МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ   
ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

«Вятский государственный университет»

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Отчет

Лабораторная работа №6 по дисциплине

«Программирование»

Реализация элементарных структур данных на основе динамической памяти

Выполнил студент группы ИВТ-1301 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Коротаев Р.С./

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

Киров

2020

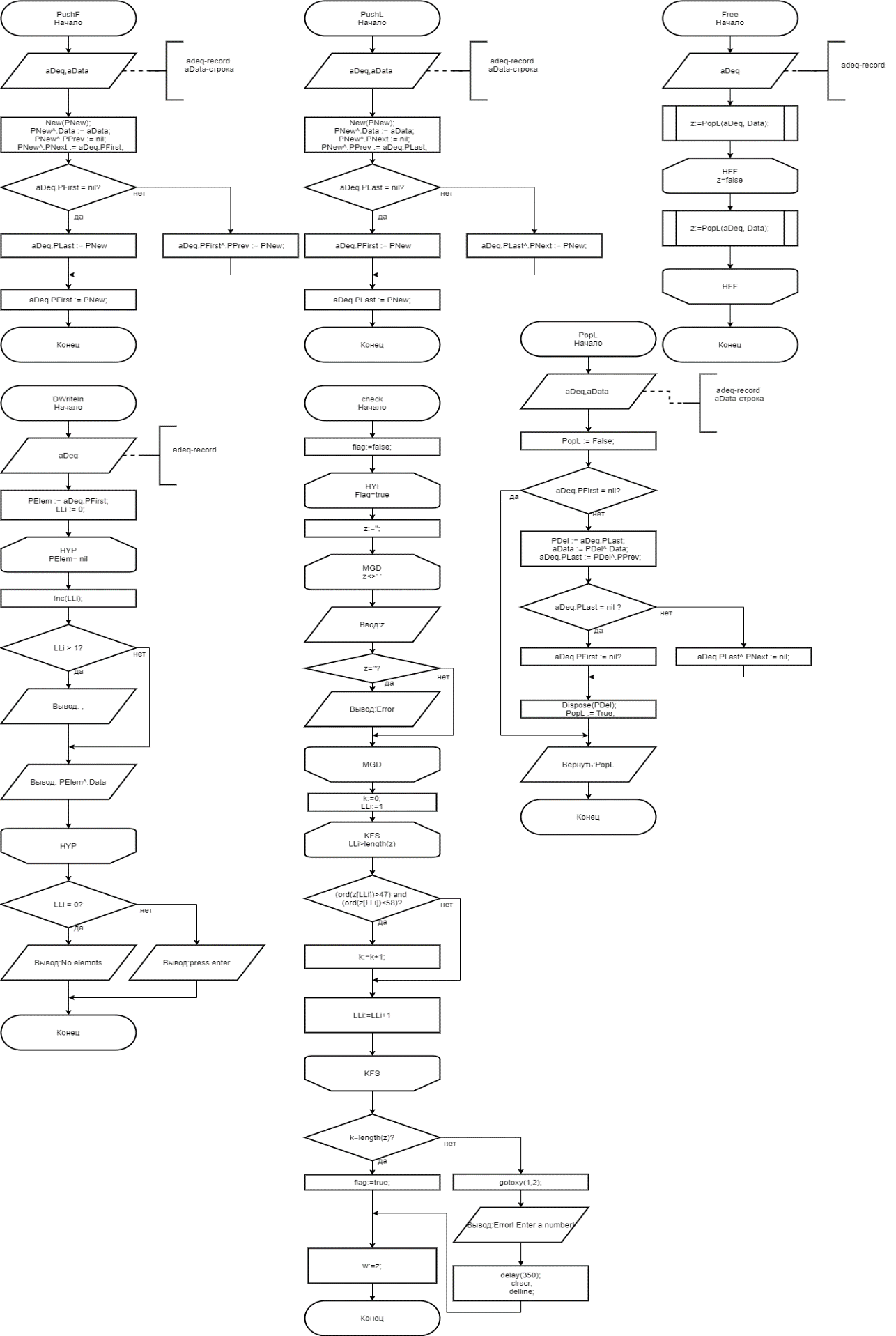
Вариант:15

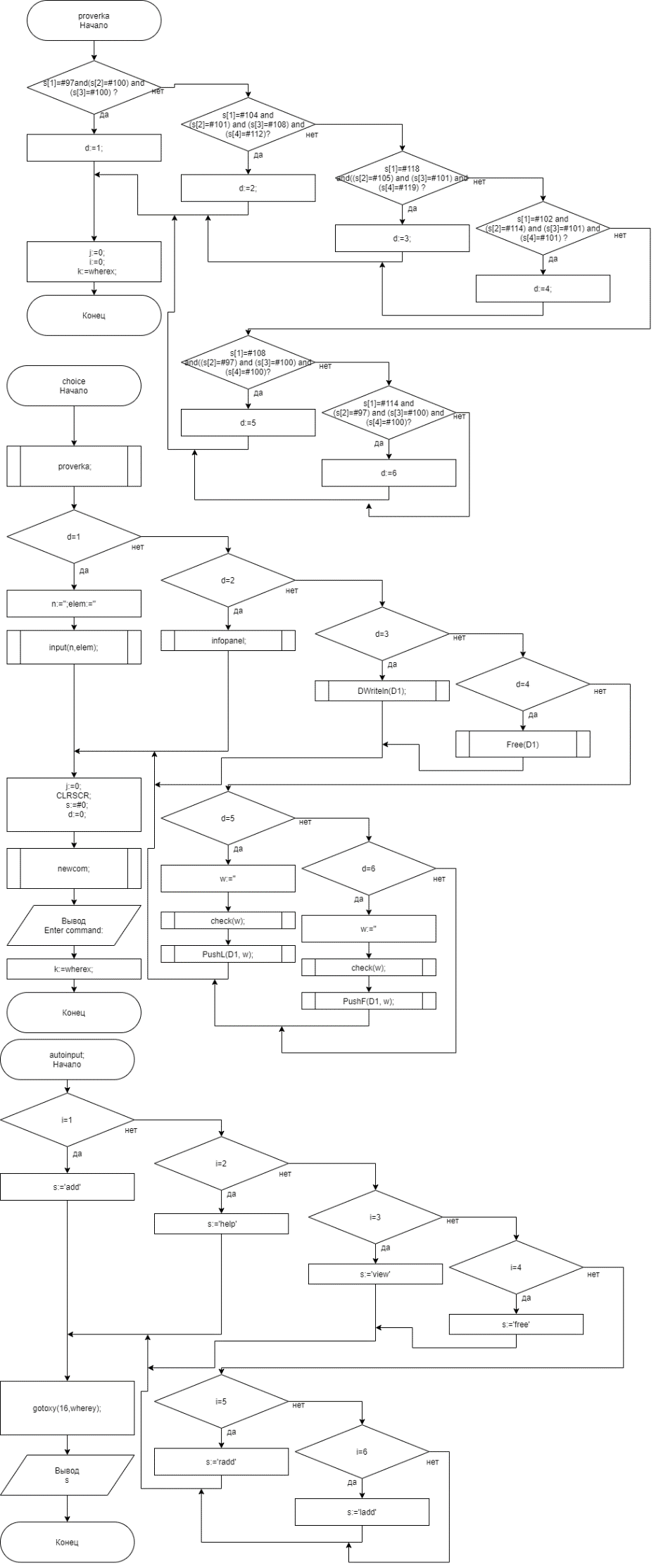
Цель работы: изучить структуры и принципов организации программных модулей, закреплением навыков работы с динамической памятью, получение базовых навыков организации работы режиме командной строки.

