ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

Кафедра Безопасные Информационные Технологии

Утверждаю

Кандидат технических наук, доцент

В. А. Безруков

“ “ 2013г.

**Отчёт по курсовой работе**

По курсу “Языки программирования”

По теме:

**«Строка»**

Исполнитель Студент группы № 1130  
 Валиев Артур Альбертович

подпись, дата

Санкт-Петербург

2013

Реферат

Отчёт 42стр.

Курсовая работа на языке С++

Цель работы – выполнение технического задания (программа на языке с++)

Дополнительные задания – переопределения операции индексации для строки идентификатора и функция перевода в верхний регистр на ассемблере

Ключевые слова: C++, WIN32 API, Строка, ООП, много файловая программа, ИТМО

Содержание

[Введение 4](#_Toc357263545)

[Основная часть 5](#_Toc357263546)

[Код программы: 5](#_Toc357263547)

[stroka.h: 5](#_Toc357263548)

[IdentStr.h:](#_Toc357263549) 10

[DesStr.h 1](#_Toc357263550)6

[main.cpp 2](#_Toc357263553)5

[down.asm](#_Toc357263554) 40

[Заключение](#_Toc357263556) 41

[Список литературы](#_Toc357263557) 42

# Введение

Цель данной курсовой работы – выполнение технического задания, а именно:

Написать тестовую программу, которая:

\* динамически выделяет массив указателей на базовый класс (4-6 шт.);

\* в режиме диалога заполняет этот массив указателями на производные классы,

при этом экземпляры производных классов создаются динамически с заданием

начальных значений;

\* для созданных экземпляров производных классов выполняется

проверка всех разработанных методов (в соответствии с вариантом задания),

с выводом исходных данных и результатов на дисплей.

\* для конструкторов копирования каждого класса предусмотреть

диагностическую печать количества его вызовов в заданное место дисплея.

Иерархия классов

Строка

Десятичная строка

Строка идентификатор

Для начала работы с программой, необходимо ввести число элементов, выбрав пункт меню Инициализация

Для тестирования выберите пункт меню Тестирование, задайте операнды и протестируйте интересующий вас метод класса.

Для завершения программы используйте пункт меню Выход.

# Основная часть

Код программы:

**stroka.h**

#ifndef HEADER\_INCLUDED

#include <stdio.h>

#include <string.h>

#include <conio.h>

#include <stdlib.h>

#include <windows.h>

#include <algorithm>

#define HEADER\_INCLUDED

#define \_CRT\_SECURE\_NO\_WARNINGS

HWND hMemo;

extern "C" void sum(char\*, char\*,char\*); // assembler

class stroka

{

protected:

char \*pCh;

int len;

public:

stroka();

stroka(const char ch);

stroka(const char \*s);

stroka(const stroka&);

virtual ~stroka();

int empty(void);

char\* GetStr(void) const { pCh[len] = '\0'; return pCh; }

int GetLen(void) const { return len; }

void Show(void) const;

char\* str(void);

operator int() {return len;}

operator char\*() {return pCh;}

friend stroka operator+(stroka,stroka);

friend stroka operator+(stroka,char\*);

friend stroka operator+(char\*,stroka);

friend stroka operator-(stroka,stroka);

friend stroka operator-(stroka,char\*);

friend stroka operator-(char\*,stroka);

friend stroka operator~(stroka &);

virtual void operator=(stroka);

char operator[](int);

virtual char\* ToDown(void);

virtual void Down(void);

stroka FindChar(char find);

virtual int Int(void);

virtual int CanInt(void);

};

stroka::stroka()

{

len=0;

pCh=new char [len+1];

pCh[0]='\0';

#ifndef \_DEBUG\_INFO

SendMessage (hMemo, EM\_REPLACESEL, 0, (LPARAM) ((LPSTR)L"Конструктор класса stroka::stroka()\r\n"));

#endif

}

stroka::stroka(const char ch)

{

len = 1;

pCh = new char [len + 1];

pCh [0] = ch;

pCh [1] = '\0';

#ifndef \_DEBUG\_INFO

SendMessage (hMemo, EM\_REPLACESEL, 0, (LPARAM) ((LPSTR)L"Конструктор класса stroka::Stroka(const char ch)\r\n"));

#endif

}

stroka::stroka(const char \*s)

{

pCh=new char[strlen(s)+1];

strcpy(pCh,s);

len=strlen(s);

#ifndef \_DEBUG\_INFO

SendMessage (hMemo, EM\_REPLACESEL, 0, (LPARAM) ((LPSTR)L"Конструктор класса stroka::stroka()\r\n"));

#endif

}

stroka::stroka(const stroka &from)

{

char \*tmpstr=from.GetStr();

int L=from.GetLen();

pCh=new char[L+1];

strcpy(pCh,tmpstr);

len=L;

#ifndef \_DEBUG\_INFO

SendMessage (hMemo, EM\_REPLACESEL, 0, (LPARAM) ((LPSTR)L"Конструктор класса stroka::stroka(const stroka& from)\r\n"));

#endif

}

stroka::~stroka()

{

if(len != 0)

delete []pCh;

len=0;

#ifndef \_DEBUG\_INFO

SendMessage (hMemo, EM\_REPLACESEL, 0, (LPARAM) ((LPSTR)L"Деструктор класса stroka::~stroka()\r\n"));

#endif

}

void stroka::Show(void) const

{

int i;

for (i=0;i<len;i++)

printf("%c",pCh[i]);

printf(" %d\n",len);

}

char\* stroka::str(void)

{

return(pCh);

}

int stroka::empty(void)

{

if (len==0) return(1); else return(0);

}

stroka operator+(stroka a,stroka b)

{

stroka temp;

temp.len=a.GetLen()+b.GetLen();

temp.pCh=new char[temp.len+1];

sum(a.pCh,b.pCh,temp.pCh);

return temp;

}

stroka operator+(stroka a,char\* b)

{

stroka temp;

temp.len=a.GetLen()+strlen(b);

temp.pCh=new char[temp.len+1];

sum(a.pCh,b,temp.pCh);

return temp;

}

stroka operator+ (char\* a,stroka b)

{

stroka temp;

temp.len=strlen(a)+b.GetLen();

temp.pCh=new char[temp.len+1];

sum(a,b.pCh,temp.pCh);

return(temp);

}

stroka operator-(stroka a,stroka b)

{

stroka temp1;

char \*temp=new char[a.len];

int length=0,i,j,proverka=1;

for (i=0;i<a.len;i++)

{

proverka=1;

for (j=0;j<b.len;j++)

if (a.pCh[i]==b.pCh[j]) {proverka=0;break;}

if (proverka==1) {

temp[length]=a.pCh[i];length++;proverka=0;}

}

temp1.pCh=new char[length+1];

for (i=0;i<length;i++) temp1.pCh[i]=temp[i];

temp1.len=length;

delete []temp;

return(temp1);

}

stroka operator-(stroka a,char\* b)

{

stroka temp1(b);

stroka temp=a-temp1;

return temp;

}

stroka operator-(char\* a,stroka b)

{

stroka temp1(a);

stroka temp=temp1-b;

return temp;

}

void stroka::operator=(stroka a)

{

len=a.GetLen();

delete []pCh;

pCh=new char[len+1];

strcpy(pCh,a.pCh);

return;

}

char stroka::operator[](int L)

{

L--;

try

{

if (L<0) throw 1;

if (L>=len) throw 2;

}

catch (int)

{

SendMessage (hMemo, EM\_REPLACESEL, 0, (LPARAM) ((LPSTR)L"ERROR!!!\r\n"));

return NULL;

}

return pCh[L];

}

stroka operator~(stroka &a)

{

stroka temp;

temp.len = a.len;

temp.pCh = new char[temp.len + 1];

strcpy(temp.pCh, a.pCh);

std::reverse(&temp.pCh[0], &temp.pCh[temp.len]);

return(temp);

}

char\* stroka::ToDown(void) {return(NULL);};

void stroka::Down(void) {};

int stroka::Int(void)

{

return(atoi(pCh));

};

int stroka::CanInt(void)

{

return(NULL);

};

stroka stroka::FindChar(char find)

{

int position = 500;

for(int i=strlen(pCh);i>=0;i--)

{

if(pCh[i] == find)

{

position = i;

break;

}

}

stroka temp;

int number=0,temp\_int=position;

while(temp>0)

{

number++;

temp\_int=temp\_int/10;

}

if(position!=500)

{

temp.len = number;

temp.pCh = new char[number + 6];

char\* b = new char[7];

sprintf(b,"Index\_%d",position+1);

strcpy(temp.pCh,b);

}

else

{

temp.len = 9;

temp.pCh = new char[temp.len + 1];

strcpy(temp.pCh,"Not found\0");

}

return(temp);

}

#endif

**IdentStr.h**

#ifdef HEADER\_INCLUDED

#include <stdio.h>

#include <string.h>

#include <conio.h>

#include <stdlib.h>

#include <windows.h>

#include "stroka.h"

#define HEADER\_INCLUDED

#endif

#define \_CRT\_SECURE\_NO\_WARNINGS

class IdentStr:public stroka

{

public:

IdentStr();

IdentStr(char \*);

IdentStr(char sI);

IdentStr(const IdentStr&);

~IdentStr();

char\* ToDown(void);

void Down(void);

void operator=(stroka);

friend IdentStr operator+(IdentStr&,IdentStr&);

friend IdentStr operator+(IdentStr&,char\*);

friend IdentStr operator+(char\*,IdentStr&);

friend IdentStr operator-(IdentStr&,IdentStr&);

friend IdentStr operator-(IdentStr&,char\*);

friend IdentStr operator-(char\*,IdentStr&);

friend IdentStr operator~(IdentStr &);

friend IdentStr operator>(const IdentStr&,const IdentStr&);

