МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«**Вятский государственный университет**»

**(ФГБОУ ВО «ВятГУ»)**

Факультет автоматики и вычислительной техники

Кафедра ЭВМ

Отчёт

### Лабораторная работа № 6 по дисциплине

«Программирование»

«Реализация элементарных структур данных на основе динамической памяти»

Выполнил студент группы ИВТб-1301 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Жеребцов К. А.

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

Киров 2021

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

***Задания:***

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:***

*Основная программа:*

program STACK;

uses

crt, module;

type

commands = array of string;

strr = record

text: string;

countspaces: int64;

end;

const

max:integer = 255;

cmdlist: array of string = (

'exit',

'help',

'insert',

'delete',

'print',

'number',

'clear'

);

windowSize=121;

var

top: Tptr;

sel : integer;

buffer: commands;

cmdEl: commands;

curCom: int64 = -1;

bs:string;

ch: Char;

s: String;

x, y: Integer;

curY: Integer;

w:boolean = true;

h:string = '>';

Procedure WriteCom(Var s:String;y:integer); // вывод команды

var i:integer=0;

a,b,c:integer;

Begin

gotoxy(1,curY);

clreol();

while i<y Do

begin

gotoxy(1,wherey()+1);

clreol();

inc(i);

end;

a:=(x+length(h)+2) mod windowSize;

if a=0 then

begin

a:=x+length(h)+2;

b:=0;

end

else

begin

b:=(x+length(h)+2) div windowSize;

end;

c:=(x+length(h)+3) div windowSize;

if curY+c > 25 then

begin

curY:=curY-c;

i:=0;

while i<c do

begin

writeln();

inc(i);

end;

end;

gotoxy(1,curY);

write(h,' ');

write(s);

gotoxy(a,curY+b);

End;

Procedure DeleteChar(var s:string); // удаление символов

var y:integer;

Begin

if x>=length(h) then

begin

x:=x-1;

delete(s,x+1,1);

y:=(length(h)+length(s)+2) div windowSize;

WriteCom(s,y);

end;

End;

Procedure AddChar(var s:string;ch:char); // добавление символа в команду

var y:integer;

Begin

if length(s)<max then

begin

x:=x+1;

s:=copy(s,1,x-1)+ch+copy(s,x,length(s));

y:=(length(h)+length(s)+2) div windowSize;

WriteCom(s,y);

end;

end;

procedure SaveCommand(); // сохранение команд

var c:int64;

begin

c:=length(buffer);

buffer[c-1]:=s;

setlength(buffer,c+1);

curCom:=c;

end;

procedure ChangeCommand(c:int64); // замена команд

begin

if curCom >= 0 then

begin

y:=(length(h)+length(s)+2) div windowSize;

buffer[curCom]:=s;

if (c = 1) and (curCom > 0) then

begin

curCom:=curCom-1;

end

else if (c = 2) and (curCom < length(buffer)-1) then

begin

curCom:=curCom+1;

end;

s:=buffer[curCom];

x:=length(s);

WriteCom(s,y);

end;

end;

function RemoveSpaces(s:string):strr; // удаление пробелов

var i:integer;

str:strr;

begin

i:=0;

while s[i+1]=' ' do i:=i+1;

delete(s,1,i);

str.text := s;

str.countSpaces := i;

Exit(str);

end;

procedure ParseString(str:string);

var i:int64 = 2;

c:int64 = 0;

s:strr;

begin

s:=RemoveSpaces(str);

if length(s.text)>0 then

begin

setLength(cmdEl,1);

cmdEl[c]:=s.text[1];

end;

while i <= length(s.text) do

begin

if (s.text[i-1]=' ') and (s.text[i]<>' ') then

begin

Inc(c);

setLength(cmdEl,length(cmdEl)+1);

cmdEl[c]:=s.text[i];

end

else if s.text[i]<>' ' then

begin

cmdEl[c]:=cmdEl[c]+s.text[i];

end;

Inc(i);

end;

end;

procedure Info(); // вывод информации

begin

writeln('Commands: ');

writeln('help - shaw all commands;');

writeln('number - total number of elements in the stack;');

Writeln('exit - exit the program;');

Writeln('clear - clear the stack;');

Writeln('insert <element> - inserting an item at the top of the stack;');

Writeln('print - printing stack elements on screen from top to down;');

Writeln('delete - removes an item from the top of the stack;');

end;

function NeedArgs(c:int64):boolean;//проверка требования командой аргумента

var i:integer;

begin

if length(cmdEl)-1 = c then

begin

Exit(true);

end

else

begin

for i:=c+1 to length(cmdEl)-1 do

begin

writeln('Unknown argument: ', cmdEl[i]);

end;

end;

Exit(false);

end;

procedure Action(cmd:string); // обработка команд

var a:int64;

b:integer;

begin

case cmd of

'clear':

begin

if NeedArgs(0) then

begin

delAllStack;

end;

end;

'exit':

begin

if NeedArgs(0) then

begin

writeln();

halt();

end;

end;

'number':

begin

if NeedArgs(0) then

begin

getCountElem();

end;

end;

'help':

begin

if NeedArgs(0) then

begin

Info();

end;

end;

'insert':

begin

if NeedArgs(1) then

begin

val(cmdEl[1],a,b);

if b = 0 then

begin

push(a);

end

else

begin

writeln('The argument must be a number!');

end;

end;

end;

'print':

begin

if NeedArgs(0) then

begin

printFromTop;

writeln();

end;

end;