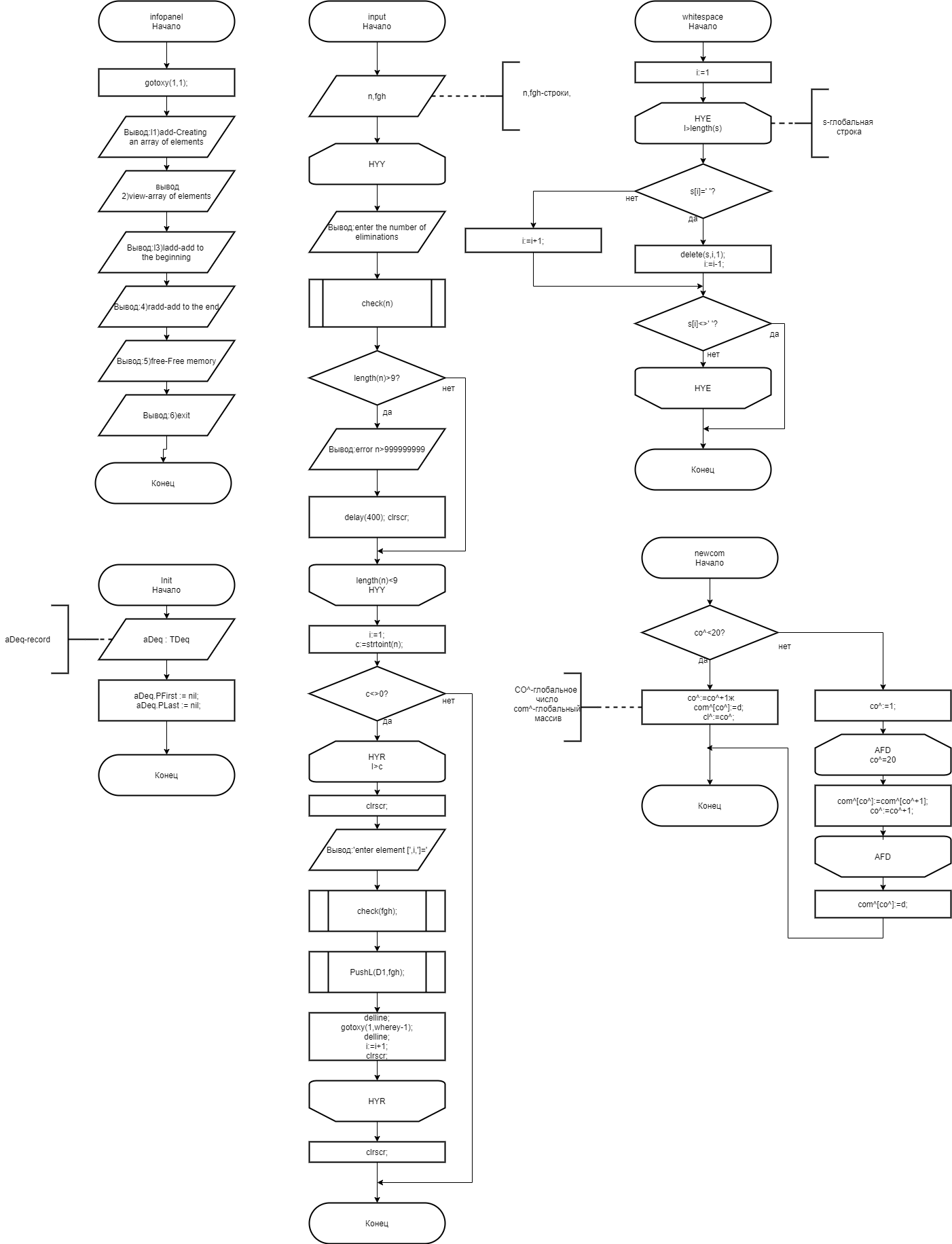
Задание:

