

# Структуры данных

## Стек (stack)

**Стеки** – последовательный список переменной длины, включение и исключение элементов из которого выполняются только с одной стороны списка, называемого **вершиной стека**. Применяются и другие названия стека - магазин и очередь, функционирующая по принципу **LIFO**: Last – In, First- Out - "последним пришел - первым исключается (выбирается)".

### Основные операции над стеком:

- включение нового элемента (английское название push - заталкивать)
- исключение элемента из стека (англ. pop - выскакивать).

Полезными могут быть также вспомогательные операции:

- определение текущего числа элементов в стеке;
- очистка стека;
- неразрушающее чтение элемента из вершины стека, которое может быть реализовано, как комбинация основных операций: `x:=pop(stack); push(stack,x)`.

**Пример: принцип включения элементов в стек и исключения элементов из стека.**

На рисунке изображены состояния стека:

а) пустого;

б-г) после последовательного включения в него элементов с именами 'А', 'В', 'С';

д, е) после последовательного удаления из стека элементов 'С' и 'В';

ж) после включения в стек элемента 'D'.



# Структуры данных

Pascal

## Стек (stack)

Pascal

Через указатели и элементы списка

### Практическое занятие

Сформировать стек, добавить в него N элементов, а затем считать элементы и вывести их на экран. Элементы должны быть символьного типа. Ввод элементов с клавиатуры, признак конца ввода – точка.

```

type Pt=k; k=record d: char; p: pt end;
var ph: Pt; c: char;
Procedure FormSt (var ph: Pt; var c: char);
(*Формирование первого элемента стека *)
begin
  New (ph); (* создать динамический стек *)
  ph.p := nil; (* указатель конца стека *)
  ph.d := c; (* значение первого элемента *)
end;
Procedure AddSt (var ph: Pt; var c: char);
(* Занесение следующего элемента в стек *)
  var px: Pt; (*вспомогательный указатель*)
begin
  New(px); (*создание вспомогательного указателя*)
  px.p := ph; (* присвоение вспомогательному
указателю адреса очередного элемента стека *)
  ph := px; (* сдвиг указателя на вершину стека *)
  ph.d := c; (*занесение элемента в вершину
стека*)
end;
Procedure DelSt (var ph: Pt; var c: char);
(* Выборка следующего элемента из стека *)
  var px: Pt; (*вспомогательный указатель*)
begin
  px := ph; (*сохранение адреса вершины во
вспомогательном указателе *)
  c := ph.d; (* чтение значения из вершины *)
  ph := ph.p; (* смещение адреса вершины *)
  Dispose (px); (*освобод. вспомогат. указателя*)
end;

```

```

Program Stack-P;
type Pt=k; k=record d: char; p: pt end;
var ph: Pt; c: char; n: integer;

begin
  n:=1;
  writeln ('Введите первый элемент ', n:3);
  readln (c);
  FormSt (ph, c);
  repeat
    n:=n+1;
    writeln (' Введите ', n:3,'-й элемент');
    readln (c);
    if c<>'.'
    then AddSt (ph, c) else break;
    AddSt (ph, c);
  until c='.';
  n:=1;
  repeat
    DelSt (ph, c);
    writeln (n:3,'-й компонент = ', c);
    n:=n+1;
  until ph=nil;
end.

```

# Структуры данных

Pascal

Стек (stack)  
Через массив

C / C++

## Практическое занятие

```

const
  MAX=100;

var
  stack: array [1..MAX] of char;
  tos: integer; {Вершина стека }

procedure push (i: char); {Занесение
элемента в стек}
begin
  if (tos >= MAX) then
    writeln('Стек полон');
  stack[tos] := i; {вставляем элемент в стек}
  inc(tos);
end;

function pop : char; {Выборка элемента из
стека }
begin
  dec(tos);
  if (tos <= 0) then
    writeln('Стек пуст');
  pop:=stack[tos];
end;

```

```

int stack [MAX];
int tos=0; // вершина стека

void push (int i)
/* Занесение элемента в стек */
{
  if (tos >= MAX)
  {
    printf (" Стек полон \n");
    return 0;
  }
  stack[tos] = i; // заносим элемент в стек
  tos++;
return 0;
}

int pop (void)
/* Выборка элемента из стека */
tos--;
if (tos < 0)
{
  printf (" Стек пуст \n");
  return 0;
}
return stack [tos];
}

```