'delete':

begin

if NeedArgs(0) then

begin

pop;

end;

end;

else

begin

writeln('There is no such command!');

end;

end;

end;

procedure RunCommand(); // запуск команды

begin

ParseString(s);

if length(cmdEl) > 0 then

begin

Action(cmdEl[0]);

end;

end;

function EnterCommand(y:int64):int64; // применение команды

var a,b:int64;

begin

if y<25 then begin

y:=wherey+1;

end;

x:=0;

b:=(length(s)+length(h)+2) div windowSize;

if b>0 then

begin

a:=(length(s)+length(h)+2) mod windowSize;

end

else

begin

a:=length(s)+length(h)+2;

end;

gotoxy(a,curY+b);

writeln();

RunCommand();

cmdEl:=nil;

SaveCommand();

s:='';

write(h,' ');

Exit(y);

end;

procedure AutoCompletion(); // автодополнение

var

i:integer;

str:strr;

begin

str:=RemoveSpaces(s);

for i:=0 to length(cmdList)-1 do

begin

if pos(str.text,cmdList[i]) = 1 then

begin

Delete(s,str.countSpaces+1,Length(s));

Insert(cmdList[i],s,str.countSpaces+1);

x:=length(s);

WriteCom(s,1);

break;

end;

end;

end;

procedure cursorLeft(); // курсор влево

var y: int64;

a,b:int64;

begin

if x > 0 then begin

x:=x-1;

a:=(x+length(h)+2) mod windowSize;

b:=(x+length(h)+2) div windowSize;

if a = 0 then

begin

a:=(x+length(h)+2) div b;

b:=b-1;

end;

gotoxy(a,curY+b);

end;

end;

procedure cursorRight(s:string); // курсор вправо

var y: integer;

a,b:integer;

begin

if x < length(s) then begin

x:=x+1;

a:=(x+length(h)+2) mod windowSize;

b:=(x+length(h)+2) div windowSize;

if a = 0 then

begin

a:=(x+length(h)+2) div b;

b:=b-1;

end;

gotoxy(a,curY+b);

end;

end;

procedure SpecialKey(s:string); // передвижение указателя стрелками

var code:integer;

ch:char;

begin

ch:=ReadKey();

code:=ord(ch);

Case code of

75: //<-

begin

cursorLeft();

end;

77://->

begin

cursorRight(s);

end;

72: //up

begin

ChangeCommand(1);

end;

80: //down

begin

ChangeCommand(2);

end;

end;

end;

function choiceCommand(ch:char;var y:integer):boolean; // выбор действия

var

code: Integer;

begin

code := ord(ch);

Case code Of

13: //Enter

begin

y:=EnterCommand(y);

curY:=wherey();

end;

8: //Backspace

begin

DeleteChar(s);

end;

9: //Tab

begin

AutoCompletion();

end;

0: SpecialKey(s);

Else

begin

AddChar(s,ch);

end;

End;

Exit(true);

end;

begin

clrscr;

top:= NIL;

write(h,' ');

s := '';

x := 0;

setlength(buffer,1);

curY := WhereY();

While w Do

Begin

ch := ReadKey;

w:=choiceCommand(ch,curY);

End;

buffer:=nil;

End.

delAllStack;

end.

*Модуль:*

unit MODULE;

interface

type

Tptr = ^Telem;

Telem = record

inf : int64;

link : Tptr;

end;

var

top: Tptr;

procedure push(k:int64); // вставка элемента

procedure pop; // удаление элемента

procedure printFromTop; // вывод всех элементов

procedure delAllStack; //очистить структуру

function getCountElem : int64; // подсчет количества элементов

implementation

procedure push(k: int64);

var

p : Tptr;

begin

new(p);

p^.link := NIL;

p^.inf:= k;

p^.link := top;

top := p;

end;

procedure pop;

var

p : Tptr;

begin

p := top;

top := p^.link;

p^.link := NIL;

dispose(p);

end;

procedure printFromTop;

var

p : Tptr;

begin

p := top;

while(p <> NIL) do

begin

write(p^.inf, ' ');

p := p^.link;

end;

end;

procedure delAllStack;

var

p : Tptr;

begin

p := top;

while(p <> NIL) do

begin

top := p^.link;

p^.link := NIL;

dispose(p);

p := top;

end;

end;

function getCountElem : int64;

var

p : Tptr;

k : integer;

begin

k := 0;

p := top;

while(p <> NIL) do

begin

k := k + 1;

p := p^.link;

end;

