程序;

```
public static void main(String[] args) {
    // 软件：处理数据的程序

    // 程序 = 数据 + 逻辑代码

    // 数据进入CPU的过程：
    // 硬盘 -> 内存 -> CPU缓存 -> CPU
```

数据类型:

```
// 数据类型的作用
// 1.提供多种数据类型，最大化使用存储空间
// 2.决定所占的空间大小
// 3.决定存放什么样的数据

// Java数据类型的分类：
// 1.基本数据类型
// 2.引用数据类型


// Java数据类型的分类：
// 1.基本数据类型
// 2.引用数据类型

/*
8种基本数据类型
1.整型
    byte    1字节    字节      -2^7 ~ 2^7-1
    short   2字节    短整型    -2^15 ~ 2^15-1
    int     4字节    整型      -2^31 ~ 2^31-1
    long    8字节    长整型    -2^63 ~ 2^63-1
2.浮点型
    float   4字节    单精度浮点型
    double  8字节    双精度浮点型
3.字符型
    char    2字节    字符型
4.布尔型
    boolean 1字节    布尔型
*/
```

常量:

```
// 常量：程序运行期间，不能改变的量

/*
1.整型常量
    十进制：123, -123
    二进制：0b00001010, 0b10001010
        注：有符号区分的二进制，第一位为0是正数，为1是负数
    八进制：0123, 0521
    十六进制：0x3a, 0x3A
2.浮点型常量
    单精度：3.14f
    双精度：3.14, 3.14d
3.字符型常量:'a', '1', '?', '🤣', '你', ' '
    注：用单引号包括起来，只能占一个位置
    注：Java中的字符型支持Unicode编码
4.布尔常量：true, false
5.字符串常量："王麻子"
```

文字编码:

```
// 文字编码：定义了字符存到计算机中的转换规则

// 常见的文字编码
// 1.ASCII：支持存储128个转换规则
// 2.Unicode：万国码
// 3.GBK：中国制定，支持汉字转换规则+ASCII转换规则
// 4.UTF-8：Unicode的一种算法，优化了存储结构
```

变量:

```
// 变量：程序运行期间，可以改变的量

// 变量需要先定义，后使用

// 变量定义的三要素(格式)
// 数据类型 变量名 = 初始值
// 注：变量使用小驼峰命名法

// 数据类型的选择
// 1.根据要存储的数据决定
// 2.整型默认使用int，浮点型默认使用double

int count = 10;
int count1 = 10;
// int 1count = 10;
int renshu = 10;
int rs = 10;
int number = 10;
int 人数 = 10;
// int int = 10;
```

标识符:

```
// 标识符(文件名，工程名，类名，变量名等等)的命名规则：
// 1.由数字，字母，下划线，美元符号组成，不能以数字开头
// 2.不能使用系统的关键字(保留字)
// 3.见名知意，建议使用英文命名，不推荐使用拼音和汉字
// 4.使用命名法则
//    a.匈牙利命名法：把数据类型的首字母作为变量名的首字母，比如：iNumber，fNumber
//    b.驼峰命名法：又叫小驼峰命名法，除第一个单词外，每一个单词的首字母大写，比如：appleCount
//    c.帕斯卡命名法：又叫大驼峰命名法，每一个单词的首字母大写，比如：AppleCount
//    d.蛇形命名法：每个单词使用小写，多个单词用下划线连接在一起，比如：apple_count
//    e.尖叫蛇形命名法：每个单词使用大写，多个单词用下划线连接在一起，比如：APPLE_COUNT
// 5.在同一作用域内，不能重名

// Java变量的作用域：块作用域
```

转义字符:

```
// 转义字符：避免破坏字符串的结构，实现特殊效果
// \"   一个双引号
// \'   一个单引号
// \n   换行
// \t   空一个tab键的距离
// \u0000   0代表1位16进制位，整体代表Unicode编码值
```

输出(三种):

```
// 初始化: 第一个赋值的过程
// int c;
// System.out.println(c);

// 输出语句
// 1.print, 不换行输出
System.out.print("辉哥最帅!");
// 2.println, 换行输出, 快捷键: sout
System.out.println("辉哥最帅!");
// 3.printf, 格式串输出, 不换行输出, 快捷键: souf
System.out.printf("辉哥最帅!");

int price = 998;
char color = '金';
String color1 = "土豪金";

System.out.println("我的" + color + "色手机价格是" + price + "元")
System.out.print("我的" + color + "色手机价格是" + price + "元\n")
System.out.printf("我的%c色手机价格是%d元\n", color, price);
```

格式串

```
// 格式串
// %c: 字符
// %d: 十进制整数
// %o: 八进制整数
// %x: 十六进制整数, 字母小写
// %X: 十六进制整数, 字母大写
// %f: 浮点型
// %s: 字符串
```

运算符:

```
// 运算符
// 1.赋值运算符, =, 从右向左执行, 赋值的过程是拷贝的过程
int a1 = 10;
int a2 = 20;

// 简化
int b1 = 10, b2 = 20;
```

自增,

```java
// ++
int c = 10;
c++;// 等价于 c = c + 1
System.out.println("c = " + c);// 11

++c;// 等价于 c = c + 1
System.out.println("c = " + c);// 12

// ++在后，先执行其他运算，再加1
result = c++;
System.out.println("c = " + c);// 13
System.out.println("result = " + result);// 12

// ++在前，先加1，再执行其他运算
result = ++c;
System.out.println("c = " + c);// 14
System.out.println("result = " + result);// 14
```

表达式:

```java
// 表达式：由变量，常量，运算符组成
// 表达式有最终的结果，比如 1 + 2

// 根据表达式的结果类型，分为
// 1.整型表达式，比如 1 + 2，100，result，result + 20
// 2.浮点型表达式
// 3.字符型表达式
// 4.条件表达式：表达式的结果为布尔类型

// 语句：程序执行的最小单位，语句以分号结束
```

语句

```java
// 语句：程序执行的最小单位，语句以分号结束
int m = 10;
;// 空语句

// 输入语句：接收你向控制台输入的值

/*
// 输入语句的使用
// 1.创建输入工具，输入工具可以反复使用
Scanner scanner = new Scanner(System.in);
// 2.提示用户输入
System.out.println("请输入一个整数:");
// 3.获取用户在控制台输入的内容
int n = scanner.nextInt();
// 4.操作
```