МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

**«Вятский государственный университет»**

**(ФГБОУ ВО «ВятГУ»)**

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Реализация элементарных структур данных на основе динамической памяти.

Отчет

Лабораторная работа №6 по дисциплине

«Программирование»

Выполнил студент группы ПИб-1301 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Суров К. С./

Проверил преподаватель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Чистяков Г.А/

Киров 2020

**1.Цель работы**

Цель работы: изучение структуры и принципов организации программных модулей, закрепление навыков работы с динамической памятью, получение базовых навыков организации работы в режиме командной строки.

**2.Задание**

1. Написать программу для работы со структурой данных "Односвязный список".

2. Структура данных должна быть реализована на основе динамической памяти.

3. Структура данных (поля и методы) должна быть описана в отдельном модуле.

4. Работа со структурой должна осуществляться в режиме командной строки (с реализацией автодополнения и истории команд). Предусмотреть наглядную визуализацию содержимого структуры.

**3.Программа**

**1)**

unit unitlaba6\_1;

interface

uses crt,SysUtils;

type

st=string;

int=longint;

tlist=^list;

list=record

next:tlist;

data:string;

end;

comm=array[1..20] of integer;

procedure CreateNewElem(var a:tlist;x:st);

Procedure add(var first:tlist;var a:tlist;x:st);

Procedure view(var first:tlist);

Procedure free(var first:tlist);

Procedure select(var first:tlist;var t:st);

procedure item(var first,a:tlist;var x, z:st);

procedure check(var w:string);

procedure infopanel;

procedure input(var n,elem:string);

Procedure newcom(s:integer);

Procedure mov;

var

ch:char;

d,j,k,i,l:int;

s:string;

first,a:tlist;

x,c:st;

m1,co,cl:^integer;

m,k2:tlist;

//n2:int;

up,dn,au:boolean;

com:^comm;

aupn,updn,cm:int;

w,q,n,elem:string;

IMPLEMENTATION

Procedure mov;

Begin

Case com^[cl^] of

1: s:='add';

2: s:='help';

3: s:='view';

4: s:='free';

5: s:='check';

6: s:='item';

end;

end;

procedure input(var n,elem:string);

var x,c:int;

begin

writeln('enter the number of eliminations');

check(n);

x:=1;

c:=strtoint(n);

if c<>0 then

begin

while x<=c do

begin

clrscr;

write('enter element');

delay(200);

clrscr;

check(elem);

add(m,k2,elem);

delline;

gotoxy(1,wherey-1);

delline;

x:=x+1;

end;

end;

clrscr;

end;

Procedure newcom(s:integer);

Begin

If co^<20 then begin

co^:=co^+1;

com^[co^]:=s;

cl^:=co^;

end

else begin

co^:=1;

Repeat

com^[co^]:=com^[co^+1];

co^:=co^+1;

Until co^=20;

com^[co^]:=s;

end;

end;

procedure infopanel;

begin

writeln('list of commands');

writeln('1) add (select the number of items,then enter them)');

writeln('2) view (shows list items)');

writeln('3) free(frees up memory)');

writeln('4) check (show an item in memory)');

writeln('5) item (position\insert an element)');

writeln('6) history (to open the story click PageDown \scroll up click PageUp)');

writeln('7) Exit');

writeln;

writeln('ATTENTION!');

writeln('you can move using the side arrows');

writeln('you can erase written with the BACKSPACE button');

writeln('not forget to press enter to read the command');

readln;

end;

procedure check(var w:string);

var

flag:boolean;

k,x:integer;z:string;

begin

flag:=false;

while flag<>true do

begin

readln(z);

k:=0;

for x:=1 to length(z) do

begin

if(ord(z[x])>47) and (ord(z[x])<58) then

k:=k+1;

end;

if k=length(z) then

flag:=true

else begin

gotoxy(1,2);

write('Error! Enter a number!');

delay(350);

clrscr;

delline;

end;

end;

w:=z;

end;

Procedure CreateNewElem(var a:tlist;x:st);

begin

new(a);

a^.data:=x;

a^.next:=nil;

end;

Procedure add(var first:tlist;var a:tlist;x:st);

var b:tlist;

begin

if first=nil then

begin

CreateNewElem(first,x);

exit;

end;

CreateNewElem(a,x);

b:=first;

while b^.next<> nil do

begin

b:=b^.next;

end;

b^.next:=a;

a^.next:=nil;

end;

Procedure view(var first:tlist);

var b:tlist;

begin

if first=nil then

begin

writeln('the list is empty');

exit;

end;

b:=first;

while b <> nil do

begin

write(b^.data,' ');

b:=b^.next;

end;

readln;

end;

Procedure select(var first:tlist;var t:st);

var b:tlist;flag:boolean;

begin

if first=nil then

begin

writeln('the list is empty');

exit;

end

else

begin

b:=first;

flag:=false;

while (b <> nil) and(flag<>true) do

begin

if t=b^.data then

begin

flag:=true;

gotoxy(1,2);

clrscr;

write('Exist');

readln;

exit;

end else

b:=b^.next;

end;

gotoxy(1,2);

clrscr;

writeln('does not exist');

end;

readln;

end;

