Practice2: implement with singly linked list

class ListStack<T> {

private class Node<T>{

private T data;

private Node<T> next;

public Node(T t,Node<T> next){

this.data=t;

this.next=next;

}

}

private Node<T> top;

public ListStack(){

top=null;

}

// 向栈中添加新的结点元素: 新节点的next指向原本的top结点;将新的top结点指向新插入的节点

public void push(T t){

Node<T> node=new Node<T>(t,top);

top=node; }

// 栈顶出栈并将新栈顶结点指向原栈顶结点

public T pop() throws Exception{

if(top==null){

throw new Exception("栈为空，元素不可出栈");

}

Node<T> node=top;

top=top.next;

return node.data;

}

//遍历打印栈

public void printfStack(){

Node<T> node=top;

while(node!=null){

System.out.println(node.data);

node=node.next;

}

}

public static void main(String[] args) throws Exception {

ListStack<String> list=new ListStack<String>();

list.push("1");

list.push("2");

list.push("3");

list.pop();

list.printfStack();

}

}