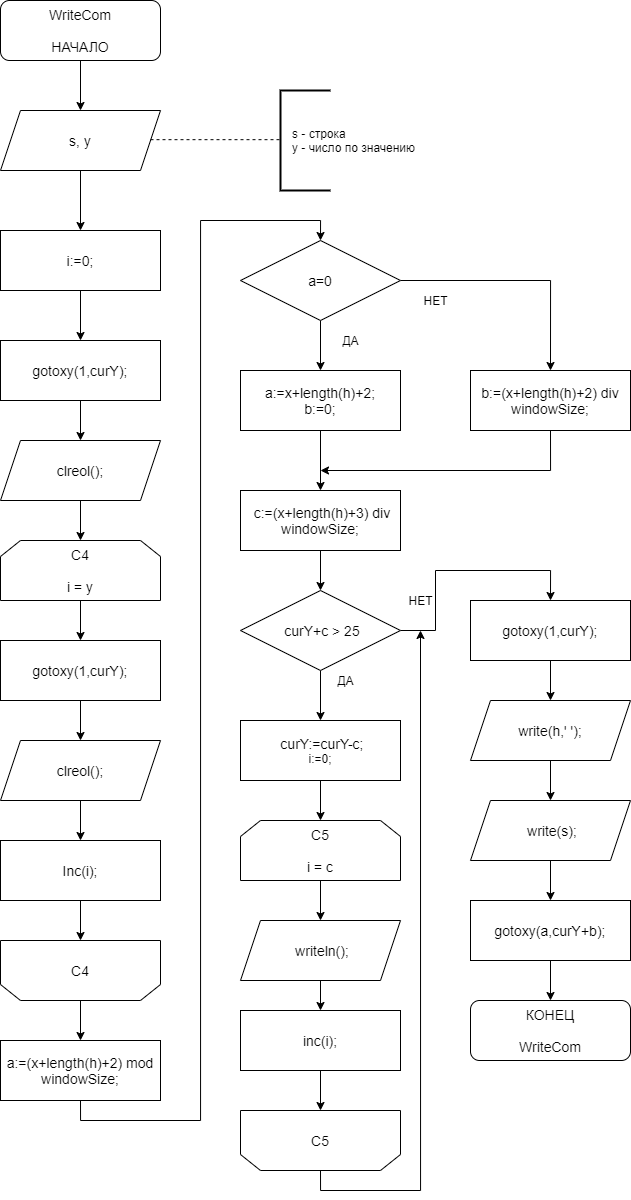
writeln(k);

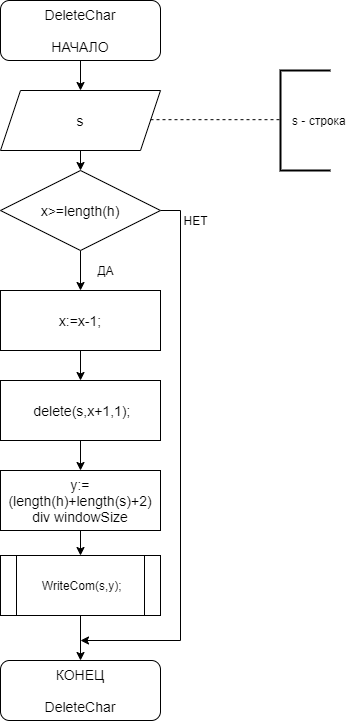
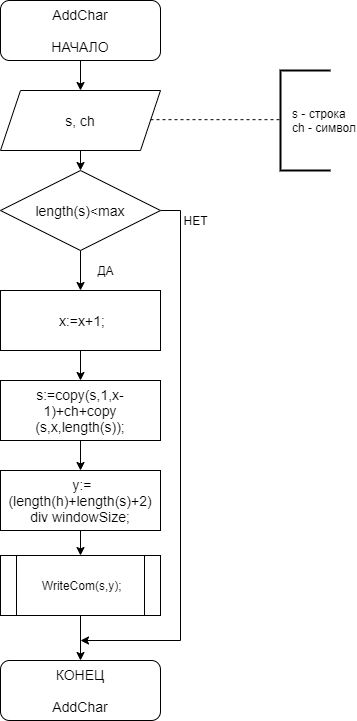
end;

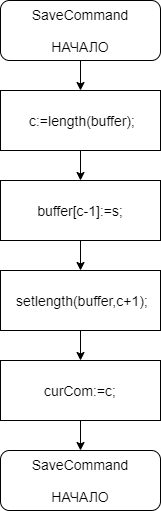
end.

