# 基本编程结构

## 进制转换

int n = 10;

String strN = Integer.**toString**(n, 2); //转成2进制

out.**println**(strN);

strN = "1010";

int m = Integer.**parseInt**(strN, 2); //按照2进制转

out.**println**(m);

## 格式化输出

out.**printf**("%,+.2f\n", 10000.0/3.0);

String message = String.**format**("The averag number is: %,+.2f, total number: %d ", 10000.0/3, 10000);

out.**println**(message);

%d、%x、%o分别是10、16和8进制，没有2进制。

## 数组的复制

int[] array = new int[]{1,2,3};

int[] arrayCopy = array;

arrayCopy[2] = 30;

for(int i : array)

out.**print**(i);

int[] arrayDeepCopy = Arrays.**copyOf**(array, array.length);

arrayDeepCopy[2] = 300;

for(int i : array)

out.**print**(i);

## 实例变量和局部变量

实例变量是在class中定义的变量，属于实例对象。**实例变量**可以不初始化，系统会自动给它复赋值，比如数字为0，boolean为假，对象引用为null。而局部变量必须要初始化之后才可以使用。

# 接口

1. 接口是任何实现该接口的父类；可以将类的实例赋值给接口类型的变量；
2. 接口可以包含静态方法，所有的方法默认是public；
3. 接口所有变量默认是public **static** final的（为什么是static的？因为接口指定的是行为，不是状态，在接口中没有实例变量）；
4. 接口可以包含默认方法，实现类可以继承或者覆盖该默认方法（默认方法是指声明为default的方法，如果没有声明为default，那么子类必须要实现，而default方法如果在子类中没有实现，将采用接口中定义的）；
5. 实现类必须将接口**方法**声明为public（如果没有声明为public，默认是default，会报错）；
6. 如果一个类实现接口一部分方法，那么这个类是abstract class；

类型T的变量不需要转型就能够赋值给类型S的变量，这S是T的父类。所以强制转型是从父类转为子类。不会存在类型为接口的对象，所有的对象都是类的实例，也就是说不能new一个Interface。

一个类实现了两个接口，

1. 两个接口定义了同样的default方法。由于default方法在接口中是有具体的实现的，编译器没有办法决定谁该覆盖谁。所以只能在子类重写解决歧义；
2. 两个接口定义了同样的public方法。这样是没有冲突的，因为要么在子类实现public的方法，要么不实现，该类作为abstract class。

如果一个类继承了一个类并实现了一个接口中有同样的方法，这种情况下只关心父类的方法，忽视来着接口的默认方法。