IdentStr FindChar(char find);

};

IdentStr::IdentStr():stroka()

{

#ifndef \_DEBUG\_INFO

SendMessage (hMemo, EM\_REPLACESEL, 0, (LPARAM) ((LPSTR)L"Конструктор произв. класса IdentStr::IdentStr():stroka()\r\n"));

#endif

}

IdentStr::IdentStr(char sI)

{

#ifndef \_DEBUG\_INFO

SendMessage (hMemo, EM\_REPLACESEL, 0, (LPARAM) ((LPSTR)L"Конструктор произв. класса IdentStr::IdentStr(char sI)\r\n"));

#endif

if (!(((sI>='a')&&(sI<='z'))||((sI>='A')&&(sI<='Z'))||((sI>='0')&&(sI<='9'))||(sI=='\_')))

{

len=0;

pCh=new char[len+1];

pCh[0]='\0';

return;

}

len=1;

pCh = new char[len+1];

pCh[0] = sI;

pCh[1] = '\0';

}

IdentStr::IdentStr(char \*sI)

{

int i=0;

if (strlen(sI)<20)

{

char proverka[20];

proverka[0]=' ';

int j=strlen(sI);

for (i=0;i<j;i++)

proverka[i+1]=sI[i];

proverka[strlen(sI)+1]=' ';

proverka[strlen(sI)+2]='\0';

//printf("\n\n-----------------\n--------%s---(%d)-----\n-------------%s----(%d)-----\n\n",proverka,strlen(proverka),sI,strlen(sI));

char \*sReserved=" asm auto bad\_cast bad\_typeid bool break case catch char class const const\_cast continue default delete do double dynamic\_cast else enum except explicit extern false finally float for friend goto if inline int int mutable namespace new operator private protected public register reinterpret\_cast return short signed sizeof static static\_cast struct switch template this throw true try type\_info typedef typeid typename union unsigned using virtual void volatile while ";

if (strstr(sReserved,proverka)!=NULL)

{

len=0;

pCh=new char[len+1];

pCh[0]='\0';

return;

}

}

#ifndef \_DEBUG\_INFO

SendMessage (hMemo, EM\_REPLACESEL, 0, (LPARAM) ((LPSTR)L"Конструктор произв. класса IdentStr::IdentStr(char \*sI)\r\n"));

#endif

if (!(((sI[0]>='a')&&(sI[0]<='z'))||((sI[0]>='A')&&(sI[0]<='Z'))||(sI[0]=='\_')))

{

len=0;

pCh=new char[len+1];

pCh[0]='\0';

return;

}

int j;

len=strlen(sI);

for (j=1;j<len;j++)

if (!(((sI[j]>='a')&&(sI[j]<='z'))||((sI[j]>='A')&&(sI[j]<='Z'))||((sI[j]>='0')&&(sI[j]<='9'))||(sI[j]=='\_')))

{

len=0;

pCh=new char[len+1];

pCh[0]='\0';

return;

}

len=0;

while (sI[len])len++;

pCh=new char[len+1];

for (int i=0;i<len;i++)

pCh[i]=sI[i];

pCh[len]='\0';

}

IdentStr::IdentStr(const IdentStr &from)

{

char \*tmpstr = from.GetStr();

int L=from.GetLen();

pCh=new char[L+1];

strcpy(pCh,tmpstr);

len=L;

#ifndef \_DEBUG\_INFO

SendMessage (hMemo, EM\_REPLACESEL, 0, (LPARAM) ((LPSTR)L"Конструктор произв. класса IdentStr::IdentStr(const IdentStr &from)\r\n"));

#endif

}

IdentStr::~IdentStr()

{

if(len != 0)

{

delete []pCh;

}

len=0;

#ifndef \_DEBUG\_INFO

SendMessage (hMemo, EM\_REPLACESEL, 0, (LPARAM) ((LPSTR)L"Деструктор произв. класса IdentStr::~IdentStr()\r\n"));

#endif

}

char\* IdentStr::ToDown(void)

{

char \*temp=new char[len];

int i;

for (i=0;i<len;i++)

{

if ((pCh[i]>='A')&&(pCh[i]<='Z')) temp[i]=char(pCh[i]+32); else temp[i]=pCh[i];

}

temp[len]='\0';

return temp;

delete [] temp;

}

void IdentStr::Down(void)

{

int i;

for (i=0;i<len;i++)

{

if ((pCh[i]>='A')&&(pCh[i]<='Z')) pCh[i]=char(pCh[i]+32);

}

}

void IdentStr::operator=(stroka a)

{

IdentStr temp(a.GetStr());

len=temp.GetLen();

delete []pCh;

pCh=new char[len+1];

strcpy(pCh,temp.pCh);

return;

}

IdentStr operator+(IdentStr &a,IdentStr &b)

{

int i;

int length=a.GetLen()+b.GetLen();

char \*str=new char[length];

for (i=0;i<a.len;i++)

str[i]=a.pCh[i];

for (i=0;i<b.len;i++)

str[i+a.len]=b.pCh[i];

str[length]='\0';

IdentStr temp(str);

return(temp);

}

IdentStr operator+(IdentStr &a,char\* b)

{

int i; int charlen=strlen(b);

int length=a.GetLen()+strlen(b);

char\* str=new char[length];

for (i=0;i<a.GetLen();i++)

str[i]=a.pCh[i];

for (i=0;i<charlen;i++)

str[i+a.len]=b[i];

str[length]='\0';

IdentStr temp(str);

return(temp);

}

IdentStr operator+ (char\* a,IdentStr &b)

{

int i; int charlen=strlen(a);

int length=b.GetLen()+strlen(a);

char\* str=new char[length];

for (i=0;i<charlen;i++)

str[i]=a[i];

for (i=0;i<b.GetLen();i++)

str[i+strlen(a)]=b.pCh[i];

str[length]='\0';

IdentStr temp(str);

return(temp);

}

IdentStr operator-(IdentStr &a,IdentStr &b)

{

if (b.GetLen()==0) return(a);

if (a.GetLen()==0) return(a);

char \*temp=new char[a.len];

int length=0,i,j,proverka=1;

for (i=0;i<a.len;i++)

{

proverka=1;

for (j=0;j<b.len;j++)

if (a.pCh[i]==b.pCh[j]) {proverka=0;break;}

if (proverka==1) {

temp[length]=a.pCh[i];length++;proverka=0;}

}

temp[length]='\0';

IdentStr temp1(temp);

delete []temp;

return(temp1);

}

IdentStr operator-(IdentStr &a,char\* b)

{

int charlen=strlen(b);

if (strlen(b)==0) return(a);

if (a.len==0) return(a);

char \*temp=new char[a.len];

int length=0,i,j,proverka=1;

for (i=0;i<a.len;i++)

{

proverka=1;

for (j=0;j<charlen;j++)

if (a.pCh[i]==b[j]) {proverka=0;break;}

if (proverka==1) {

temp[length]=a.pCh[i];length++;proverka=0;}

}

temp[length]='\0';

IdentStr temp1(temp);

delete []temp;

return(temp1);

}

IdentStr operator-(char\* a,IdentStr &b)

{

int charlen=strlen(a);

if (strlen(a)==0) return(a);

if (b.len==0) return(a);

char \*temp=new char[charlen];

int length=0,i,j,proverka=1;

for (i=0;i<charlen;i++)

{

proverka=1;

for (j=0;j<b.len;j++)

if (a[i]==b.pCh[j]) {proverka=0;break;}

if (proverka==1) {

temp[length]=a[i];length++;proverka=0;}

}

temp[length]='\0';

IdentStr temp1(temp);

delete []temp;

return(temp1);

}

IdentStr operator>(const IdentStr &a,const IdentStr &b)

{

IdentStr temp;

if( strcmp(a.pCh, b.pCh) > 0 )

{

temp.len = 6;

temp.pCh = new char[temp.len + 1];

strcpy(temp.pCh,"Bigger\0");

return(temp);

}

else if ( strcmp(a.pCh, b.pCh) == 0 )

{

temp.len = 5;

temp.pCh = new char[temp.len + 1];

strcpy(temp.pCh,"Equal\0");

return(temp);

}

temp.len = 6;

temp.pCh = new char[temp.len + 1];

strcpy(temp.pCh,"Less\0");

return(temp);

}

IdentStr operator~(IdentStr &a)

{

IdentStr temp;

temp.len = a.len;

temp.pCh = new char[temp.len + 1];

strcpy(temp.pCh, a.pCh);

std::reverse(&temp.pCh[0], &temp.pCh[temp.len]);

return(temp);

}

IdentStr IdentStr::FindChar(char find)

{

int position = 500;

for(int i=strlen(pCh);i>=0;i--)

{

if(pCh[i] == find)

{

position = i;

break;

}

}

IdentStr temp;

int number=0,temp\_int=position;

while(temp>0)

{

number++;

temp\_int=temp\_int/10;

}

if(position<500)

{

temp.len = number;

temp.pCh = new char[number + 6];

char\* b = new char[7];

sprintf(b,"Index\_%d",position+1);

strcpy(temp.pCh,b);

}

else

{

temp.len = 9;

temp.pCh = new char[temp.len + 1];

strcpy(temp.pCh,"Not found\0");

}

return(temp);

