МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Вятский государственный университет»
(ФГБОУ ВО «ВятГУ»)
Институт математики и информационных систем
Факультет автоматики и вычислительной техники
Кафедра электронных вычислительных машин

Отчёт
Лабораторная работа № 6 по дисциплине
«Программирование»

«Реализация элементарных структур данных на основе
динамической памяти»

Выполнил студент группы Пиб-1301__________/Соловей Р.О.
Проверил преподаватель________________/Чистяков.Г.А.

Киров 2022

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

Задание:

1. Написать программу для работы со структурой данных “Стек”.
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

Листинг кода и экранная форма: листинг кода и экранная форма приложены ниже.

Схема алгоритма: схема алгоритма приложена в конце отчета.

Вывод: в ходе лабораторной работы была изучена структура и принципы организации программных модулей, закреплены навыки работы с динамической памятью, получены базовые навыки организации работы в режиме командной строки.

Подключаемый модуль

```
unit sollab6modul;

interface

type
  Tptr = ^Telem;
  Telem = record
    inf: string;
    link: Tptr;
  end;

var
  top: Tptr;
  check:integer = 0;

procedure push;
procedure pop;
procedure printFromTop;
procedure delAllStack;
function getCountElem:integer;
procedure help;

implementation

uses crt;

procedure push;
var
  p: Tptr;
  k:char;
  i:integer=0;
  ii:integer;
  st:string;
  flag: boolean;
begin
  flag:=true;
  new(p);
  p^.link := nil;
  writeln('The stack element has a length of 10 characters');
  write('Enter the element to be added: ');
  readln(p^.inf);
  for ii:= 1 to length(p^.inf) do begin if p^.inf[ii] = ' ' then
begin flag:= false; break; end;end;
  if length(p^.inf) < 11 then
  begin
  if (p^.inf <> '') and (flag=true) then
  begin
   p^.link := top;
   top := p;
   inc(check);
  end
  else
  begin
   textcolor(red);
   writeln('Error!');
   textcolor(white);
  end;
  end
  else
  begin
   textcolor(red);
   writeln('Length of stack element more then 10!');
   textcolor(white);
  end;
end;


procedure pop;
var
  p: Tptr;
begin
  p := top;
  top := p^.link;
  p^.link := nil;
  dispose(p);
  dec(check);
end;


procedure printFromTop;
var
  p: Tptr;
begin
  p := top;
  write('Stack elements have the form: ');
  while(p <> nil) do
  begin
    write('{',p^.inf,'}', ' ');
    p := p^.link;
  end;
  writeln();
end;


procedure delAllStack;
var
  p: Tptr;
begin
  p := top;
  while(p <> nil) do
  begin
    top := p^.link;
    p^.link := nil;
    dispose(p);
    p := top;
    check := 0;
  end;
end;


function getCountElem: integer;
var
  p: Tptr;
  k: integer;
begin
  k := 0;
  p := top;
  while(p <> nil) do
  begin
    k := k + 1;
    p := p^.link;
  end;
  getCountElem := k;
  writeln('Number of elements: ',k);
end;


procedure help;
begin
  writeln('Commands for working with the stack: ');
  writeln('<help> - calling this menu');
  writeln('<push> - adding an item to the stack');
  writeln('<pop> - removing an item from the stack');
  writeln('<count> - count of stack elements');
  writeln('<delete> - deleting the entire stack');
  writeln('<print> - stack output');
  writeln('<scrcln> - screen cleaning');
  writeln('<exit> - exit');
end;


begin
end.
```

