//==========================

#pragma once

template<class type>

\_\_interface IComparable {

int compareTo(type\* obj);

};

//===========================

#pragma once

#include"IComparable.h"

#include<string>

using namespace std;

template<typename type>

class Stack:public IComparable<type>

{

type\* data;

int maxLength, length;

void set\_maxLength(int value)

{

if (value > 0)

{

maxLength = value;

data = new type[maxLength];

length = 0;

}

else

throw "error";

}

public:

int compareTo(type\* obj)

{

return maxLength - (\*obj);

}

int get\_maxLength()

{

return maxLength;

}

int get\_length()

{

return length;

}

Stack(int maxLength)

{

set\_maxLength(maxLength);

}

Stack() :Stack(15) {}

void push(type value)

{

if (length<maxLength)

{

data[length] = value;

length++;

}

else

{

throw "out of memory";

}

}

//type pop();

bool pop(type& value)

{

if (length > 0)

{

value = data[length - 1];

length--;

return true;

}

else

return false;

}

string toString()

{

if (length==0)

{

return "stack is empty";

}

string s = "";

try {

for (int i = 0; i < length; i++)

{

s += " " + data[i].toString();

}

}

catch{

for (int i = 0; i < length; i++)

{

s += " " + to\_string( data[i]);

}

}

return s;

}

~Stack() {

delete[] data;

}

};

//===========================================

// ConsoleApplication53.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include<iostream>

#include "Stack.h"

using namespace std;

int main()

{

//З клавіатури вводиться певна кількість символів, поки не буде введено 0. Знайти суму елментів, які більші за останній.

Stack<int> stack;

int number;

do

{

printf("number=");

cin >> number;

if (number!=0)

{

stack.push(number);

}

} while (number!=0);

int lastNumber;

int sum = 0;

stack.pop(lastNumber);

while (stack.pop(number))

{

if (number>lastNumber)

{

sum += number;

}

}

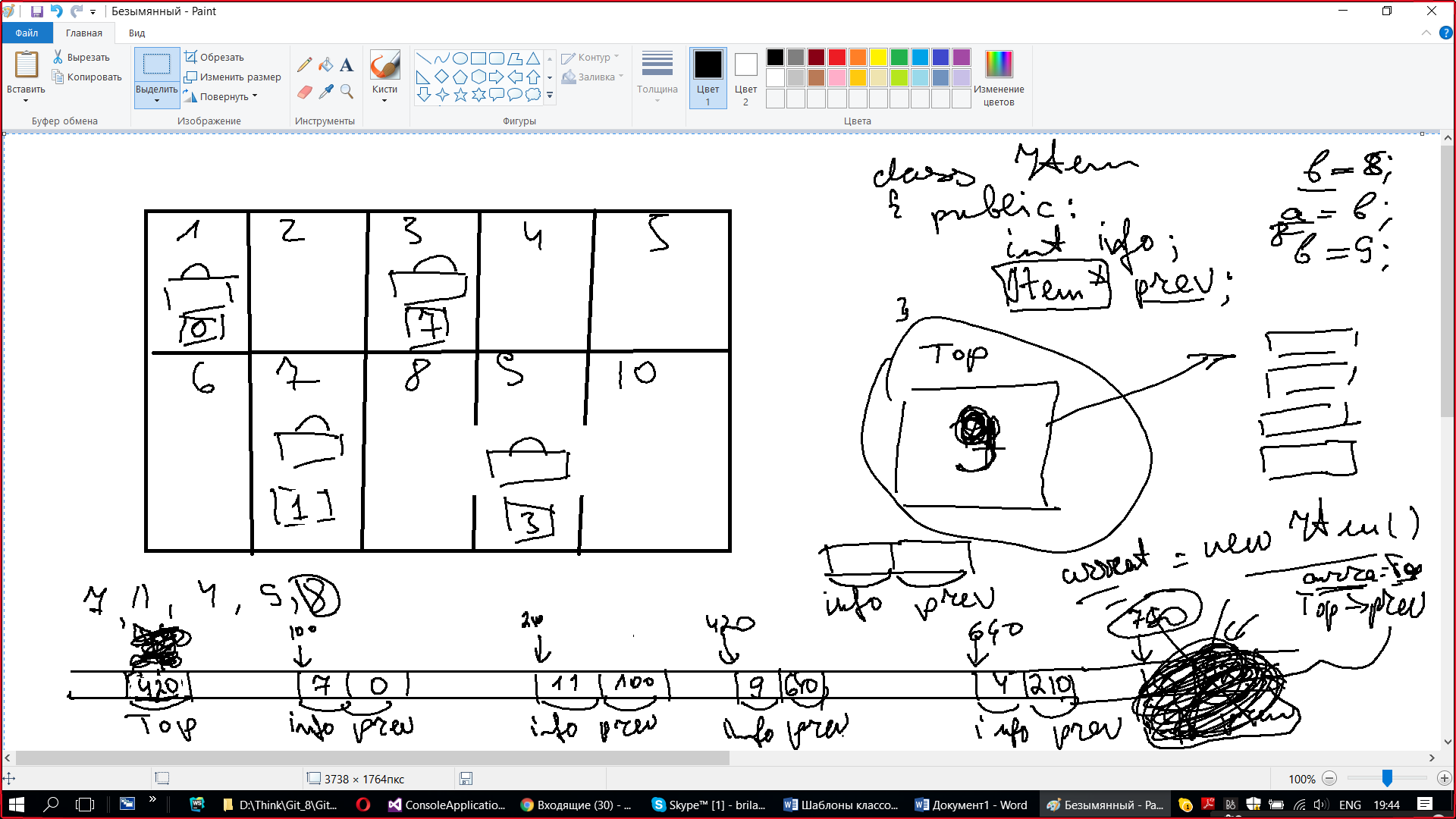
printf("Sum=%d\n",sum);

system("pause");

return 0;

}

//=========================================



#pragma once

template<typename type>

class Item

{

public:

type Info; //Поле даних

Item\* Prev; //Поле зв"язку (містить адресу попереднього елемента)

Item(type info, Item\* prev):Info(info), Prev(prev) {}

};

//---------------------------------

#pragma once

#include"Item.h"

template<typename type>

class Stack

{

public:

Item<type>\* Top;

Stack() {

Top = 0;

}

void push(type value) {

Item<type>\* current = new Item<type>(value, Top);

Top = current;

}

bool pop(type& value)

{

if (Top!=0)

{

value = Top->Info; //Зчитуємо значення з останнього елемента (його адреса знаходиться у Тор)

Item<type>\* current = Top;

Top = Top->Prev; //У Тор записуємо адресу, яка була адресою попереднього елемента.

delete current;

return true;

}

else

{

return false;

}

}

};

//-------------------------------------

// ConsoleApplication54.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include<iostream>

#include "Stack.h"

using namespace std;

int main()

{

//З клавіатури вводиться певна кількість символів, поки не буде введено 0. Знайти суму елментів, які більші за останній.

Stack<int> stack;

int number;

do

{

printf("number=");

cin >> number;

if (number != 0)

{

stack.push(number);

}

} while (number != 0);

int lastNumber;

int sum = 0;

stack.pop(lastNumber);

while (stack.pop(number))

{

if (number > lastNumber)

{

sum += number;

}

}

printf("Sum=%d\n", sum);

system("pause");

}

//---====================================

ЗРОБИТИ САМОСТІЙНО

