

## Лабораторная работа N21

Тема: Использование указателей для организации связанных списков

Цель работы: изучение динамических структур данных на примере односвязных списков, процедур для выделения и освобождения памяти.

### Теоретические сведения

Рассмотренные ранее структуры данных являлись статическими. Это означает, что память под эти данные выделялась на этапе компиляции. Такое выделение памяти не всегда удобно, т.к. заранее трудно предвидеть, например, размер массива для сортировки или число уравнений в системе. Поэтому в Турбо Паскале существует возможность выделения памяти не на этапе компиляции, а на этапе исполнения программы - динамическая память.

Процедура выделения памяти связана с процедурой ее освобождения после использования. В Турбо Паскале имеется три пары процедур выделения-освобождения памяти: New - Dispose, GetMem - FreeMem, Mark - Release. Чаще всего используется пара New для выделения памяти и Dispose для ее освобождения. Подчеркнем, что использование Dispose для освобождения выделенной динамической памяти не является обязательным, после выхода из программы динамическая память освобождается автоматически, но частое использование процедуры выделения памяти без ее освобождения может привести к состоянию невозможности дальнейшего выделения памяти и к остановке программы.

Рассмотрим работу с динамической памятью на примере односвязного списка. Такой список содержит записи с информационной частью (например, фамилия, имя, отчество, адрес, номер телефона и т.д.) и ссылочной частью (например, адрес следующей или предыдущей записи). Такой список принципиально отличается от массива тем, что в массиве записи упорядочены, удаление одной записи приводит к необходимости переписывать остальные записи на место предыдущих, в то время как в связном списке следует лишь переставить указатель. Число записей в массиве фиксировано, в связном списке - нет.

```
program spisok;
  type point = ^rec;
  rec = record
    fio : string[15]; { ФИО пациента }
    davl : integer; { давление }
    v : point; { указатель на следующую запись }
  end;
  var tek, pred, perv, rab : point; { указатели на текущую, предыдущую, первую, рабочую
записи }
  k : integer;
begin
  new (tek); { выделение памяти }
  write ('введите фамилию пациента: '); readln (tek^.fio);
  write ('введите давление: '); readln (tek^.davl);
  tek^.v := nil; { следующей записи пока нет }
  pred:=tek;
  perv:=tek;
  for k:=2 to 5 do begin
    new (tek); { выделение памяти }
    pred^.v := tek;
```

```

write ('введите фамилию пациента: ');
readln (tek^.fio);
write ('введите давление: ');
readln (tek^.davl);
tek^.v:=nil; {следующей записи пока нет}
pred:=tek;
end;
{удаление записей пациентов с давлением большим 140}
tek:=perv; {указатель на начало списка}
pred:=nil; {предыдущей записи пока нет}
while tek<>nil do begin
    if tek^.davl>140 then begin
        rab:=tek;
        if tek<>perv then pred^.v:=tek^.v else perv:=tek^.v;
        if tek^.v=nil then {запись последняя}
        if tek<>perv then pred^.v:=nil {список не пуст}
        else perv:=nil; {список пуст}
        dispose (rab);
        end
    else pred:=tek;{переставляем указатель предыдущей записи, если текущую
запись не удаляли}
    tek:=tek^.v; {переход к следующей записи}
    end;
{печать оставшихся записей}
writeln ('Оставшиеся записи в односвязном списке');
if perv<>nil then begin {список не пуст}
tek:=perv;
while tek<>nil do begin
    writeln('ФИО ',tek^.fio,' давление ', tek^.davl);
    tek:=tek^.v {переход к след. записи}
end end end.

```

В данной программе `rec` - запись, содержащая сведения о пациенте: `fio` - фамилия, имя, отчество (15 символов), `davl` - давление (целое число), `v` - указатель на следующую запись. Тип указатель занимает в памяти 4 байта (2 байта сегментный регистр, 2 байта смещение), независимо от размера переменной, на которую он указывает. При описании типа указателя `point : ^rec`; допускается употреблять неописанный тип `rec` (тип `rec` описан позже). Перед типом `rec` ставится символ `^` (карат), что означает: `point` является указателем на тип `rec`. Когда речь идет о конкретной переменной, на которую указывает указатель, знак карата переносится в конец имени указателя, например, `tek^.davl` означает поле "давление" записи, на которую указывает указатель `tek`.

