

# Chapter 2 基本程序设计

## 2.1 编写简单的程序

编写程序，给定半径，计算圆的面积

```
package www.learnjava.garfield.ch2; // java的类都在包中
// 默认导入 java.lang.System类
public class Area {
    public static void main(String[] args) {
        double radius = 10;
        double area;
        double pi = 3.1415;
        area = radius * radius * pi;
        System.out.println(area);
    }
}
```

## 2.2 简单IO

- 标准输入/输出流
  - System.out: 标准输出流类OutputStream的对象
  - System.in: 标准输入流类InputStream的对象
- Scanner类 (java.util.Scanner)

```
Scanner scanner = new Scanner(System.in);
// 从控制台读入一个double类型数
double d = scanner.nextDouble();
// 从控制台读取一个字符串
String s = scanner.next();
// Scanner还有很多方法，如nextInt, nextByte等
```

## 2.3 标识符，变量，常量

### 2.3.1 标识符

标识符被用来命名常量，变量，方法，类等实体，命名规则有以下几点：

- 由字母、数字、下划线、美元符构成
- 开头不能是数字
- 不能是保留字

### 2.3.2 变量

变量用于保存数据输入，输出，中间值等，变量声明语法：`datatype variableName`，例如 `int x`。

### 2.3.3 常量

常量一旦初始化后就不能再改变，常量声明语法：`final datatype CONSTANT_NAME = value`，注意常量的声明和初始化必须同时完成。

## 2.4 赋值语句和赋值表达式

- 赋值语句

形如 `variable = expression` 的语句，其中 `expression` 是包含字面量、变量、方法调用和操作符的表达式。赋值语句的结果是将表达式的值赋值给左边的变量。例如 `x=1`

- 赋值表达式

赋值表达式形式和赋值语句相同，赋值表达式的结果等于表达式的值，赋值表达式是右结合的。例如 `i=j=k=1`。

## 2.5 JAVA数据类型

### 2.5.1 基本数据类型

- 整数类型：byte, short, int, long
- 字符类型：char
- 浮点类型：float, double
- 逻辑类型：boolean

### 2.5.2 引用类型

- 类
- 接口
- 数组

### 2.5.3 数值字面值 (literal)

定义：字面值是直接出现在程序中的常量值。例如 `long l = 10000L` 中的 `10000L` 就是字面值。

整数字面值

- 以0开头表示八进制
- 以0x或者0X开头表示十六进制
- 以1-9开头表示十进制
- 以l或L结尾表示long类型
- 无后缀表示int类型

浮点数字面值

- 以d或D结尾或者无后缀表示double类型
- 以f或者F结尾表示float类型

## 2.5.4 数值类型转换

如果二元操作符的两个操作数的数据类型不同，那么根据下面的规则对操作数进行转换：

- 数据转换总是向较大范围的数据类型转换，避免精度损失。

```
byte i = 10;
long k = i*3 + 4; // i转成int参与表达式计算，计算结果转long
double d = i*3.1 + k/2; // i转成double，k/2转double
```

- 将值赋值给较大取值范围的变量时，自动进行类型转换。  
byte < char < short < int < long < float < double
- 将值赋值给较小取值范围的变量时，必须进行强制类型转换。

```
float f = (float)10.1; // 10.1默认double类型，double>float
int i = (int)f;
```

### 注意

- 整数操作时，除数不能位0，否则会产生ArithmeticException异常
- 浮点数操作上溢至Infinity，下溢至0  
浮点数除0等于Infinity  
0.0除0.0等于NaN

## 2.5.5 字符数据类型

- char表示16位的单个Unicode字符（Java中字符都是Unicode编码）
- char类型的字面值
  - 以两个单引号界定的单个Unicode字符。如'A'
  - 可以用\uxxxx形式表示，xxxx为16进制。如\u7537
  - 可以用转移字符表示：\n \t \b等
- String表示一个字符序列，是基于String类实现的，不是java内置的数据类型，是引用类型

## 2.6 编程风格和常见错误类型

### 2.6.1 编程风格

- 注释：
  - 类和方法前使用javadoc文档注释
  - 方法步骤前使用行注释
- 命名：
  - 类名首字母大写，驼峰命名
  - 变量和方法名小写，多个单词，第一个单词首字母小写，其他单词首字母大写
  - 常量使用大写，单词间以下划线分割
- 缩进为4个空格

### 2.6.2 java常见错误类型

- 语法错误 (syntax error) : 编译期间产生的错误, 编译报错
- 运行时错误 (runtime error) : 导致程序非正常终止的错误, 编译不会报错
- 逻辑错误 (logic error) : 程序不能按预期的方式执行, 编译不会报错