}

**DesStr.h**

#ifdef HEADER\_INCLUDED

#include <stdio.h>

#include <string.h>

#include <conio.h>

#include <stdlib.h>

#include <windows.h>

#include "stroka.h"

#define HEADER\_INCLUDED

#endif

#define \_CRT\_SECURE\_NO\_WARNINGS

class DesStr:public stroka

{

public:

DesStr();

DesStr(char \*);

DesStr(const DesStr&);

~DesStr();

operator int() { return(atoi(pCh));}

friend DesStr operator+(DesStr,DesStr);

friend DesStr operator+(int, DesStr);

friend DesStr operator+(DesStr,int);

friend DesStr operator+(char\*, DesStr);

friend DesStr operator+(DesStr,char\*);

friend DesStr operator-(DesStr,DesStr);

friend DesStr operator-(int, DesStr);

friend DesStr operator-(DesStr,int);

friend DesStr operator-(char\*, DesStr);

friend DesStr operator-(DesStr,char\*);

friend DesStr operator\*(DesStr,DesStr);

friend DesStr operator\*(int, DesStr);

friend DesStr operator\*(DesStr,int);

friend DesStr operator\*(char\*, DesStr);

friend DesStr operator\*(DesStr,char\*);

friend DesStr operator/(DesStr,DesStr);

friend DesStr operator/(int, DesStr);

friend DesStr operator/(DesStr,int);

friend DesStr operator/(char\*, DesStr);

friend DesStr operator/(DesStr,char\*);

friend DesStr operator%(DesStr,DesStr);

friend DesStr operator%(int, DesStr);

friend DesStr operator%(DesStr,int);

friend DesStr operator%(char\*, DesStr);

friend DesStr operator%(DesStr,char\*);

friend DesStr operator>(const DesStr &in1,const DesStr &in2);

void operator=(stroka);

int Int(void);

int CanInt(void);

};

DesStr::DesStr():stroka()

{

#ifndef \_DEBUG\_INFO

SendMessage (hMemo, EM\_REPLACESEL, 0, (LPARAM) ((LPSTR)L"Конструктор произв. класса DesStr::DesStr():stroka()\r\n"));

#endif

}

DesStr::DesStr(char \*sD)

{

#ifndef \_DEBUG\_INFO

SendMessage (hMemo, EM\_REPLACESEL, 0, (LPARAM) ((LPSTR)L"Конструктор произв. класса DesStr::DesStr(char \*sD)\r\n"));

#endif

if ((sD[0]=='+')||(sD[0]=='-')||((sD[0]>='0')&&(sD[0]<='9')))

{

len=strlen(sD);

for (int i=1;i<len;i++)

if (!((sD[i]>='0')&&(sD[i]<='9')))

{

len=0;

pCh=new char[len+1];

pCh[0]='\0';

return;

}

pCh=new char[len+1];

lstrcpyA(pCh,sD);

}

else

{

len=0;

pCh=new char[len+1];

pCh[0]='\0';

return;

}

}

DesStr::DesStr(const DesStr &from)

{

char \*tmpstr = from.GetStr();

int L=from.GetLen();

pCh=new char[L+1];

lstrcpyA(pCh,tmpstr);

len=L;

#ifndef \_DEBUG\_INFO

SendMessage (hMemo, EM\_REPLACESEL, 0, (LPARAM) ((LPSTR)L"Конструктор произв. класса DesStr::DesStr(const DesStr &from)\r\n"));

#endif

}

DesStr::~DesStr()

{

if (len!=0)

delete []pCh;

len=0;

#ifndef \_DEBUG\_INFO

SendMessage (hMemo, EM\_REPLACESEL, 0, (LPARAM) ((LPSTR)L"Деструктор произв. класса DesStr::~DesStr()\r\n"));

#endif

}

void DesStr::operator=(stroka a)

{

char \*sD=a.GetStr();

delete []pCh;

if ((sD[0]=='+')||(sD[0]=='-')||((sD[0]>='0')&&(sD[0]<='9')))

{

len=a.GetLen();

pCh=new char[len+1];

for (int i=1;i<len;i++)

if (!((sD[i]>='0')&&(sD[i]<='9')))

{

len=0;

pCh=new char[len+1];

pCh[0]='\0';

return;

}

lstrcpyA(pCh,sD);

}

else

{

len=0;

pCh=new char[len+1];

pCh[0]='\0';

return;

}

return;

}

DesStr operator+(DesStr a,DesStr b)

{

int tempint=a.Int()+b.Int();

char word[50];

sprintf(word, "%d", tempint);

DesStr temp;

temp.len=strlen(word);

temp.pCh=new char[strlen(word)+1];

strcpy(temp.pCh,word);

return(temp);

}

DesStr operator+ (DesStr a,int b)

{

int tempint=a.Int()+b;

char word[50];

sprintf(word, "%d", tempint);

DesStr temp;

temp.len=strlen(word);

temp.pCh=new char[strlen(word)+1];

strcpy(temp.pCh,word);

return(temp);

}

DesStr operator+ (int a,DesStr b)

{

int tempint=a+b.Int();

char word[50];

sprintf(word, "%d", tempint);

DesStr temp;

temp.len=strlen(word);

temp.pCh=new char[strlen(word)+1];

strcpy(temp.pCh,word);

return(temp);

}

DesStr operator+ (DesStr a,char\* b)

{

int tempint=a.Int()+atoi(b);

char word[50];

sprintf(word, "%d", tempint);

DesStr temp;

temp.len=strlen(word);

temp.pCh=new char[strlen(word)+1];

strcpy(temp.pCh,word);

return(temp);

}

DesStr operator+ (char\* a,DesStr b)

{

int tempint=atoi(a)+b.Int();

char word[50];

sprintf(word, "%d", tempint);

DesStr temp;

temp.len=strlen(word);

temp.pCh=new char[strlen(word)+1];

strcpy(temp.pCh,word);

return(temp);

}

DesStr operator-(DesStr a,DesStr b)

{

int tempint=a.Int()-b.Int();

char word[50];

sprintf(word, "%d", tempint);

DesStr temp;

temp.len=strlen(word);

temp.pCh=new char[strlen(word)+1];

strcpy(temp.pCh,word);

return(temp);

}

DesStr operator- (DesStr a,int b)

{

int tempint=a.Int()-b;

char word[50];

sprintf(word, "%d", tempint);

DesStr temp;

temp.len=strlen(word);

temp.pCh=new char[strlen(word)+1];

strcpy(temp.pCh,word);

return(temp);

}

DesStr operator- (int a,DesStr b)

{

int tempint=a-b.Int();

char word[50];

sprintf(word, "%d", tempint);

DesStr temp;

temp.len=strlen(word);

temp.pCh=new char[strlen(word)+1];

strcpy(temp.pCh,word);

return(temp);

}

DesStr operator- (DesStr a,char\* b)

{

int tempint=a.Int()-atoi(b);

char word[50];

sprintf(word, "%d", tempint);

DesStr temp;

temp.len=strlen(word);

temp.pCh=new char[strlen(word)+1];

strcpy(temp.pCh,word);

return(temp);

}

DesStr operator- (char\* a,DesStr b)

{

int tempint=atoi(a)-b.Int();

char word[50];

sprintf(word, "%d", tempint);

DesStr temp;

temp.len=strlen(word);

temp.pCh=new char[strlen(word)+1];

strcpy(temp.pCh,word);

return(temp);

}

DesStr operator\*(DesStr a,DesStr b)

{

int tempint=a.Int()\*b.Int();

char word[50];

sprintf(word, "%d", tempint);

DesStr temp;

temp.len=strlen(word);

temp.pCh=new char[strlen(word)+1];

strcpy(temp.pCh,word);

return(temp);

}

DesStr operator\* (DesStr a,int b)

{

int tempint=a.Int()\*b;

char word[50];

sprintf(word, "%d", tempint);

DesStr temp;

temp.len=strlen(word);

temp.pCh=new char[strlen(word)+1];

strcpy(temp.pCh,word);

return(temp);

}

DesStr operator\* (int a,DesStr b)

{

int tempint=a\*b.Int();

char word[50];

sprintf(word, "%d", tempint);

DesStr temp;

temp.len=strlen(word);

temp.pCh=new char[strlen(word)+1];

strcpy(temp.pCh,word);

return(temp);

}

DesStr operator\* (DesStr a,char\* b)

{

int tempint=a.Int()\*atoi(b);

char word[50];

sprintf(word, "%d", tempint);

DesStr temp;

temp.len=strlen(word);

temp.pCh=new char[strlen(word)+1];

strcpy(temp.pCh,word);

return(temp);

}

DesStr operator\* (char\* a,DesStr b)

{

int tempint=atoi(a)\*b.Int();

char word[50];

sprintf(word, "%d", tempint);

DesStr temp;

temp.len=strlen(word);

temp.pCh=new char[strlen(word)+1];

strcpy(temp.pCh,word);

return(temp);

}

DesStr operator/(DesStr a,DesStr b)

{

int tempint=a.Int()/b.Int();

char word[50];

sprintf(word, "%d", tempint);

DesStr temp;

temp.len=strlen(word);

temp.pCh=new char[strlen(word)+1];

strcpy(temp.pCh,word);

return(temp);

}

DesStr operator/ (DesStr a,int b)

{

int tempint=a.Int()/b;

char word[50];

sprintf(word, "%d", tempint);

DesStr temp;

temp.len=strlen(word);

temp.pCh=new char[strlen(word)+1];

strcpy(temp.pCh,word);

return(temp);

}

DesStr operator/ (int a,DesStr b)