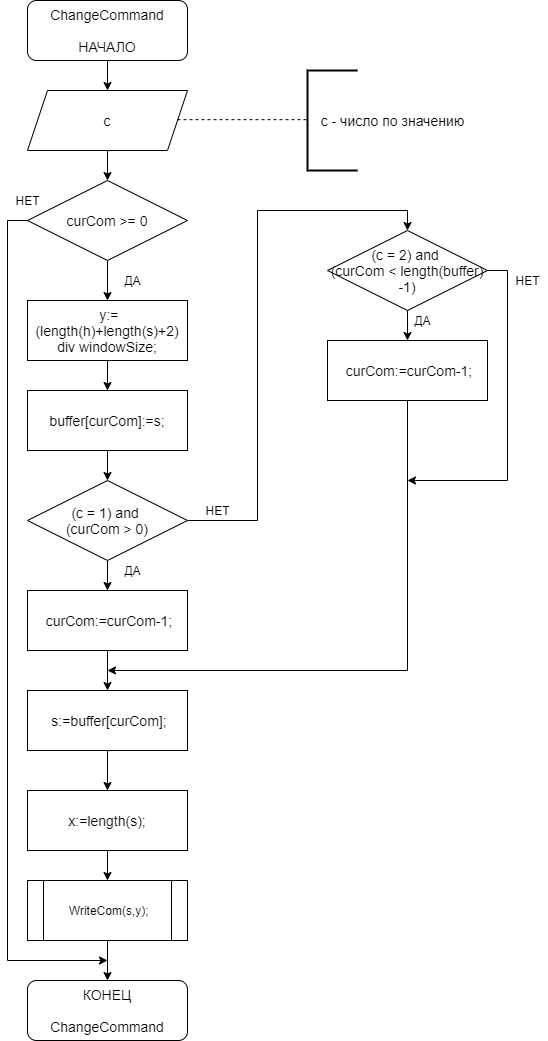
***Схема алгоритма:***

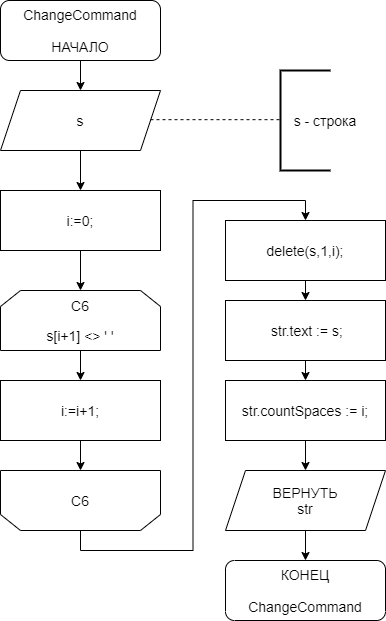
*Основная программа:*

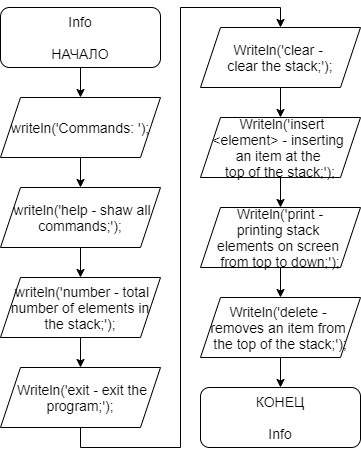
**

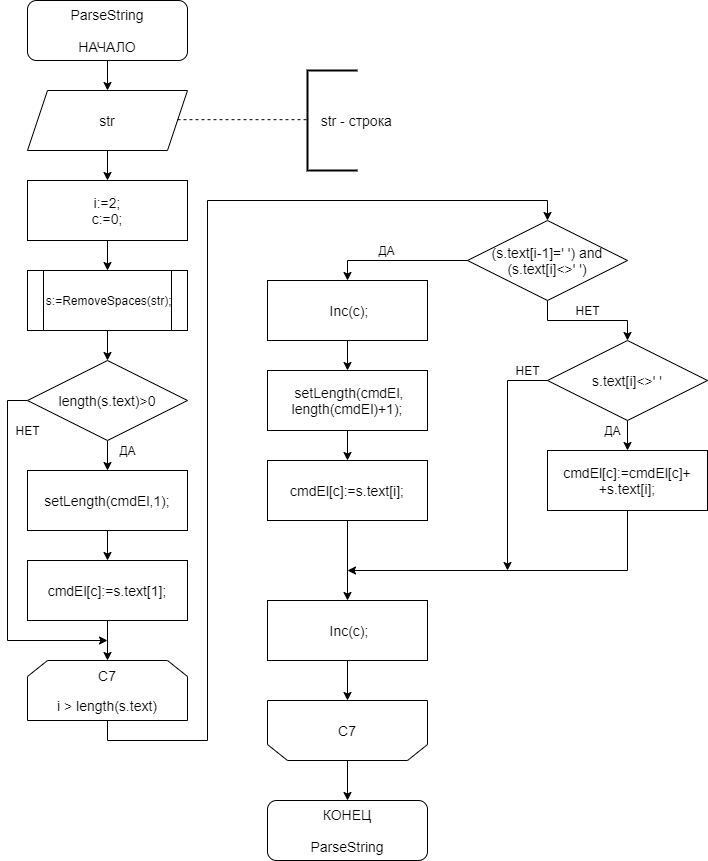