Procedure free(var first:tlist);

var b,temp:tlist;

begin

if first<>nil then

begin

b:=first;

while b<>nil do

begin

temp:=b;

b:=b^.next;

dispose(temp);

end;

first:=nil;dispose(first);

writeln('cleaning completed')

end

else writeln('the list is empty');

end;

function quantity(first:tlist):integer;

Var k : integer;

cur :tlist;

begin

if first = Nil then

k:=0

Else

begin

k:=1;

cur:=first;

while cur^.next <> nil do

begin

k:=k+1;

cur:=cur^.next;

end;

end;

quantity:=k;

end;

procedure item(var first,a:tlist;var x, z:st);

var

temp,cur:tlist;

k,n,c:integer;

begin

n:=quantity(first);

c:=strtoint(x);

if (c<1)or(c>n)then

begin

writeln ('invalid x');

readln;

exit;

end

else

begin

CreateNewElem(a,z);

if c=1 then

begin

cur:=first;

first:=a;

a^.next:=cur;

end

else

if c=n then

begin

cur:=first;

while cur^.next <> nil do

begin

cur:=cur^.next;

end;

cur^.next:=a;

a^.next:=nil;

end

else

begin

cur := first;

for k:=2 to c do

begin

cur := cur^.next;

end;

temp := cur^.next;

cur^.next := a;

a^.next := temp;

end;

end;

end;

Initialization

Begin

New(com);

New(co);

co^:=0;

New(cl);

end;

Finalization

Begin

Dispose(com);

Dispose(co);

Dispose(cl);

end;

END.

**2)**

program laba6;

uses crt,unitlaba6\_1,SysUtils;

{$CODEPAGE CP866}

{$mode delphi}

type

int=longint;

Procedure proverka;

Begin

Case s[1] of

#97: If (s[2]=#100) and (s[3]=#100) then d:=1;

#104: If (s[2]=#101) and (s[3]=#108) and (s[4]=#112) then d:=2;

#118: If (s[2]=#105) and (s[3]=#101) and (s[4]=#119) then d:=3;

#102: If (s[2]=#114) and (s[3]=#101) and (s[4]=#101) then d:=4;

#99: If (s[2]=#104) and (s[3]=#101) and (s[4]=#99) and (s[5]=#107) then d:=5;

#105: If (s[2]=#116) and (s[3]=#101) and (s[4]=#109) then d:=6;

end;

j:=0;

i:=0;

k:=wherex;

end;

procedure choice;

var w,q,n,elem:string;

begin

proverka;

Writeln;

Case d of

1: begin n:='';elem:='';input(n,elem)end;

2: infopanel;

3: view(m);

4: free(m);

5: begin w:=''; check(w);select(m,w);end;

6: begin w:=''; check(w); q:='';check(q);item(m,k2,w,q);end;

end;

j:=0;

newcom(d);

CLRSCR;

s:=#0;

d:=0;

Write('Enter command: ');

k:=wherex;

end;

procedure autoinput;

begin

Case i of

1: s:='add';

2: s:='help';

3: s:='view';

4: s:='free';

5: s:='check';

6: s:='item';

end;

gotoxy(16,wherey);

Write(s);

end;

begin

CLRSCR;

Write('Enter command: ');

i:=0;

k:=wherex;

d:=0;

j:=0;

Repeat

begin

Ch:=Readkey;

if Ch<>#0 then

begin

case ch of

#48..#57,#32, #97..#122,#192..#255,#65..#90:

begin

If k<>(j+16) then

begin

j:=j+1;

l:=j;

Repeat

s[j]:=s[j-1];

j:=j-1;

Until j=(k-15);

j:=l;

s[k-15]:=Ch;

k:=k+1;

CLRSCR;

Write('Enter command:');

l:=0;

While l<=j do

begin

Write(s[l]);

l:=l+1;

end;

gotoxy(k,wherey);

end

else

begin

j:=j+1;

s[j]:=Ch;

k:=k+1;

Write(Ch);

end;

Case Ch of

#97: i:=1;

#104: i:=2;

#118: i:=3;

#102: i:=4;

#99: i:=5;

#105: i:=6;

end;

end;

#127,#8:

If j>0 then

begin

if k<>(j+16) then

begin

l:=k;

Repeat

s[k-16]:=s[k-15];

k:=k+1;

Until k=16+j;

k:=l;

end;

k:=k-1;

s[j]:=#0;

j:=j-1;

CLRSCR;

Write('Enter command:');

l:=0;

While l<=j do

begin

Write(s[l]);

l:=l+1;

end;

gotoxy(k,wherey);

end;

#9: autoinput;

#13: choice;

end;

end

else

begin

Ch:=Readkey;

case ch of

#77 :

begin

gotoxy(wherex+1,wherey);

k:=k+1;

end;

