class DynamicArrayStackImpl {

private int capacity;

private final static int DEFAULT\_CAPACITY = 16;

private final int INITIAL\_CAPACITY;

private int top = -1;

private int[] stackArray;

public DynamicArrayStackImpl() {

this (DynamicArrayStackImpl.DEFAULT\_CAPACITY);

}

public DynamicArrayStackImpl(int capacity) {

this .capacity = capacity;

this .INITIAL\_CAPACITY = capacity;

stackArray = new int[ this .capacity];

}

public int size() {

return top + 1;

}

public boolean isEmpty(){

return top == -1;

}

public void expand() {

capacity = capacity \* 2 ;

int[] newStack = new int[capacity];

System.arraycopy(stackArray, 0, newStack, 0, size());

stackArray = newStack;

}

public void shrink() {

if (INITIAL\_CAPACITY <= (capacity >> 2 )) {

int minSize = capacity >> 2;

if (top < minSize) {

capacity = capacity / 2;

int[] newStack = new int[capacity];

System.arraycopy(stackArray, 0 , newStack, 0 , size());

stackArray = newStack;

}

}

}

public void push(int data) throws Exception {

if (size() == capacity) {

expand();

}

stackArray[++top] = data;

}

public int top() throws Exception {

if (isEmpty()) {

throw new Exception("Stack is empty!" );

}

return stackArray[top];

}

public int pop() throws Exception {

if (isEmpty()) {

throw new Exception("Stack is empty!" );

}

int data = stackArray[top];

stackArray[top--] = 0;

shrink();

return data;

}

public String toString() {

String arrayString = "[";

for (int index = 0 ; index <= top; index++) {

if (index == 0 ) {

arrayString += stackArray[index];

}

else {

arrayString += "," + stackArray[index];

}

}

arrayString += "]";

return arrayString;

}

}

public class DynamicArrayBasedImplementation {

public static void main( String [] args) {

DynamicArrayStackImpl stack = new DynamicArrayStackImpl();

try {

System.out.println( "isEmpty: " +stack.isEmpty());

stack.push( 5 );

stack.push( 10 );

stack.push( 15 );

stack.push( 20 );

stack.push( 25 );

System.out.println( "Stack: " +stack);

System.out.println( "Top: " +stack.top());

System.out.println( "Stack: " +stack);

System.out.println( "Pop data: " +stack.pop());

System.out.println( "Stack: " +stack);

System.out.println( "Size is " +stack.size());

System.out.println( "isEmpty: " +stack.isEmpty());

} catch (Exception e) {

}

}

}