

# 变量与常量



# 主要内容

- 标识符
- 关键字
- 变量
- 数据类型
- 类型转换
- 常量

# 标识符

- 我们所认识的标识符
- 如：类名 HelloWorld

# 标识符的命名规则

- 标识符可以由字母、数字、下划线 ( \_ ) 和美元符 ( \$ ) 组成，不能以数字开头
- 标识符严格区分大小写
- 标识符不能是Java关键字和保留字
- 标识符的命名最好能反映出其作用

# 标识符

- 请区分下列标识符哪些是合法的，哪些是非法的？
- Hello , hello\_imooc , \_hello , class , Void , 123abc , abc\$123 , abc 123

# 关键字

<b>abstract</b>	<b>boolean</b>	<b>break</b>	<b>byte</b>	<b>case</b>	<b>catch</b>
char	class	continue	default	do	double
else	extends	false	final	finally	float
for	if	implements	import	native	int
interface	long	instanceof	new	null	package
private	protected	public	return	short	static
super	switch	synchronized	this	throw	throws
transient	true	try	void	volatile	while

# 变量

- 什么是变量
- 变量名的命名规则

# 什么是变量

- **变量的三个元素：变量类型、变量名和变量值。**

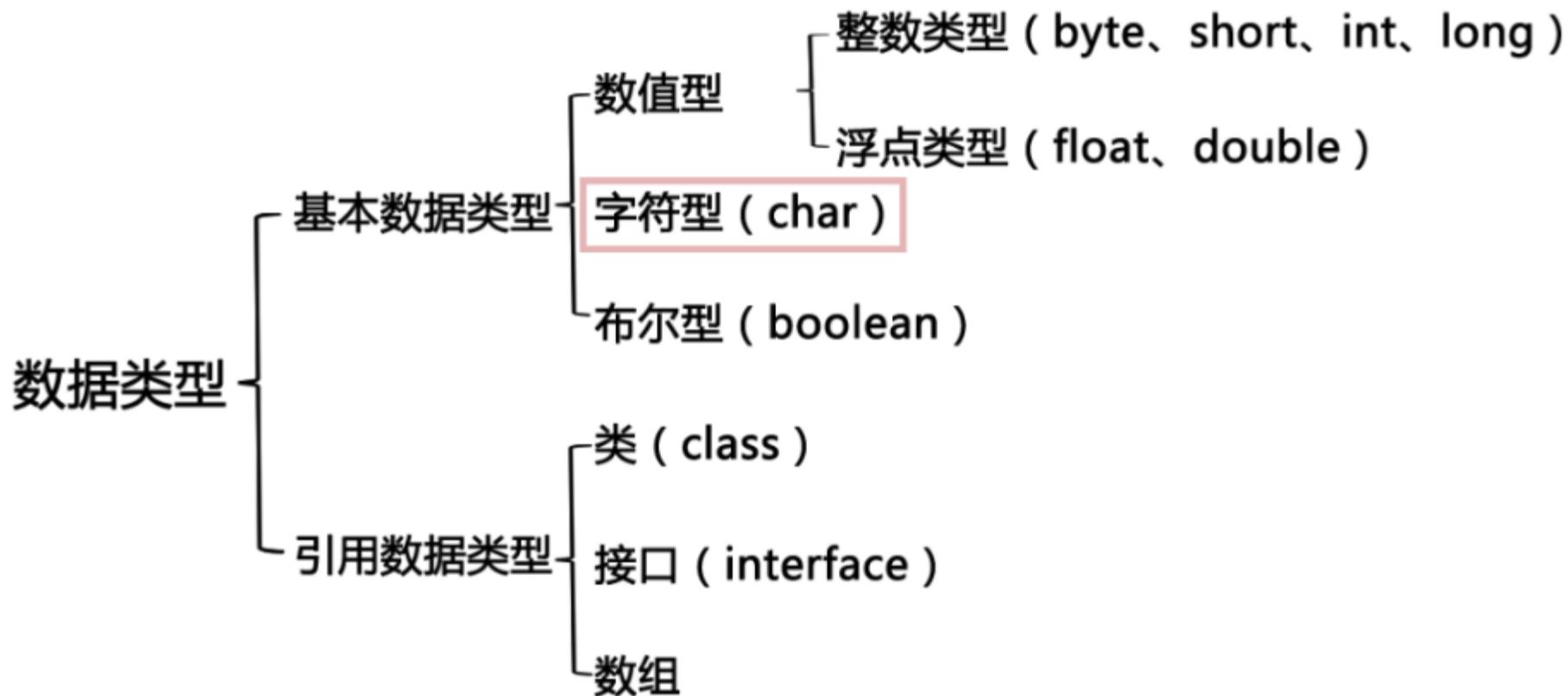
# 变量名的命名规则

- 满足标识符命名规则
- 符合驼峰法命名规范
- 尽量简单，做到见名知意
- 变量名的长度没有限制

# 类的命名规则

- 满足Pascal命名法规范

# 数据类型



# 基本数据类型

数据类型	说 明	字 节
byte	字节型	1
short	短整型	2
int	整型	4
long	长整型	8
float	单精度浮点型	4
double	双精度浮点型	8
char	字符型	2
boolean	布尔型	1

# 进制表示

**八进制：以0开头，包括0-7的数字**

**如：037，056**

**十六进制表示：以0x或0X开头，包括0-9的数字，及字母a-f，A-F**

**如：0x12，0xabcf，0XABCFF**

# 整型字面值

- Java中有三种表示整数的方法：十进制、八进制、十六进制
