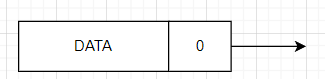
**ЛЕКЦИЯ 4**

**ДИНАМИЧЕСКИЕ СТРУКТУРЫ ДАННЫХ. ОДНО И ДВУНАПРАЛВЕННЫЕ СПИСКИ**

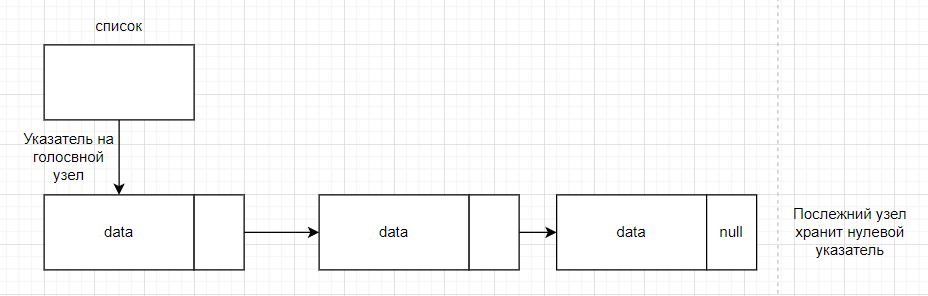
1. Динамическая структура – это объект, который работает с динамической памятью;
2. Обычно, динамическая память выделяется сегментами в соответствии с программами;
3. Node – поля и связи между ними;
4. Для описания node и создается динамическая структура через систему указателей;
5. Если node не ссылается на другие, то указателю на него присваевается значение nullptr;

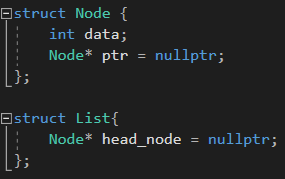
**СПИСКИ ОДНОНАПРАВЛЕННЫЕ**



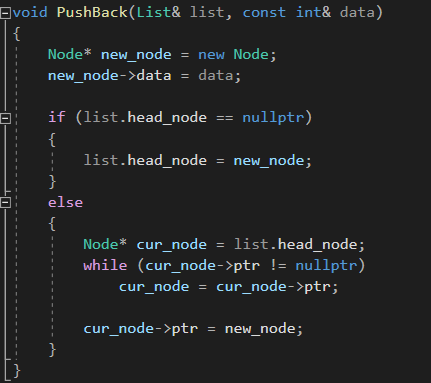
1. Выше – пример node;
2. Два поля: с данными и указателем;
3. Работа с node организована через тип данных список;
4. В лабораторной сделать через **ВСЕ 4** типа списков;
5. Если есть классификация в коде, то список её в этой классификации имеет поле с указателем на начальный узел;
6. Возможно создать указатель на любой нужный узел.