* *

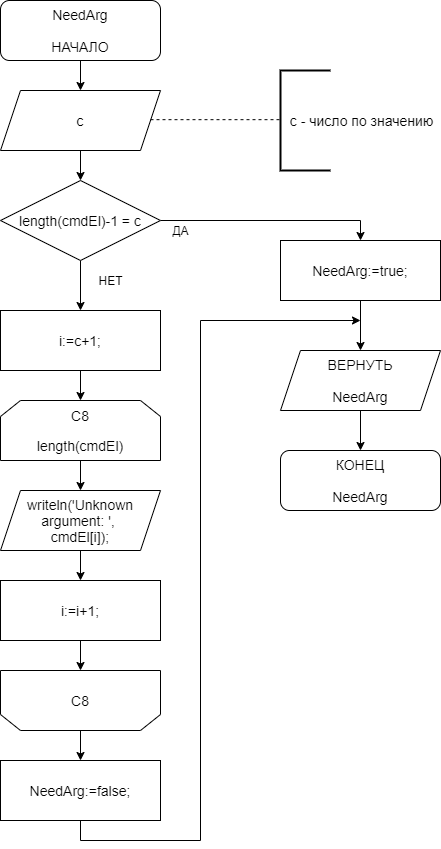
**

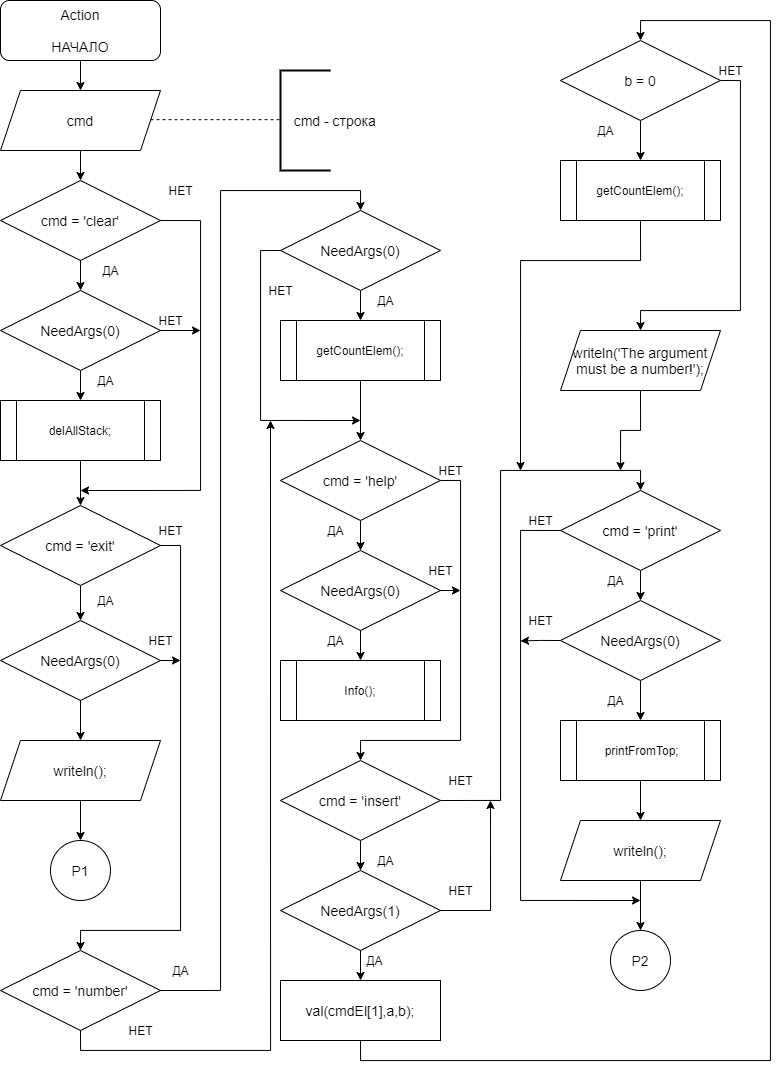
**

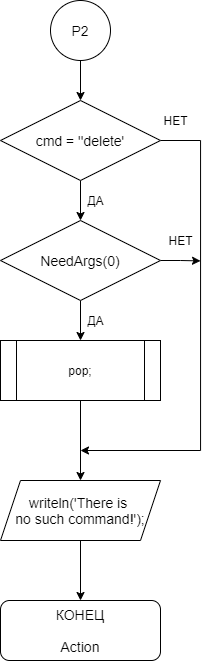
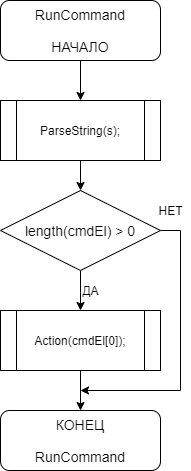
**