{

int tempint=a/b.Int();

char word[50];

sprintf(word, "%d", tempint);

DesStr temp;

temp.len=strlen(word);

temp.pCh=new char[strlen(word)+1];

strcpy(temp.pCh,word);

return(temp);

}

DesStr operator/ (DesStr a,char\* b)

{

int tempint=a.Int()/atoi(b);

char word[50];

sprintf(word, "%d", tempint);

DesStr temp;

temp.len=strlen(word);

temp.pCh=new char[strlen(word)+1];

strcpy(temp.pCh,word);

return(temp);

}

DesStr operator/ (char\* a,DesStr b)

{

int tempint=atoi(a)/b.Int();

char word[50];

sprintf(word, "%d", tempint);

DesStr temp;

temp.len=strlen(word);

temp.pCh=new char[strlen(word)+1];

strcpy(temp.pCh,word);

return(temp);

}

DesStr operator%(DesStr a,DesStr b)

{

int tempint=a.Int()%b.Int();

char word[50];

sprintf(word, "%d", tempint);

DesStr temp;

temp.len=strlen(word);

temp.pCh=new char[strlen(word)+1];

strcpy(temp.pCh,word);

return(temp);

}

DesStr operator% (DesStr a,int b)

{

int tempint=a.Int()%b;

char word[50];

sprintf(word, "%d", tempint);

DesStr temp;

temp.len=strlen(word);

temp.pCh=new char[strlen(word)+1];

strcpy(temp.pCh,word);

return(temp);

}

DesStr operator% (int a,DesStr b)

{

int tempint=a%b.Int();

char word[50];

sprintf(word, "%d", tempint);

DesStr temp;

temp.len=strlen(word);

temp.pCh=new char[strlen(word)+1];

strcpy(temp.pCh,word);

return(temp);

}

DesStr operator% (DesStr a,char\* b)

{

int tempint=a.Int()%atoi(b);

char word[50];

sprintf(word, "%d", tempint);

DesStr temp;

temp.len=strlen(word);

temp.pCh=new char[strlen(word)+1];

strcpy(temp.pCh,word);

return(temp);

}

DesStr operator% (char\* a,DesStr b)

{

int tempint=atoi(a)%b.Int();

char word[50];

sprintf(word, "%d", tempint);

DesStr temp;

temp.len=strlen(word);

temp.pCh=new char[strlen(word)+1];

strcpy(temp.pCh,word);

return(temp);

}

DesStr operator>(const DesStr &in1, const DesStr &in2)

{

DesStr temp;

int a;

if(atoi(in1.pCh) >= atoi(in2.pCh) )

{

temp.len = 1;

temp.pCh = new char[2];

strcpy(temp.pCh,"1\0");

return(temp);

}

else

{

temp.len = 1;

temp.pCh = new char[2];

strcpy(temp.pCh, "0\0");

return(temp);

}

}

int DesStr::Int()

{

return(atoi(pCh));

}

int DesStr::CanInt()

{

if (len>=12) return(0);

if (len<=9) return(1);

char MAX1[11]="2147483647",MAX2[12]="+2147483647",MIN[12]="-2147483648";

if ((!strcmp(MAX1,pCh))||(!strcmp(MAX2,pCh))||(!strcmp(MIN,pCh))) return(1);

int max=2147483647,min=2147483648;

if ((DesStr::Int()==max)||(DesStr::Int()==-min)) return(0); else return(1);

}

**main.cpp**

#include <windows.h>

#include <stdlib.h>

#include <string.h>

#include <tchar.h>

#include <commctrl.h>

#include <stdio.h>

#include "stroka.h"

#include "IdentStr.h"

#include "DesStr.h"

#define \_CRT\_SECURE\_NO\_WARNINGS

#define \_CRT\_NON\_CONFORMING\_SWPRINTFS

#define HEADER\_INCLUDED

#define UNICOD

#define ID\_INI\_NUMBER 40001

#define ID\_INI\_START 40002

#define ID\_TEST\_STRING 40003

#define ID\_TEST\_IDEN 40004

#define ID\_TEST\_DES 40005

#define ID\_TEST\_OPER 40006

#define ID\_TEST\_STRING\_FOOL 40007

#define ID\_TEST\_IDEN\_REG 40008

#define ID\_TEST\_DES\_INT 40009

#pragma comment(lib,"comctl32.lib")

HMENU hIni,hTest,hMenu,hStroka,hIdent,hDes;

int Esc=0;

static TCHAR szWindowClass[] = \_T("Stroka[winapi]");

static TCHAR szTitle[] = \_T("Курсовая работа");

HINSTANCE hInst;

UINT wId;

HWND hStatusWindow;

int i=0;

//ИНТЕРФЕЙС

HWND hText1,hButton1,hEdit1,hGroup1; //начальные значения

HWND \*hTextArray1,\*hEditArray1,\*hStatus,\*hAnswer,hInfo[10],hBox1; //после ввода N

HWND hBox; //КОНСТРУКТОРЫ ДЕСТРУКТОРЫ (окно)

HWND hText2,hEdit2,hComboBox2,hButton2,hBox2,hClose2,hButtonClear; //ИНИЦИАЛИЗАЦИЯ; //инициализация

HWND hText3[4],hEdit3[5],hBox3,hClose3,hOAnswer,hButton3; //операнды

wchar\_t string[100];wchar\_t msg[100];wchar\_t buff1[100];wchar\_t buff2[100];wchar\_t tempstring[100];

char \*oper1,\*oper2;char buffO[100];char cbuff[100];int otstup=30;

int otsupY=0;int Nmax=0;int update=1;int init=0,oper=0;

int Answer=0;int PrewParam=0;int OAnswer=0;

int Operand1,Operand2,Operand3;int RW=0;int first=0;int mode=0; //класс+класс

//ТЕСТИРОВАНИЕ

stroka \*\*Stroka; //массив указателей на базовый класс

int \*klass;

char\*\* input;

int N=0,Pos=0;

int current=1;

LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);

INT\_PTR CALLBACK InitProc(HWND hDlg,UINT message, WPARAM wParam,LPARAM lParam);

int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nCmdShow)

{

WNDCLASSEX wcex;

wcex.cbSize = sizeof(WNDCLASSEX);

wcex.style = CS\_VREDRAW; //CS\_HREDRAW |

wcex.lpfnWndProc = WndProc;

wcex.cbClsExtra = 0;

wcex.cbWndExtra = 0;

wcex.hInstance = hInstance;

wcex.hIcon = LoadIcon(hInstance, MAKEINTRESOURCE(IDI\_APPLICATION));

wcex.hCursor = LoadCursor(NULL, IDC\_ARROW);

wcex.hbrBackground = (HBRUSH)(DC\_BRUSH);//CreateSolidBrush(RGB(22,121,242));//(HBRUSH)(COLOR\_WINDOW+1);

wcex.lpszMenuName = NULL;

wcex.lpszClassName = szWindowClass;

wcex.hIconSm = LoadIcon(wcex.hInstance, MAKEINTRESOURCE(IDI\_APPLICATION));

if (!RegisterClassEx(&wcex))

{

MessageBox(NULL, \_T("Call to RegisterClassEx failed!"), \_T("Win32 Guided Tour"), NULL);

return 1;

}

hInst = hInstance;

HWND hWnd = CreateWindow( //главное окошко

szWindowClass,

szTitle,

//WS\_OVERLAPPEDWINDOW|

CS\_HREDRAW|CS\_VREDRAW|WS\_SYSMENU|WS\_VSCROLL,

//WS\_OVERLAPPED | WS\_SYSMENU,// | WS\_MINIMIZEBOX, // WS\_OVERLAPPEDWINDOW, //СТИЛЬ

0,140,//CW\_USEDEFAULT, CW\_USEDEFAULT, //относительно экрана/родителя

1300, 510, //размер окна

NULL, //дескриптор родителя

NULL, //hMenu

hInstance,

NULL

);

if (!hWnd)

{

MessageBox(NULL, \_T("Call to CreateWindow failed!"), \_T("Win32 Guided Tour"), MB\_ICONEXCLAMATION);

return 1;

}

hStatusWindow=CreateStatusWindow(WS\_CHILD|WS\_VISIBLE, (LPTSTR)(L"Валиев Артур БИТ 1130"),hWnd, wId);

ShowWindow(hWnd, nCmdShow);

UpdateWindow(hWnd);

//МЕНЮ

hIni=CreatePopupMenu();

hTest=CreatePopupMenu();

hMenu=CreateMenu();

hStroka=CreateMenu();

hIdent=CreateMenu();

hDes=CreateMenu();

AppendMenu(hIni,MF\_ENABLED|MFT\_STRING,ID\_INI\_NUMBER,\_T("Число элементов"));

AppendMenu(hIni,MF\_GRAYED|MFT\_STRING,ID\_INI\_START,\_T("Проинициализировать"));

AppendMenu(hStroka,MF\_ENABLED|MFT\_STRING,ID\_TEST\_STRING\_FOOL,\_T("Пуста ли строка?"));

AppendMenu(hStroka,MF\_ENABLED|MFT\_STRING,1001,\_T("Сложение"));

AppendMenu(hStroka,MF\_ENABLED|MFT\_STRING,1002,\_T("Вычитание"));

AppendMenu(hStroka,MF\_ENABLED|MFT\_STRING,1003,\_T("[ ]"));

AppendMenu(hStroka,MF\_ENABLED|MFT\_STRING,1005,\_T("String Reverse"));

