**Лабораторная работа №2**

**Однонаправленные и двунаправленный списки**

**Пример 1.** Создать однонаправленный список, вывести его содержимое на экран, добавить новые компоненты и удалить первый элемент, стоящий за значением, равным 3.

|  |
| --- |
| uses crt;  type ref=^elem;  elem=record  date:integer;  next:ref  end;  var p,q:ref;  i:integer;  procedure newlist(var p:ref); {*создание списка*}  var q:ref;  i:integer;  begin p:=nil;  writeln('input i or -1');  readln(i);  while i<>-1 do  begin new(q);  q^.date:=i;  q^.next:=p;  p:=q;  readln(i);  end;  end;  procedure print(p:ref); {*вывод содержимого списка на экран*}  var q:ref;  begin q:=p;  while q<>nil do  begin write(q^.date,' ');  q:=q^.next  end;  writeln  end;  procedure insert2(p:ref); {*вставка новых компонент в конец списка*}  var q,l,k:ref;  i:integer;  begin q:=p;  while q<>nil do  begin l:=q;  q:=q^.next  end;  writeln('input i or -1');  readln(i);  while i<>-1 do  begin new(k);  k^.date:=i;  k^.next:=q;  l^.next:=k;  q:=l^.next;  readln(i);  end;  end;  procedure delete2(p:ref); {*удаление компоненты из списка*}  var q:ref;  begin if p^.next<>nil then  begin q:=p^.next;  p^.next:=p^.next^.next;  dispose(q)  end;  end;  begin clrscr;  newlist(p);  print(p);  insert2(p);  print(p);  q:=p;  while (q^.date<>3) and(q^.next<>nil) do q:=q^.next;  delete2(q);  print(p);  end. |

**Пример 2.** Создается двунаправленный список, выводится его содержимое на экран, добавляются новые компоненты и удаляется первый элемент, равный 3.

|  |
| --- |
| uses crt;  type ref=^elem;  elem=record  pred:ref;  next:ref;  date:integer;  end;  var p,ring:ref; d:integer;  procedure insertl(d:integer; p:ref); {*Процедура создания двунаправленного списка}*  var q:ref;  begin new(q);  q^.date:=d;  q^.next:=p^.next;  q^.pred:=p^.next^.pred;  p^.next^.pred:=q;  p^.next:=q  end;  procedure deletel( p:ref); {*Процедура удаления звена из списка*}  begin p^.next^.pred:=p^.pred;  p^.pred^.next:=p^.next;  dispose(p);  end;  begin new(p);  p^.next:=p;  p^.pred:=p;  ring:=p;  writeln(*'Ввод элементов списка'*);  writeln('Введите d or -1');  readln(d);  while d<>-1 do  begin insertl(d,ring^.pred);  readln(d);  end;  Writeln(*'Вывод созданного списка'*);  p:=ring^.next;  while p<> ring do  begin write(p^.date:3);  p:=p^.next;  end;  writeln;  ring:=p;  writeln(*'Добавление новых звеньев в список'*);  writeln('Введите d or -1');  readln(d);  while d<>-1 do  begin insertl(d,ring^.pred);  readln(d);  end;  Writeln(*'Вывод дополненного списка'*);  p:=ring^.next;  while p<> ring do  begin write(p^.date:3);  p:=p^.next;  end;  writeln;  p:=ring^.next;  while (p<> ring)and (p^.date<>3) do  p:=p^.next;  deletel(p);  Writeln(*'Вывод списка после удаления звена'*);  p:=ring^.next;  while p<> ring do  begin write(p^.date:3);  p:=p^.next;  end;  writeln;  end. |

**Вариант 1.**

Создать однонаправленный список, добавить элементы вывести на экран. Найти элементы равные 6 и удалить их. Вывести полученный список на экран.

**Вариант 2.**

Создать однонаправленный список, добавить элементы вывести на экран. Найти элементы равные 3 и удалить их. Вывести полученный список на экран.

**Вариант 3.**

Создать двунаправленный список, добавить элементы вывести на экран. Найти элементы равные 3 и удалить их. Вывести полученный список на экран.

**Вариант 4.**

Создать двунаправленный список, добавить элементы вывести на экран. Найти элементы равные 6 и удалить их. Вывести полученный список на экран.

**Вариант 5.**

Создать двунаправленный список, добавить элементы вывести на экран. Найти элементы равные 10 и удалить их. Вывести полученный список на экран.

**Вариант 6.**

Создать двунаправленный список, добавить элементы вывести на экран. Найти элементы равные 15 и удалить их. Вывести полученный список на экран.

**Вариант 7.**

Создать двунаправленный список, добавить элементы вывести на экран. Найти элементы равные 17 и удалить их. Вывести полученный список на экран.