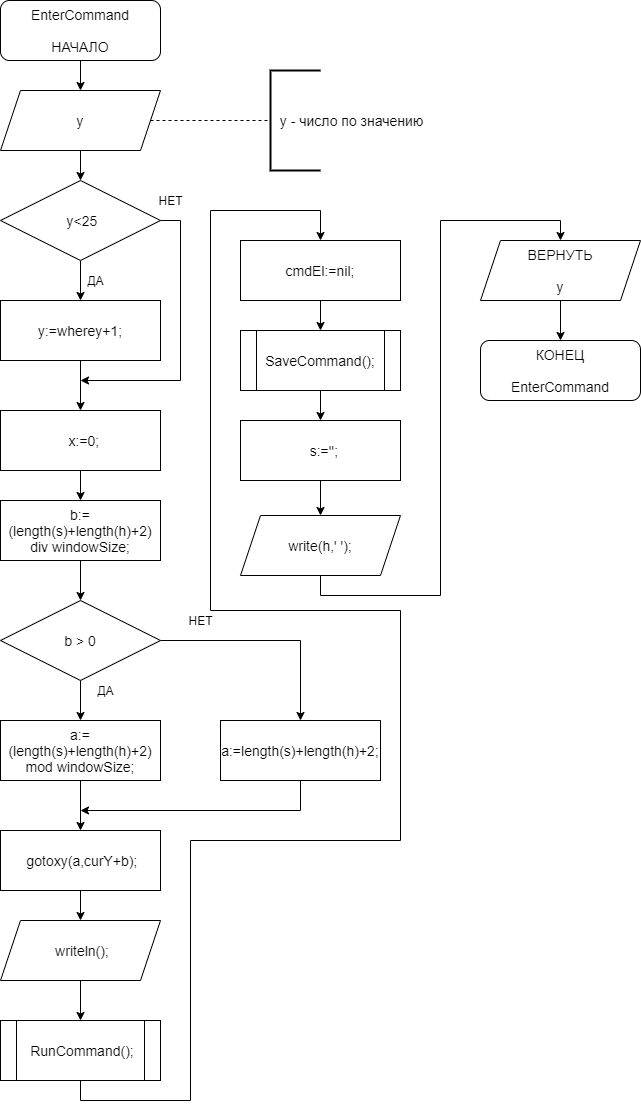
**

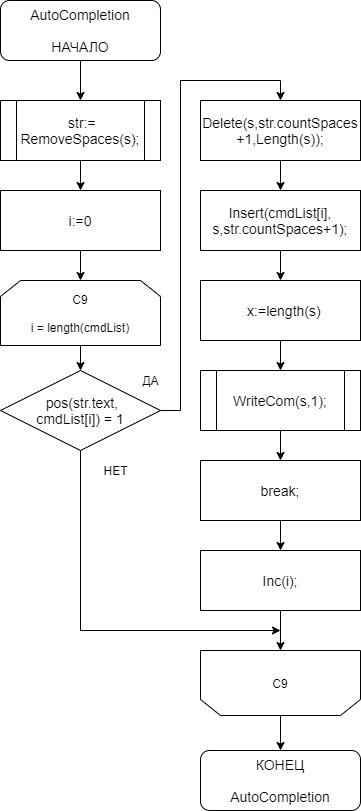
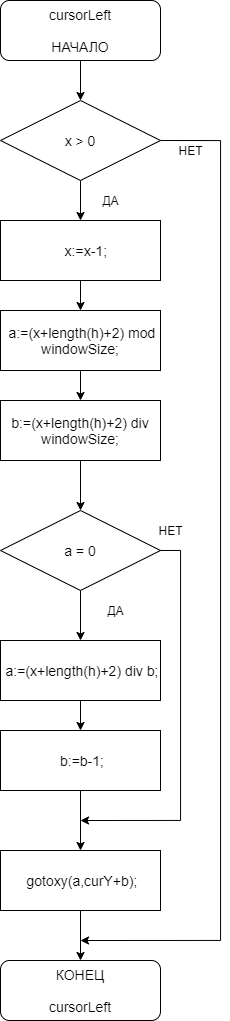
**

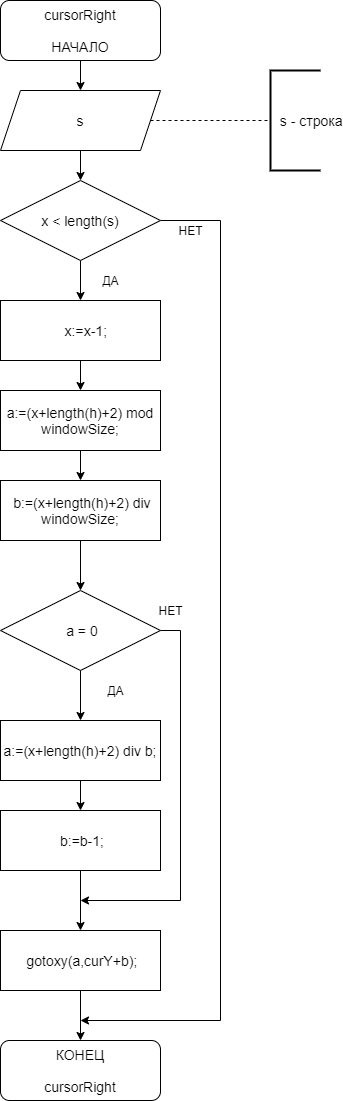
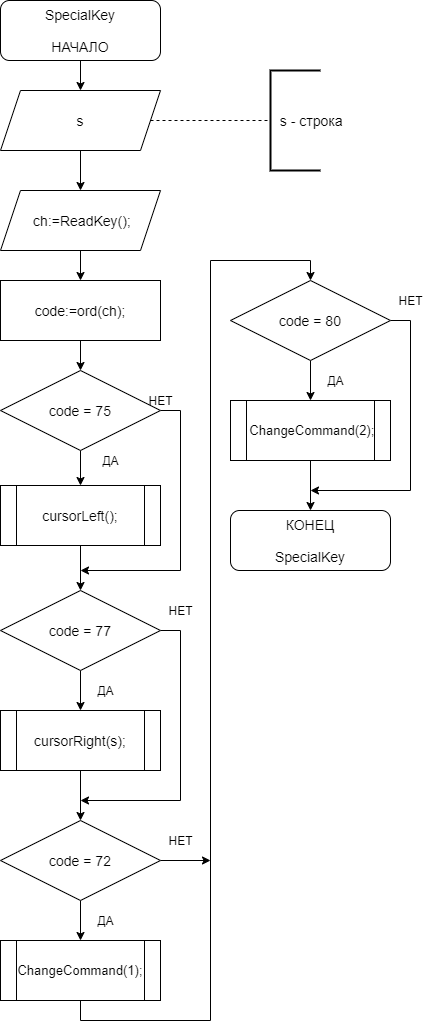
**