AppendMenu(hIdent,MF\_ENABLED|MFT\_STRING,ID\_TEST\_IDEN\_REG,\_T("Перевод в нижний регистр"));

AppendMenu(hIdent,MF\_ENABLED|MFT\_STRING,2001,\_T("Сложение"));

AppendMenu(hIdent,MF\_ENABLED|MFT\_STRING,2002,\_T("Вычитание"));

AppendMenu(hIdent,MF\_ENABLED|MFT\_STRING,2003,\_T("[ ]"));

AppendMenu(hIdent,MF\_ENABLED|MFT\_STRING,2004,\_T("Проверка на больше"));

AppendMenu(hIdent,MF\_ENABLED|MFT\_STRING,2005,\_T("Поиск символа"));

AppendMenu(hIdent,MF\_ENABLED|MFT\_STRING,2006,\_T("String Reverse"));

AppendMenu(hDes,MF\_ENABLED|MFT\_STRING,ID\_TEST\_DES\_INT,\_T("Можно ли представить в формате int?"));

AppendMenu(hDes,MF\_ENABLED|MFT\_STRING,3001,\_T("Сложение"));

AppendMenu(hDes,MF\_ENABLED|MFT\_STRING,3002,\_T("Вычитание"));

AppendMenu(hDes,MF\_ENABLED|MFT\_STRING,3003,\_T("Умножение"));

AppendMenu(hDes,MF\_ENABLED|MFT\_STRING,3004,\_T("Деление"));

// AppendMenu(hDes,MF\_ENABLED|MFT\_STRING,3005,\_T("Остаток от деления"));

AppendMenu(hDes,MF\_ENABLED|MFT\_STRING,3006,\_T("[ ]"));

AppendMenu(hDes,MF\_ENABLED|MFT\_STRING,3007,\_T("Проверка на больше"));

AppendMenu(hTest,MF\_ENABLED|MF\_POPUP,(UINT\_PTR)hStroka,\_T("Строка"));

AppendMenu(hTest,MF\_ENABLED|MF\_POPUP,(UINT\_PTR)hIdent,\_T("Строка идентификатор"));

AppendMenu(hTest,MF\_ENABLED|MF\_POPUP,(UINT\_PTR)hDes,\_T("Десятичная строка"));

AppendMenu(hTest,MF\_ENABLED|MF\_POPUP,1004,\_T("Присваивание"));

AppendMenu(hTest,MF\_ENABLED|MFT\_STRING,ID\_TEST\_OPER,\_T("Задать операнд"));

AppendMenu(hMenu,MF\_ENABLED|MF\_POPUP,(UINT\_PTR)hIni,\_T("Инициализация"));

AppendMenu(hMenu,MF\_GRAYED|MF\_POPUP,(UINT\_PTR)hTest,\_T("Тестирование"));

AppendMenu(hMenu,MF\_ENABLED|MF\_POPUP,(UINT\_PTR)12,\_T("О программе"));

AppendMenu(hMenu,MF\_ENABLED|MF\_POPUP,(UINT\_PTR)11,\_T("Выход"));

SetMenu(hWnd,hMenu);

// Main message loop:

MSG msg;

while (GetMessage(&msg, NULL, 0, 0))

{

TranslateMessage(&msg); //клавиатура

DispatchMessage(&msg); //

}

return (int) msg.wParam;

}

void Operand(int kl,int operation) //ОПЕРАЦИИ НАД ОПЕРАНДАМИ

{

mode=0;

GetWindowTextA(hEdit3[0], buffO, 100);

if (buffO[0]=='"')

{

oper1=new char[strlen(buffO)+1];

mode=mode+1;

for (i=0;i<(int)strlen(buffO)-1;i++) oper1[i]=buffO[i+1];

oper1[strlen(buffO)-2]='\0';

}

Operand1=atoi(buffO);

if (klass[Operand1-1]==0)

{

SetWindowText(hOAnswer,L"Ошибка. Один из операндов не определён!");

return;

}

GetWindowTextA(hEdit3[1], buffO, 100);

if (buffO[0]=='"')

{

oper2=new char[strlen(buffO)];

mode=mode+2;

for (i=0;i<(int)strlen(buffO)-1;i++) oper2[i]=buffO[i+1];

oper2[strlen(buffO)-2]='\0';

}

Operand2=atoi(buffO);

if (((operation==3)&&(kl!=3))||((operation==6)&&(kl==3))) {} else

if ((kl!=1)&&(operation!=4)) if (klass[Operand2-1]==0)

{

SetWindowText(hOAnswer,L"Ошибка. Один из операндов не определён!");

return;

}

GetWindowText(hEdit3[2], string, 100);

Operand3=\_wtoi(string);

if (wcscmp(string,L"")!=0) RW=1;

else Operand3=N+1;

if (klass[Operand3-1]==0)

{

Stroka[Operand3-1]=new stroka();

klass[Operand3-1]=1;

}

if (RW==1) if (Operand3-1>N)

{

SetWindowText(hOAnswer,L"Ошибка. Операнда с таким номером не существует!");

return;

}

switch (kl)

{

case 1:

switch (operation)

{

case 1: switch (mode) {

case 0:\*Stroka[Operand3-1]=\*Stroka[Operand1-1]+\*Stroka[Operand2-1];break;

case 1:\*Stroka[Operand3-1]=oper1+\*Stroka[Operand2-1];break;

case 2:\*Stroka[Operand3-1]=\*Stroka[Operand1-1]+oper2;break;

case 3:\*Stroka[Operand3-1]=(stroka)oper1+oper2;break;

}break;

case 2:switch (mode) {

case 0:\*Stroka[Operand3-1]=\*Stroka[Operand1-1]-\*Stroka[Operand2-1];break;

case 1:\*Stroka[Operand3-1]=oper1-\*Stroka[Operand2-1];break;

case 2:\*Stroka[Operand3-1]=\*Stroka[Operand1-1]-oper2;break;

case 3:\*Stroka[Operand3-1]=(stroka)oper1-oper2;break;

}break;

case 3:switch (mode) {

case 0:\*Stroka[Operand3-1]=(\*Stroka[Operand1-1])[Operand2];break;

case 1:SetWindowText(hOAnswer,L"Ошибка. 1-ый операнд не правильный!");return; break;

case 2:SetWindowText(hOAnswer,L"Ошибка. 2-ой операнд не правильный!");return; break;

case 3:SetWindowText(hOAnswer,L"Ошибка. Операнды не правильные!");return; break;

}break;

case 4:SetWindowText(hEdit3[1],L"Операнд не используется!");

switch (mode) {

case 0:\*Stroka[Operand3-1]=\*Stroka[Operand1-1];break;

case 1:\*Stroka[Operand3-1]=oper1;break;

case 2:\*Stroka[Operand3-1]=\*Stroka[Operand1-1];break;

case 3:\*Stroka[Operand3-1]=oper1;break;

}break;

case 5:SetWindowText(hEdit3[1],L"Операнд не используется!");

switch (mode) {

case 0:\*Stroka[Operand3-1]=~\*Stroka[Operand1-1];break;

}break;

}

break;

case 2: switch (operation)

{

case 1: switch (mode) {

case 0:\*Stroka[Operand3-1]=(IdentStr)\*Stroka[Operand1-1]+(IdentStr)\*Stroka[Operand2-1];break;

case 1:\*Stroka[Operand3-1]=oper1+(IdentStr)\*Stroka[Operand2-1];break;

case 2:\*Stroka[Operand3-1]=(IdentStr)\*Stroka[Operand1-1]+oper2;break;

case 3:\*Stroka[Operand3-1]=(IdentStr)oper1+oper2;break;

}break;

case 2:switch (mode) {

case 0:\*Stroka[Operand3-1]=(IdentStr)\*Stroka[Operand1-1]-(IdentStr)\*Stroka[Operand2-1];break;

case 1:\*Stroka[Operand3-1]=oper1-(IdentStr)\*Stroka[Operand2-1];break;

case 2:\*Stroka[Operand3-1]=(IdentStr)\*Stroka[Operand1-1]-oper2;break;

case 3:\*Stroka[Operand3-1]=(IdentStr)oper1-oper2;break;

}break;

case 3:switch (mode) {

case 0:\*Stroka[Operand3-1]=(\*Stroka[Operand1-1])[Operand2];break;

case 1:SetWindowText(hOAnswer,L"Ошибка. 1-ый операнд не правильный!");return; break;

case 2:SetWindowText(hOAnswer,L"Ошибка. 2-ой операнд не правильный!");return; break;

case 3:SetWindowText(hOAnswer,L"Ошибка. Операнды не правильные!");return; break;

}break;

case 4: switch(mode){

case 0: \*Stroka[Operand3-1]=(IdentStr)\*Stroka[Operand1-1]>(IdentStr)\*Stroka[Operand2-1];break;

}break;

case 5: switch(mode){

case 0: \*Stroka[Operand3-1]=(IdentStr)Stroka[Operand1-1]->FindChar('a');break;

}break;

case 6:SetWindowText(hEdit3[1],L"Операнд не используется!");

switch(mode){

case 0: \*Stroka[Operand3-1]=~\*Stroka[Operand1-1];break;

}break;

}

break;

case 3: switch (operation)

{

case 1:switch (mode) {

case 0:\*Stroka[Operand3-1]=(DesStr)\*Stroka[Operand1-1]+(DesStr)\*Stroka[Operand2-1];break;

case 1:\*Stroka[Operand3-1]=oper1+(DesStr)\*Stroka[Operand2-1];break;

case 2:\*Stroka[Operand3-1]=(DesStr)\*Stroka[Operand1-1]+oper2;break;

case 3:\*Stroka[Operand3-1]=(DesStr)oper1+oper2;break;