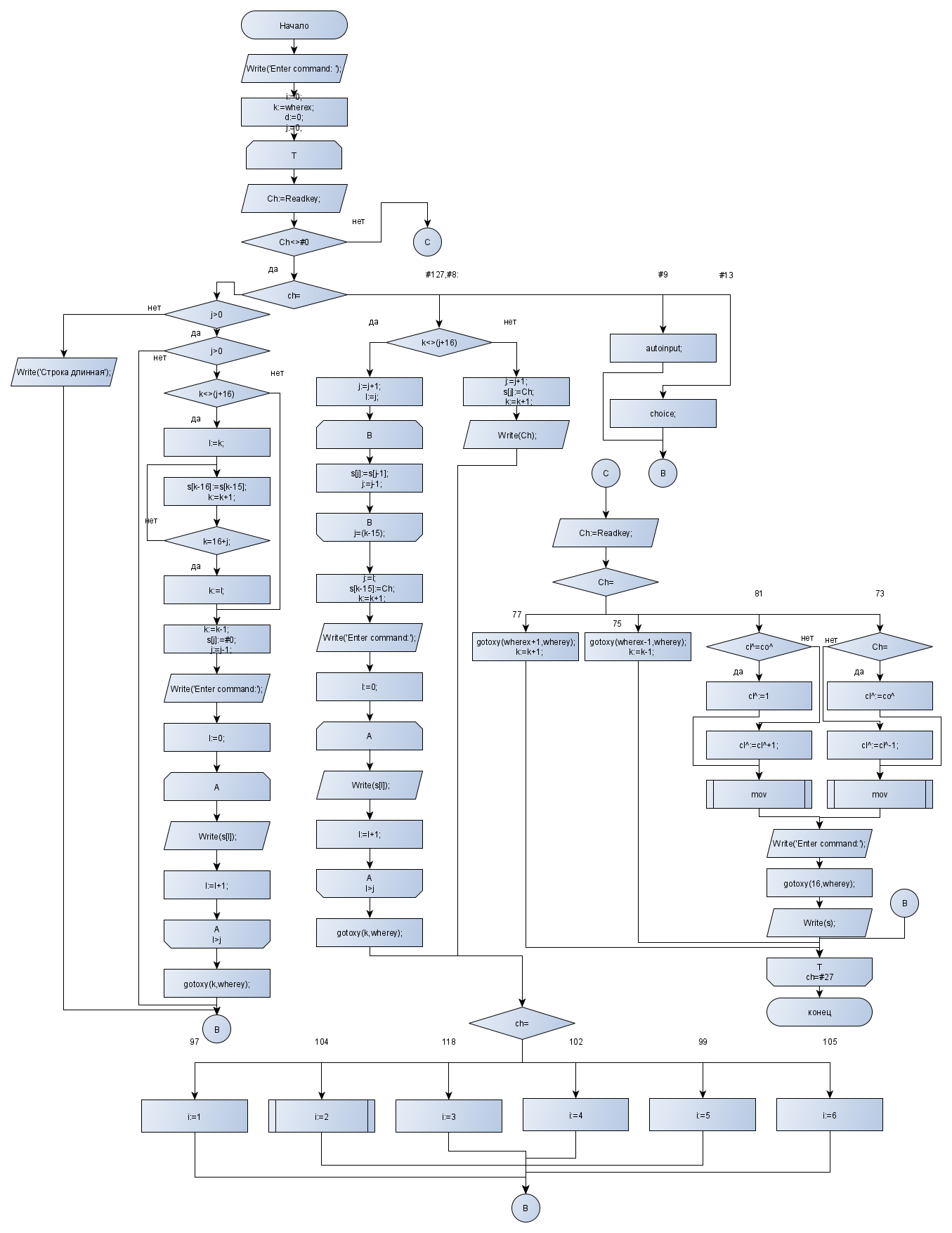
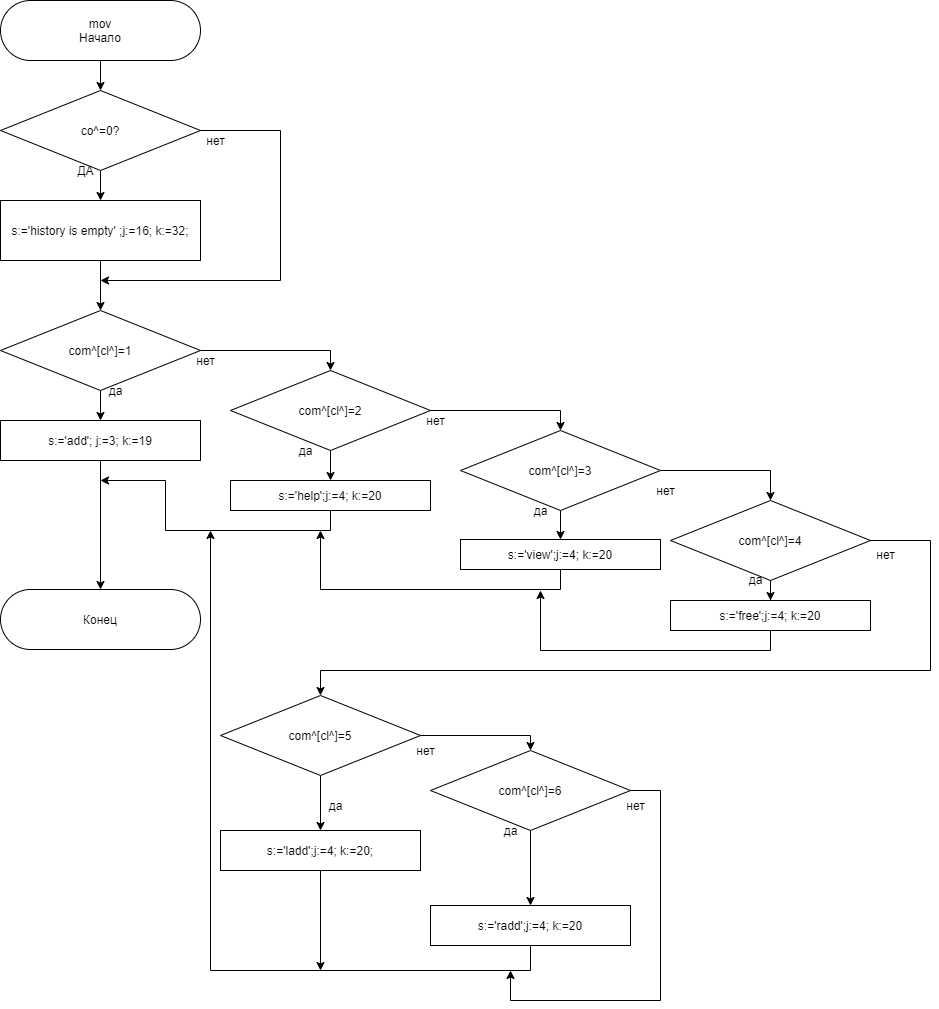
1. Написать программу для работы со структурой данных “Дек”
2. Структура данных должна быть реализована на основе динамической памяти.
3. Структура данных(поля и методы) должны быть описана в отдельном модуле.
4. Работа со структурой должна осуществляться в режиме командной строки(с реализацией автодополнения и истории команд). Предусмотреть наглядную визуализацию содержимого структуры.