**КЛАССИФИКАЦИЯ СПИСКОВ: ОДНОСВЯЗНЫЕ**

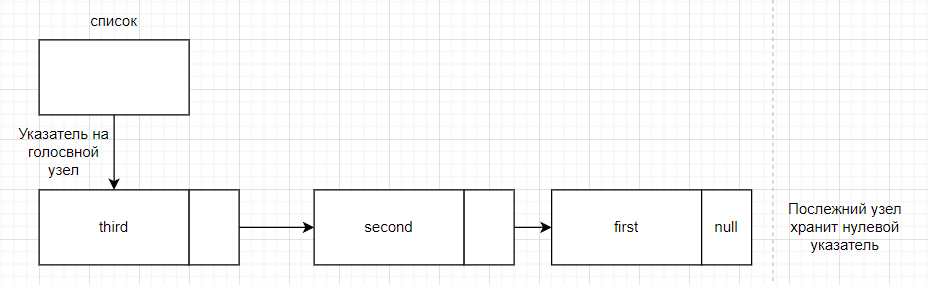




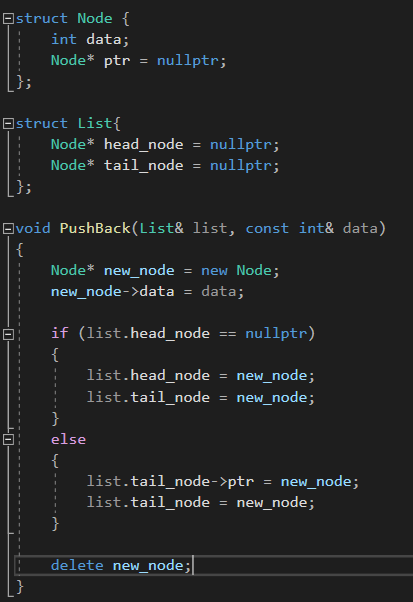
**ДОБАВЛЕНИЕ ЭЛЕМЕНТА В КОНЕЦ СПИСКА**



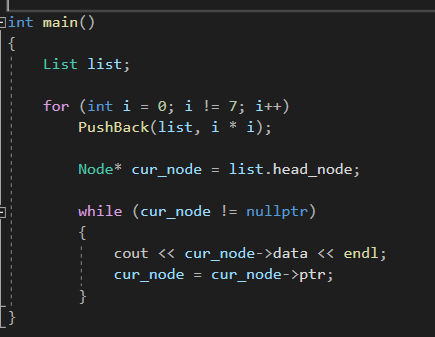
**ДОБАВЛЕНИЯ ЭЛЕМЕНТА В НАЧАЛО СПИСКА**



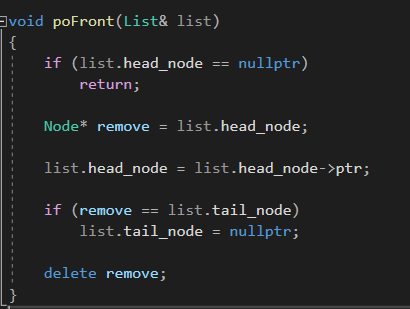
**ЕЩЕ СПОСОБ ДОБАВИТЬ ЭЛЕМЕНТ В КОНЕЦ СПИСКА**