Для выделения памяти используется процедура `new(<указатель>)`. В данном случае `new(tek)` означает выделение памяти для записи, на которую указывает `tek`. Таким образом, `tek` - это адрес в оперативной памяти, с которой начинается вновь образованная переменная `tek^`, имеющая тип записи `rec`. Конечно, заранее знать этот адрес невозможно, поэтому выделение динамической памяти производится на этапе исполнения (RunTime). Тип `point` (указатель на `rec`) имеют переменные `tek`, `pred`, `perv`, `rab` для текущей, предыдущей, первой, рабочей записей. Односвязный список всегда проходится в одном направлении, в данном случае от начала к концу, для установки на начало служит указатель на первую запись `perv`. В записи имеется ссылочная (адресная)

часть  $v$  - указатель на следующую запись. Для последней записи списка, это - указатель в никуда, имеющий имя  $nil$ .

В начале программы вводится 5 записей пациентов. Ввод первой записи описан отдельно от остальных для инициализации указателей (в частности, указателя  $pred$  на предыдущую запись, которая ранее не существовала и определяется только после ввода первой записи).

Для каждой записи вначале ссылочному полю  $v$  присваиваем  $nil$ , что означает отсутствие следующей записи. В дальнейшем, при выделении памяти следующей записи процедурой  $new$  в это поле записывается текущий указатель, но не для текущей, а для предыдущей записи. В последней же записи на месте  $v$  остается  $nil$ .

Затем в программе описано удаление записей пациентов с давлением, большим 140. Односвязный список просматривается с начала, для чего текущему указателю  $tek$  присваивается значение  $prev$  указателя на первую запись. Просматривается поле давление  $tek^.dav1$  для текущей записи и если оно больше 140, запись подлежит удалению. При этом действия различны, является ли удаляемая запись первой или нет.

Если удаляемая запись первая, то указатель на первую запись  $prev$  нужно передвинуть на следующую запись  $prev:=tek^.v$ . Если требуется удалить и следующую запись, она вновь будет первой, и вновь указатель первой записи будет изменен, как и следует.

Если же удаляемая запись не первая, то указатель предыдущей записи устанавливаем на адрес, на который указывает ссылка в текущей записи, таким образом, текущая запись будет пропущена в списке.

Если удаляемая запись последняя ( $tek^.v=nil$ ), то следует в ссылочной части предыдущей записи установить признак конца списка  $pred^.v:=nil$ , если эта предыдущая запись существует (список не пуст, т.е. удалены не все записи). Если же удалены все записи, следует установить  $prev:=nil$ .

Далее в программе следует печать оставшихся записей, если список не пуст. Устанавливаем текущий указатель на первую запись и выводим на экран информационные поля  $fio$  и  $dav1$ , затем переходим к следующей записи  $tek:=tek^.v$ .

#### Индивидуальные задания

Задание: 1. Изучить предоставленный теоретический материал, набрать предоставленную программу в среде Паскаль. Внимательно разобраться в примере программы и ее работе в различных ситуациях.

2. Сформировать односвязный список из 5 записей, удалить записи, удовлетворяющие некоторому условию, и вывести список на экран.

Вар	Задание
1.	Запись содержит марку магнитофона, качество, цену. Удалить из списка записи с ценой ниже 800 руб.
2.	Запись содержит фамилию и 4 оценки. Удалить из списка записи с неудовлетворительными оценками
3.	Запись содержит информацию о лекарственных средствах аптечного пункта (наименование, количество, срок годности, цена). Удалить информацию о лекарственных средствах, которых нет в аптеке (количество равно 0).
4.	Запись содержит марку автомобиля, мощность, скорость. Удалить из списка записи со скоростью ниже 110 км/ч.
5.	Запись содержит наименование периодической литературы библиотеки: наименование, тип (газета, журнал), год выпуска., дата выпуска. Удалить информацию о газетах

