ПРАКТИЧЕСКАЯ РАБОТА № 14

(4 академических часа)

Тема: Динамические структуры данных. Линейные списки.

Цель задания:

изучение теоретических основ построения и методов программной реализации линейных динамических структур данных – списков.

Теоретическая часть.

Динамической структурой называется множество объектов, состав и взаимное расположение которых в процессе выполнения программы может динамически изменяться.

Операции по модификации динамических структур:

- создание / разрушение структуры,
- включение объектов в структуру / исключение объектов из структуры,
- выделение подмножества объектов структуры по определенным признакам,
- объединение нескольких подмножеств объектов в определенном порядке в единую структуру.

Если на множестве объектов определено отношение порядка, различают линейные и нелинейные структуры данных.

Линейной динамической структурой (связанным списком) называется множество объектов (элементов, узлов) S={si}, i=1,...,n, на котором определены отношения предшествования / следования, причем для любого объекта si, i=2,...,n-1 существует единственный “предшественник” si-1 и единственный “последователь” si+1. Объект s1 имеет последователя, но не имеет предшественника и является первым элементом списка, объект sn имеет предшественника, но не имеет последователя и является “хвостом” списка. Ситуация n=0 определяет особое состояние: “список пуст”.

Реализация динамической структуры линейного списка на связанной памяти требует включения в структуру каждого его элемента полей для связи с соседними элементами. В зависимости от того, с каким количеством соседних объектов связан каждый объект в списке, различаются односвязные, двусвязные и многосвязные списки.

Односвязные линейные списки

Каждый элемент односвязного списка содержит одно или несколько информационных полей и единственное поле связи. Следовательно, одно из полей элемента хранения односвязного списка должно иметь тот же тип, что и сам элемент. Элементом связанного списка не может быть структура, т.к. структура непосредственно содержит свои данные. Например, невозможно объявить следующую структуру:

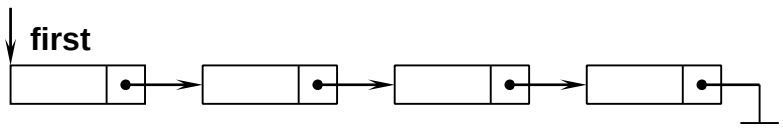
```
Struct MyType
{    int info;
   MyType t;
}
```

Внутри объекта типа MyType будет содержаться поле t, которое, в свою очередь, будет содержать поле t и т.д. до бесконечности. Чтобы получить возможность использовать объект в объекте того же типа, необходимо использовать ссылки и объекты классов.

Элемент хранения узла односвязного списка описывается следующим образом:

```
public class Node                          // Класс «Узел односвязного списка»
{   private int info;                      // информационное поле узла
    private Node link;                     // поле связи узла

    public int Info {...}                  // свойства
    public Node Link {...}

 public Node () {}   public Node (int info)   // конструкторы
 {
   Info = info;
 }

 public Node (int info, Node link)
 {
   Info = info;  Link = link;
 }
}
```

```
public class SingleLinkedList              // Класс «Односвязные списки»
{
   private Node first;                     // ссылка на первый узел списка

   public SingleLinkedList()               // конструктор: создание пустого списка
  {
     first = null;
   }
}
```



На первый элемент списка указывает ссылка ***first***. Если список пуст, то ***first == null***. Если список не пуст, то к полю любого его элемента (например, первого) можно получить доступ через ссылку.

```
int x = first.Info;                  // значение информационного поля первого
Node p = first.Link;                 элемента // значение поля связи первого элемента –
                                                адрес второго //элемента
```

К полям любого элемента списка, кроме первого, возможен дистанционный доступ.

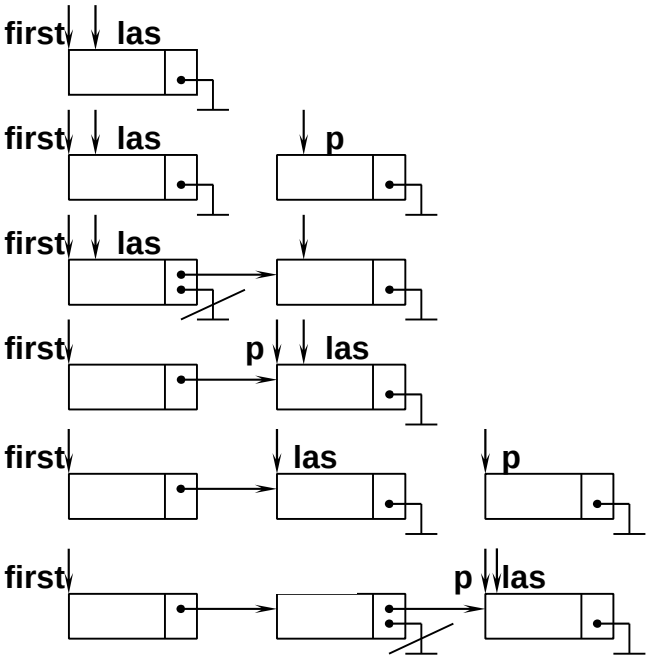
```
int x = first.Link.Info;          // значение информационного поля второго элемента
```