#75:

begin

gotoxy(wherex-1,wherey);

k:=k-1;

end;

#81:

begin

If cl^=co^ then cl^:=1

else cl^:=cl^+1;

mov;

CLRSCR;

Write('Enter command:');

gotoxy(16,wherey);

Write(s);

end;

#73:

begin

If cl^=1 then cl^:=co^

else cl^:=cl^-1;

mov;

CLRSCR;

Write('Enter command:');

gotoxy(16,wherey);

Write(s);

end;

end;

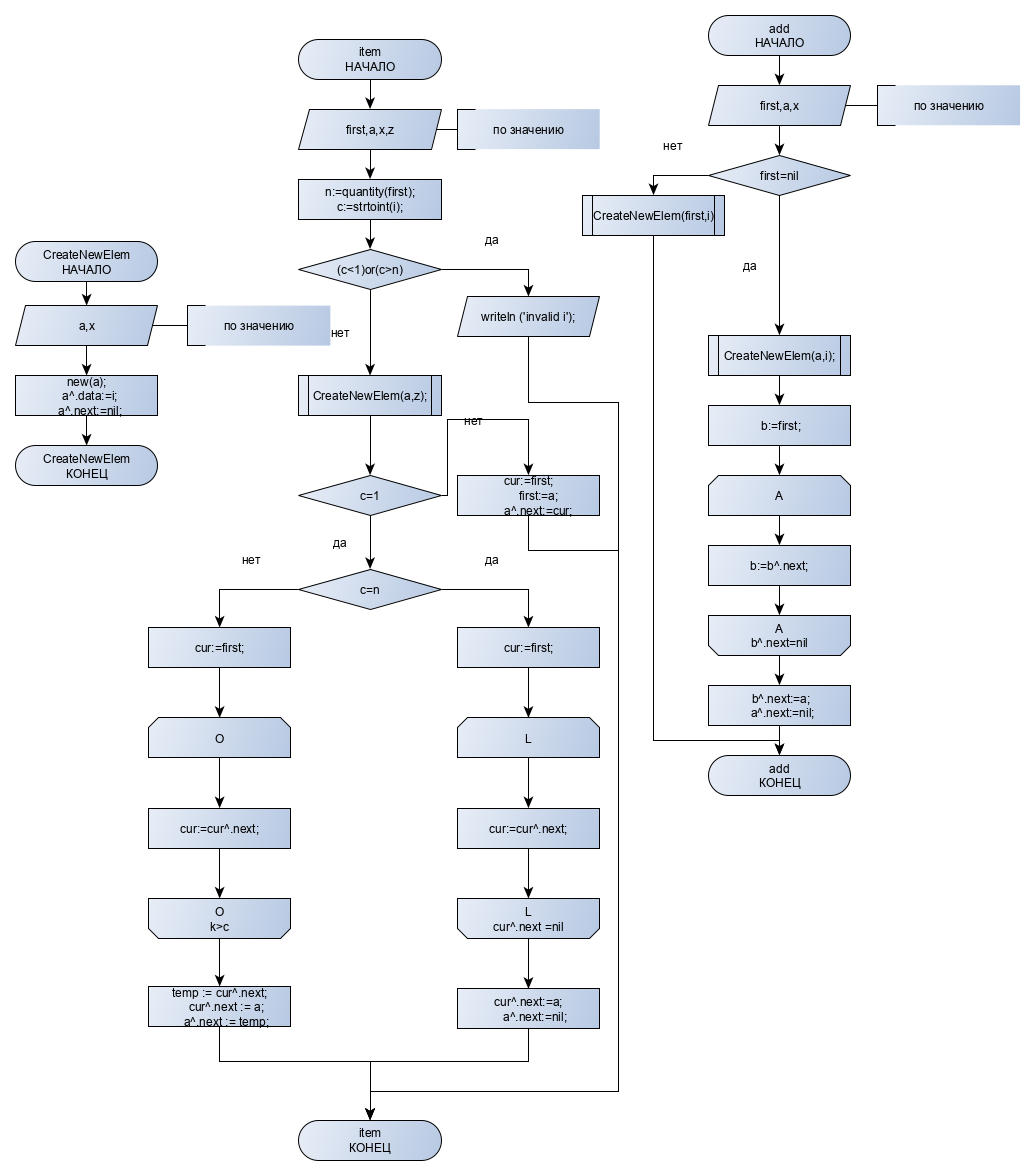
