**Тема № 2**

Индивидуальное задание:

Написать программу по созданию, добавлению, просмотру и решению поставленной задачи (в рассмотренных примерах это действие отсутствует) для однонаправленного линейного списка типа СТЕК. Реализовать сортировку стека двумя рассмотренными выше методами.

Решение поставленной задачи представить в виде блок-схемы.

Во всех заданиях создать список из положительных и отрицательных случайных целых чисел.

1. Разделить созданный список на два: в первом – положительные числа, во втором – отрицательные.

* 1. **Создание консольного приложения**

Текст программы:

#include <iostream>

#include <iomanip>

struct Stack

{

int value;

Stack\* Next, \* Head;

};

void Push(int value, Stack\*& MyList)

{

Stack\* temp = new Stack;

temp->value = value;

temp->Next = MyList->Head;

MyList->Head = temp;

}

int\* Pop(Stack\* MyList)

{

if (MyList->Head)

{

int value = MyList->Head->value;

Stack\* next = MyList->Head->Next;

delete MyList->Head;

MyList->Head = next;

return &value;

}

else

{

return NULL;

}

}

void PrintStack(Stack\* MyList)

{

Stack\* temp = MyList->Head;

std::cout.setf(std::ios::showpos);

while (temp != NULL)

{

std::cout << std::setw(3) << temp->value << " ";

temp = temp->Next;

}

std::cout << std::endl;

}

void ClearStack(Stack\* MyList)

{

while (MyList->Head != NULL)

{

Stack\* temp = MyList->Head->Next;

delete MyList->Head;

MyList->Head = temp;

}

}

void SortByValue(Stack\* MyList)

{

Stack\* a;

int tmpData;

for (bool go = true; go; )

{

go = false;

a = MyList->Head;

while (a->Next != NULL)

{

if (a->value > a->Next->value)

{

tmpData = a->value;

a->value = a->Next->value;

a->Next->value = tmpData;

go = true;

}

a = a->Next;

}

}

}

void SortByAdress(Stack\* srtStack)

{

Stack\* t, \* m, \* a, \* b;

for (bool go = true; go; )

{

go = false;

a = t = srtStack->Head;

b = srtStack->Head->Next;

while (b != NULL)

{

if (a->value > b->value)

{

if (t == a)

srtStack->Head = b;

else

t->Next = b;

a->Next = b->Next;

b->Next = a;

m = a, a = b, b = m;

go = true;

}

t = a;

a = a->Next;

b = b->Next;

}

}

}

void SplitStack(Stack\* srtStack, Stack\* StackPositive, Stack\* StackNegative)

{

Stack\* temp = srtStack->Head;

int Number;

while (temp != NULL)

{

Number = temp->value;

if (Number >= 0)

{

Push(Number, StackPositive);

}

else

{

Push(Number, StackNegative);

}

temp = temp->Next;

}

}

void FillStack(Stack\* MyList)

{

ClearStack(MyList);

std::cout << "Заполнение стека: ";

for (int i = 0; i < 10; i++)

{

Push(rand() % 20 - 10, MyList);

}

PrintStack(MyList);

}

int main()

{

setlocale(LC\_ALL, "RUS");

Stack\* MyList = new Stack;

MyList->Head = NULL;

FillStack(MyList);

std::cout << "Сортировка по адресам: ";

SortByAdress(MyList);

PrintStack(MyList);

FillStack(MyList);

std::cout << "Сортировка по значениям: ";

SortByValue(MyList);

PrintStack(MyList);

FillStack(MyList);

Stack\* stkPositive = new Stack, \* stkNegative = new Stack;

stkPositive->Head = NULL; stkNegative->Head = NULL;

std::cout << "Разделение стека:" << std::endl;

SplitStack(MyList, stkPositive, stkNegative);

std::cout << " Стек содержащий положительный числа: ";

PrintStack(stkPositive);

std::cout << " Стек содержащий отрицательный числа: ";

PrintStack(stkNegative);

system("pause");

return 0;

}

Результаты выполнения:

