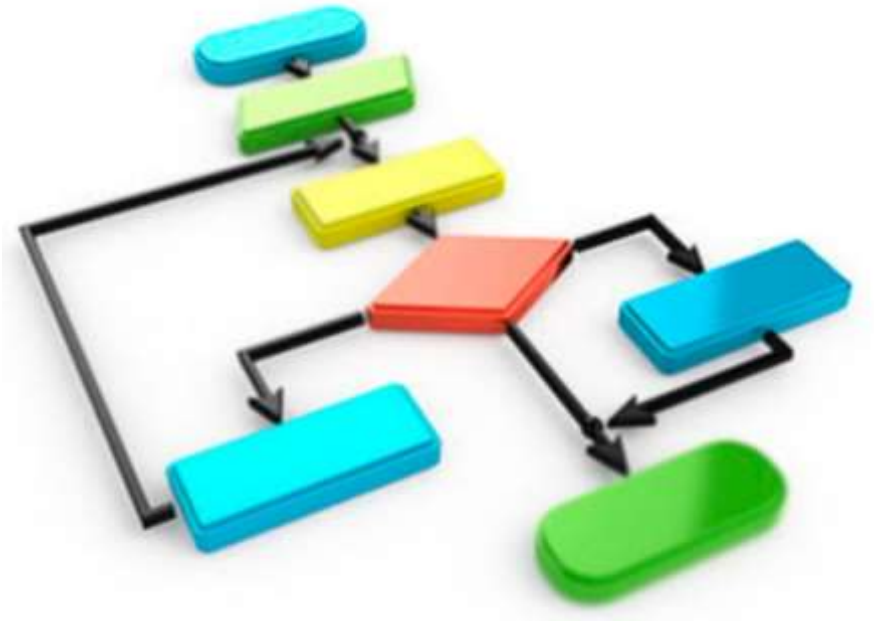
Толстова Н.С.

Динамические структуры данных

Лабораторный практикум



Оглавление

Лабораторная работа№1
Указатели

1. Понятие указателя

Все рассмотренные ранее типы данных содержат непосредственно данные, размещенные в памяти компьютера. Для организации динамической памяти применяются особые переменные, называемые указателями. Назначение их состоит в том, чтобы указывать местоположение какой-то другой переменной заранее определенного типа. Таким образом,

> **Указатель** – это переменная, которая в качестве своего значения содержит адрес первого байта памяти, по которому записанны данные.

Сам по себе указатель занимает в памяти всего четыре байта, а данные на которые он указывает, могут простираться в памяти на десятки килобайт.

При выполнении функции (даже самых простых) обмен данными осуществляется через стек – область памяти компьютера, специально выделяемую для этой цели. При вызове такой функции копии значений переменных записываются в стек в соответствии с типами указанными типами и после их выполнения результат сначала заносится в стек, а потом помещается в область памяти, предназначенную для хранения значений переменных. Таким образом, при выполнении программы занимается в два раза большая область памяти если не использовать указатели.

В свою очередь, при использовании указателей, в стеке хранятся только адреса ячеек памяти, в которых хранятся данные, адреса занимают в любом случае не более 4 байт, в то время, когда данные в десятки раз больше.

2. Объявление и инициализация указателей.

При определении (объявлении) указателя надо стремиться выполнить его инициализацию, то есть присвоение начального значения. Непреднамеренное использование неинициализированных указателей – распространенный источник ошибок в программе.

Задание 1.

Считайте с диска файл ***Ukaz1*** прочтите комментарии и рассмотрите способы объявления и инициализации указателей.

Задание 2.

Считайте с диска файл ***Ukaz2*** и ответьте на следующие вопросы:

- Какие указатели присутствуют в данной программе?
- На какие типы значений они указывают?
- На адреса каких переменных они ссылаются?

Ответы озвучьте преподавателю.

3. Операции с указателями.

С указателями можно выполнять арифметические действия. Но эти операции имеют смысл в основном при работе со структурами данных, последовательно размещенных в памяти данных, например, с массивами.

Задание 3.

1) Считайте файл ***Ukaz2.*** Определите в каком месте происходит изменение индекса элемента массива.
2) Измените программу так, чтобы элементы массива выводились в обратном порядке (начиная с элемента с индексом 5 и заканчивая элементом с индексом 0).

Результат продемонстрируйте преподавателю.

4. Контрольное задание

В компьютерной программе, предназначенной для автостоянок, заведен массив 50x100 чисел. В каждом элементе этого массива хранится 0 (место на стоянке свободно) или 1 (место занято). Составьте фрагмент программы, использующей указатели, определяющий сумму выручки, если одно место на стоянке стоит 15 руб.

Лабораторная работа №2
Динамические структуры данных. Формирование линейного односвязного списка. Просмотр элементов списка.

1. Понятие списка

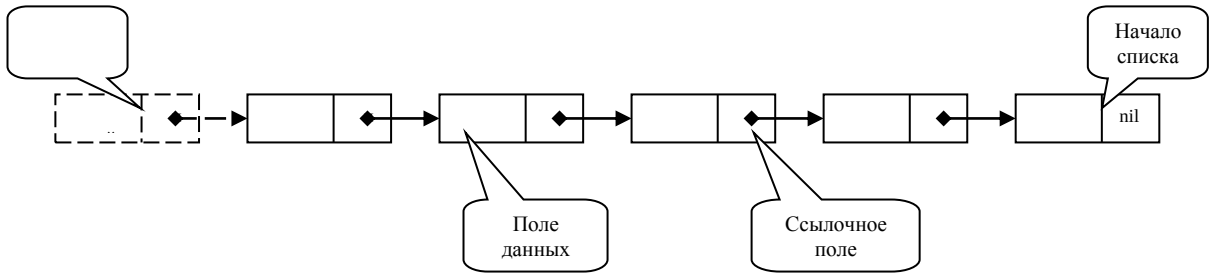
Разберем следующий пример. В процессе физического эксперимента многократно снимаются показания прибора (допустим, термометра) и записываются в компьютерную память для дальнейшей обработки. Заранее неизвестно, сколько будет произведено измерений.

Если для обработки таких данных не использовать внешнюю память (файлы), то разумно расположить их в динамической памяти. Во-первых, динамическая память позволяет хранить больший объем информации, чем статическая. А во-вторых, в динамической памяти эти числа можно организовать в связанный список, который не требует предварительного указания количества чисел, подобно массиву.

Линейный список – это данные динамической структуры, которые представляют собой совокупность линейно связанных однородных элементов и для которых разрешены следующие действия:

Добавление элемента в начало (голову) списка;
Добавление элемента в конец (хвост) списка;
Вставка элемента между двумя любыми другими элементами списка;
Удаление любого как крайнего, так и среднего элемента списка.

Из определения следует, что каждый элемент списка содержит поле данных (оно может иметь сложную структуру) и поле ссылки на следующий элемент. После ссылки последнего элемента должно содержать пустой указатель (nil).



2. Формирование списка (добавление элемента в начало)

Чтобы добавить новый элемент в список, необходимо:

1. Получить память для него;
2. Поместить туда информацию;
3. Добавить элемент в конец списка.

Задание 1.

Запустите приложение **mod_dync.exe**, выберите пункт «Ввод и формирование динамической цепочки», разберитесь в приведенном фрагменте программы, выполнив пошагово предложенный алгоритм.

Задание 2.

Считайте с диска файл **add_nach,** проанализируйте приведенную там программу и ответьте на следующие вопросы:

- Какого типа являются указатели на элементы линейного списка?
- Сколько информационных полей содержит элемент списка? Какого они типа?
- Какое количество ссылочных полей содержит элемент списка?
- Какие указатели на элементы списка участвуют в работе с ним?
- Какой указатель ссылается на первый элемент, последний элемент, а какой на текущий?
- Где в тексте программы происходит заполнение информационных полей?
- Где в тексте программы устанавливается связь с вновь созданным элементом списка?
- Где в тексте программы происходит перемещение указателя начала на вновь созданный элемент?
- До каких пор элементы будут добавляться в список?

Задание 3.

Модифицируйте текст программы файла **add_nach** так, чтобы:

1) элемент списка содержал два информационных поля;
2) значение второго задавалось случайным образом из диапазона целых чисел от 50 до 100.
3) При добавлении элемента перемещался не pbeg указатель, а pend.

Задание 4.

Составить программу, которая при вводе чисел формирует два списка, помещая в первый из них положительные, а во второй — отрицательные числа.

3. Просмотр элементов списка

Просмотр элементов списка осуществляется последовательно, начиная с его конца. Указатель **Tek** последовательно ссылается на последний, предпоследний и т. д. элементы списка до тех пор, пока весь список не будет пройден (т.е. ссылочное поле не будет **nil**). При этом с каждым элементом списка выполняется некоторая операция, например, печать элемента.

Задание 5.

Запустите приложение **mod_dync.exe**, выберите пункт «Вывод данных из списка», разберитесь в приведенном фрагменте программы, выполнив пошагово предложенный алгоритм.

Задание 6.

Считайте с диска файл **beg_show,** проанализируйте приведенную там программу и ответьте на следующие вопросы:

- С какого элемента начинается просмотр списка?
- Какой указатель перемещается от элемента к элементу? Как выглядит эта строка программы?
- До каких пор происходит просмотр элементов списка?
- Что происходит с данными во время их просмотра? Где это осуществляется в программе?

Задание 7.

Модифицируйте текст программы файла **beg_show** так, чтобы:

1) Значение информационного поля увеличивалось в 2 раза;
2) Выводились значения элементов, отстоящих друг от друга на 2 шага, начиная с конца;
3) Выводилось значение элемента с указанным номером (отсчет производится с конца).

Задание 8.

Составить программу, которая выводит предыдущий и последующий элементы элемента, указанного пользователем.

4. Контрольное задание[1]

1) Сформировать список строк из текстового файла.
2) Написать функцию, которая вычисляет среднее арифметическое элементов непустого списка.
3) Написать процедуру присоединения списка L2 к списку L1.
4) Написать функцию, которая создает список L2, являющийся копией списка L1, начинающегося с данного узла.
5) Многочлен $P(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$ задан своими коэффициентами, которые хранятся в форме списка. Написать функцию:

Equal(p, q), проверяющую на равенство многочлены p и q.

6) Многочлен $P(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$ задан своими коэффициентами, которые хранятся в форме списка. Написать функцию:

Summa(p, q, r), которая строит многочлен r = p + q.

7) *Вычислить значение многочлена $P(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$ в целочисленной точке x. Коэффициенты вводятся с клавиатуры и динамически размещаются в памяти.
8) Сформировать список вещественных чисел и вычислить сумму.
9) Написать процедуры проверки наличия в списке заданного числа.
10) Написать функцию, которая проверяет, упорядочены ли элементы списка по алфавиту.
11) Определить симметричность произвольного текста любой длины. Текст должен оканчиваться точкой. Задачу решить с помощью двух списков.
12) Вычислить значение выражения $x_1 x_n + x_2 x_{n-1} + \ldots + x_n x_1$. Значения $x_1, x_2, \ldots, x_n$ вводятся с клавиатуры и динамически размещаются в памяти.
13) Написать функцию, которая использует исходный список L и создает два новых списка L1 и L2. L1 содержит нечетные элементы, а L2 – четные.
14) Написать функцию, которая использует исходный список L и создает два новых списка L1 и L2. L1 содержит нечетные числа, а L2 – четные.

5. Вопросы для самоконтроля

1. Что понимают под "связанным списком"?
2. Как классифицируют связанные списки?
3. Сколько элементов может содержать список? Как заканчивается список?
4. Сколько полей может содержать элемент списка? От чего зависит количество полей? Приведите примеры.
5. Какого типа могут быть поля элементов списка? Приведите примеры.

[1] Номер задания указывает преподаватель

6. Можно ли ссылку одного элемента направить сразу на два или больше других элемента? Как необходимо поменять тип указателя, чтобы решить эту проблему?
7. Может ли элемент списка быть такого типа, чтобы содержать несколько полей типа указателя? Если - да, то приведите пример, для чего это может быть нужно.
8. Какие операции требуется выполнить для вставки и удаления элемента списка?
9. Можно ли для построения списка обойтись одной переменной?
10. Можно ли последовательно "связать" два списка разного типа и почему?
11. Можно ли одновременно работать с несколькими списками сразу?
12. Как Вы считаете, на что нужно обращать особое внимание при работе со списками?

Лабораторная работа №3
Динамические структуры данных. Добавление элемента списка в конец. Вставка элемента

1. Добавление элемента в конец списка

Чтобы добавить новый элемент в конец списка (см. **Ошибка! Источник ссылки не найден.**), необходимо:

1. Получить память для него;
2. Поместить туда информацию;
3. Установить связь с последним элементом списка.

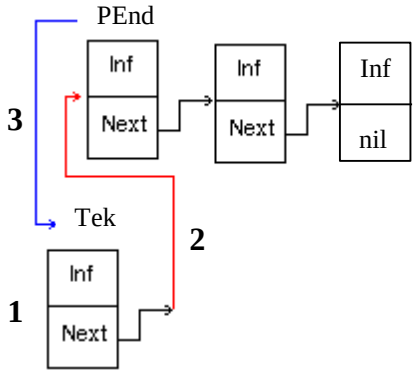


Рис. 1. Добавление элемента в конец списка

Задание 1.

Запустите приложение **mod_dyn.exe**, выберите пункт «Дополнение списка», разберитесь в приведенных фрагментах программы, выполнив пошагово предложенные алгоритмы.

Задание 2.

