

Списки, стеки, очереди

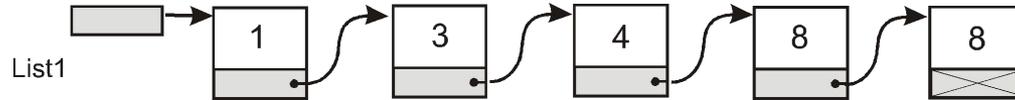
Списки

Самая простая динамическая структура данных — это линейный список.

Линейные списки находят широкое применение в приложениях, где непредсказуемы требования на:

- размер памяти, необходимой для хранения данных;
- большое число сложных операций над данными, особенно включений и исключений.

Вставка элементов в список



Вставка

- в начало списка (как в стек);
- в конец списка (как в очередь);
- в упорядоченный список с сохранением упорядоченности. *(DemoList)*

Упорядоченный список

Для получения упорядоченного списка вовсе необязательно сортировать его после построения, достаточно добавлять новый элемент таким образом, чтобы список все время оставался упорядоченным.

Такой метод позволяет иметь упорядоченный список на каждом шаге, не заботясь о том, закончились ли элементы, которые следует добавить в список.

Будем считать ключом, по которому производится упорядочивание, значение информационного поля.

Упорядоченный список

При добавлении элемента в список необходимо сначала найти место, куда его следует поместить. При этом возможны следующие варианты:

- список пуст;
- новый элемент меньше первого элемента существующего списка, и его следует поместить в начало;
- новый элемент требуется поместить в середину существующего списка.

В последнем варианте требуется сначала определить место, куда следует вставить новый элемент. Для этого нужно двигаться по списку до тех пор, пока либо не найдется элемент, больший или равный вставляемому, либо не будет достигнут конец списка.

Для удобства вставки используется дополнительный указатель — на предыдущий элемент.

см. DemoList

Стек

Использование стека в повседневной ЖИЗНИ

- грузовой отсек транспортного самолета
- тупиковый железнодорожный разъезд для сортировки вагонов
- винтовочный патронный магазин

Использование стека в программировании

Любая операционная система содержит так называемый **системный стек**, куда помещаются адреса возврата и ряд других значений при вложенном вызове процедур и функций.

Использование стека в программировании

- системный стек: вызов процедур и функций
- компилятор на этапе синтаксического разбора текста программы
- интерпретатору стек необходим для вычисления выражений

Принцип работы стека

Таким образом, *стек* — это структура, работа с которой происходит по принципу LIFO:

последним пришел — первым ушел

(от англ. Last — In — First — Out). Включение элемента в стек и исключение элемента из стека выполняются только с одной стороны, которая называется *вершиной* стека.

Для работы со стеком необходимы следующие операции:

- инициализация стека, то есть подготовка структуры;
- включение нового элемента в стек
(англ. *push* – заталкивать);
- проверка стека на пустоту;
- исключение элемента из стека
(англ. *pop* – выталкивать).

При решении задач, использующих стек, совершенно неважно, каким образом организован сам стек.

Мы рассмотрим два способа реализации стека:

- динамическая реализация
- реализация с использованием массива

Задача

На двух стержнях перемешаны кольца двух цветов.

Используя третий стержень, переместить на разные стержни кольца, одинаковые по цвету

Очередь

Очередь — это структура, работа с которой происходит по принципу *FIFO*:

первым пришел — первым ушел

(от англ. *First — In — First — Out*).

Принцип работы очереди

Очередь — это структура, в которую новый элемент добавляется только с одной стороны. Эта сторона называется *концом* или *хвостом* (*tail*) очереди. Говорят, что *элемент добавляется в конец очереди*. Взятие элемента из очереди происходит с другой стороны — из *начала* (или из *головы* (*head*)) очереди.

В качестве примера очереди в программировании можно назвать очередь процессов к разделяемому ресурсу под управлением операционной системы

Основные операции с очередью

- инициализация очереди;
- добавление элемента в очередь;
- проверка очереди на пустоту;
- взятие элемента из очереди.

В зависимости от характера решаемой задачи очередь можно организовать статически или динамически.

Если в процессе работы очередь то очень длинная (несколько десятков или сотен элементов), то короткая (один-два элемента), имеет смысл реализовать очередь с использованием динамической структуры (списка).

Если заранее известна максимальная длина очереди, то можно использовать статический массив.

Рассмотрим оба этих способа.

Динамическая организация очереди

При динамической реализации основой очереди является линейный односвязный список.

Для работы с очередью необходимы два указателя: на **голову** очереди и на ее **хвост**.

Очередь на массиве

В этом случае для хранения значений элементов очереди используется массив размерностью `SizeQueue`. Необходимо еще два указателя:

- *Голова*, указывающая на первый элемент в очереди, и
- *Хвост* — на элемент массива, в который можно поместить очередное значение.

Если очередь пуста, то *Голова* и *Хвост* указывают на один и тот же элемент массива.

Очередь на массиве

Поскольку в процессе работы при помещении значений в очередь массив заполняется с правой стороны, а при извлечении из очереди — освобождается слева, то имеет смысл сделать массив кольцевым. Это означает, что когда значение помещается в последний элемент массива, указатель *Хвост* перемещается к первому элементу массива (аналогично для *Головы*). Таким образом, *Хвост* может быть левее *Головы*. см. *QueueArray*

Очередь пуста — это значит, что в массиве не содержится ни одного значения. Очередь может стать пустой в процессе работы, при этом *Голова* и *Хвост* будут указывать на один и тот же элемент, но не обязательно на первый элемент массива. Поэтому для проверки очереди на пустоту необходим еще счетчик элементов очереди, который равен нулю, если очередь пуста, и имеет значение *SizeQueue*, если очередь забита полностью.

Задача

Завод скоропортящейся продукции, например, глазированных сырков, имеет склад, куда поступает готовая продукция: коробки с сырками. В магазин по заявке следует отправить первой ту коробку, которая раньше всех поступила на склад. Но если заявок поступило много и склад опустел, то вывозить сырки можно сразу из цеха, минуя склад. Незаявленная продукция остается на складе. В отчете требуется представить характеристики коробок, отправленных на продажу.

Блок-схема задачи

