Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Курсовая работа**

**по курсу «Языки и методы программирования»**

**II семестр**

**Задание №8**

**«Линейные списки»**

|  |  |
| --- | --- |
| **Студент:** | Ватулин Валентин |
| **Группа:** | М8О-106Б-19 |
| **Преподаватель:** | Дубинин Алексей |
| **Оценка:** |  |
| **Дата:** |  |

Москва, 2020

**Содержание**

[Введение 3](#_Toc43639262)

[Списки и их виды 4](#_Toc43639263)

[Описание программы 6](#_Toc43639264)

[Заключение 8](#_Toc43639265)

[Список литературы 9](#_Toc43639266)

# Введение

Существует несколько структур данных для хранения информации. Все они, не смотря на свою некоторую схожесть, предназначаются для разных нужд и задач. Некоторые больше подходят для хранения данных, к которым необходим быстрый доступ, как например вектор. Некоторые ориентируются на частую изменчивость данных, как например список. Для эффективной работы программисту необходимы знания этих структур данных, их возможностей и особенностей.

# Списки и их виды

Связный список является простейшим типом данных динамической структуры, состоящей из элементов (узлов). Каждый узел включает в себя в классическом варианте два поля:

* данные (в качестве данных может выступать переменная, объект класса или структуры и т. д.)
* указатель на следующий узел в списке.

Элементы связанного списка можно помещать и исключать произвольным образом.

Связный список, содержащий только один указатель на следующий элемент, называется односвязным.

Связный список, содержащий два поля указателя – на следующий элемент и на предыдущий, называется двусвязным.

**Линейный однонаправленный список** — это структура данных, состоящая из элементов одного типа, связанных между собой последовательно посредством указателей. Каждый элемент списка имеет указатель на следующий элемент. Последний элемент списка указывает на NULL. Элемент, на который нет указателя, является первым (головным) элементом списка. Здесь ссылка в каждом узле указывает на следующий узел в списке. В односвязном списке можно передвигаться только в сторону конца списка. Узнать адрес предыдущего элемента, опираясь на содержимое текущего узла, невозможно.



**Двусвязный список.** Здесь ссылки в каждом узле указывают на предыдущий и на последующий узел в списке. Как и односвязный список, двусвязный допускает только последовательный доступ к элементам, но при этом дает возможность перемещения в обе стороны. В этом списке проще производить удаление и перестановку элементов, так как легко доступны адреса тех элементов списка, указатели которых направлены на изменяемый элемент.



**Кольцевой список.** Разновидностью связных списков является кольцевой (циклический, замкнутый) список. Он тоже может быть односвязным или двусвязным. Последний элемент кольцевого списка содержит указатель на первый, а первый (в случае двусвязного списка) — на последний.



**Двусвязный циклический список.** Каждый узел двунаправленного (двусвязного) циклического списка (ДЦС) содержит два поля указателей — на следующий и на предыдущий узлы. Указатель на предыдущий узел корня списка содержит адрес последнего узла. Указатель на следующий узел последнего узла содержит адрес корня списка.



**Список с барьерным элементом.** Это список с одним фиктивным элементом - барьером. В таком случае, если список пуст, то ссылки списка замыкаются на самом барьере, в случаях, когда длина списка больше нуля, поле барьерного элемента, которое указывает на следующий элемент, указывает на первый элемент списка, а последний элемент списка указывает на барьер. Такая реализация позволяет не обрабатывать отдельно граничный случаи и позволяет обходить ошибки NULL

**Преимущества:**

Выделение памяти осуществляется по мере ввода новых элементов. Удаление/добавление элемента осуществляется переустановкой указателей, при этом сами данные не копируются. Вставка за О(1)

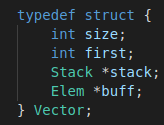
**Недостатки:**

Возможен только последовательный доступ к элементам. Обращение к элементу за О(n)

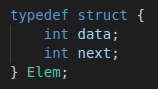
# Описание программы

Задание: составить и отладить программу на языке Си для обработки линейного односвязного списка с отображением списка на массив. Навигацию по списку следует реализовать с применением итераторов. Нестандартное действие: добавить k экзепляров последнего элемента в начало списка.

Односвязный список представляет из себя структуру, состоящую из вектора, стека, номера первого элемента и размера списка.

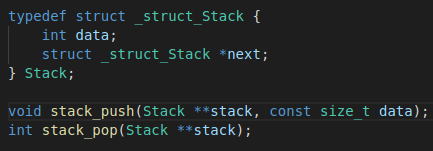


Элемент списка представляет из себя структуру, состоящую из содержащихся в нем данных и индекса следующего элемента.

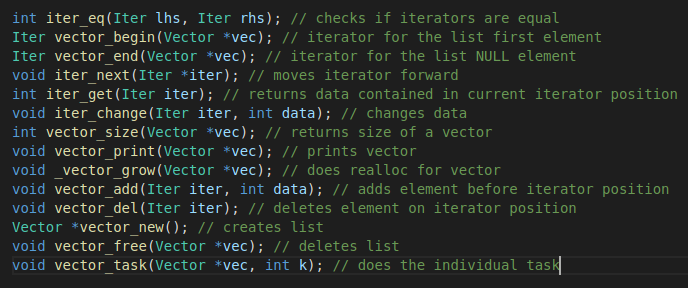


Итератор представляет из себя структуру, состоящую из указателя на список и позиции, в которой он находится.

В стеке хранятся индексы свободных ячеек.



Функции для работы со списком:



# Заключение

Списки имеют свои преимущества: нет определенного порядка в памяти (если он не на массиве), добавление элемента за O(1) и недостатки: обращение к элементу за O(n), дополнительная память под хранение указателей. На списках, как и на векторах можно реализовать другие полезные структуры: stack, queue, double end queue.

Линейный список на массиве является неудобной структурой данных, дающей преимущество только в очень специфичных случаях. Я уверен, что эта структура данных не пригодится мне в будущем.

# Список литературы

1. Поляков К.​“Динамические структуры данных”.
2. Б. Керниган, Д. Ритчи. “Язык программирования Си”
3. <https://prog-cpp.ru/data-dcs/>
4. <https://ru.wikipedia.org/wiki/Связанный_список>