}break;

case 2:switch (mode) {

case 0:\*Stroka[Operand3-1]=(DesStr)\*Stroka[Operand1-1]-(DesStr)\*Stroka[Operand2-1];break;

case 1:\*Stroka[Operand3-1]=oper1-(DesStr)\*Stroka[Operand2-1];break;

case 2:\*Stroka[Operand3-1]=(DesStr)\*Stroka[Operand1-1]-oper2;break;

case 3:\*Stroka[Operand3-1]=(DesStr)oper1-oper2;break;

}break;

case 3:switch (mode) {

case 0:\*Stroka[Operand3-1]=(DesStr)\*Stroka[Operand1-1]\*(DesStr)\*Stroka[Operand2-1];break;

case 1:\*Stroka[Operand3-1]=oper1\*(DesStr)\*Stroka[Operand2-1];break;

case 2:\*Stroka[Operand3-1]=(DesStr)\*Stroka[Operand1-1]\*oper2;break;

case 3:\*Stroka[Operand3-1]=(DesStr)oper1\*oper2;break;

}break;

case 4:switch (mode) {

case 0:\*Stroka[Operand3-1]=(DesStr)\*Stroka[Operand1-1]/(DesStr)\*Stroka[Operand2-1];break;

case 1:\*Stroka[Operand3-1]=oper1/(DesStr)\*Stroka[Operand2-1];break;

case 2:\*Stroka[Operand3-1]=(DesStr)\*Stroka[Operand1-1]/oper2;break;

case 3:\*Stroka[Operand3-1]=(DesStr)oper1/oper2;break;

}break;

case 5:switch (mode) {

case 0:\*Stroka[Operand3-1]=(DesStr)\*Stroka[Operand1-1]%(DesStr)\*Stroka[Operand2-1];break;

case 1:\*Stroka[Operand3-1]=oper1%(DesStr)\*Stroka[Operand2-1];break;

case 2:\*Stroka[Operand3-1]=(DesStr)\*Stroka[Operand1-1]%oper2;break;

case 3:\*Stroka[Operand3-1]=(DesStr)oper1%oper2;break;

}break;

case 6:switch (mode) {

case 0:\*Stroka[Operand3-1]=(\*Stroka[Operand1-1])[Operand2];break;

case 1:SetWindowText(hOAnswer,L"Ошибка. 1-ый операнд не правильный!");return; break;

case 2:SetWindowText(hOAnswer,L"Ошибка. 2-ой операнд не правильный!");return; break;

case 3:SetWindowText(hOAnswer,L"Ошибка. Операнды не правильные!");return; break;

}break;

case 7:switch(mode){

case 0:\*Stroka[Operand3-1]=operator>((DesStr)\*Stroka[Operand1-1],(DesStr)\*Stroka[Operand2-1]);break;

}break;

}

break;

}

SetWindowTextA(hOAnswer,Stroka[Operand3-1]->GetStr());

if (RW==1)

{

if ((Operand3-1-Pos>=0)&&(Operand3-1-Pos<=15))

{

SetWindowTextA(hEditArray1[Operand3-1-Pos],Stroka[Operand3-1]->GetStr());

switch (klass[Operand3-1])

{

case 0:swprintf(msg,L" ");break;

case 1:swprintf(msg,L"stroka");break;

case 3:swprintf(msg,L"IdentStr");break;

case 2:swprintf(msg,L"DesStr");break;

}

SetWindowText(hStatus[Operand3-1-Pos],msg);

}

} else RW=0;

}

LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)

{

PAINTSTRUCT ps;

HDC hdc;

switch (message)

{

case WM\_PAINT:

hdc = BeginPaint(hWnd, &ps);

SetScrollRange(hWnd,SB\_VERT,0,N-Nmax,TRUE);

SetTextColor(hdc, RGB(254, 254, 254));

SetBkColor(hdc, RGB(22,121,242));

EndPaint(hWnd, &ps);

break;

case WM\_CLOSE:

if (Esc==1)

{if (MessageBox(hWnd,\_T("Вы уверены, что хотите выйти из приложения?"),\_T("Выход"), MB\_OKCANCEL | MB\_ICONQUESTION)==IDOK)

DestroyWindow(hWnd); else Esc=0;}

else DestroyWindow(hWnd);

break;

case WM\_KEYDOWN:

switch(wParam)

{

case VK\_ESCAPE:

Esc=1;

SendMessage(hWnd,WM\_CLOSE,0,0);

default: Esc=2;SendMessage(hWnd,WM\_PAINT,0,0);

break;

}break;

case WM\_DESTROY:

PostQuitMessage(0);

break;

case WM\_VSCROLL: //ПРОКРУТКА

update=1;

wchar\_t tempbuff[50];

switch(LOWORD(wParam))

{

case SB\_LINEDOWN: if (Pos<N-Nmax) {Pos++;} else update=0; break;

case SB\_LINEUP: if (Pos>0) {Pos--;} else update=0;break;

case SB\_THUMBTRACK:

Pos=HIWORD(wParam);break;

}

SetScrollPos(hWnd,SB\_VERT,Pos,TRUE);

if (update==1)

for (i=0;i<Nmax;i++)

{

swprintf(msg,L"%d",i+Pos+1);

SetWindowText(hTextArray1[i],msg);

if (klass[i+Pos]!=0) SetWindowTextA(hEditArray1[i],Stroka[i+Pos]->GetStr());

else SetWindowTextA(hEditArray1[i],"");

switch (klass[i+Pos])

{

case 0:swprintf(msg,L" ");break;

case 1:swprintf(msg,L"stroka");break;

case 3:swprintf(msg,L"IdentStr");break;

case 2:swprintf(msg,L"DesStr");break;

}

GetWindowText(hStatus[i],tempbuff,50);

if (wcscmp(tempbuff,msg)!=0)

SetWindowText(hStatus[i],msg);

if (Answer==1)

{

GetWindowText(hAnswer[i],tempbuff,50);

if (klass[i+Pos]==0) swprintf(msg,L""); else if (Stroka[i+Pos]->empty()==1) swprintf(msg,L"Да"); else swprintf(msg,L"Нет");

if (wcscmp(tempbuff,msg)!=0)

SetWindowText(hAnswer[i],msg);

}

else

if (Answer==2)

{

GetWindowText(hAnswer[i],tempbuff,50);

if (klass[i+Pos]!=2) swprintf(msg,L""); else if (Stroka[i+Pos]->CanInt()==1) swprintf(msg,L"Да"); else swprintf(msg,L"Нет");

if (wcscmp(tempbuff,msg)!=0)

SetWindowText(hAnswer[i],msg);

}

}

break;

break;

case WM\_MOUSEWHEEL:

if (((short) HIWORD(wParam))<0)

SendMessage (hWnd,WM\_VSCROLL,SB\_LINEDOWN,NULL);

else

SendMessage (hWnd,WM\_VSCROLL,SB\_LINEUP,NULL);

break;

case WM\_COMMAND:

switch (LOWORD(wParam))

{

case ID\_INI\_NUMBER: //Число элементов //Число элементов

SendMessage(hStatusWindow,SB\_SETTEXT,(WPARAM)0,(LPARAM)(L"Введите число элементов"));

if (oper==1) SendMessage(hWnd,WM\_COMMAND,50000,NULL);

if (init==1) SendMessage(hWnd,WM\_COMMAND,50001,NULL);

if (first==1) SendMessage(hWnd,WM\_COMMAND,50002,NULL);

for (i=0; i<N;i++) if (klass[i]!=0) {delete Stroka[i];}

hGroup1= CreateWindow(L"button", L"", WS\_CHILD | WS\_VISIBLE |BS\_GROUPBOX, 10, 10, 360, 50, hWnd, NULL, hInst, NULL);

hText1 = CreateWindow(L"STATIC",L"Количество элементов:",WS\_CHILD|WS\_VISIBLE|SS\_LEFT,30,30,230,20,hWnd,NULL,hInst,NULL);

hButton1 = CreateWindow(L"button", L"OK", WS\_CHILD | WS\_VISIBLE | BS\_PUSHBUTTON, 260, 30, 80, 20, hWnd, (HMENU)10, hInst, NULL);

hEdit1 = CreateWindow(L"EDIT", NULL, WS\_CHILD|WS\_VISIBLE|WS\_TABSTOP|ES\_MULTILINE|ES\_NUMBER,200,30,50,20,hWnd,NULL,hInst,NULL);

break;

case 10: //button "OK" //button "OK

GetWindowText(hEdit1, string, 100);

N=\_wtoi(string);

if (N<2) N=2;

if (N>=1000) N=999;

if (N>15) Nmax=15; else Nmax=N;

Stroka = new stroka\*[N+1];

klass = new int[N+1];

for (i=0;i<N+1;i++)

{

klass[i]=0;

}

hTextArray1= new HWND[Nmax];

hEditArray1= new HWND[Nmax];

hStatus=new HWND[Nmax];

hAnswer=new HWND[Nmax];

swprintf(msg,L"%d элементов",N);

SendMessage(hStatusWindow,SB\_SETTEXT,(WPARAM)0,(LPARAM)(msg));

DestroyWindow(hEdit1);

DestroyWindow(hButton1);

DestroyWindow(hText1);

DestroyWindow(hGroup1);