6.	Запись содержит марку автобуса, скорость, вместимость. Удалить из списка записи с вместимостью ниже 25 пассажиров.
7.	Запись содержит марку компьютера, тактовую частоту, цену. Удалить из списка записи с ценой ниже 1000\$.
8.	Запись содержит количество учащихся в учебном заведении и о типе этого заведения (школа, техникум или училище). Удалить все записи о учебных заведения с количеством учащихся менее 500 человек
9.	Запись содержит фамилию спортсмена, вид спорта, год рождения. Удалить из списка боксеров рождения до 1980 года
10.	Запись содержит марку фотоаппарата, разрешающую способность, цену. Удалить из списка записи с ценой выше 600\$
11.	Запись содержит фамилию писателя, произведение, год издания, число страниц. Удалить из списка записи с изданием до 1995 года и числом страниц более 500
12.	Запись содержит марку принтера, разрешающую способность, цену. Удалить из списка записи с ценой ниже 80\$ .
13.	Запись содержит фамилию ученика (с инициалами), название класса (года обучения и буквы), в котором он учится, оценок, полученных учеником по предмету. Удалить из списка информацию об учениках получивших оценку 2.
14.	Запись содержит фамилию пассажира, количество и общий вес его вещей. Удалите из списка сведения о багаже, общий вес вещей в котором меньше чем 10 кг
15.	Запись содержит фамилию сотрудника фирмы, должность, домашний адрес и отношение к военной службе (военнообязанный или нет). Удалить из списка фамилии всех невоеннообязанных сотрудников.
16.	Запись содержит фамилию ученика (с инициалами), название класса (года обучения и буквы), в котором он учится, и рост ( в см). Удалить из списка информацию об учениках ростом ниже 160 см.
17.	Запись содержит количество учащихся в учебном заведении и о типе этого заведения (школа, техникум или училище). Удалить все записи о школе.
18.	Запись содержит марку фотоаппарата, разрешающую способность, цену. Удалить из списка записи с разрешающей способностью ниже 4 MegaPixels
19.	Запись содержит марку машины, год выпуска, информацию о расходе топлива. Удалить из списка записи с расходом топлива более 10 л
20.	Запись содержит фамилию писателя, произведение, год издания, число страниц. Удалить из списка все записи о заданном авторе
21.	Запись содержит фамилию спортсмена, вид спорта, год рождения и количество лет , которые он занимается спортом . Удалить из списка записи о спортсменах , которые занимаются спортом менее 5 лет
22.	Запись содержит фамилию, адрес и номер телефона. Удалить все записи, номера которых начинаются с цифры 7
23.	Запись содержит названия городов и страны , которых они находятся. Удалить все записи о городах расположенных в Италии
24.	Запись содержит фамилию сотрудника фирмы, должность, домашний адрес и семейное положение(женат/замужем или нет). Удалить из списка фамилии всех холостых сотрудников
25.	Запись содержит информацию об автомобилях предприятия( марка, цвет, пробег, год выпуска, водитель). Удалить из списка записи пробег которых более 100000 км
26.	Запись содержит фамилии детей детского сада, год рождения, домашний адрес и телефон. Удалить из списка фамилии детей старше 6 лет.

27.	Запись содержит информацию о лекарственных средствах аптечного пункта (наименование, количество, срок годности, цена). Удалить информацию о лекарственных средствах срок годности которых истекает в текущем году.
28.	Запись содержит наименование периодической литературы библиотеки: наименование, тип (газета, журнал), год выпуска., дата выпуска. Удалить информацию о газетах и журналах изданных до 2000 года
29.	Запись содержит марку компьютера, тактовую частоту, цену. Удалить из списка записи с тактовой частотой ниже 200 МГц
30.	Запись содержит марку принтера, разрешающую способность, цену. Удалить из списка записи с разрешающей способностью ниже 600 dpi
31.	Запись содержит фамилию служащего, образование( высшее, среднее специальное, среднее), профессию. Удалить все записи о служащих, не имеющих высшего образования

### Контрольные вопросы

1. Статические и динамические переменные.
2. Три пары процедур выделения-освобождения памяти.
3. Процедура new. Процедура dispose.
4. Односвязные списки. Двухсвязные списки.
5. Употребление символа ^ (карат).
6. Основные фрагменты программы для обработки списков.