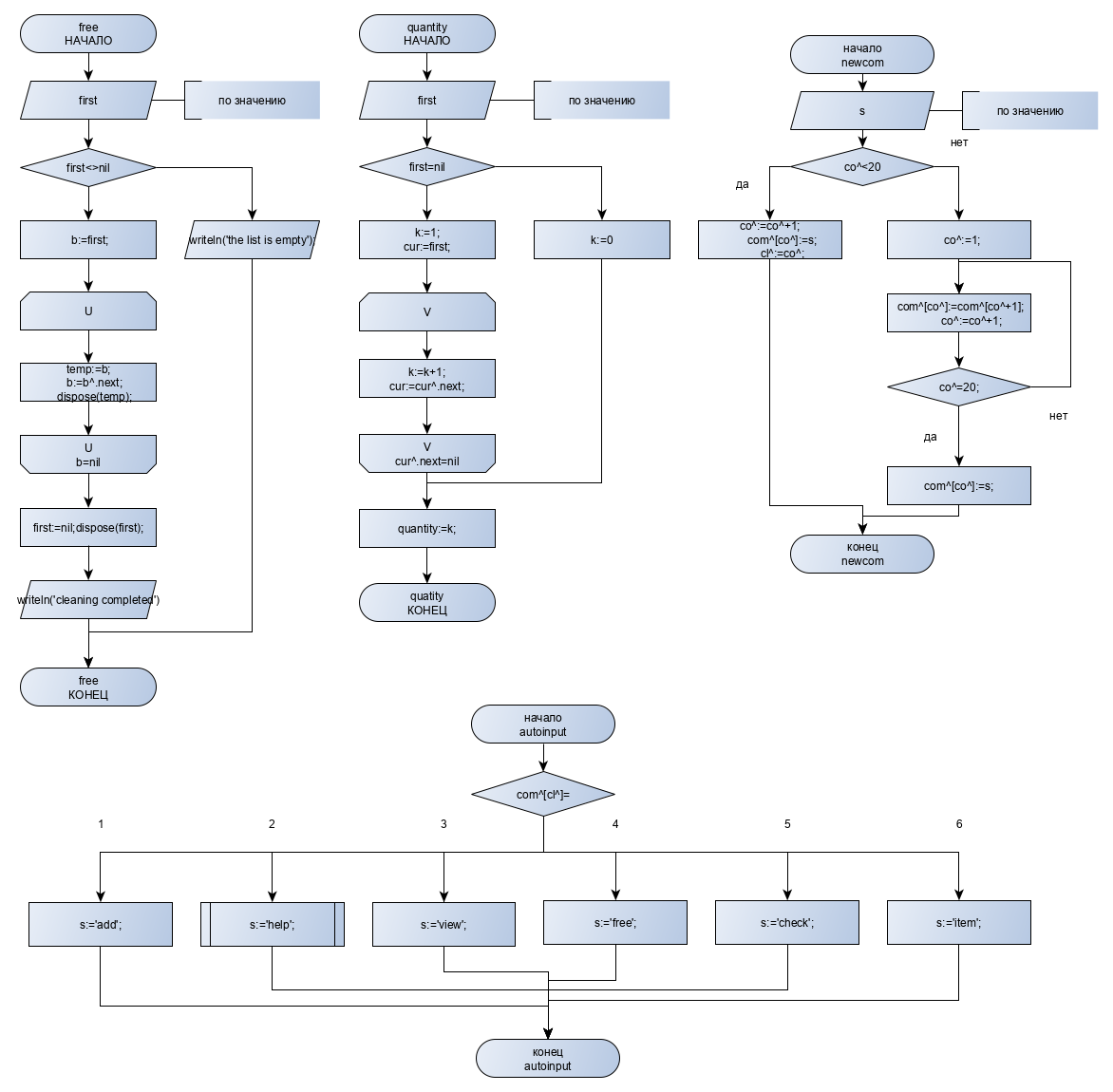
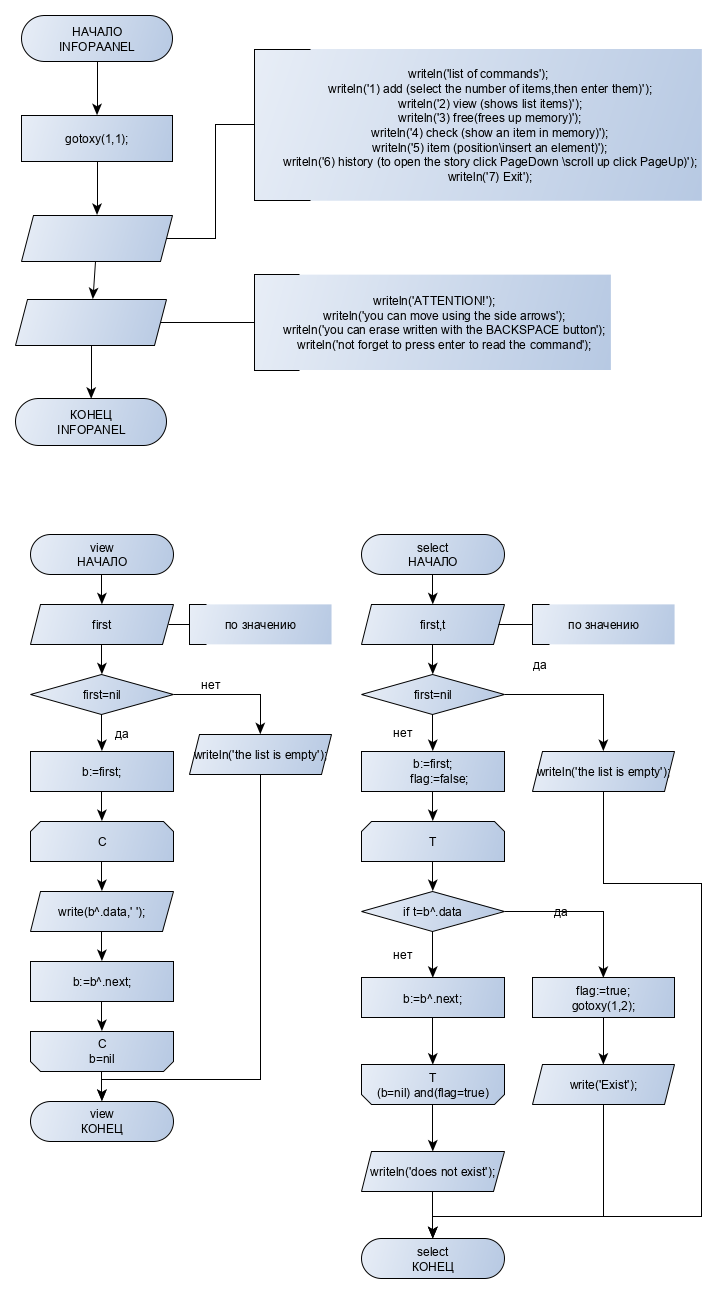
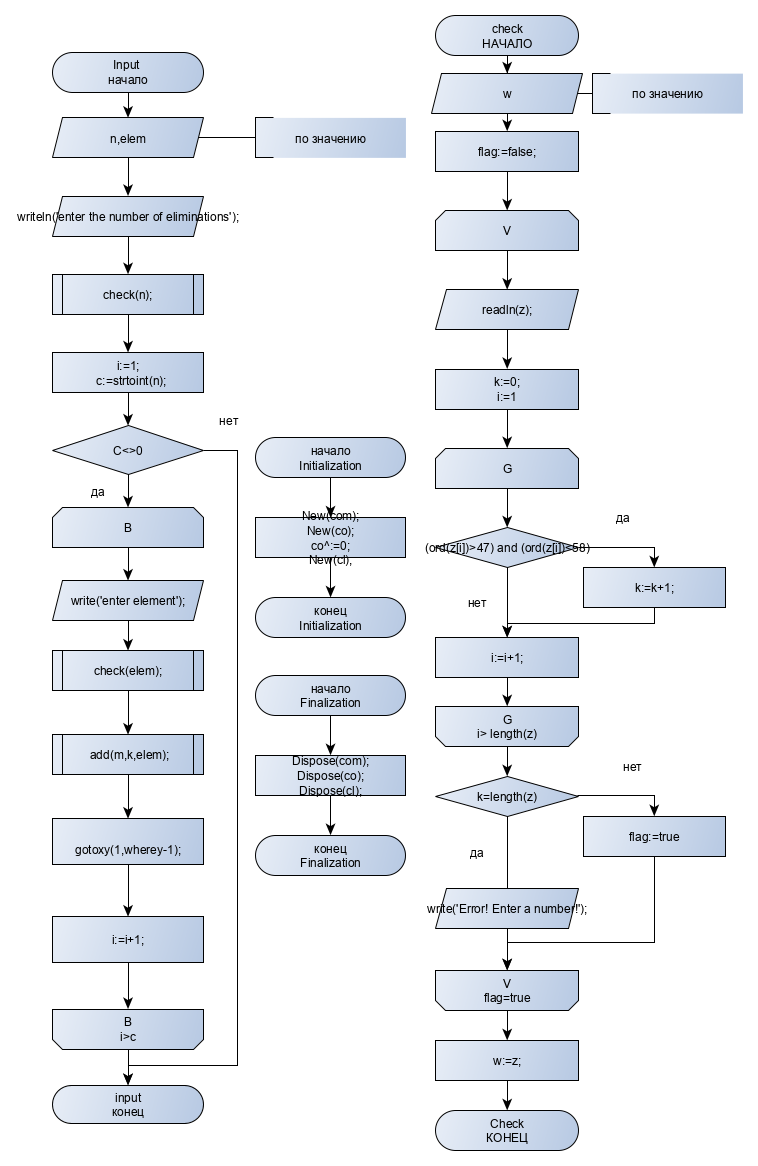