EnableMenuItem(hIni,ID\_INI\_START,MF\_BYCOMMAND|MF\_ENABLED);

hBox1=CreateWindow(L"button", L"", WS\_CHILD | WS\_VISIBLE |BS\_GROUPBOX, 10, 10, 420, 420, hWnd, NULL, hInst, NULL);

hInfo[0]=CreateWindow(L"STATIC", L"№", WS\_CHILD|WS\_VISIBLE|SS\_LEFT,20,20,40,20,hWnd,NULL,hInst,NULL);

hInfo[1]=CreateWindow(L"STATIC", L"Ввод", WS\_CHILD|WS\_VISIBLE|SS\_LEFT,10+50,20,160,20,hWnd,NULL,hInst,NULL);

hInfo[2]=CreateWindow(L"STATIC", L"Класс", WS\_CHILD|WS\_VISIBLE|SS\_LEFT,10+50+170,20,160,20,hWnd,NULL,hInst,NULL);

for (i=0;i<Nmax;i++)

{

swprintf(msg,L"%d",i+1);

hTextArray1[i]= CreateWindow(L"STATIC",(LPCWSTR)msg,WS\_CHILD|WS\_VISIBLE|SS\_LEFT,20,otstup+20+i\*25,40,20,hWnd,NULL,hInst,NULL);

if (klass[i+Pos]!=0) hEditArray1[i]= CreateWindowA("EDIT", Stroka[i+Pos]->GetStr(),ES\_AUTOHSCROLL|WS\_CHILD|WS\_VISIBLE|WS\_TABSTOP,60,otstup+20+i\*25,160,20,hWnd,(HMENU)(101+i),hInst,NULL);//ES\_READONLY|

else hEditArray1[i]= CreateWindowA("EDIT", NULL,ES\_AUTOHSCROLL|WS\_CHILD|WS\_VISIBLE|WS\_TABSTOP,60,otstup+20+i\*25,160,20,hWnd,(HMENU)(101+i),hInst,NULL);

switch (klass[i+Pos])

{

case 0:swprintf(msg,L" ");break;

case 1:swprintf(msg,L"stroka");break;

case 3:swprintf(msg,L"IdentStr");break;

case 2:swprintf(msg,L"DesStr");break;

}

hStatus[i]= CreateWindow(L"EDIT",msg,ES\_READONLY|WS\_CHILD|WS\_VISIBLE|WS\_TABSTOP,60+170,otstup+20+i\*25,50,20,hWnd,NULL,hInst,NULL);

}

hBox= CreateWindow(L"button", L"", WS\_CHILD | WS\_VISIBLE |BS\_GROUPBOX, 835, 10, 420, 420, hWnd, NULL, hInst, NULL);

if (first==0) hMemo=CreateWindow(L"EDIT",msg,ES\_READONLY|WS\_VSCROLL|ES\_MULTILINE|WS\_CHILD|WS\_VISIBLE|WS\_TABSTOP|ES\_AUTOVSCROLL,840,20,410,400,hWnd,NULL,hInst,NULL);

first=1;

break;

case ID\_INI\_START: //ИНИЦИАЛИЗАЦИЯ ОКНО

EnableMenuItem(hTest,(UINT\_PTR)hTest,MF\_BYCOMMAND|MF\_ENABLED);

EnableMenuItem(hMenu,(UINT\_PTR)hTest,MF\_BYCOMMAND|MF\_ENABLED);

EnableMenuItem(hTest,ID\_TEST\_IDEN,MF\_BYCOMMAND|MF\_ENABLED);

EnableMenuItem(hTest,ID\_TEST\_DES,MF\_BYCOMMAND|MF\_ENABLED);

EnableMenuItem(hTest,ID\_TEST\_OPER,MF\_BYCOMMAND|MF\_ENABLED);

if (init==1) break; else

{

init=1;

hComboBox2=CreateWindow(L"listbox", NULL,WS\_CHILD | WS\_VISIBLE | LBS\_STANDARD |LBS\_WANTKEYBOARDINPUT,otsupY+520,90, 180, 50,hWnd, (HMENU)14, hInst, NULL);

SendMessage(hComboBox2, LB\_ADDSTRING, 0,(LPARAM)(LPSTR)L"Cтрока");

SendMessage(hComboBox2, LB\_ADDSTRING, 0,(LPARAM)(LPSTR)L"Десятичная строка");

SendMessage(hComboBox2, LB\_ADDSTRING, 0,(LPARAM)(LPSTR)L"Строка идентификатор");

SendMessage(hComboBox2, CB\_SETCURSEL, 0, 0L);

swprintf(buff1,L"№%d",current);

hBox2=CreateWindow(L"button", L"Инициализация", WS\_CHILD | WS\_VISIBLE |BS\_GROUPBOX, otsupY+450, 10, 360, 200, hWnd, NULL, hInst, NULL);

hClose2= CreateWindow(L"button", L"Закрыть", WS\_CHILD | WS\_VISIBLE | BS\_PUSHBUTTON, otsupY+749, 11, 61, 20, hWnd, (HMENU)50001, hInst, NULL);

hText2= CreateWindow(L"STATIC", buff1,WS\_CHILD|WS\_VISIBLE|WS\_TABSTOP,otsupY+470,15+40,40,20,hWnd,NULL,hInst,NULL);

hEdit2= CreateWindow(L"EDIT",NULL,ES\_AUTOHSCROLL|WS\_CHILD|WS\_VISIBLE|WS\_TABSTOP,otsupY+520,15+40,180,20,hWnd,NULL,hInst,NULL);

hButton2= CreateWindow(L"button", L"OK", WS\_CHILD | WS\_VISIBLE | BS\_PUSHBUTTON, otsupY+720,15+ 40, 80, 20, hWnd, (HMENU)5000, hInst, NULL);

hButtonClear=CreateWindow(L"button", L"Очистить", WS\_CHILD | WS\_VISIBLE | BS\_PUSHBUTTON, otsupY+720,25+15+ 40, 80, 20, hWnd, (HMENU)5005, hInst, NULL);

}

break;

case 5000: //ПРОИНИЦИАЛИЗИРОВАТЬ

if (klass[current-1]!=0) {delete Stroka[current-1];}

klass[current-1]= (int)SendMessage(hComboBox2,LB\_GETCURSEL, 0, 0L)+1;

GetWindowTextA(hEdit2,cbuff,100);

//if (klass[current-1]!=0) {delete Stroka[current-1];}

switch (klass[current-1])

{

case 0: Stroka[current-1]=new stroka(cbuff);klass[current-1]=1;break;

case 1: Stroka[current-1]=new stroka(cbuff);break;

case 3: Stroka[current-1]=new IdentStr(cbuff);break;

case 2: Stroka[current-1]=new DesStr(cbuff);break;

}

switch (klass[current-1])

{

case 0:swprintf(msg,L" ");break;

case 1:swprintf(msg,L"stroka");break;

case 3:swprintf(msg,L"IdentStr");break;

case 2:swprintf(msg,L"DesStr");break;

}

if ((current-1-Pos>=0)&&(current-1-Pos<=15)) SetWindowText(hStatus[current-1-Pos],msg);

if ((current-1-Pos>=0)&&(current-1-Pos<=15)) SetWindowTextA(hEditArray1[current-1-Pos],Stroka[current-1]->GetStr());

if (current<N) current++;

swprintf(buff2,L"№%d",current);

SetWindowText(hText2,buff2);

break;

case 5005: // ОЧИСТИТЬ

if (klass[current-1]!=0)

{

klass[current-1]=0;

delete Stroka[current-1];

if ((current-1-Pos>=0)&&(current-1-Pos<=15))

{SetWindowText(hStatus[current-1-Pos],L"");

SetWindowText(hEditArray1[current-1-Pos],L"");}

}

else SendMessage (hMemo, EM\_REPLACESEL, 0, (LPARAM) ((LPSTR)L"Удаление не возможно\r\n"));

break;

case (UINT)11: // MENU: Exit

SendMessage(hStatusWindow,SB\_SETTEXT,(WPARAM)0,(LPARAM)(L"Выход"));

Esc=1;SendMessage(hWnd,WM\_CLOSE,0,0);break;

case 12: // MENU: About

MessageBox(hWnd,L"Курсовая работа\nСтудента группы №1130\nВалиева А.А.\nВариант №4",L"О программе",MB\_ICONINFORMATION);

break;

case ID\_TEST\_STRING\_FOOL: //пуста ли строка

if (Answer==0)

{

hInfo[3]=CreateWindow(L"STATIC", L"Пустая?", WS\_CHILD|WS\_VISIBLE|SS\_LEFT,170+20+140,20,60,20,hWnd,NULL,hInst,NULL);

for (i=0;i<Nmax;i++)

{

if (klass[i+Pos]==0) swprintf(msg,L""); else

if (Stroka[i+Pos]->empty()==1) swprintf(msg,L"Да"); else swprintf(msg,L"Нет");

hAnswer[i]=CreateWindow(L"STATIC", msg, WS\_CHILD|WS\_VISIBLE|SS\_LEFT,20+170+140,otstup+20+i\*25,60,20,hWnd,NULL,hInst,NULL);

}

} else

{

SetWindowText(hInfo[3],L"Пустая?");

for (i=0;i<Nmax;i++)

{

if (klass[i+Pos]==0) swprintf(msg,L""); else

if (Stroka[i+Pos]->empty()==1) swprintf(msg,L"Да"); else swprintf(msg,L"Нет");

SetWindowText(hAnswer[i],msg);

}

}

Answer=1;

break;

case ID\_TEST\_IDEN\_REG: //в нижний регистр

if (Answer==1)

{

Answer=0;

DestroyWindow(hInfo[3]);

for (i=0;i<Nmax;i++)