Основная программа

```
program shell;

uses crt, sollab6modul;


var
  s: string;
  x, y, i, updown: integer;
  bufi : integer = 1;
  buf_k : integer = 1;
  buffer,revbuf : array[1..50] of string;
  key : char;
  flg : byte;
  mn : byte = 15;



function check_char(ch:char):boolean;
var  i:byte;
 begin
  for i := 97 to 122 do
   begin
    if (ord(ch) = i) or (ch = #8) or (ch = #13) or (ch = #72) or
(ch = #80) {or (ch = #32)} then
     begin
      check_char := True;
      break;
     end
    else
     begin
      check_char := False;
     end;
   end;
 end;


procedure auto;
var x,y:integer;
 begin
  if (s = 'h') or (s = 'he') or (s = 'hel')  then
   begin
       y := wherey;
       x := wherex;
       textcolor(blue);
       textbackground(white);
       if s = 'h' then begin  gotoxy(7,y); write('elp'); flg:=1;
end;
       if s= 'he' then begin gotoxy(8,y); write('lp'); flg:=1;
end;
       if s = 'hel' then begin gotoxy(9,y); write('p'); flg:=1;
end;
       textcolor(mn);
       textbackground(black);
       gotoxy(x,y);
   end
   else
    begin
     if (flg <> 2) and (flg <> 3) and (flg <> 4) and (flg <> 5)
```

```
and (flg <> 6) and (flg <> 7) and (flg <> 8) then clreol;
    end;
   if (s = 'e') or (s = 'ex') or (s = 'exi')  then
    begin
        y := wherey;
        x := wherex;
        textcolor(blue);
        textbackground(white);
        if s = 'e' then begin  gotoxy(7,y); write('xit'); flg:=2;
end;
        if s= 'ex' then begin gotoxy(8,y); write('it'); flg:=2;
end;
        if s = 'exi' then begin gotoxy(9,y); write('t'); flg:=2;
end;
        textcolor(mn);
        textbackground(black);
        gotoxy(x,y);
    end
   else
    begin
     if (flg <> 1) and (flg <> 3) and (flg <> 4) and (flg <> 5)
and (flg <> 6) and (flg <> 7) and (flg <> 8) then clreol;
    end;
     if (s = 'c') or (s = 'co') or (s = 'cou') or (s = 'coun')
then
     begin
       y := wherey;
       x := wherex;
       textcolor(blue);
       textbackground(white);
       if s = 'c' then begin  gotoxy(7,y); write('ount'); flg:=3;
end;
       if s= 'co' then begin gotoxy(8,y); write('unt'); flg:=3;
end;
       if s = 'cou' then begin gotoxy(9,y); write('nt'); flg:=3;
end;
       if s = 'coun' then begin gotoxy(10,y); write('t'); flg:=3;
end;
       textcolor(mn);
       textbackground(black);
       gotoxy(x,y);
     end
   else
    begin
    if (flg <> 1) and (flg <> 2) and (flg <> 4) and (flg <> 5) and
(flg <> 6) and (flg <> 7) and (flg <> 8) then clreol;
    end;
    if (s = 'd') or (s = 'de') or (s = 'del') or (s = 'dele') or
(s = 'delet')  then
   begin
       y := wherey;
       x := wherex;
       textcolor(blue);
       textbackground(white);
       if s = 'd' then begin  gotoxy(7,y); write('elete'); flg:=4;
end;
       if s= 'de' then begin gotoxy(8,y); write('lete'); flg:=4;
end;
```

```
       if s = 'del' then begin gotoxy(9,y); write('ete'); flg:=4;