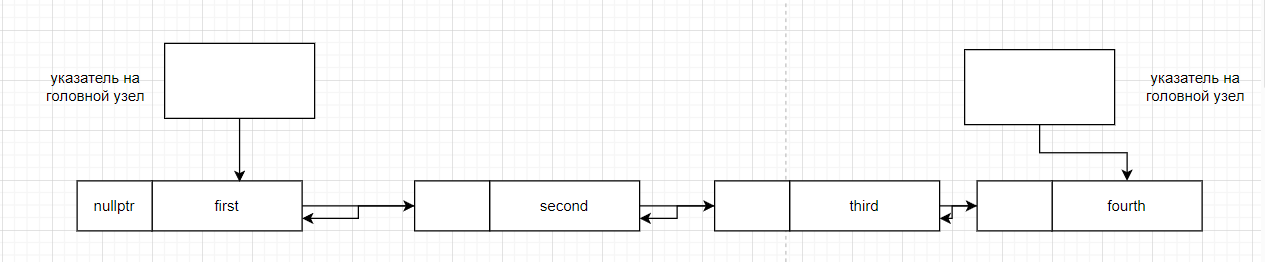
**ВЫВЕДЕНИЕ СПИСКА**



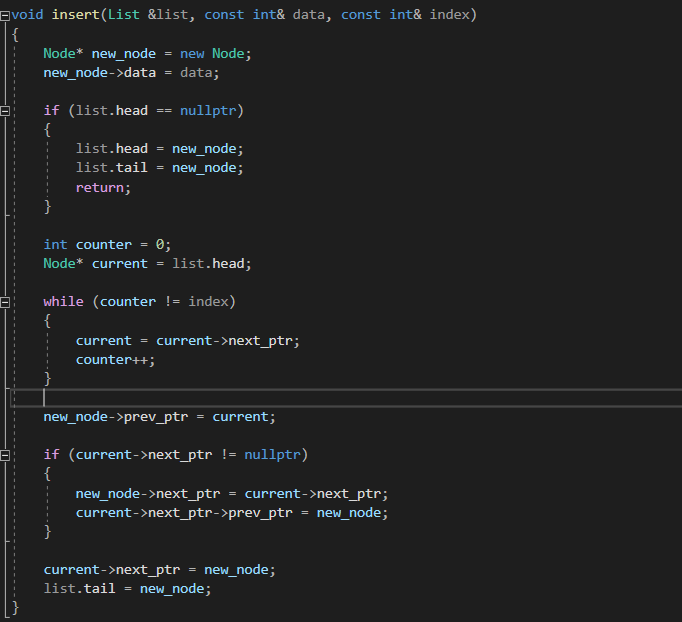
**УДАЛЕНИЕ ИЗ СПИСКА**



**ДВУСВЯЗНЫЕ СПИСКИ**

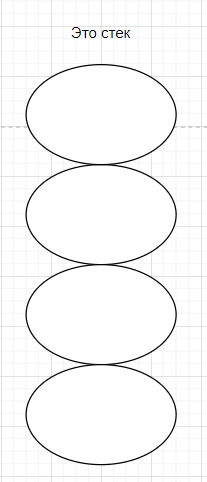


**ДОБАВЛЕНИЕ ЭЛЕМЕНТА В СЕРЕДИНУ СПИСКА**

****

**СТЕКИ**

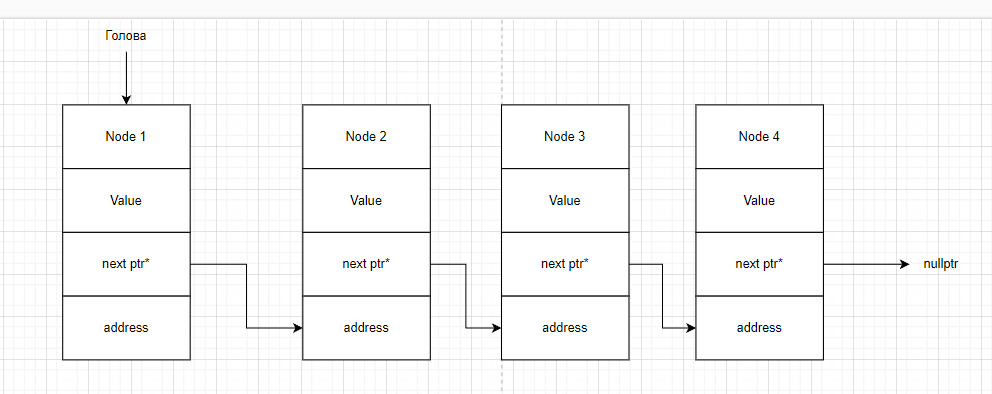
1. Линейная структура данных, когда добавление и удаление элементов идет с вершины стека;

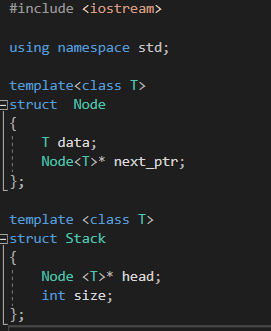


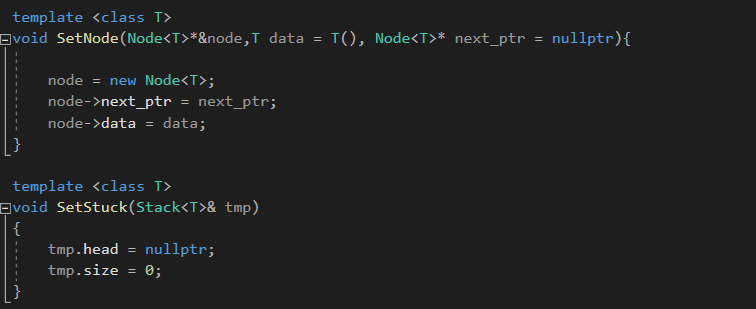
1. **Last in, first out**. Последним пришел = первым вышел;

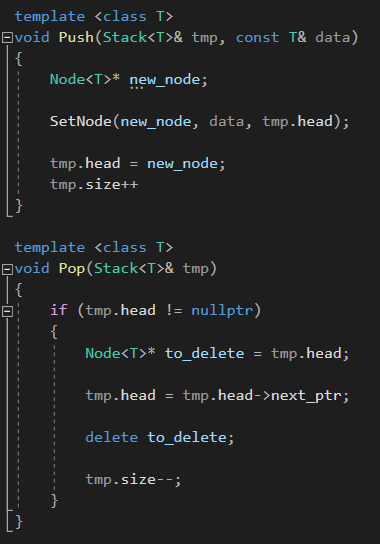
**СТЕК КАК ОДНОСВЯЗНЫЙ СПИСОК**

1. Элемент стека – узел, который содержит:
2. Данные;
3. Указатель на следующий узел;
4. Собственный адрес, с помощью которого элементы стека связаны друг с другом;
5. Первый элемент стека – голова, второй – размер;

****

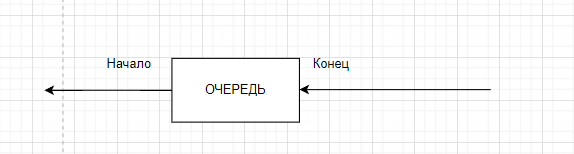
****





**ОЧЕРЕДИ**

1. Абстрактный тип данных, в которой элементы располагаются по закону “первый пришёл, первый ушел”;



1. Добавлять элементы в начало, убирать из конца;