DestroyWindow(hAnswer[i]);

}

for (i=0;i<N;i++)

{

if (klass[i]==3) Stroka[i]->Down();

}

for (i=0;i<Nmax;i++)

{

if (klass[i+Pos]==3) SetWindowTextA(hEditArray1[i],Stroka[i+Pos]->GetStr());

}

break;

case ID\_TEST\_DES\_INT: // можно ли в Int

if (Answer==0)

{

hInfo[3]=CreateWindow(L"STATIC", L"Int?", WS\_CHILD|WS\_VISIBLE|SS\_LEFT,170+20+140,20,60,20,hWnd,NULL,hInst,NULL);

for (i=0;i<Nmax;i++)

{

if (klass[i+Pos]!=2) swprintf(msg,L""); else

if (Stroka[i+Pos]->CanInt()==1) swprintf(msg,L"Да"); else swprintf(msg,L"Нет");

hAnswer[i]=CreateWindow(L"STATIC", msg, WS\_CHILD|WS\_VISIBLE|SS\_LEFT,20+170+140,otstup+20+i\*25,60,20,hWnd,NULL,hInst,NULL);

}

} else

{

SetWindowText(hInfo[3],L"Int?");

for (i=0;i<Nmax;i++)

{

if (klass[i+Pos]!=2) swprintf(msg,L""); else

if (Stroka[i+Pos]->CanInt()==1) swprintf(msg,L"Да"); else swprintf(msg,L"Нет");

SetWindowText(hAnswer[i],msg);

}

}

Answer=2;

break;

case ID\_TEST\_OPER: // выбрать операнд

if (oper==1) break;

else

{

oper=1;

hBox3= CreateWindow(L"button", L"Выбор операндов", WS\_CHILD | WS\_VISIBLE |BS\_GROUPBOX, otsupY+450, 227, 360, 202, hWnd, NULL, hInst, NULL);

hClose3= CreateWindow(L"button", L"Закрыть", WS\_CHILD | WS\_VISIBLE | BS\_PUSHBUTTON, otsupY+751, 228, 60, 20, hWnd, (HMENU)50000, hInst, NULL);

hText3[0] = CreateWindow(L"STATIC",L"1-ый операнд",WS\_CHILD|WS\_VISIBLE|WS\_TABSTOP,otsupY+470,255,100,20,hWnd,NULL,hInst,NULL);

hEdit3[0] = CreateWindow(L"EDIT",NULL,WS\_CHILD|WS\_VISIBLE|WS\_TABSTOP,otsupY+575,255,200,20,hWnd,NULL,hInst,NULL);

hText3[1] = CreateWindow(L"STATIC",L"2-ой операнд",WS\_CHILD|WS\_VISIBLE|WS\_TABSTOP,otsupY+470,280,100,20,hWnd,NULL,hInst,NULL);

hEdit3[1] = CreateWindow(L"EDIT",NULL,WS\_CHILD|WS\_VISIBLE|WS\_TABSTOP,otsupY+575,280,200,20,hWnd,NULL,hInst,NULL);

hText3[2] = CreateWindow(L"STATIC",L"3-ий операнд",WS\_CHILD|WS\_VISIBLE|WS\_TABSTOP,otsupY+470,305,100,20,hWnd,NULL,hInst,NULL);

hEdit3[2] = CreateWindow(L"EDIT",NULL,WS\_CHILD|WS\_VISIBLE|WS\_TABSTOP|ES\_NUMBER,otsupY+575,305,200,20,hWnd,NULL,hInst,NULL);

hText3[3] = CreateWindow(L"STATIC",L"Результат",WS\_CHILD|WS\_VISIBLE|WS\_TABSTOP,otsupY+470,370,100,20,hWnd,NULL,hInst,NULL);

hOAnswer = CreateWindow(L"EDIT",NULL,ES\_AUTOHSCROLL|WS\_CHILD|WS\_VISIBLE|WS\_TABSTOP|ES\_NUMBER,otsupY+470,395,310,20,hWnd,NULL,hInst,NULL);

}

break;

case 50000: //КНОПКА ЗАКРЫТЬ ОПЕРАНДЫ

oper=0;

DestroyWindow(hBox3);

for (i=0;i<=2;i++)

{DestroyWindow(hText3[i]);DestroyWindow(hEdit3[i]);}

DestroyWindow(hClose3);

DestroyWindow(hOAnswer);

DestroyWindow(hText3[3]);

break;

case 50001: //КНОПКА ЗАКРЫТЬ ИНИЦИАЛИЗАЦИЯ

init=0;

DestroyWindow(hComboBox2);

DestroyWindow(hBox2);

DestroyWindow(hClose2);

DestroyWindow(hText2);

DestroyWindow(hEdit2);

DestroyWindow(hButton2);

DestroyWindow(hButtonClear);

break;

case 50002: //ЗАКРЫТЬ ВСЕ

for (i=0;i<=3;i++)

DestroyWindow(hInfo[i]);

for (i=0;i<=Nmax;i++)

{

DestroyWindow(hTextArray1[i]);

DestroyWindow(hEditArray1[i]);

DestroyWindow(hStatus[i]);

DestroyWindow(hAnswer[i]);

}

DestroyWindow(hBox1);

Answer=0;

break;

case 1001:if (oper==0) SendMessage(hWnd,WM\_COMMAND,ID\_TEST\_OPER,NULL); else Operand(1,1); break;

case 1002:if (oper==0) SendMessage(hWnd,WM\_COMMAND,ID\_TEST\_OPER,NULL); else Operand(1,2); break;

case 1003:if (oper==0) SendMessage(hWnd,WM\_COMMAND,ID\_TEST\_OPER,NULL); else Operand(1,3); break;

case 1005:if (oper==0) SendMessage(hWnd,WM\_COMMAND,ID\_TEST\_OPER,NULL); else Operand(1,5); break;

case 2001:if (oper==0) SendMessage(hWnd,WM\_COMMAND,ID\_TEST\_OPER,NULL); else Operand(2,1); break;

case 2002:if (oper==0) SendMessage(hWnd,WM\_COMMAND,ID\_TEST\_OPER,NULL); else Operand(2,2); break;

case 2003:if (oper==0) SendMessage(hWnd,WM\_COMMAND,ID\_TEST\_OPER,NULL); else Operand(2,3); break;

case 2004:if (oper==0) SendMessage(hWnd,WM\_COMMAND,ID\_TEST\_OPER,NULL); else Operand(2,4); break;

case 2005:if (oper==0) SendMessage(hWnd,WM\_COMMAND,ID\_TEST\_OPER,NULL); else Operand(2,5); break;

case 2006:if (oper==0) SendMessage(hWnd,WM\_COMMAND,ID\_TEST\_OPER,NULL); else Operand(2,6); break;

case 3001:if (oper==0) SendMessage(hWnd,WM\_COMMAND,ID\_TEST\_OPER,NULL); else Operand(3,1); break;

case 3002:if (oper==0) SendMessage(hWnd,WM\_COMMAND,ID\_TEST\_OPER,NULL); else Operand(3,2); break;

case 3003:if (oper==0) SendMessage(hWnd,WM\_COMMAND,ID\_TEST\_OPER,NULL); else Operand(3,3); break;

case 3004:if (oper==0) SendMessage(hWnd,WM\_COMMAND,ID\_TEST\_OPER,NULL); else Operand(3,4); break;

case 3005:if (oper==0) SendMessage(hWnd,WM\_COMMAND,ID\_TEST\_OPER,NULL); else Operand(3,5); break;

case 3006:if (oper==0) SendMessage(hWnd,WM\_COMMAND,ID\_TEST\_OPER,NULL); else Operand(3,6); break;

case 3007:if (oper==0) SendMessage(hWnd,WM\_COMMAND,ID\_TEST\_OPER,NULL); else Operand(3,7); break;

case 1004:if (oper==0) SendMessage(hWnd,WM\_COMMAND,ID\_TEST\_OPER,NULL); else Operand(1,4); break;

break;

default:

int temp=wParam-16777317;

if ((temp<16)&&(temp>=0))

{

SendMessage(hWnd,WM\_COMMAND,ID\_INI\_START,NULL);

current=temp+Pos+1;

swprintf(buff2,L"№%d",current);

SetWindowText(hText2,buff2);

GetWindowText(hEditArray1[current-1],buff2,100);

SetWindowText(hEdit2,buff2);

}

break;

}

case WM\_SIZE: SendMessage(hStatusWindow,WM\_SIZE,wParam,lParam);

break;

default:

return DefWindowProc(hWnd, message, wParam, lParam);

break;

}

return 0;

}

**down.asm**

.686

.MODEL FLAT, C

PUBLIC ASMDOWN

.CODE

ASMDOWN PROC C x:DWORD, y:BYTE

mov ESI, x

mov CL, y

BODY:

mov AL, byte ptr [ESI]

cmp AL, 'A'

jl I

cmp AL, 'Z'

jg I

add AL,20h

mov byte ptr[ESI], AL

I:

inc ESI

loop BODY

mov EAX , x

RET

ASMDOWN ENDP

END

# Заключение

В ходе проделанной работы мы выполнили техническое задание и создали приложение под WIN32, способное тестировать наши классы.

# Список литературы

1.Березин Б.И., Березин С.Б. Начальный курс Си и C++. -М.: ДИАЛОГ-МИФИ, 2001.

2. Безруков В.А. Win32 API. Программирование /учебное пособие.–СПб:СПбГУ ИТМО,2009

3. http://msdn.microsoft.com/