end;
       if s = 'dele' then begin gotoxy(10,y); write('te'); flg:=4;
end;
       if s = 'delet' then begin gotoxy(11,y); write('e'); flg:=4;
end;
       textcolor(mn);
       textbackground(black);
       gotoxy(x,y);
   end
   else
    begin
     if (flg <> 1) and (flg <> 2) and (flg <> 3) and (flg <> 5)
and (flg <> 6) and (flg <> 7) and (flg <> 8) then clreol;
    end;
     if (s = 'po')   then
   begin
       y := wherey;
       x := wherex;
       textcolor(blue);
       textbackground(white);
       if s = 'po' then begin  gotoxy(8,y); write('p'); flg:=5;
end;
       textcolor(mn);
       textbackground(black);
       gotoxy(x,y);
   end
   else
    begin
     if (flg <> 1) and (flg <> 2) and (flg <> 3) and (flg <> 4)
and (flg <> 6) and (flg <> 7) and (flg <> 8) then clreol;
    end;
    if (s = 'pu') or (s = 'pus') then
    begin
       y := wherey;
       x := wherex;
       textcolor(blue);
       textbackground(white);
       if s = 'pu' then begin  gotoxy(8,y); write('sh'); flg:=6;
end;
       if s= 'pus' then begin gotoxy(9,y); write('h'); flg:=6;
end;
       textcolor(mn);
       textbackground(black);
       gotoxy(x,y);
   end
   else
    begin
     if (flg <> 1) and (flg <> 2) and (flg <> 3) and (flg <> 4)
and (flg <> 5) and (flg <> 7) and (flg <> 8) then clreol;
    end;
    if (s = 'pr') or (s = 'pri') or (s = 'prin') then
    begin
       y := wherey;
       x := wherex;
       textcolor(blue);
       textbackground(white);
       if s = 'pr' then begin  gotoxy(8,y); write('int'); flg:=7;
```

```
end;
       if s= 'pri' then begin gotoxy(9,y); write('nt'); flg:=7;
end;
       if s= 'prin' then begin gotoxy(10,y); write('t'); flg:=7;
end;
       textcolor(mn);
       textbackground(black);
       gotoxy(x,y);
   end
   else
    begin
     if (flg <> 1) and (flg <> 2) and (flg <> 3) and (flg <> 4)
and (flg <> 5) and (flg <> 6) and (flg <> 8) then clreol;
    end;
    if (s = 's') or (s = 'sc') or (s = 'scr') or (s = 'scrc') or
(s = 'scrcl') then
    begin
       y := wherey;
       x := wherex;
       textcolor(blue);
       textbackground(white);
       if s = 's' then begin  gotoxy(7,y); write('crcln'); flg:=8;
end;
       if s= 'sc' then begin gotoxy(8,y); write('rcln'); flg:=8;
end;
       if s= 'scr' then begin gotoxy(9,y); write('cln'); flg:=8;
end;
       if s= 'scrc' then begin gotoxy(10,y); write('ln'); flg:=8;
end;
       if s= 'scrcl' then begin gotoxy(11,y); write('n'); flg:=8;
end;
       textcolor(mn);
       textbackground(black);
       gotoxy(x,y);
   end
   else
    begin
     if (flg <> 1) and (flg <> 2) and (flg <> 3) and (flg <> 4)
and (flg <> 5) and (flg <> 6) and (flg <> 7)  then clreol;
    end;
 end;


procedure keys();
var
copy_s : string;
checkend : char;
 begin
  s := '';
  textcolor(mn);
  write('>>>> ');
  repeat
   if length(s) < 24 then
    begin
     key := readkey();