- 如：123，023，0x1357，0X3c，0x1abcL

# 变量声明

- **格式：数据类型 变量名；**
- **例：**
- **int n; 声明整型变量n**
- **long count; 声明长整型变量count**

# 赋值

- 使用 “=” 运算符进行赋值
- “=” 叫作赋值运算符，将运算符右边的值赋给左边的变量。
- 例：`int n;` 定义int型变量n
- `n=3;` 将3赋值给n
- 可以在定义变量的同时给变量赋值，即变量的初始化。
- 例：`int n=3;`
- 数据类型 变量名 = 变量值;

# 变量定义

• `int octal=037; //定义int类型变量存放八进制数据`

• `long longNumber=0xa2cdf3ffL;`

`//定义变量存放十六进制长整型数据`

• `short shortNumber=123;`

`//定义变量存放短整型数据`

• `byte b=10; //定义变量存放byte类型数据`

- **说明：**
- **整型字面值默认情况下是int类型，如果表示长整型则在末尾加l或L**

# 语句

- 以分号结束
- 不能换到一行写

- **单行注释，注释一行，在练习题中引入**

# 浮点型字面值

- 浮点型字面值默认情况下表示double类型，也可以在值后加d或D
- 如：123.43d或123.43D
- 如表示float类型，则需要字面值后加f或F
- 如：23.4f或23.4F

# 基本数据类型变量的存储

- 局部变量
- `int n=100;`如何在内存中存储

# 字符型字面值

- 字符型字面值用单引号内的单个字符表示。
- 如： 'a' , 'b' , '\$'
- 如何定义字符型变量？
- `char a = 'a' ;`
- `char ch = 65;`
- 疑问：怎么整型还能赋值给char类型？

# ASCII码

- **ASCII ( American Standard Code for Information Interchange , 美国标准信息交换代码 )**
- **基于拉丁字母的一套电脑编码系统**
- **主要用于显示现代英语和其他西欧语言**

# ASCII码表示

- 使用7位或8位二进制数组合来表示128或256种可能的字符。
- 标准ASCII码使用7位二进制来表示所有的大写字母和小写字母，数字0-9，标点符号，以及在美式英语种使用的控制字符。
- 后128个称为扩展ASCII码，用于表示特殊符号、外来语字母和图形符号。

