Assessment 3 part 2 Code

//Robert Lothian

//HND Software Development 2

//26.04.22

//This excercise uses an interface to develop a stack using an int Array List

using System;

namespace Assessment\_3.\_2

{

internal class Program

{

interface StackADT

{

void push(int value);

int pop();

bool isEmpty();

bool ifFull();

int size();

void display();

}

class intArrayList

{

private int count;

private int[] values;

/// <summary> Creates array list </summary>

/// <param name="capacity"></param>

public intArrayList(int capacity)

{

values = new int[capacity];

count = 0;

}

/// <summary>Instanciates Array List</summary>

public intArrayList()

{

values = new int[20];

count = 0;

}

/// <summary>Is the Array List empty </summary>

/// <returns>0 if true</returns>

public bool isEmpty()

{

return (count == 0);

}

/// <summary> Is the array list full?</summary>

/// <returns>The length of value is equal to count</returns>

public bool isFull()

{

return values.Length == count;

}

/// <summary>Adds to the array list in the first position</summary>

/// <param name="value"></param>

/// <exception cref="Exception"></exception>

public void addFirst(int value)

{

if (isFull())

{

throw new Exception("list full");

}

if (isEmpty())

{

addLast(value);

}

else

{

for (int pos = count; pos > 0; pos--)

{

values[pos] = values[pos - 1];

}

values[0] = value;

count++;

}

}

/// <summary>adds to the array list at the last position</summary>

/// <param name="value"></param>

/// <exception cref="Exception"></exception>

public void addLast(int value)

{

if (isFull())

{

throw new Exception("list full");

}

values[count++] = value;

}

/// <summary>Removes last value in Array List </summary>

/// <returns>Returns value decrements count</returns>

/// <exception cref="Exception"></exception>

public int removeLast()

{

if (isEmpty())

{

throw new Exception("list full");

}

return values[--count];

}

/// <summary>Displays how large the Array List is</summary>

/// <returns>the count on the array list</returns>

public int size()

{

return count;

}

/// <summary>

/// displays array list details

/// </summary>

public void display()

{

if (count == 0)

{

Console.WriteLine("list is empty");

}

else

{

Console.WriteLine("list has " + count + " items");

for (int i = count-1; i >=0; i--)

{

Console.WriteLine("value:" + values[i]);

}

}

}

}

class MyStack : StackADT

{

public intArrayList list;

/// <summary>

/// instanciates an object list as intArrayList using interface StackADT

/// </summary>

public MyStack()

{

list = new intArrayList(20);

}

/// <summary>

/// calls display

/// </summary>

public void display()

{

list.display();

}

/// <summary>

/// calls isFull from intArrayList

/// </summary>

/// <returns>Value true if Stack is full</returns>

public bool ifFull()

{

return list.isFull();

}

/// <summary>

/// calls isEmpty from intArrayList

/// </summary>

/// <returns>value true if the stack has no contents</returns>

public bool isEmpty()

{

return list.isEmpty();

}

/// <summary>

/// Calls removeLast from intArrayList

/// </summary>

/// <returns>Removes the last value</returns>

public int pop()

{

return list.removeLast();

}

/// <summary>

/// Calls add last from intArrayList

/// </summary>

/// <param name="value"></param>

public void push(int value)

{

list.addLast(value);

}

/// <summary>

/// calls size from intArrayList

/// </summary>

/// <returns></returns>

public int size()

{

return list.size();

}

}

/// <summary>

/// Driver to enter test data

/// </summary>

static void stackdriver()

{

MyStack astack = new MyStack();

Console.WriteLine( "testing Stack ");

Console.WriteLine( "testing is empty " +astack.isEmpty());

for (int i = 1; i < 6; i++)

astack.push(i);

Console.WriteLine( "num values in stack: " +astack.size());

astack.display();

Console.WriteLine("popping value" +astack.pop());

Console.WriteLine( "value 5 should have been removed");

astack.display();

}

/// <summary>

/// Main class calls driver

/// </summary>

/// <param name="args"></param>

static void Main(string[] args)

{

stackdriver();

}

}

}