    end
   else
    begin
```

```
     key:=#8;
    end;
    if check_char(key) = True then
     begin
     if length(s) = 25 then
      begin
       checkend := s[24];
       key := #8;
      end;
     if key <> #72  then
     if key <> #80 then
       write(key);
       if wherex > 30 then begin gotoxy(wherex-1,wherey); clreol;
delete(s,length(s),1);  end;
      if (key = #13)  then
       begin
        bufi:=buf_k+1;
        if flg = 1 then begin gotoxy(6,wherey); clreol;
write('help'); s:='help'; end;
        if flg = 2 then begin s := 'exit'; end;
        if flg = 3 then begin gotoxy(6,wherey); clreol;
write('count'); s:='count'; end;
        if flg = 4 then begin gotoxy(6,wherey); clreol;
write('delete'); s:='delete'; end;
        if flg = 5 then begin gotoxy(6,wherey); clreol;
write('pop'); s:='pop'; end;
        if flg = 6 then begin gotoxy(6,wherey); clreol;
write('push'); s:='push'; end;
        if flg = 7 then begin gotoxy(6,wherey); clreol;
write('print'); s:='print'; end;
        if flg = 8 then begin gotoxy(6,wherey); clreol;
write('scrcln'); s:='scrcln'; end;
        flg := 0;
        break;
       end;
      if (key = #80)  then
       begin
        if (bufi<buf_k) then
         begin
          inc(bufi);
          gotoxy(6,wherey);
          clreol;
          gotoxy(6,wherey);
          write(buffer[bufi]);
          s := buffer[bufi];
         end;
        updown:=2;
       end;
      if (key = #72) then
       begin
        if (bufi > 1) then
         begin
          if buffer[bufi-1] = '' then dec(bufi);
          dec(bufi);
          gotoxy(6,wherey);
          clreol;
          gotoxy(6,wherey);
          write(buffer[bufi]);
          s:= buffer[bufi];
         end;
        updown:=1;
       end;
      if key = #8 then begin
                        dec(x);
                        flg:=0;
                        x := wherex;
                        if x > 5 then
                        begin
                         clreol;
                         delete(s,length(s),1);
                        end
                        else
                         begin
                         flg:=0;
                         gotoxy(6,wherey);
                         end;
                       end;

      if (key<>#8) and (key <> #72) and (key <> #80) then
       begin
        s := s + key;
        inc(x);
        flg:=0;
       end;
     auto;
     end;
  until key = #13;
  writeln();
 end;


procedure wrstr();
var i : integer;
 begin
  keys;
  if buf_k < 11 then
   begin
   if (s <> '') and (s <> buffer[buf_k-1]) then
    begin
     buffer[buf_k] := s;
     inc(buf_k);
    end;
   end
   else
   begin
   if s<> '' then
    begin
     buf_k := 1;
     for i:=1 to 11 do begin buffer[i] := ''; end;
     buffer[buf_k] := s;
     bufi := buf_k+1;
     inc(buf_k);
    end;
   end;
  for i:=1 to 2 do
   begin
```

