**8.2. Очереди**

Очередь — это частный случай однонаправленного списка, добавление элементов в который выполняется в один конец, а выборка — из другого конца. Другие операции с очередью не определены. При выборке элемент исключается из очереди. Говорят, что очередь реализует принцип обслуживания FIFO (first in — first out, первым пришел — первым ушел). Очередь проще всего представить себе, постояв в ней час-другой. В программировании очереди применяются, например при моделировании, диспетчеризации задач операционной системой, буферизованном вводе/выводе.

Ниже приведена программа, которая формирует очередь из пяти целых чисел и выводит ее на экран. Функция помещения в конец очереди называется add, а выборки — del. Указатель на начало очереди называется pbeg, указатель на конец — pend.

#include <iostream.h>

struct Node{

int d;

Node \*p;

};

Node \* first(int d);

void add(Node \*\*pend, int d);

int del(Node \*\*pbeg);

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

int main(){

Node \*pbeg = first(1);

Node \*pend = pbeg;

for (int i = 2; i<6; i++) add(&pend; i);

while (pbeg)

cout « del(&pbeg) « ' ';

return 0;

}

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

// Начальное формирование очереди

Node \* first(int d){

Node \*pv = new Node;

pv->d = d;

pv->p = 0;

return pv;

}

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

/ / Добавление в конец

void add(Node \*\*pend, int d)(

Node \*pv = new Node;

pv->d = d;

pv->p = 0;

(\*pend)->p = pv;

\*pend = pv;

}

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

// Выборка

int del(Node \*\*pbeg){

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

Node \*pv = \*pbeg;

\*pbeg = (\*pbeg)->p;

delete pv;

return temp;

}

Результат работы программы:

12 3 4 5