|  |
| --- |
| 0hnQGAuNPo4wojnr77c2x8cWiaT70FGHBZ2h-3__FYQ.png |
| Stack |
| Data Structures Made Easy |
|  |

Dublin city university

# 1. *Bounded Stack*

class bounded\_Stack<T>{

private T[] sequence = (T[])(new Object[10000]);

private int size = 0;

public boolean isEmpty(){

return size == 0;

}

public boolean push(T t){

if(size >= sequence.length){

T[] sequence\_2 = (T[])(new Object[sequence.length \* 2]);

System.arraycopy(sequence, 0, sequence\_2, 0, sequence.length);

sequence = sequence\_2;

}

sequence[size] = t;

size++;

return true;

}

public T pop(){

if(isEmpty())

return null;

else{

T temporary = sequence[size];

size--;

return sequence[size];

}

}

public static void main(String [] args){

bounded\_Stack<Integer> stack = new bounded\_Stack<Integer>();

System.out.println('\n' + "ELEMENTS");

System.out.println("===============");

for(int index = 10; index <= 100; index += 10){

System.out.print(index + " ");

stack.push(index);

}

System.out.println();

System.out.println('\n' + "REVERSED ELEMENTS");

System.out.println("=================");

while(!stack.isEmpty())

System.out.print(stack.pop() + " ");

}

}

# 2. *Unbounded Stack*

class unbounded\_Stack<T>{

private static class Node<T>{

private T item;

private Node<T> next = null;

Node(T item0, Node<T> next0){

item = item0;

next = next0;

}

}

private Node<T> head = null;

public boolean isEmpty(){

return head == null;

}

public boolean push(T t){

head = new Node<T>(t, head);

return true;

}

public T pop(){

if(isEmpty())

return null;

T t = head.item;

head = head.next;

return t;

}

public static void main(String [] args){

unbounded\_Stack<Integer> stack = new unbounded\_Stack<Integer>();

System.out.println('\n' + "ELEMENTS");

System.out.println("===============");

for(int index = 10; index <= 100; index += 10){

System.out.print(index + " ");

stack.push(index);

}

System.out.println();

System.out.println('\n' + "REVERSED ELEMENTS");

System.out.println("=================");

while(!stack.isEmpty()){

System.out.print(stack.pop() + " ");

}

}

}