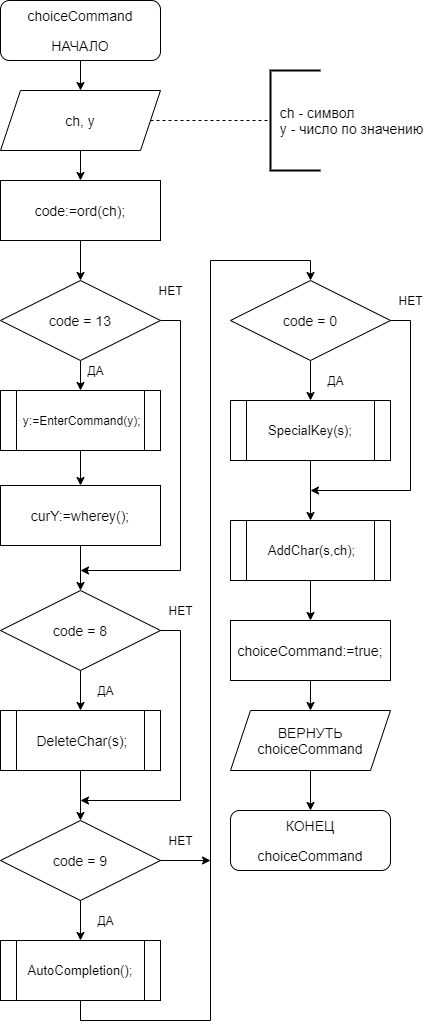
**

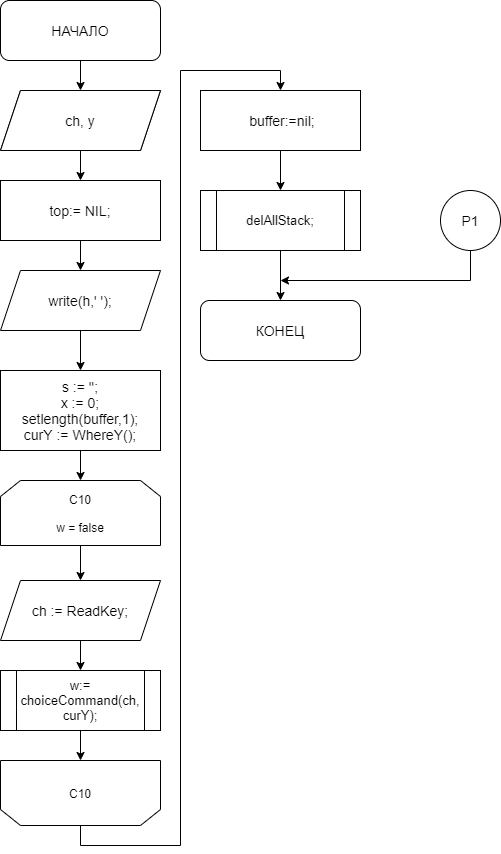
* *

**

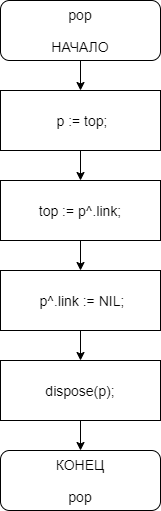
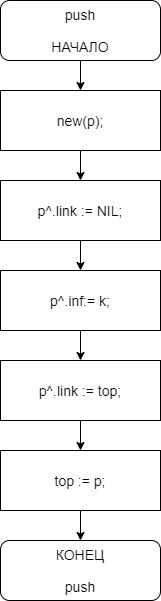
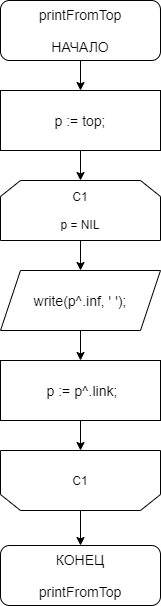
* *

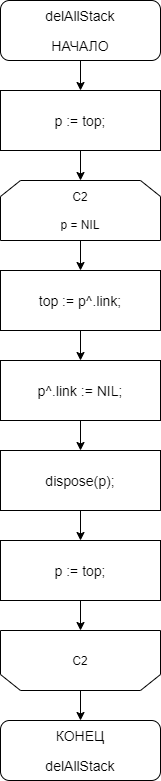
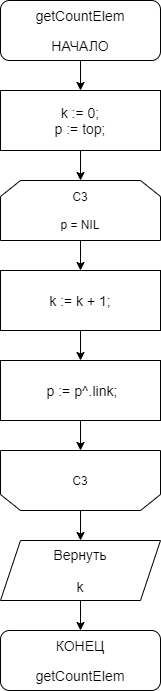
* *

