**8.1. Стеки**

Стек — это частный случай однонаправленного списка, добавление элементов в который и выборка из которого выполняются с одного конца, называемого вершиной стека. Другие операции со стеком не определены. При выборке элемент исключается из стека. Говорят, что стек реализует принцип обслуживания LIFO (last in — first out, последним пришел — первым ушел). Стек проще всего представить себе как закрытую с одного конца узкую трубу, в которую бросают мячи. Достать первый брошенный мяч можно только после того, как вынуты все остальные. Стеки широко применяются в системном программном обеспечении, компиляторах, в различных рекурсивных алгоритмах.

Ниже приведена программа, которая формирует стек из пяти целых чисел (1,2,3,4,5) и выводит его на экран. Функция помещения в стек по традиции называется push, а выборки – pop. Указатель для работы со стеком (top) всегда ссылается на его вершину.

#include <iostream.h>

struct Node{

int d;

Node \*p;

};

Node \* first(int d);

void push(Node \*\*top, int d);

int pop(Node \*\*top);

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

int main(){

Node \*top = first(1);

for (int i = 2; i<6; i++)push(&top, i);

while (top)

cout « pop(&top) « ' ';

return 0;

}

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

// Начальное формирование стека

Node *\** first(int d){

Node \*pv = new Node;

pv->d = d;

pv->p = 0;

return pv;

}

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

// Занесение в стек

void push(Node \*\*top, int d){

Node \*pv = new Node;

pv->d = d;

pv->p = \*top;

\*top = pv;

}

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

// Выборка из стека

int pop (Node \*tор){

int temp = (\*top)->d;

Node \*pv = tор;

tор = (\*top)->p;

delete pv;

return temp;

}

Результат работы программы:

5 4 3 2 1