Схема алгоритма









Исходный код

1. Основная программа:

**program** qq;

**uses** crt,Laba\_6\_KorotaevRoman;

{$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;

#108: **If** (s[2]=#97) **and** (s[3]=#100) **and** (s[4]=#100) **then** d:=5;

#114: **If** (s[2]=#97) **and** (s[3]=#100) **and** (s[4]=#100) **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: DWriteln(D1);

4: **begin** Free(D1);**end**;

5: **begin** w:=''; check(w); PushL(D1, w); **end**;

6: **begin** w:=''; check(w); PushF(D1, w);**end**;

**end**;

j:=0;

newcom;

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:='radd';

6: s:='ladd';

**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;

#108: i:=5;

#114: 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**.

1. Модуль:

**unit** Laba\_6\_KorotaevRoman;

**interface**

**uses** crt,SysUtils;

**type**

st=string;

TData = string;

int=longint;

TPElem = ^TElem;

TElem = **record**

Data : TData;

PPrev, PNext : TPElem;

**end**;

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

{Тип, описывающий дек.}

TDeq = **record**

PFirst, PLast : TPElem; {Указатели на первый и на последний элементы дека.}

**end**;

**procedure** check(**var** w:string);

**procedure** PushL(**var** aDeq : TDeq; **const** aData : TData);

**procedure** PushF(**var** aDeq : TDeq; **const** aData : TData);

**function** PopL(**var** aDeq : TDeq; **var** aData : TData) : Boolean;

**procedure** DWriteln(aDeq : TDeq);

**procedure** Free(**var** aDeq : TDeq);

**procedure** Init(**var** aDeq : TDeq);

**procedure** infopanel;

**procedure** input(**var** n,fgh:string);

**procedure** whitespace;

**Procedure** newcom;

**Procedure** mov;

**var**

D1 : TDeq;

Data : TData;

co,cl:^integer;

com:^comm;

s:string;

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

Ch:char;

w,q,n,elem:string;

**IMPLEMENTATION**

**procedure** whitespace;

**var** i:int;

**begin**

i:=1;

**while** i<=length(s) **do**

**begin**

**if** s[i]=' ' **then**

**begin**

delete(s,i,1);

i:=i-1;

**end**

**else**

i:=i+1;

**if** s[i]<>' ' **then break**;

**end**;

**end**;

**procedure** input(**var** n,fgh:string);

**var** i,c:int;

flag22:boolean;

**begin**

**repeat**

writeln('enter the number of eliminations');

check(n);

**if** (length(n)>9) **then begin** writeln('error n>999999999'); delay(400); clrscr; **END**;

**until** (length(n)<=9);

i:=1;

c:=strtoint(n);

**if** c<>0 **then**

**begin**

**while** i<=c **do**

**begin**

clrscr;

write('enter element [',i,']=');

check(fgh);

PushL(D1,fgh);

delline;

gotoxy(1,wherey-1);

delline;

i:=i+1;

clrscr;

**end**;

**end**;

clrscr;

**end**;

**procedure** infopanel;

**begin**

writeln('list of commands');

writeln('1)add-Creating an array of elements');

writeln('2)view-array of elements');

writeln('3)ladd-add to the beginning');

writeln('4)radd-add to the end');

writeln('5)free-Free memory');

writeln('6)exit');

readln;

**end**;

**procedure** Init(**var** aDeq : TDeq);

**begin**

aDeq.PFirst := nil;

aDeq.PLast := nil;

**end**;

**procedure** check(**var** w:string);

**var**

flag:boolean;

k,LLi:int;z:string;

**begin**

flag:=false;

**while** flag<>true **do**

**begin**

z:='';

**while** z='' **do**

**begin**

readln(z);

**if** z='' **then begin** writeln('Error');delay(300); clrscr; **end**;

**end**;

k:=0;

**for** LLi:=1 **to** length(z) **do**

**begin**

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

k:=k+1;

**end**;

**if** k=length(z) **then begin**

flag:=true;

**end**

**else begin**

gotoxy(1,2);

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

delay(350);

clrscr;

delline;

**end**;

**end**;

w:=z;

**end**;

{Добавление элемента в начало дека.}

**procedure** PushF(**var** aDeq : TDeq; **const** aData : TData);

**var**

PNew : TPElem;

**begin**

**New**(PNew);

PNew^.Data := aData;

PNew^.PPrev := nil;

PNew^.PNext := aDeq.PFirst;

**if** aDeq.PFirst = nil **then**

aDeq.PLast := PNew

**else**

aDeq.PFirst^.PPrev := PNew;

aDeq.PFirst := PNew;

**end**;

{Добавление элемента в конец дека.}

**procedure** PushL(**var** aDeq : TDeq; **const** aData : TData);

**var**

PNew : TPElem;