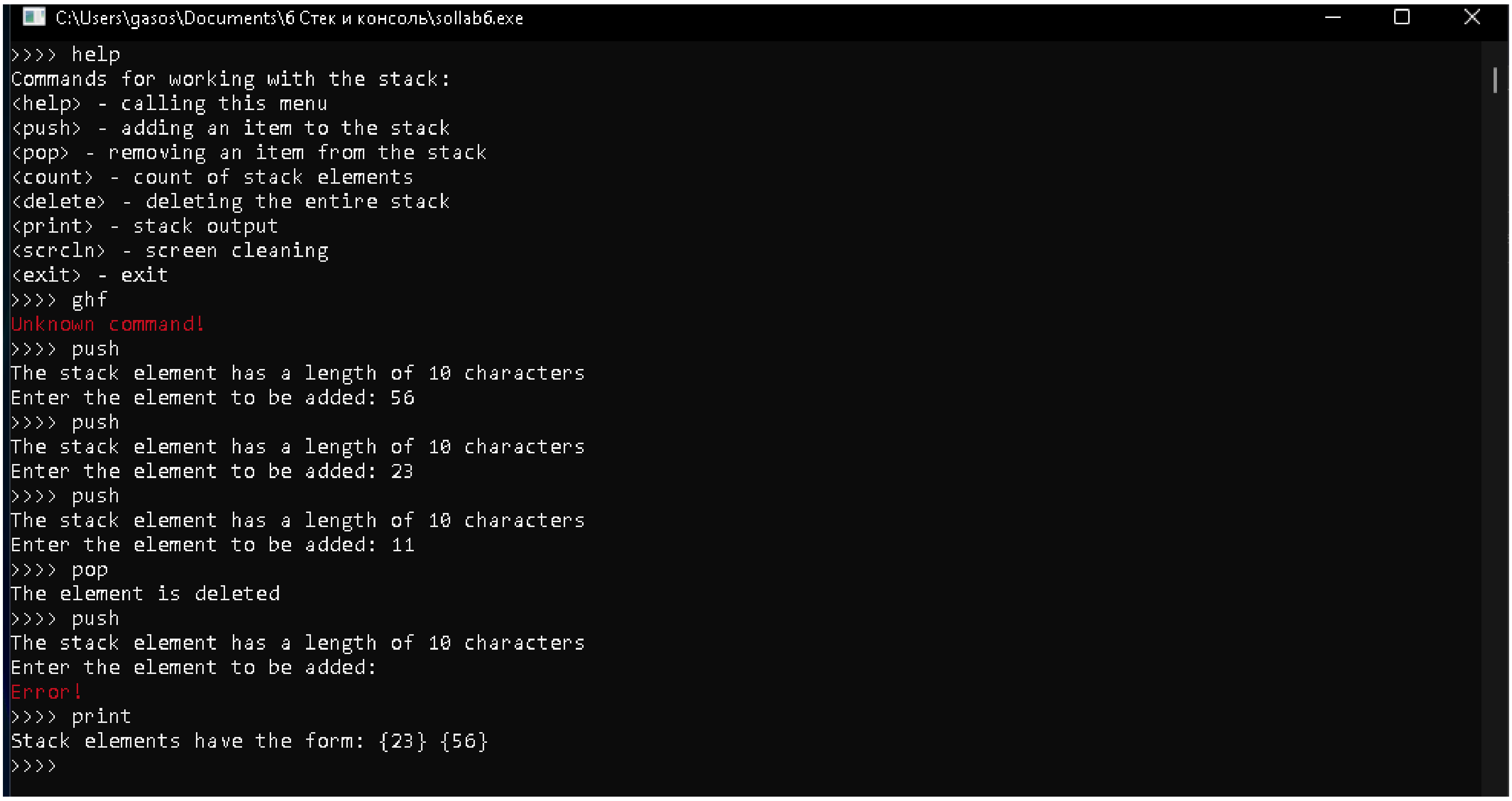
```
    if s = 'exit' then begin delallstack; break; end;
    if (s = 'help') or (s = 'push') or (s = 'pop') or (s =
'count') or (s = 'print') or (s = 'delete') or (s = 'scrcln') or
(s = 'changecolor') then
     begin
       if s = 'help' then begin  help;  end;
       if s = 'push' then begin  push;   end;
       if s = 'scrcln' then begin clrscr; end;
       if s = 'changecolor' then begin if mn = 15 then mn:=2 else
mn:=15; end;
       if s = 'pop' then begin
                            if check = 0 then begin
                            textcolor(red);
                            writeln('Stack is empty!');
                            textcolor(mn); end
                            else begin pop; writeln('The element
is deleted'); end;
                         end;
       if s = 'count' then begin
                            if check = 0 then begin
                            textcolor(red);
                            writeln('Stack is empty!');
                            textcolor(mn); end
                            else getCountElem;
                            end;
       if s = 'delete' then begin
                            if check = 0 then  begin
                            textcolor(red);
                            writeln('Stack is empty!');
                            textcolor(mn);    end
                            else begin delAllStack; writeln('Stack
cleared'); end;
                            end;
       if s = 'print' then begin
                            if check = 0 then begin
                            textcolor(red);
                            writeln('Stack is empty!');
                            textcolor(mn); end
                            else printFromTop;
                            end;
       wrstr();
     end
     else
     begin
      textcolor(red);
      writeln('Unknown command! ');
      textcolor(white);
      wrstr;
     end;
   end;
 end;



begin
  textcolor(15);
  textbackground(blue);
  writeln('Press "Enter" if you want to use auto-completion');
```

