Java基础

java程序编写–编译–运行的过程

编写：我们将编写的java代码保存在".java"结尾的源文件中

编译：使用javac.exe命令编译我们的java源文件。格式：javac 源文件名.java

编译的过程：编译以后，会生成一个或多个字节码文件。字节码文件的文件名与java源文件中的类名相同。

运行：使用java.exe命令解释运行我们的字节码文件。格式：java 类名

java 程序注意

在一个java源文件中可以声明多个class。但是，只能最多有一个类声明为public

而且要求声明为public的类的类名必须与源文件名相同

程序的入口是main()方法。格式是固定的

每一行执行语句都以";"结束。

java数据类型

long型需要在数字后面加L或者l

如：

```
long a = 212312312312124L;
//或者
long b = 2132124123123121;
```

float 型 末尾加 “F”或者“f”

输出语句

System.out.println():先输出数据，然后换行
System.out.print():只输出数据

输出数组

- 调用Array.toString(a)，返回一个包含数组元素的字符串，这些元素被放置在括号内，并用逗号分开

```
int[] arr = {1,2,3,4,5};
  System.out.println(Arrays.toString(arr));
```

输出：[1, 2, 3, 4, 5]

- 传统的for循环方式

```
for(int i=0;i<array.length;i++)
{
      System.out.println(array[i]);
}
```

- for each循环

```
for(int a:array)
    System.out.println(a);
```

注释

单行注释

//xxxx

多行注释

/* ..... */

文档注释

/** .... */

说明注释允许你在程序中嵌入关于程序的信息。你可以使用 javadoc 工具软件来生成信息，并输出到HTML文件中。

说明注释，使你更加方便的记录你的程序信息。

在开始的 /** 之后，第一行或几行是关于类、变量和方法的主要描述。

之后，你可以包含一个或多个各种各样的 @ 标签。每一个 @ 标签必须在一个新行的开始或者在一行的开始紧跟星号(*).

多个相同类型的标签应该放成一组。例如，如果你有三个 **@see** 标签，可以将它们一个接一个的放在一起。

@author 作者

@version 版本

例子：

```
/*
 * <p>项目名称: ${project_name} </p>
 * <p>文件名称: ${file_name} </p>
 * <p>描述: [类型描述] </p>
 * <p>创建时间: ${date} </p>
 * <p>公司信息: ************公司 *********部</p>
 * @author <a href="mail to: *******@******.com" rel="nofollow">作者</a>
 * @version v1.0
 * @update [序号][日期YYYY-MM-DD] [更改人姓名][变更描述]
 */
```

```
/**
 * @Title: ${enclosing_method}
 * @Description: [功能描述]
 * @Param: ${tags}
 * @Return: ${return_type}
 * @author <a href="mail to: *******@******.com" rel="nofollow">作者</a>
 * @CreateDate: ${date} ${time}</p>
 * @update: [序号][日期YYYY-MM-DD] [更改人姓名][变更描述]
 */
```

```
/**
 * 获取  ${bare_field_name}
 */



/**
 * 设置   ${bare_field_name}
 * (${param})${field}
 */
```

Java 关键字

定义：被java语言赋予了特殊含义，用作专门用途的字符串（单词）

特点：关键字中所有字母均为小写

	数据	类型	关键字	
class	interface	enum	byte	short
int	long	float	double	char
boolean	void			
	流程控制	关键字		
if	else	switch	case	default
while	do	for	break	continue
return				
	访问权限	关键字		
private	protect	public		
	类、函数	变量修饰符	的关键字	
abstract	final	static	synchronized	
	类与类	之间关系	的关键字	
extends	implements			
	建立实例	判断实例	的关键字	
new	this	super	instanceof	
	异常处理	的关键字		
try	catch	finally	throw	throws
	包	的关键字		
package	import			
	其他修饰符	的关键字		
native	strictfp	tracnsient	volatile	assert数据
	数据类型值	的字面值		
true	false	null		