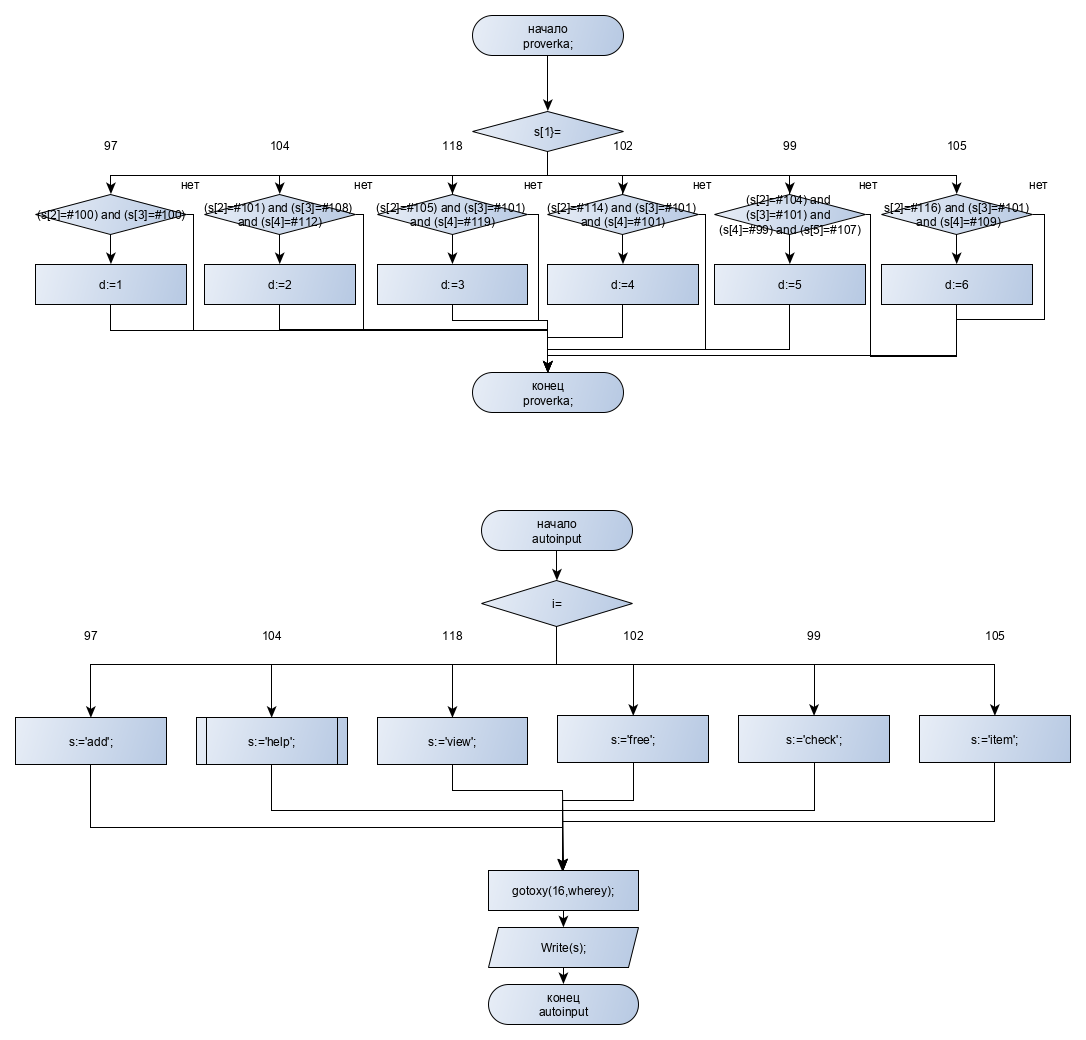
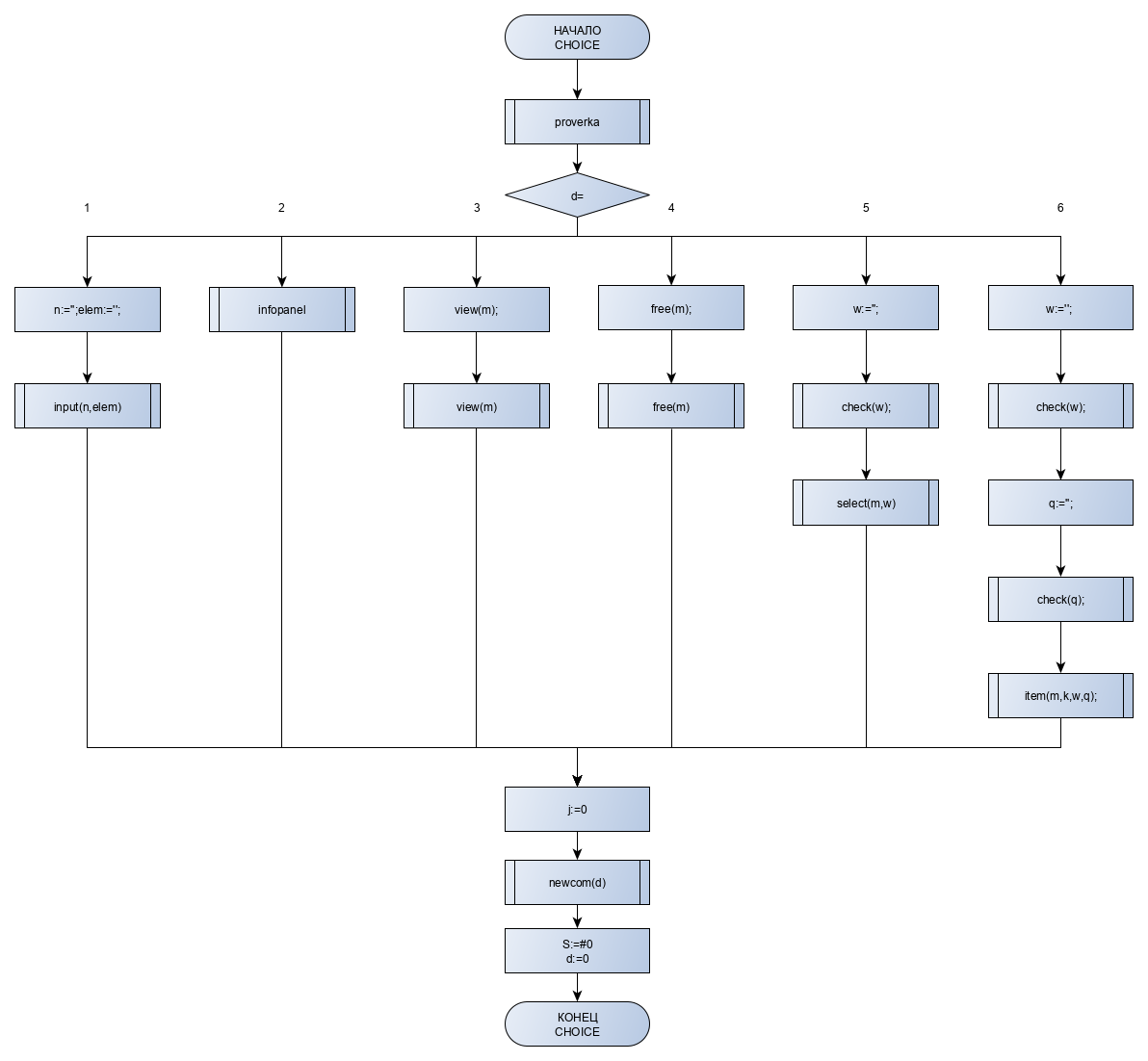
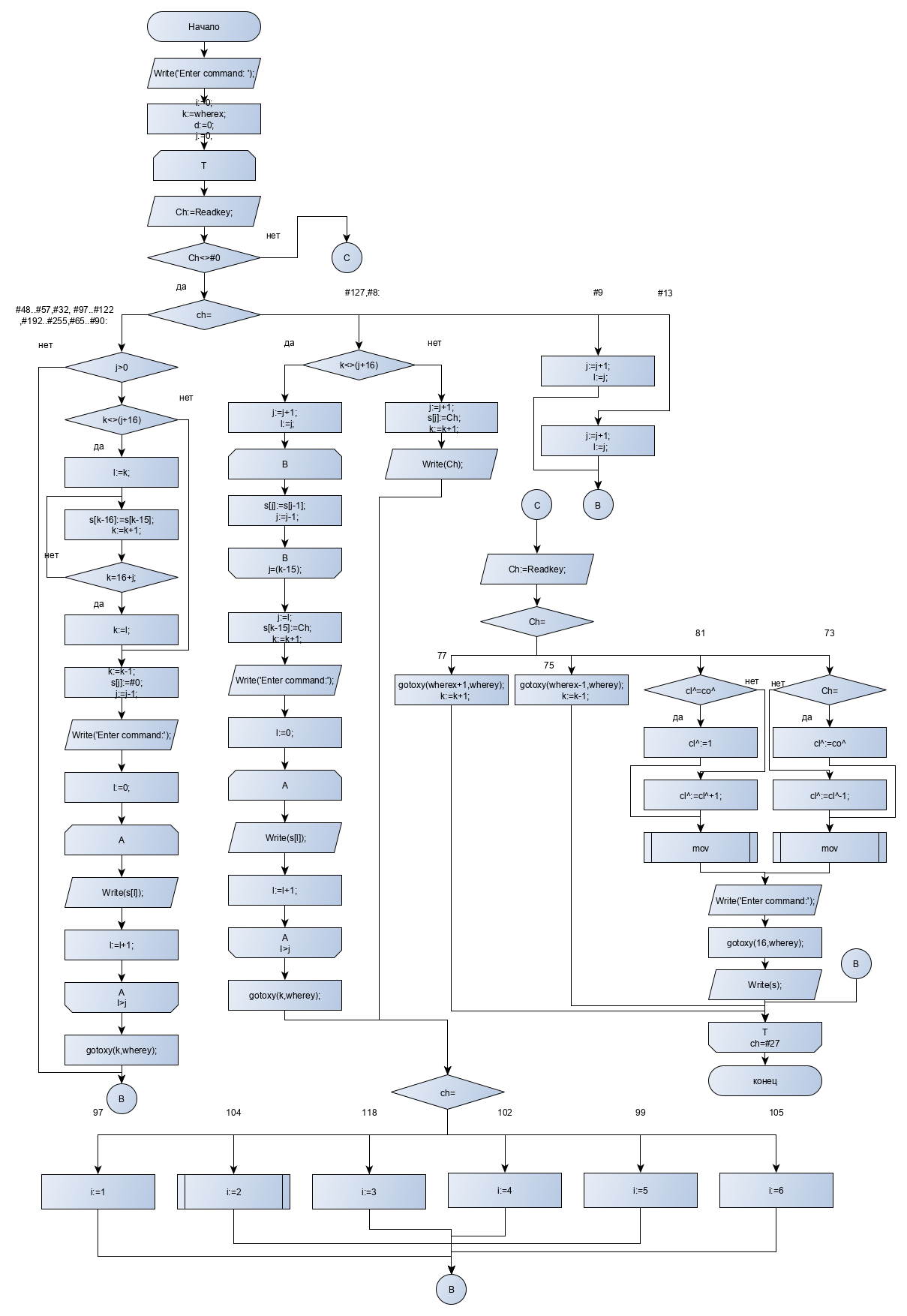
end;

