

第二章 Java基础语法

- 标识符
- 关键字
- Java基本数据类型
- 运算符
- 表达式和语句
- 分支
- 循环
- 方法
- 变量的作用域
- 递归调用

Java使用的字符集

- **Unicode**字符集是一个国际化的字符集，全世界多种语言所使用字母、符号、表意文字等都被收入其中，当然也包括汉字。
- 在Unicode字符集中，每个字符编码由**16**位二进制数组成，共有**65536**个编码。
- 优点：在程序中使用全球多种语言符号，不会因使用了不同的系统而产生表示符号的混乱。

标识符

- Java 语言中，对各种**变量**、**方法**和**类**等要素命名时使用的字符序列称为标识符。（如HelloWorld.java中）
 - 凡是自己可以起名字的地方都叫标识符，都遵守标识符的规则。
- Java 标识符有如下命名规则：
 - 标识符由字母、下划线“_”、美元符“\$”或数字组成。
 - 标识符应以字母、下划线、美元符开头。
 - Java 标识符大小写敏感，长度无限制。
- 约定俗成：Java 标识符选取因注意“见名知意”且不能与 Java 语言的关键字重名。

合法的标识符	不合法的标识符
HelloWorld	class
DataClass	DataClass#
_983	98.3
\$bS5_c7	Hell World

关键字

- Java 中一些赋以特定的含义，用做专门用途的字符串称为关键字（keyword）。
- 大多数编辑器会将关键字用特殊方式标出。
- 所有Java关键字都是小写英文字符串。
- goto 和 const 虽然从未使用，但也作为 Java 关键字保留。

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

Java常量

- Java 的常量值用字符串表示，区分不同的数据类型。
 - 如整型常量 123
 - 实型常量 3.14
 - 字符常量 'a'
 - 逻辑常量 true、false
 - 字符串常量 "helloworld"
- 注意：
 - 区分字符常量和字符串常量
 - “常量”这个名词还会用在另外其他语境中表示值不可变的变量。

Java变量

- Java变量是程序中最基本的存储单元，其要素包括**变量名**，**变量类型**和**作用域**。
- 不同的变量类型分配不同的长度，Java程序中每一个变量都属于特定的数据类型，在使用前必须对其声明，声明格式为：

`type varName [=value][{,varName[=value]}]`

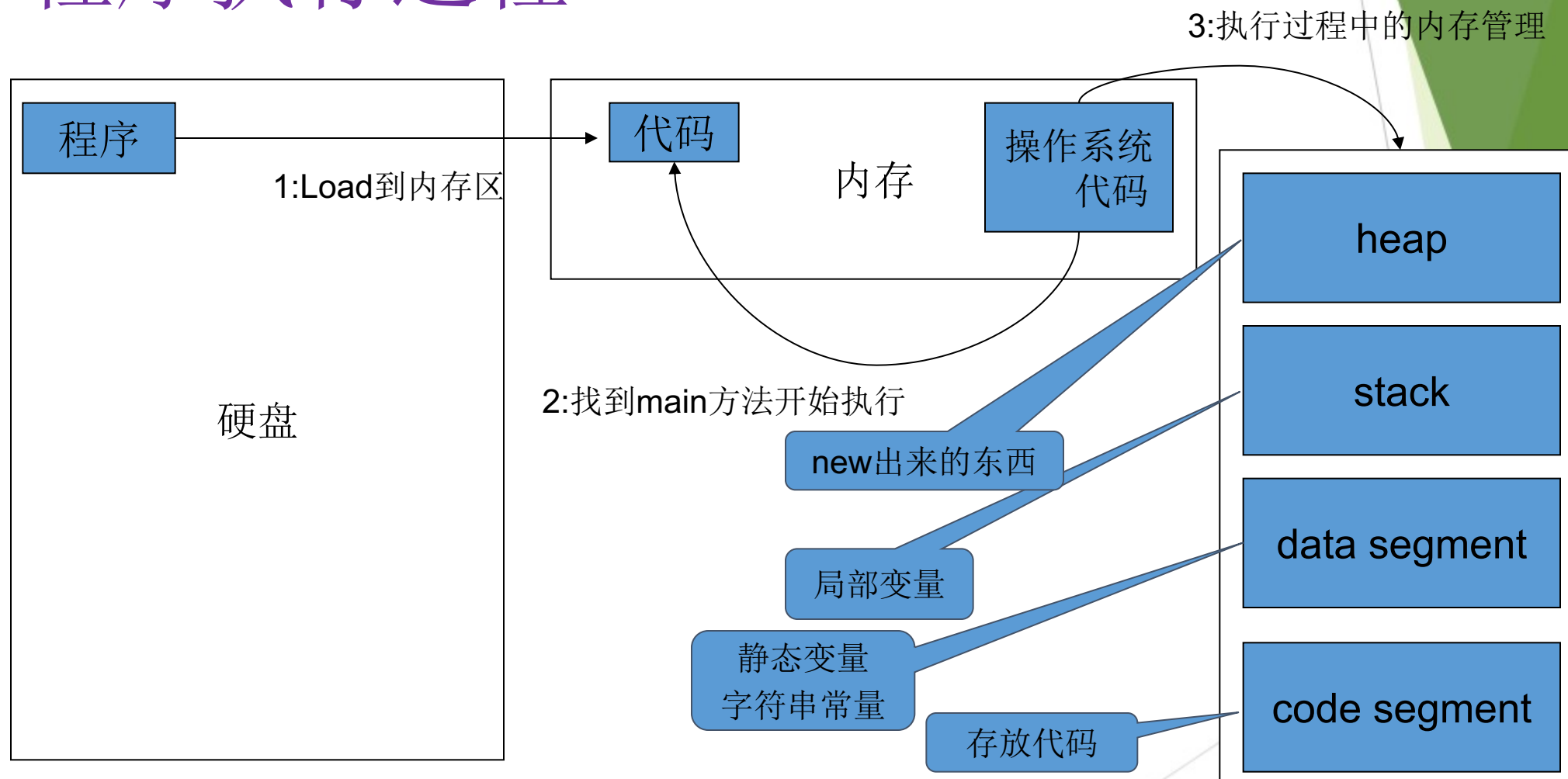
- 例如：

```
int i = 100;  
float f = 12.3f;  
double d1, d2, d3 = 0.123;  
String s = "hello";
```