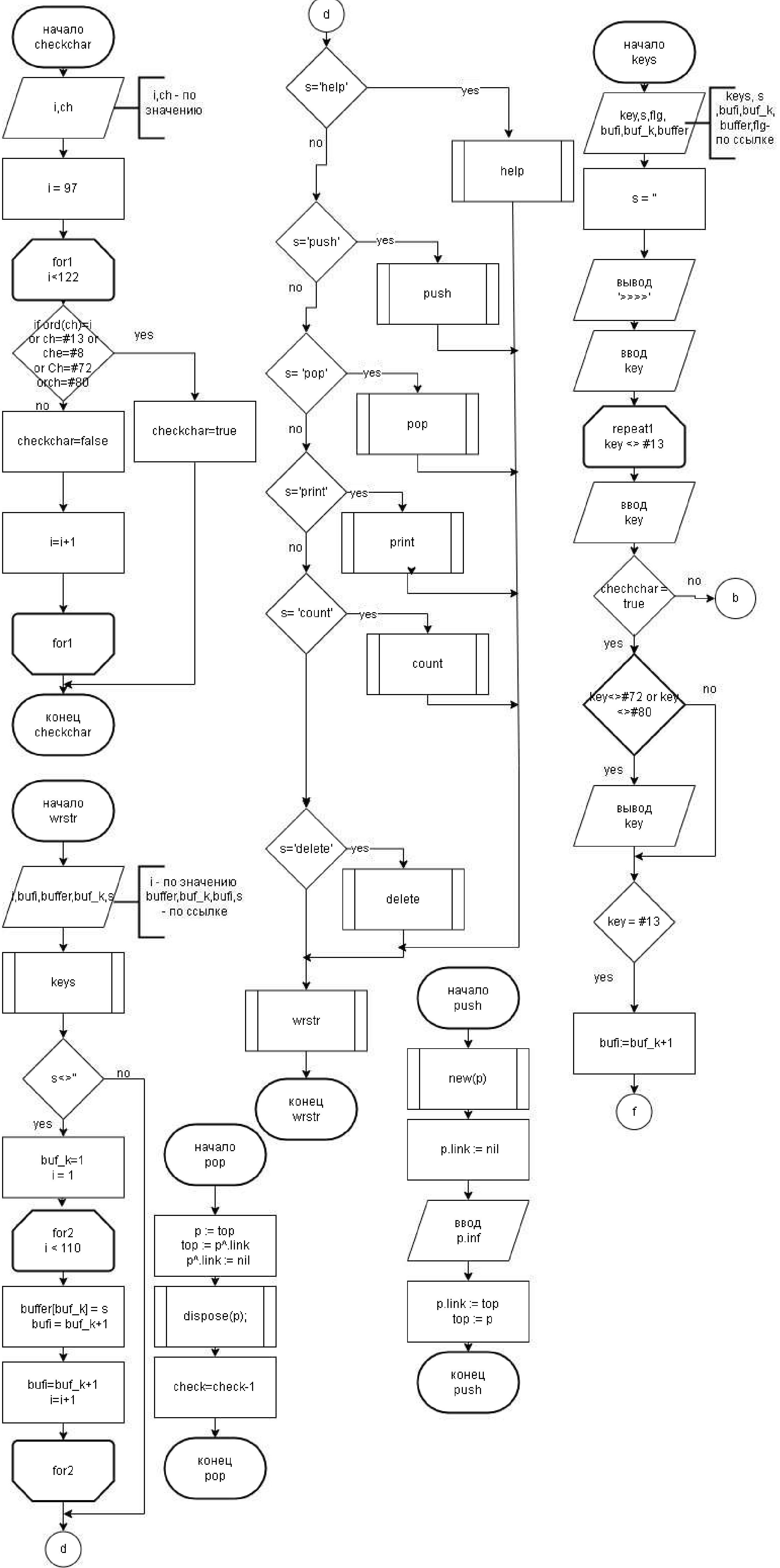
```
    textcolor(mn);
    textbackground(black);
    wrstr;
  end.
```