**begin**

**New**(PNew);

PNew^.Data := aData;

PNew^.PNext := nil;

PNew^.PPrev := aDeq.PLast;

**if** aDeq.PLast = nil **then**

aDeq.PFirst := PNew

**else**

aDeq.PLast^.PNext := PNew;

aDeq.PLast := PNew;

**end**;

**procedure** DWriteln(aDeq : TDeq);

**var**

PElem : TPElem;

LLi : Integer;

**begin**

PElem := aDeq.PFirst;

LLi := 0;

**while** PElem <> nil **do begin**

Inc(LLi);

**if** LLi > 1 **then** Write(', ');

Write(PElem^.Data);

PElem := PElem^.PNext;

**end**;

**if** LLi = 0 **then**

Writeln('No elemnts')

**else**

Writeln;

Writeln('press enter');

readln();

**end**;

**procedure** Free(**var** aDeq : TDeq);

**var**

Data : TData;

**begin**

**while** PopL(aDeq, Data) **do**;

**end**;

**function** PopL(**var** aDeq : TDeq; **var** aData : TData) : Boolean;

**var**

PDel : TPElem;

**begin**

PopL := False;

**if** aDeq.PFirst = nil **then Exit**;

PDel := aDeq.PLast;

aData := PDel^.Data;

aDeq.PLast := PDel^.PPrev;

**if** aDeq.PLast = nil **then**

aDeq.PFirst := nil

**else**

aDeq.PLast^.PNext := nil;

Dispose(PDel);

PopL := True;

**end**;

**Procedure** newcom;

**Begin**

**If** co^<20 **then begin**

co^:=co^+1;

com^[co^]:=d;

cl^:=co^;

**end**

**else begin**

co^:=1;

**Repeat**

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

co^:=co^+1;

**Until** co^=20;

com^[co^]:=d;

**end**;

**end**;

**Procedure** mov;

**Begin**

**if** co^=0 **then begin** s:='history is empty' ;j:=16; k:=32; **end**

**else**

**Case** com^[cl^] **of**

1:**begin** s:='add'; j:=3; k:=19;**end**;

2: **begin** s:='help';j:=4; k:=20;**end**;

3: **begin** s:='view';j:=4; k:=20;**end**;

4: **begin** s:='free';j:=4; k:=20;**end**;

5: **begin** s:='ladd';j:=4; k:=20;**end**;

6: **begin** s:='radd';j:=4; k:=20;**end**;

**end**;

**end**;

**Initialization**

**Begin**

**New**(com);

**New**(co);

co^:=0;

**New**(cl);

**end**;

**Finalization**

**Begin**

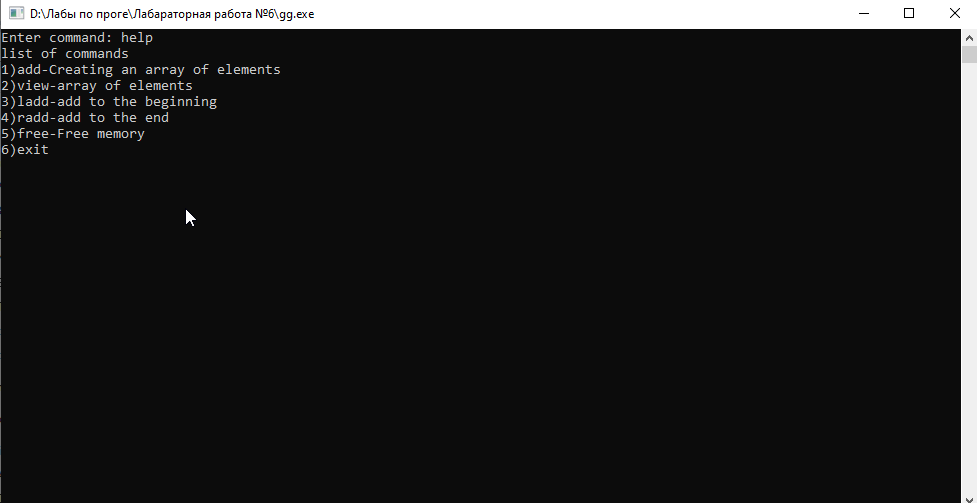
Dispose(com);

