操作符详解：

&与

将数字转化为二进制，同真为真，否则为假

|或

同假为假，否则为真

~非

取反运算，1为0，0变1

^异或

同为假，异为真

HashMap的K和V都可以为空

底层实现

JDK7是数组加链表：头插法

JDK8数组加链表加红黑树，尾插法

HashMap中数组长度的计算：

HashMap中数组的长度必须要是2的幂次方的正整数

如果你设定数组长度的参数不是2的幂次方，HashMap会调用Integer的hightestOneBit转换为一个比设定值小但最接近设定值的整数

代码实现：

int i;

i |=(i>>1);

i |=(i>>2);

i |=(i>>4);

i |=(i>>8);

i |=(i>>16);

return i –(i>>>1);

具体计算：

0001 \*\*\*\* >>1

0000 1\*\*\* |=

0001 1\*\*\* >>2

0000 011\* |=

0001 111\* >>4

0000 0001 |=

0001 1111 >>8

0000 0000 |=

0001 1111

......

0001 1111 >>>1

0000 1111 i-(i>>>1)

0001 0000

因为2的幂次方的整数的二进制均为100000..的形式，所以做后结果就一定是比原数小且最接近原数的2的幂次方。

该算法的精髓在于通过位移找到原整数中，最靠左边的1的位置。

扩容：

当HashMap中存储的值大于总容量的0.75，就会进行扩容