Экранная форма

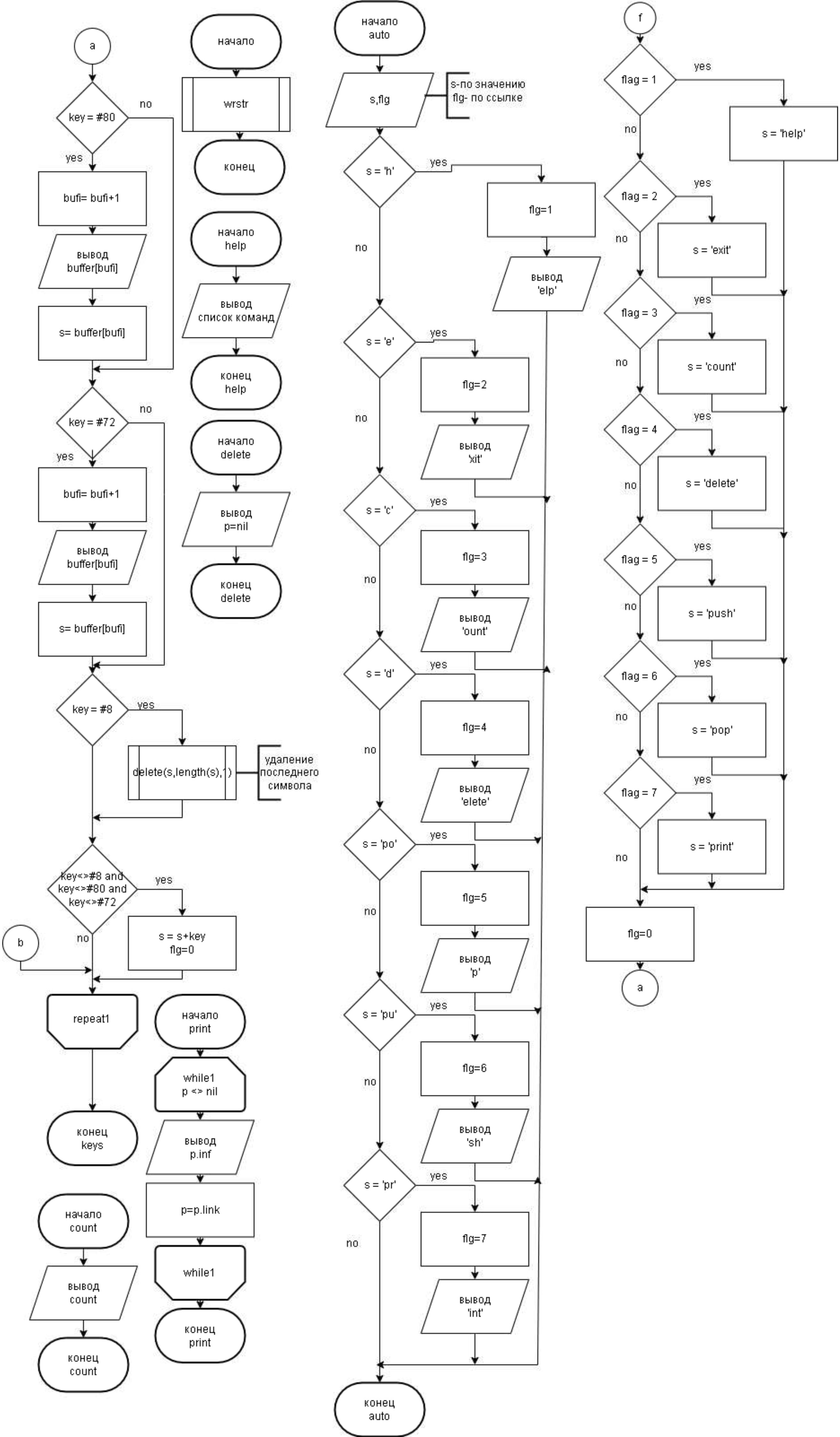


C:\Users\gasos\Documents\6 Стек и консоль\sollab6.exe

```
>>>> help
Commands for working with the stack:
<help> - calling this menu
<push> - adding an item to the stack
<pop> - removing an item from the stack
<count> - count of stack elements
<delete> - deleting the entire stack
<print> - stack output
<scrcln> - screen cleaning
<exit> - exit
>>>> ghf
Unknown command!
>>>> push
The stack element has a length of 10 characters
Enter the element to be added: 56
>>>> push
The stack element has a length of 10 characters
Enter the element to be added: 23
>>>> push
The stack element has a length of 10 characters
Enter the element to be added: 11
>>>> pop
The element is deleted
>>>> push
The stack element has a length of 10 characters
Enter the element to be added:
Error!
>>>> print
Stack elements have the form: {23} {56}
>>>>
```



начало checkchar
i,ch
i,ch - по значению
i = 97
for1 i<122
if ord(ch)=i or ch=#13 or che=#8 or Ch=#72 orch=#80
yes
no
checkchar=false
checkchar=true
i=i+1
for1
конец checkchar
начало wrstr
i,bufi,buffer,buf_k,s
i - по значению buffer,buf_k,bufi,s - по ссылке
keys
s<>''
no
yes
buf_k=1 i = 1
for2 i < 110
buffer[buf_k] = s bufi = buf_k+1
bufi=buf_k+1 i=i+1
for2
d
d
s='help'
yes
no
help
s='push'
yes
no
push
s= 'pop'
yes
no
pop
s='print'
yes
no
print
s= 'count'
yes
count
s='delete'
yes
delete