Java保留字

goto

const

自己命名时要避免使用这些保留字

标识符的使用

定义：凡是自己可以起名字的地方都叫标识符

设计到的结构：包名、类名、接口名、变量名、方法名、常量名

规则：

- 由26个英文字母大小写，0-9 ，_或 $ 组成

- 数字不可以开头。
- 不可以使用关键字和保留字，但能包含关键字和保留字。
- Java中严格区分大小写，长度无限制。
- 标识符不能包含空格。

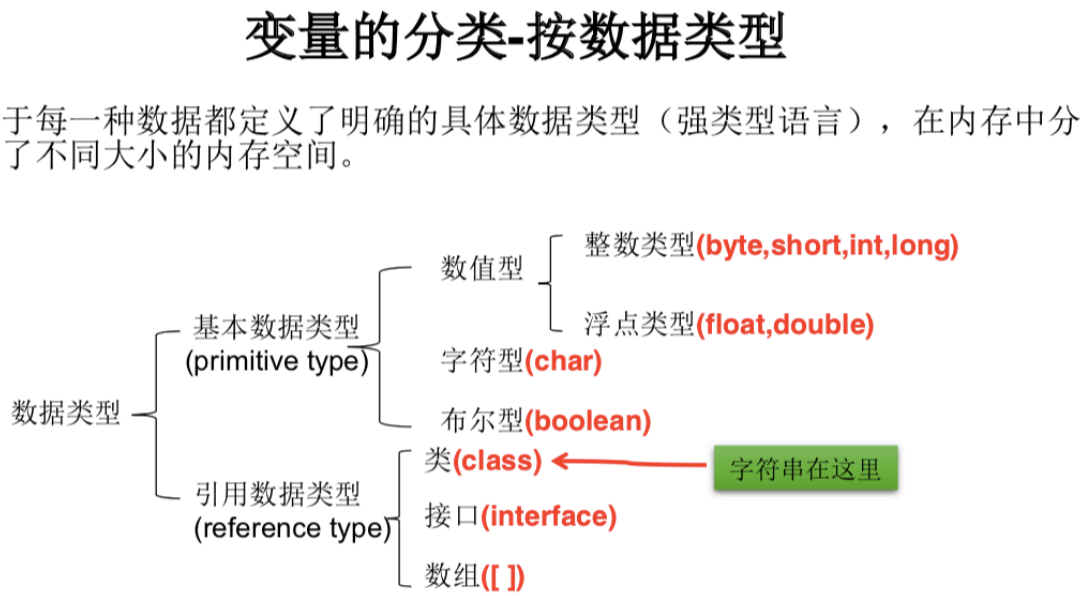
Java中的名称命名规范

- Java中的名臣命名规范
  - **包名**：多单词组成时所有字母都小写：xxxyyyzzz
  - **类名、接口名**：多单词组成时，所有单词的数字母大写：XxxYyyZzz
  - **变量名、方法名**：多单词组成时，第一个单词的首字母小写，第二个单词开始每个单词首字母大写：xxxYyyZzz
  - **常量名**：所有字母都大写。多单词时每个单词用下划线：XXX_YYY_ZZZ
- 注意1：在起名字时，为了提高阅读性，要尽量有意义，见名知意
- 注意2：java采用Unicode字符集，因此标识符也可以使用汉字，但是不建议使用

Java变量的分类

按数据类型分

基本数据类型



- 数值型
  - 整数类型(byte,short,int,long)
  - 浮点类型(float,double)
- 字符型(char)
- 布尔型(boolean)

引用数据类型

- 类(class) 比如：Sting
- 接口(interface)
- 数组([])

详细说明

1. 整型：byte(1字节=8bit)\short(2字节)\int(4字节)\long(8字节)

   ①byte范围：-128~127

   ②声明long型变量，必须以"l"或者"L"结尾

   ③通常，定义整型变量时，使用int型

   ④整型的常量，默认类型是：int型

类　型	占用存储空间	表数范围
byte	1字节=8bit位	-128 ~ 127
short	2字节	-2^{15} ~2^{15}-1
int	4字节	-2^{31} ~ 2^{31}-1 (约21亿)
long	8字节	-2^{63} ~ 2^{63}-1

2. 浮点型：float(4字节)/double(8字节)

   ①浮点型，表示带小数点的数值

   ②float表示数值的范围比long还大

   ③定义float类型变量时，变量要以"f"或者"F"结尾

   ④通常，定义浮点型变量时，使用double型

   ⑤浮点型的常量，默认类型为：double

类　型	占用存储空间	表数范围
单精度float	4字节	-3.403E38 ~ 3.403E38
双精度double	8字节	-1.798E308 ~ 1.798E308

3. 字符型：char（1字符=2字节）

   ①定义char型变量，通常使用一堆' ',内部只能写一个字符

   ②表示方式：

   1.声明一个字符

   2.转义字符

   3.直接使用Unicode值来表示字符型常量

转义字符	说明
\b	退格符
\n	换行符
\r	回车符
\t	制表符
\"	双引号
\'	单引号
\\	反斜线