概念：

栈是一种特殊的线性表。线性表的插入删除可以在表的任意位置进行，但是栈只允许在表尾进行。

栈允许插入删除的一边叫做栈顶，另一端叫做栈底。但是内部上则是在表尾进行插入和删除。每次入栈的元素都在栈顶，每次取出的元素也在栈顶。

栈是一种先进后出（后进先出）的线性表。

栈的应用,例如浏览器对用户当前访问过的地址的管理、键盘缓冲区中对键盘输入信息的管理。算法上例如，括号匹配检验、十进制转化成任意进制等。

功能：

clear(); 栈清空

isEmpty(); 栈是否为空

length(); 栈长度

peek(); 取栈顶元素

push(); 入栈

pop(); 取出栈顶元素，注意与peek()的区别，peek()只是取栈顶元素

实现：

内部实现分为两种：数组+链表

这里讨论数组实现栈：

1.栈内部的数据:

保存数据用的数组+栈顶指针。

private Object[] stackElem;

private int top=-1; //因为top指向的是当前栈顶元素，必须从-1开始。

//如果从0开始，进入一个元素后再+1,top就是栈顶元素的下一个空位。

2.peek()返回当前的栈顶元素

public Object peek() { //返回栈顶元素

if(!isEmpty())

return stackElem[top]; //top指针指向的是当前栈顶的元素

else

return null;

}

3.push()入栈

注意：因为top指针指向的是当前栈顶元素，要入栈必须先要top+1指向新的栈顶元素单元，然后入栈。

（1）top=top+1（2）入栈

public void push(Object x) { //进栈

if(top!=stackElem.length) //1.让top指针指向栈顶的下一个空位

stackElem[++top]=x; //2.进栈

else

System.out.println("栈已满");

}//这里的stackElem[++top]=x,先执行top=top+1，然后执行stackElem[top]=x;

4.pop()取出栈顶元素（出栈）

（1）出栈（2）top=top-1 注意：要返回出去的栈顶元素

public Object pop() { //取出栈顶元素

if(!isEmpty())

return stackElem[top--];

else

return null;

}// stackElem[top--]先执行stackElem[top]，然后执行top=top-1

在java中栈的建立

Stack<String> s = new Stack<>();

s.empty();

s.push("a");

s.search("a");

System.out.println(s.peek());

System.out.println(s.pop());