可以用来存储数据的，但是无论是内存还是硬盘，计算机存储设备的最小信息单元叫“位 (bit)”，我们又称之为“比特位”，通常用小写的字母“b”表示。而计算机中最小的存储单元叫“字节 (byte)”，通常用大写字母“B”表示，字节是由连续的8个位组成。

除了字节外还有一些常用的存储单位，大家比较熟悉，我们一起来看看：

1B (字节) = 8bit
1KB = 1024B
1MB = 1024KB
1GB = 1024MB
1TB = 1024GB



4.3 数据类型内存占用和取值范围

数据类型	关键字	内存占用	取值范围
整数	byte	1	-128~127
	short	2	-32768~32767
	int(默认)	4	-2的31次方到2的31次方-1
	long	8	-2的63次方到2的63次方-1
浮点数	float	4	负数: -3.402823E+38到-1.401298E-45 正数: 1.401298E-45到3.402823E+38
	double(默认)	8	负数: -1.797693E+308到-4.9000000E-324 正数: 4.9000000E-324到1.797693E+308
字符	char	2	0-65535
布尔	boolean	1	true, false

说明：E+38表示是乘以10的38次方，同样，E-45表示乘以10的负45次方

红色字体表示的“E+38；E-45”是科学计数法

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [版本 10.0.17134.407]
(c) 2018 Microsoft Corporation。保留所有权利。

C:\Users\itheima>e:

E:\>javac VariableDemo02.java
VariableDemo02.java:31: 错误: 已在方法 main(String[])中定义了变量 b
    boolean b = true;
           ^
1 个错误

E:\>_
```

变量名重复错误

```
E:\>javac VariableDemo02.java
VariableDemo02.java:38: 错误: 可能尚未初始化变量l
    System.out.println(l);
                       ^
1 个错误

E:\>_
```

没有对变量名赋值错误

```
E:\>javac VariableDemo02.java
VariableDemo02.java:37: 错误: 整数太大
    long l = 100000000000;
           ^
1 个错误

E:\>
```

整数默认类型是 int 类型，为了防止数据过大，给 long 类型变量赋值的时候需要加一个“L”

```
E:\>javac VariableDemo02.java
VariableDemo02.java:43: 错误: 不兼容的类型: 从double转换到float可能会有损失
    float f = 13.14;
           ^
1 个错误

E:\>_
```

浮点数默认类型是 double 类型，为了防止给 float 类型变量赋值时不兼容，需要加一个“F”

变量使用的注意事项:

- 在同一对花括号中，变量名不能重复。
- 变量的使用原则：先声明、赋值（初始化），后使用；声明赋值可以分步（分行）进行。
- 一行可以同时定义多个同类型的变量，中间用逗号隔开。
- 定义long类型的变量时，为了防止整数过大，需要在整数的后面加L（大小写均可，建议大写）。因为整数默认是int类型，整数太大可能超出int范围。
- 定义float类型的变量时，为了防止类型不兼容，需要在小数的后面加F（大小写均可，建议大写）。因为浮点数的默认类型是double，double的取值范围是大于float的，类型不兼容。

6.2 标识符定义规则

- 由**数字、字母、下划线(_)**和**美元符(\$)**组成
- 不能以数字开头
- 不能是关键字
- 区分大小写

6.3 常见命名约定

小驼峰命名法: **方法、变量**

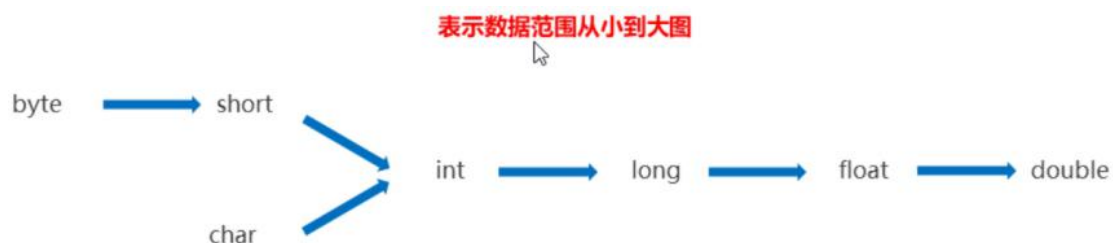
- 约定1: 标识符是一个单词的时候, 首字母小写
- 范例1: **name**
- 约定2: 标识符由多个单词组成的时候, 第一个单词首字母小写, 其他单词首字母大写
- 范例2: **firstName**

大驼峰命名法: **类**

- 约定1: 标识符是一个单词的时候, 首字母大写
- 范例1: **Student**
- 约定2: 标识符由多个单词组成的时候, 每个单词的首字母大写
- 范例2: **GoodStudent**

7.2 自动类型转换

把一个表示数据**范围小的数值**或者**变量**赋值给另一个表示数据**范围大的变量**



7.3 强制类型转换

把一个表示数据**范围大的数值**或者**变量**赋值给另一个表示数据**范围小的变量**

- 格式：**目标数据类型 变量名 = (目标数据类型)值或者变量;**
- 范例：**int k = (int)88.88;**

强制类型转换是可以的，但是数据会有所丢失，一般不建议使用

第一天目标检测：

目标检测 TARGETS DETECTION

- ◆ 能够说出Java语言的跨平台原理
- ◆ 能够独立下载并安装JDK
- ◆ 能够独立配置path环境变量
- ◆ 能够独立编写HelloWorld案例并编译执行
- ◆ 能够说出Java中的8种基本数据类型
- ◆ 能够定义基本数据类型的变量并使用变量名输出
- ◆ 能够说出标识符的命名规则
- ◆ 能够说出基本数据类型之间相互转换的规则