Дистанционный доступ ко второму узлу списка эквивалентен следующей последовательности операторов:

```
Node p = first.Link;  // установка вспомогательной ссылки на второй узел списка
int x = p.Info;   // значение информационного поля второго узла списка
```

1. Создание односвязного списка из N узлов: добавление узлов в конец списка

Первый узел создается отдельно (т.к. включить узел «за» несуществующим узлом невозможно), а остальные (n-1) узлов создаются и включаются в хвост списка одинаковым образом. При этом удобно использовать вспомогательную ссылку на последний добавленный узел. Значение этой ссылки изменяется в процессе создания списка, значение ссылки на первый узел списка не изменяется после создания первого узла. Порядок следования узлов в списке получается прямым, т.к. в начале списка находится тот узел, который был включен в список первым.



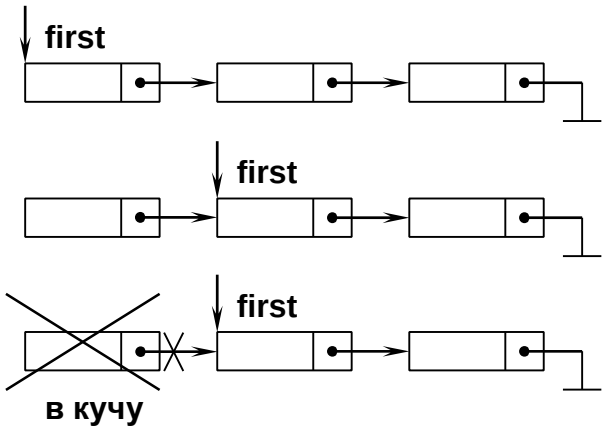
```
public void add_el(int el)     //  добавление элемента в список
{
    Node p = new Node(el);
    if (first == null)              // список пуст
    {
        first = p;                   //  добавление первого элемента в список
        first.Link = null;
        last = first;
    }
    else
    {
        last.Link = p;           // установка связи с новым элементом
        last = p;                // новый элемент становится последним
    }
}
```

2. *Просмотр односвязного списка из N узлов*

```
public void Print()             // просмотр информационных полей узлов списка
{
    Node p = first;
    while (p != null)
    {
        Console.WriteLine(p.Info);
        p = p.Link;
    }
}
```

3. *Исключение узла из начала односвязного списка*

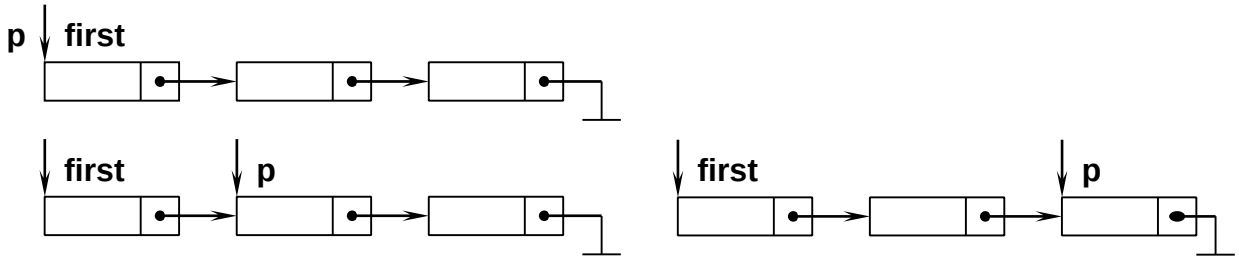
Для того чтобы исключить из списка первый элемент, необходимо переустановить ссылку на следующий элемент списка. Память, занятую бывшим первым элементом, вернет в кучу сборщик мусора, если дальнейшее использование элемента не требуется.



На исключаемый элемент следует предварительно установить ссылку, если требуется дальнейшее использование этого элемента с целью включения его в другую точку данного списка или в другой список.

4. Переустановка ссылки

Доступ к объектам динамической структуры может быть получен с помощью единственной вспомогательной ссылки, которая будет последовательно изменяться, всякий раз принимая значение адреса соседнего объекта, в направлении стрелки, изображающей связь. Адрес соседнего объекта извлекается из поля связи того элемента списка, на который в текущий момент указывает ссылка, затем полученный адрес присваивается этой ссылке, которая теперь открывает доступ к соседнему элементу списка. Такая операция называется переустановкой ссылки. Операция переустановки ссылки используется, если необходимо единообразно обработать все или несколько следующих подряд элементов списка (для этого следует организовать цикл, включающий операции обработки элемента и переустановки ссылки). В этом случае последовательность операций переустановки ссылки обеспечивает проход по списку.