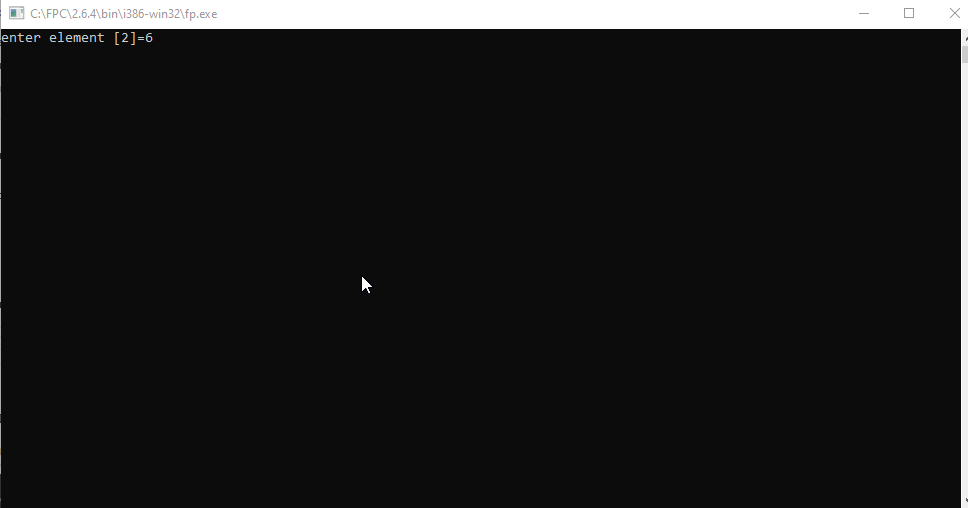
Dispose(co);

Dispose(cl);

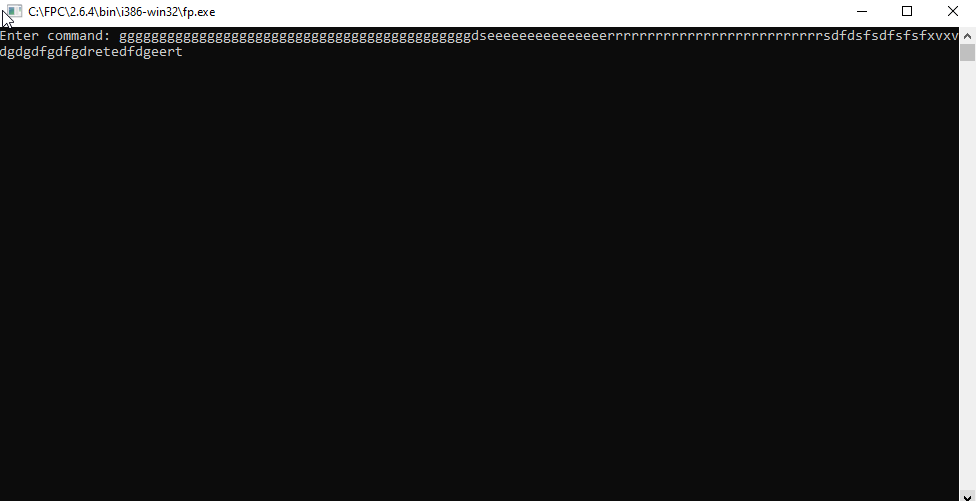
**end**;

**END**.

Фото отчет выполненой программы: 







Вывод: В ходе выполнение лабораторной работы по программированию были получены:

* Навыки работы с модулями
* Навыки работы с динамической памятью
* Навык работы с деком.

При выполнение лабораторной работы были выявлены сложности такие как:

* Работа с динамической памятью
* Работа с командной строкой и ее функциями.