# Unicode编码

- `char c= '\u005d' ;`
- Unicode表示法，在值前加前缀u

# 布尔类型字面值

- 布尔值只能定义为true和false
- 例：`boolean b=true;`

# 字符串面值

- 双引号引起来的0个或多个字符。

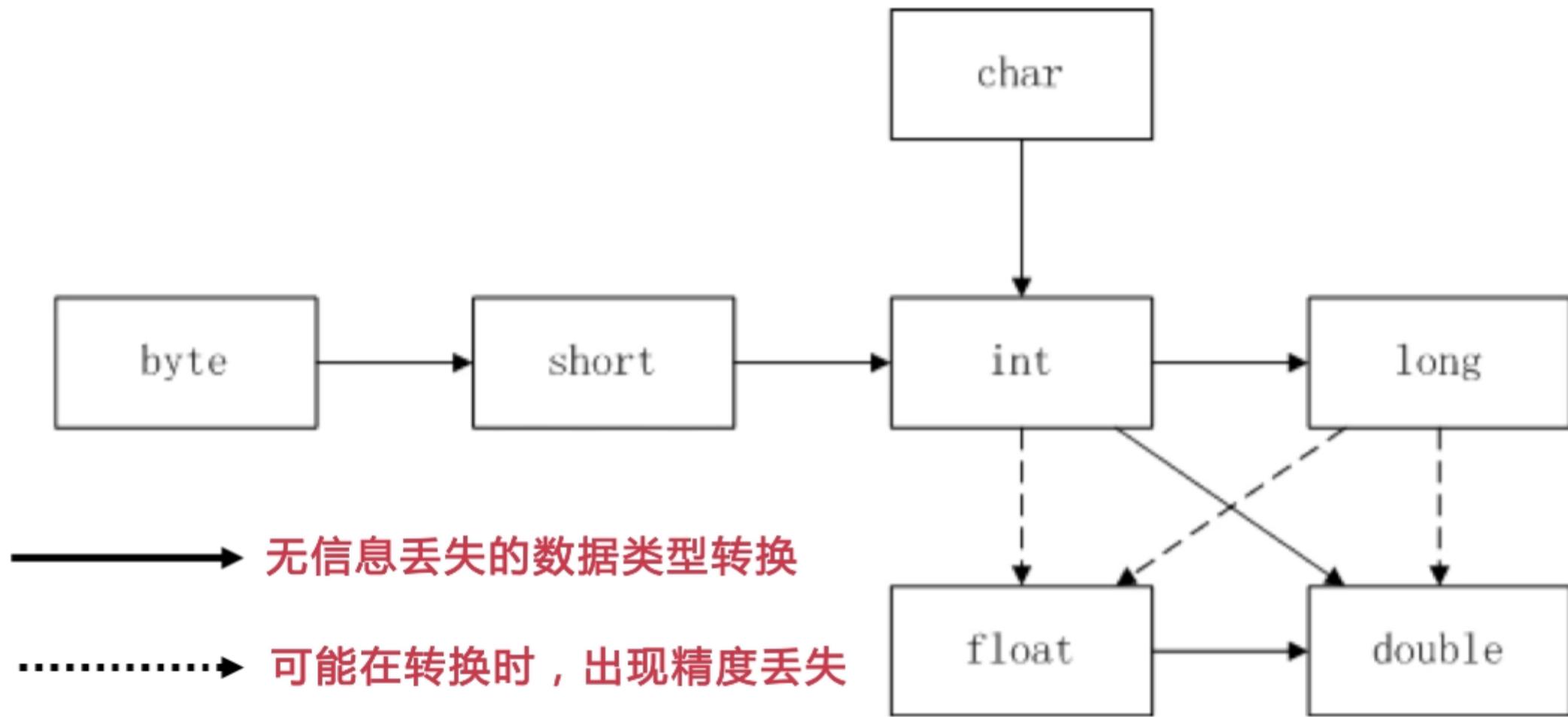
# 转义字符

转义字符	描述
<code>\uxxxx</code>	四位16进制数所表示的字符
<code>\'</code>	单引号字符
<code>\"</code>	双引号字符
<code>\\</code>	反斜杠字符
<code>\r</code>	回车
<code>\n</code>	换行
<code>\t</code>	横向跳格
<code>\b</code>	退格

# 类型转换

- **类型转换分为自动类型转换和强制类型转换**

# 自动类型转换顺序



# 强制类型转换

- 如果A类型的数据表示范围比B类型大，则将A类型的值赋值给B类型，需要强制类型转换
- 如：`double d = 123.4;`
- `float f = (float)d;`

# 常量

- **final int n = 5 ;**

# 总结

- 标识符的命名规则
- 关键字
- 数据类型
- 变量的定义和初始化
- 基本数据类型字面值
- ASCII码和Unicode编码
- 类型转换
- 常量