Считайте с диска файл **add_end,** проанализируйте приведенную там программу и ответьте на следующие вопросы:

- Какого типа являются указатели на элементы линейного списка?
- Сколько информационных полей содержит элемент списка? Какого они типа?
- Какое количество ссылочных полей содержит элемент списка?
- Какие указатели на элементы списка участвуют в работе с ним?
- Какой указатель ссылается на первый элемент, последний элемент, а какой на текущий?
- Где в тексте программы происходит заполнение информационных полей?
- Где в тексте программы устанавливается связь с вновь созданным элементом списка?
- Где в тексте программы происходит перемещение указателя начала на вновь созданный элемент?
- До каких пор элементы будут добавляться в список?

Задание 3.

Считайте с диска файл **Listadd1** и выполните перечисленные там задания.

Задание 4.

Модифицируйте текст программы файла **add_end** так, чтобы:

1) после формирования списка, к нему добавлялся элемент в конец, в информационном поле которого хранилось количество элементов списка;
2) чтобы существовало меню для выбора способа формирования списка и после выбора соответствующего пункта, список формировался соответствующим способом. Возможности добавления элемента реализуйте с помощью процедур и функций;
3) элементы списка сохранялись в файл **list.dat;**
4) существовала возможность просмотреть элементы созданного списка;
5) возможности дополнения списка были реализованы с помощью рекурсивных процедур и функций.

2. Поиск элемента в списке

Чтобы найти элемент списка необходимо:

1. Просматривать каждый элемент списка;
2. Сравнить значение определенного поля с заданным условием списка.

Задание 5.

Запустите приложение **mod_dync.exe**, выберите пункт «Поиск данных в списке», разберитесь в приведенном фрагменте программы, выполнив пошагово предложенный алгоритм.

Задание 6.

Модифицируйте текст программы файла **ListFind** так, чтобы:

1) Выводились на экран элементы, значение которого больше заданного с клавиатуры;
2) Формировался и выводился новый список из элементов, значение которых больше заданного числа.

3. Вставка элемента между двумя существующими элементами списка

Чтобы вставить новый элемент перед некоторым элементом списка (см. **Ошибка! Источник ссылки не найден.**), необходимо:

1. Найти элемент, перед которым необходимо вставить;
2. Выделить память для нового элемента Vsp;
3. Поместить туда информацию;
4. Установить связь с элементом списка, с которым связан элемент Pred;
5. установить связь элемента Pred с элементом Vsp.

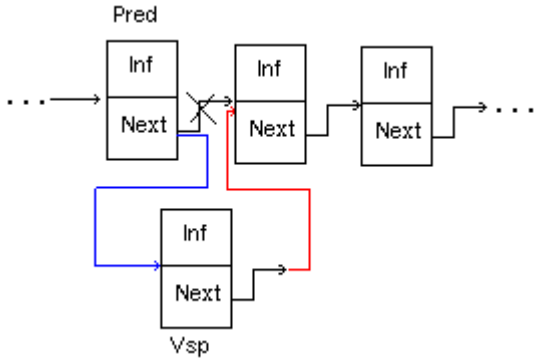


Рис. 2. Вставка элемента перед элементом

4. Контрольное задание

Написать программу присоединения списка S2 к списку S1.

1) *Слить два списка, содержащих возрастающую последовательность целых положительных чисел, в третий список так, чтобы его элементы располагались также в порядке возрастания.
2) Написать функцию, которая подсчитывает количество вхождений ключа в списке.
3) Написать рекурсивную процедуру проверки наличия в списке заданного числа.
4) Написать программу, которая проверяет, упорядочены ли элементы списка по алфавиту.
5) Написать программу, подсчитывающую количество слов в списке, которые начинаются с той же буквы, что и следующее слово.
6) Написать функцию, которая использует исходный список L и создает два новых списка L1 и L2. L1 содержит нечетные узлы, а L2 – четные (добавление в конец).
7) 19. Написать функцию, которая использует исходный список L и создает два новых списка L1 и L2. L1 содержит нечетные числа, а L2 – четные (добавление в конец).
8) Дано число N (> 0) и указатели P1 и P2 на начало и конец непустого списка. Извлечь из списка N начальных элементов и вывести их значения (если список содержит менее N элементов, то извлечь все его элементы).
9) Написать функцию, которая вычисляет среднее арифметическое элементов непустого списка.

5. Вопросы для контроля

1. Как добавить элемент в конец списка?
2. Как добавить элемент в начало списка?
3. Что нужно знать, чтобы вставить элемент после известного элемента?
4. Как осуществляется поиск элемента в списке?
5. Как будет выглядеть строка из элементов списка, если он формировался путем добавления элементов в конец списка, в начало? Исходный список элементов выглядит следующим образом: 13, 2, 7, 35, 28, 9.

