

# Реализации стека и очереди на языке C++



# Стек

**Стек** — структура данных, в которой доступ к элементам организован по принципу *LIFO* (*last in — first out*, «последним пришёл — первым вышел»). Чаще всего принцип работы стека сравнивают со стопкой тарелок: чтобы взять вторую сверху, нужно снять верхнюю.

# Операции над стеками

- 1.Добавление элемента в стек
- 2.Проверка пустоты стека
- 3.Удаление элемента из стека

# Добавление элемента в стек

Чтобы поместить элемент в стек необходимо указать сам элемент и индекс элемента стека

```
Void Push(int Top,int Element)
```

```
{
```

```
If (Top==maxstack) exit;//maxstack – размер  
стека
```

```
Stack[top]=element;//Стек заполнен
```

```
Top++;//Сдвигает на один элемент вверх
```

```
}
```

# Проверка пустоты стека

Данная функция возвращает  $p==1$  стек пуст

Или  $p==2$  возвращающая

```
Void EMPTY(int top)
```

```
{
```

```
if(top==0)p=1; //Стек пуст;
```

```
Else p=2; //Стек не пуст
```

```
}
```

# Удаление элемента из стека

```
Void remove(int top)
{
if(top==0)exit;
Top--;
}
```

# Очереди

**Очередь** — структура данных с дисциплиной доступа к элементам «первый пришёл — первый вышел» (FIFO, First In — First Out).  
Добавление элемента (принято обозначать словом `enqueue` — поставить в очередь) возможно лишь в конец очереди, выборка — только из начала очереди (что принято называть словом `dequeue` — убрать из очереди), при этом выбранный элемент из очереди удаляется.

# Операции в очереди

---

1. Добавление элементов
2. Проверка на пустоту
3. Удаление элементов



# Добавление элемента в очередь

Массив- queue

Maxqueue-размер очереди

```
Void Add(int Last,int number)
```

```
{
```

```
If(last==maxqueue)exit;
```

```
queue[last]=number;
```

```
Last++;}
```

# Проверка на очереди на наличие элементов

P=1 возвращает если очередь пуста

P=2 если очередь не пуста

```
Void Empty(int first, int Last)
```

```
{
```

```
  If(first==last)p=1;
```

```
  Else p=2;
```

```
}
```

# Удаление элемента из очереди

```
Void remove (int first, int last)
{
If (first==last)exit;//Очередь пуста
Start++;//Сдвигает на один элемент вправо

}
```