wrstr
конец wrstr
начало pop
p := top top := p^.link p^.link := nil
dispose(p);
check=check-1
конец pop
начало push
new(p)
p.link := nil
ввод p.inf
p.link := top top := p
конец push
начало keys
key,s,flg, bufi,buf_k,buffer
keys, s ,bufi,buf_k, buffer,flg- по ссылке
s = ''
вывод '>>>>'
ввод key
repeat1 key <> #13
ввод key
chechchar= true
no
b
yes
key<>#72 or key <>#80
no
yes
вывод key
key = #13
yes
bufi:=buf_k+1
f


a
key = #80
no
yes
bufi= bufi+1
вывод
buffer[bufi]
s= buffer[bufi]
key = #72
no
yes
bufi= bufi+1
вывод
buffer[bufi]
s= buffer[bufi]
key = #8
yes
delete(s,length(s),1)
удаление
последнего
символа
key<>#8 and
key<>#80 and
key<>#72
yes
no
s = s+key
flg=0
b
repeat1
конец
keys
начало
count
вывод
count
конец
count
начало
wrstr
конец
начало
help
вывод
список команд
конец
help
начало
delete
вывод
p=nil
конец
delete
начало
print
while1
p <> nil
вывод
p.inf
p=p.link
while1
конец
print
начало
auto
s,flg
s-по значению
flg- по ссылке
s = 'h'
yes
flg=1
вывод
'elp'
no
s = 'e'
yes
flg=2
вывод
'xit'
no
s = 'c'
yes
flg=3
вывод
'ount'
no
s = 'd'
yes
flg=4
вывод
'elete'
no
s = 'po'
yes
flg=5
вывод
'p'
no
s = 'pu'
yes
flg=6
вывод
'sh'
no
s = 'pr'
yes
flg=7
вывод
'int'
no
конец
auto
f
flag = 1
yes
s = 'help'
no
flag = 2
yes
s = 'exit'
no
flag = 3
yes
s = 'count'
no
flag = 4
yes
s = 'delete'
no
flag = 5
yes
s = 'push'
no
flag = 6
yes
s = 'pop'
no
flag = 7
yes
s = 'print'
no
flg=0
a