END;

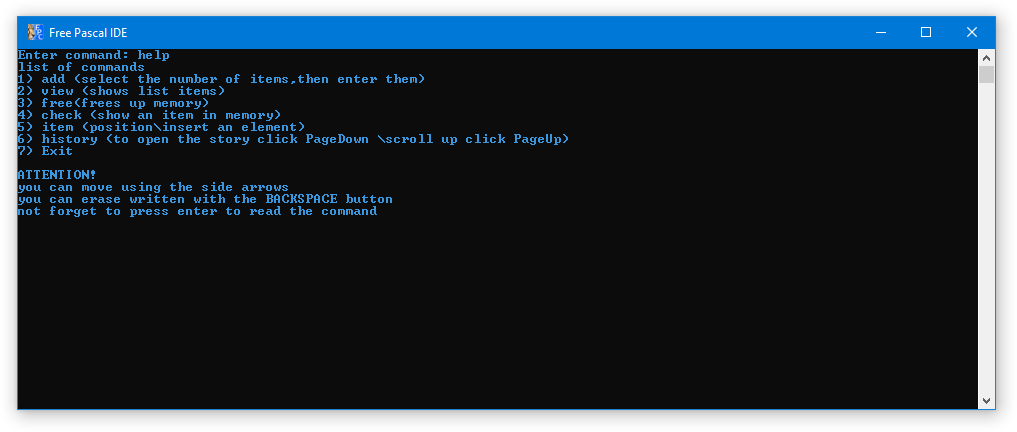
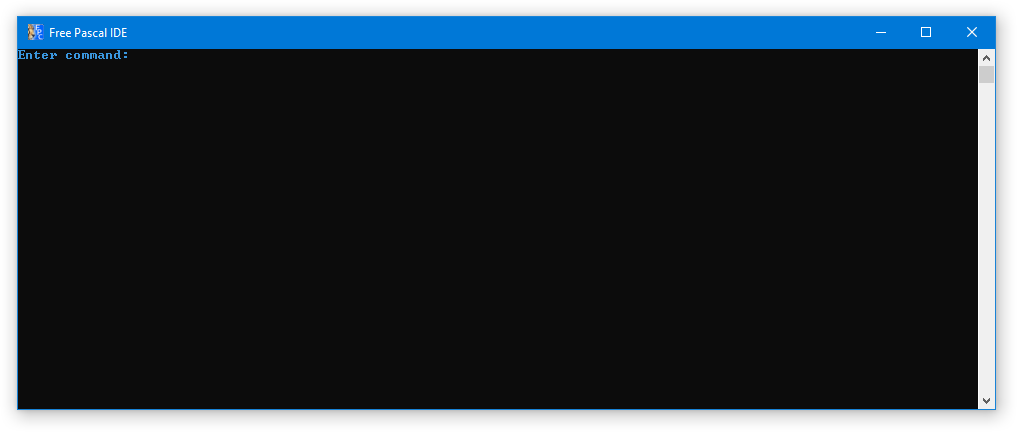
Until Ch=#27

end.

**4.Схема**



**5.Скриншоты программы.**



**6.Вывод.**

В ходе данной лабораторной работы была написана программа, осуществляющая работу в режиме командной строки.

Были изучены структуры и принципы организации программных модулей, закреплены навыки работы с динамической памятью.

Программа осуществляет работу со структурой «Односвязный список». Поля и методы программы описаны в отдельном модуле.

В реализации программы предусмотрено автодополнение команд при нажатии клавиши Tab, а также просмотр истории команд на клавишу PageDown и возможность их прокрутки в истории на клавишу PageUp.

Лабораторная работа выполнена в средах разработки [PascalABC.NET](https://vk.com/away.php?utf=1&to=http%3A%2F%2FPascalABC.NET)  и [Free Pascal.](https://www.freepascal.org/)