Лабораторная работа№4
Динамические структуры данных. Стек.

1. Общие сведения

Стек – упорядоченный набор элементов, в котором добавление новых элементов и удаление существующих производится с одного конца, называемого вершиной стека.

В любой момент времени доступен лишь один элемент стека – верхний. Извлекать элементы из стека можно только в порядке, обратном их добавления в стек (первым пришел, последним ушел).

Значением указателя, представляющего стек, является ссылка на вершину стека, каждый элемент стека содержит поле ссылки на следующий элемент.

Таким образом, описать стек, содержащий, например, целые числа, можно так как показано в файле **stack**. Если стек пуст, то значение переменной Stack равно NULL.

2. Занесение в стек.

Занесение в стек производится аналогично вставке нового элемента в начало списка (файл **ADD_NACH**).

3. Извлечение элемента из стека.

В результате выполнения этой операции некоторой переменной N должно быть присвоено значение первого элемента стека и изменено значение указателя на начало стека (файл **stack**).

4. Контрольные вопросы

1. В чем сходство и отличие динамических структур данных типа список и стек?
2. Можно ли добраться до середины или конца («дна») стека, минуя его начало («вершину»)?
3. Приведите примеры из жизни, где встречается «принцип стека».

Задача 1

Оформите в виде процедуры Push занесение в стек значения.

Задача 2

Оформите в виде процедуры Pop извлечение значения из стека.

Задача 3

Подсчитать количество элементов в стеке.

Задача 4

Сформировать стек, содержащий строки и сохранить его в текстовом файле.

Задача 5

Восстановить стек, содержащий строки, из текстового файла.

5. Контрольные задания

1. Написать функцию, которая вычисляет среднее арифметическое элементов стека.
2. Написать процедуру присоединения стека S2 к стеку S1.
3. Определить симметричность произвольного текста любой длины. Текст должен оканчиваться точкой. Задачу решить с помощью двух стеков.
4. Слить два стека, содержащих возрастающую последовательность целых положительных чисел, в третий стек так, чтобы его элементы располагались также в порядке возрастания.
5. В данном тексте проверить соответствие открытия и закрытия скобок.
6. Напечатать содержимое текстового файла, выписывая символы каждой его строки в обратном порядке.
7. Проверить, является ли строка палиндромом.

Лабораторная работа№5
Динамические структуры данных. Очередь.

Общие сведения

📖 Очередь – упорядоченный набор элементов, в котором добавление новых элементов производится с одного конца, называемого концом очереди, а удаление существующих производится с другого конца, называемого началом.

В любой момент времени доступны два элемента очереди – верхний и нижний. Извлекать элементы из очереди можно только в порядке их добавления в очередь (первым пришел, первым ушел).

Задача

На вход системы массового обслуживания (**СМО**) поступает поток заявок, которые обрабатываются обслуживающими аппаратами за некоторое время. При поступлении заявки размещаются в очередь, из которой они поступают на обработку. Интервал времени между заявками задается функцией, время обслуживания каждой заявки тоже определяется функцией. Обращение к пустой очереди заявок для выборки на обработку считается отказом выборки и приводит к простою прибора. Работа прибора производится в интервале времени [t0..t1].

Варианты

1. а) количество обслуживающих аппаратов - один;
   б) количество обслуживающих аппаратов - два
2. а) очередь на каждый аппарат одна, без приоритетов;
   б) очереди на каждый аппарат две, для приоритетных и обычных заявок;
   в) общая очередь на два аппарата.
3. а) порядок обслуживания - естественный (FIFO);
   б) порядок обслуживания - стек (LIFO)
4. а) длина очереди не ограничена;
   б) длина очереди ограничена - попытка установки в длинную очередь приводит к отказу в обслуживании.

Определить:

1. Среднюю длину очереди.
2. Максимальную длину очереди.
3. Среднее время пребывания заявки в очереди.
4. Максимальное время пребывания заявки в очереди.
5. Среднее время простоя прибора.
6. Максимальное время простоя прибора.
7. Минимальное время простоя прибора.
8. Количество отказов в обслуживании.
9. Количество отказов выборки.

По каждой задаче варианты нумеруются следующим образом:

1 - аааа	5 - абаа	9 - баaа	13- ббаа	17 - бваа
2 - аааб	6 - абаб	10 - бааб	14 - ббаб	18 - бваб
3 - ааба	7 - абба	11 - баба	15 - ббба	19 - бвба
4 - аабб	8 - аббб	12 - бабб	16 - бббб	20 - бвбб