5. Поиск узла в односвязном списке по заданному условию

Условием поиска элемента в списке может быть:

- значение информационного поля элемента,
- порядковый номер элемента в списке, начиная от первого узла,
- адрес элемента списка, который хранится в некоторой переменной ссылке.

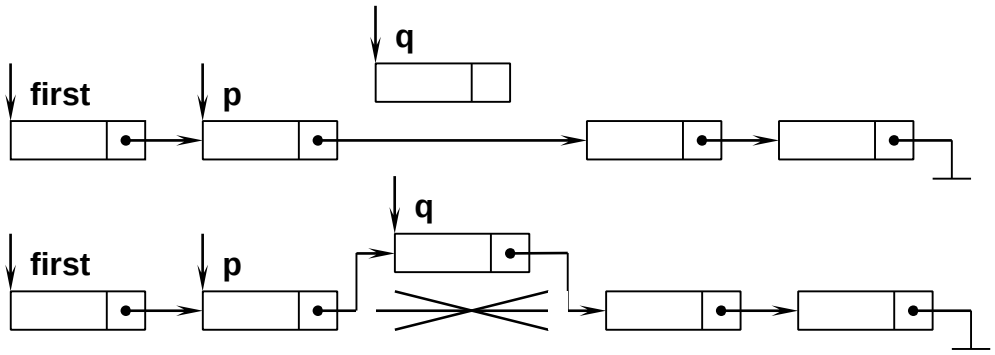
Поиск элемента в списке по заданному условию обычно организуется в цикле, включающем операции проверки выполнения условия для текущего элемента, на который указывает ссылка, и переустановки ссылки на соседний элемент списка (т.е. поиск осуществляется в процессе прохода по списку).

Поиск заканчивается либо при обнаружении элемента списка, соответствующего заданному условию (результатом поиска в этом случае является значение ссылки, установленной на искомый узел), либо при достижении конца списка, если элемент, соответствующий условию поиска, не обнаружен (результатом поиска в этом случае является null).

6. Включение в односвязный список нового узла за тем узлом, на который предварительно установлена ссылка

Для того чтобы включить в список новый элемент за тем элементом, на который предварительно установлена ссылка, необходимо разместить элемент хранения в куче и выполнить последовательность операций 23. После выполнения операции

вставки значение ссылки на первый элемент списка не изменяется. Значение ссылки на тот элемент, за которым выполнена вставка, также не изменяется.



ЗАДАНИЕ

Разработать программу, реализующую работу с линейным списком. В программе необходимо создать базу данных (список) из N записей (N – определяется при работе программы), выполнить просмотр, поиск записи по заданному критерию (вводится при работе программы), вставка записи в любое место списка (до или после записи с заданным критерием), удаление элемента из списка .

* Данные можно считывать из текстового файла (на оценку - 5).

Варианты заданий для самостоятельного решения

1. Автостоянка. Сведения о прибывающих машинах: марка, номер, ФИО владельца, дата прибытия на стоянку, время нахождения на стоянке.
2. Фирма. Анкетные данные сотрудников: ФИО, год поступления в фирму, дата рождения, оклад, адрес.
3. Аптека. Номенклатура товаров: название лекарства, внутреннее/наружное, дата изготовления, срок годности, ФИО кассира (продавца).
4. Фильмы. Список содержит следующую информацию: название фильма, режиссёр, год выпуска, продолжительность в минутах, студия, где снимался фильм.
5. Фотография. Журнал записей содержит информацию: номер заказа, дата приёма заказа, размер фотографий, ФИО фотографа, ФИО заказчика, цена заказа.
6. Гостиница. Содержится следующая информация о проживающих в гостинице: ФИО клиента, номер комнаты, дата въезда, количество дней проживания, стоимость суточного проживания (зависит от категории номера).
7. Продажа видео-аудио кассет. Хранится следующая информация: марка кассеты, фирма-изготовитель, название альбома, время записи (например: 60, 90 минут).
8. Магазин “Мебель”. Номенклатура товаров: наименование изделия, дата изготовления, цена продажи, завод-изготовитель, цвет, название материала из которого изготовлено изделие.
9. Детский сад. Информация о дошкольниках: ФИО ребёнка, дата рождения,

адрес проживания, уровень подготовки (значение 1-5).

10. Ремонт часов. Имеется следующая информация: марка часов, ФИО часовщика, дата приёмки, дата выхода из ремонта, стоимость ремонта.

11. Картинная галерея. Ведётся учёт экспонатов галереи: наименование картины, художник, инвентарный номер, год создания картины, реставратор, цена.

12. Телефонная станция. Имеется информация: ФИО абонента, номер телефона, адрес, тариф со звонка.

Контрольные вопросы

1. Описание структуры. Конструкторы.
2. Обращение к элементам структуры.
3. Чем отличается класс от структуры?
4. Как объявить массив структур?