**

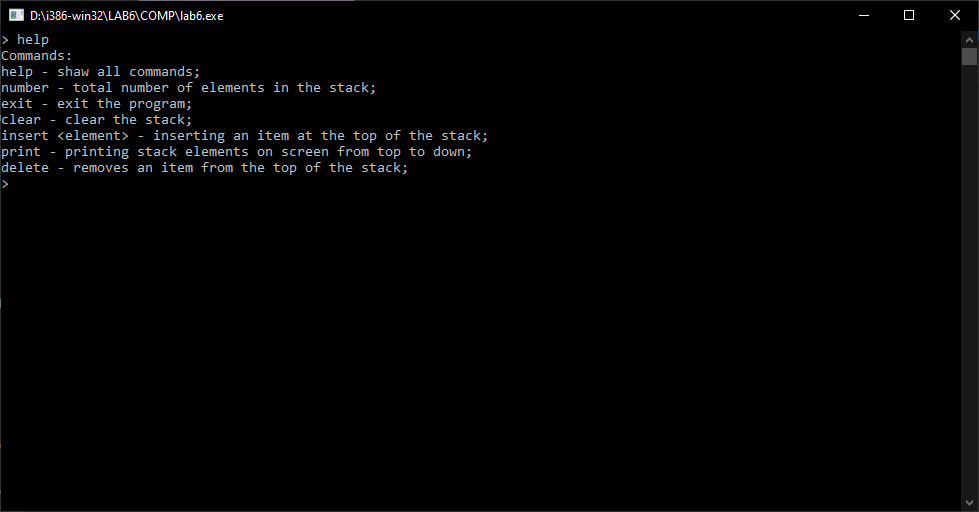
**

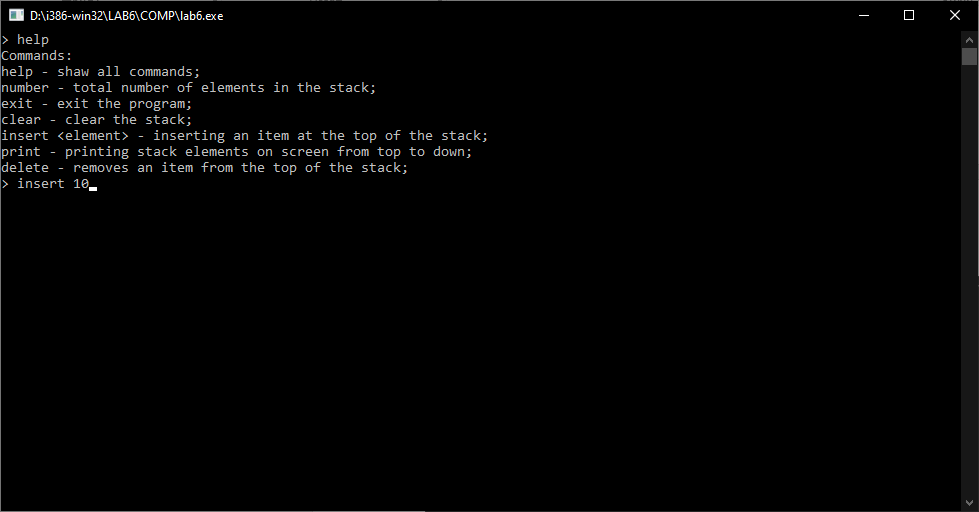
*Модуль:*

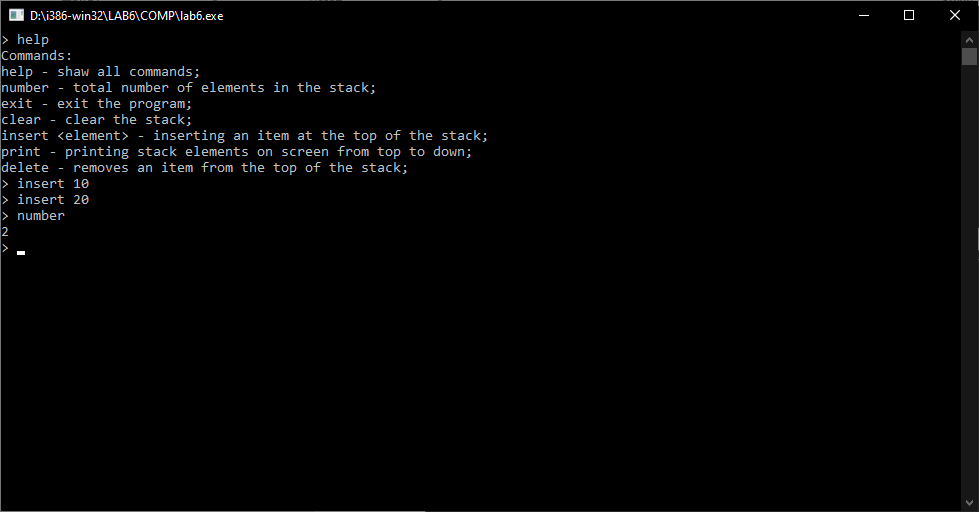
  

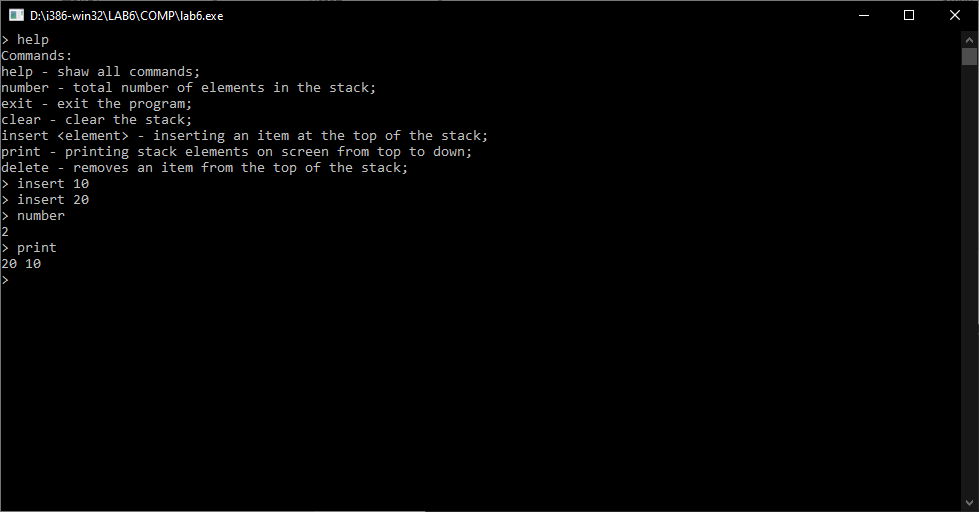
 

***Экранные формы:***









***Вывод:***

Была изучена структура и принципы организации программных модулей, были закреплены навыки работы с динамической памятью и получены базовые навыки организации работы в режиме командной строки.

Написана программа для работы со структурой данных «Стек», которая реализована на основе динамической памяти и описана в отдельном модуле.

Работа со структурой должна осуществляется в режиме командной строки (с реализацией автодополнения и истории команд). Также предусмотрена наглядная визуализация содержимого структуры.