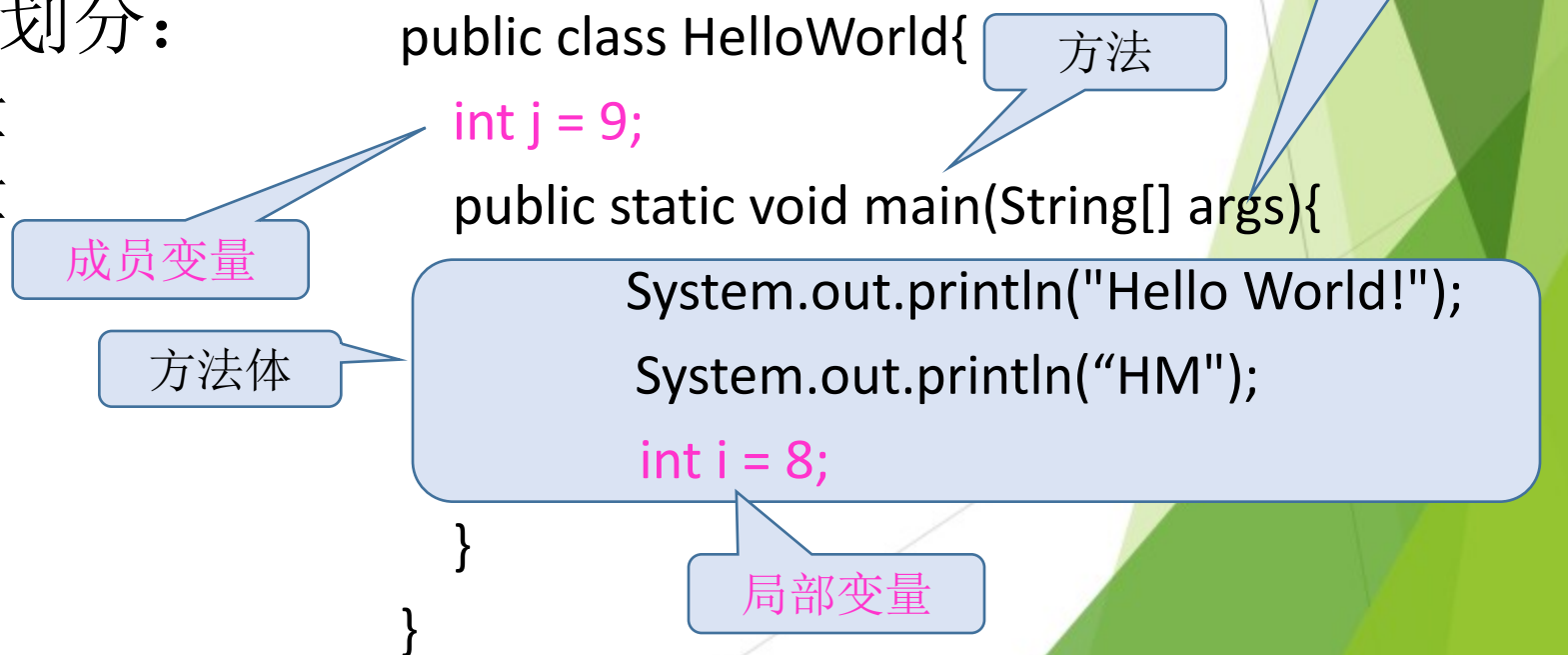
- **从本质上讲，变量其实是内存中的一小块区域**，使用变量名来访问这块区域，因此，每一个变量使用前必须要**先申请**（声明），然后必须**再进行赋值**（填充内容），才能使用。

程序执行过程



Java变量的分类

- 按被声明的位置划分：
 - 局部变量：方法或语句块内部定义的变量
(方法体内部是指与方法对应的大括号内部)
 - 成员变量：方法外部、类的内部定义的变量
 - 注意：类外面（与类对应的大括号外面）不能有变量的声明
- 按所属的数据类型划分：
 - 基本数据类型变量
 - 引用数据类型变量



Java局部变量与成员变量

- 方法体内部声明的变量（包括形参）称为局部变量
 - 方法体内部是指与方法对应的大括号内部
- 在方法体外，类体内声明的变量称为成员变量

变量的作用域

```
public void method() {  
    int i;  
    int j = i+5 ; // 编译出错  
    double d = 3.14;  
}
```

注：以上程序目前还不能编译

```
public class HelloWorld{  
    static int j = 9;  
    public static void main(String[] args){  
        System.out.println("Hello World!");  
        System.out.println("HM");  
        System.out.println(j);  
        int i = 8;  
    }  
    System.out.println(i);  
}
```

j可以打印出来

i打印不出来