Удаление комментариев из текста программ

Содержание

Введение

1. Постановка задачи

2. Основные особенности среды Visual C++

2.1 Описание алгоритма

2.2 Описание программы

3. Инструкции пользователя

Заключение

Литература

Приложение

Введение

Эволюция языков программирования с момента появления первых электронных компьютеров, построенных для выполнения расчётов траектории движения артиллерийских снарядов во время второй мировой войны, была довольно драматической. Раньше программисты работали с самыми примитивными компьютерными командами, представлявшими собой часть машинного языка. Эти команды состояли из длинных строк единиц и нулей. Вскоре были изобретены ассемблеры, которые могли отображать машинные команды в мнемоническом представлении, более понятном для человека (например, команды ADD или MOV).

Со временем появились такие языки высокого уровня, как BASIC и COBOL. Благодаря этим языкам появилась возможность программировать, используя логические конструкции из слов и предложений, например integer I = 100. Эти команды переводились в машинный язык то есть компилировались. Одно из основных преимуществ компилируемых языков программирования, состоит в том, что полученные программы могут выполняться на компьютерах без компиляции. При работе же с интерпретируемыми языками для выполнения готовой программы нужно обязательно иметь соответствующую программу-интерпретатор.

Следующей шагом в программировании стало появление визуальных сред программирования таких, как Visual C++ , позволяющих использовать визуальные компоненты, не описывая их самому , а брать их из библиотеки стандартных элементов . Это позволяет ускорить написание программ, облегчая тем самым работу программиста . А также позволяет использовать все возможности программирования под Windows, используя при этом относительно не сложные алгоритмы и функции. В связи с выше изложенным нами было выбрана для разработки проекта среда Visual C++ 6.0

1. Постановка задачи

В данном курсовом проекте необходимо разработать программу с использованием функций библиотеки MFC:

* Удаление комментариев из текстов программ;
* Автоматический выбор языка, на котором написана программа;
* Сохранение результата в файл.

2. Основные особенности среды Visual C++

Среда Visual C++ – это сложный механизм, обеспечивающий высокоэффективную работу программиста. Создание прикладных программ, или приложений выполняется в интегрированной среде разработки IDE (Integrated Development Environment). IDE служит для организации взаимодействия с программистом и включает ряд окон, содержащих различные управляющие элементы. С помощью средств интегрированной среды разработчик может проектировать интерфейсную часть приложения, а также писать программный код и связывать его с управляющими элементами. При этом вся работа по созданию приложения, включая отладку, происходит в IDE.

Интегрированная среда разработки Visual C++ представляет собой многооконную систему. Вид интегрированной среды разработки (интерфейс) может различаться в зависимости от настроек. Кроме стандартных окон, на экране могут присутствовать и другие окна, отображаемые при вызове соответствующих средств, например, Image Editor (Редактор изображений). Окна Visual C++ (но не главное) можно перемещать, убирать с экрана, а также изменять их размеры. Одной из последних версий стал Microsoft Visual C++ 6.0. Visual C++ 6.0, вобрав в себя всё самое лучшее от предыдущих версий, предоставляет ряд новых возможностей. Так, например, стал более удобным и современным интерфейс среды программирования, создаваемые Visual C++ программы учитывают архитектуру современных процессоров, существенно расширены возможности отладчика.

Visual C++ 6.0 может работать в среде операционных систем от Windows 95 до Windows 2000 и Windows XP. Особенных требований к компьютеру система не предъявляет, за исключением того, что процессор должен быть типа Pentium, оперативной памяти – не менее 32 Мбайт и достаточное количество свободной дисковой памяти (порядка 200 Мбайт).

2.1 Описание алгоритма

Согласно условию курсового проекта был разработан алгоритм удаления комментариев из текстов программ. Суть алгоритма заключается в том, что при нажатии кнопки "Удалить комментарии" происходит открытие файла, указанного в верхнем текстовом окне, определение языка, на котором написана программа, удаление из него комментариев и запись в файл, указанный в нижнем текстовом окне.

Удаление комментариев возможно из текстов программ, написанных на следующих языках: C и С++, Pascal, Basic и Assembler.

2.2 Описание программы

Данное приложение состоит из трех классов: CKursApp, CKursDlg и CSelectLangDlg. Все основные функции содержатся во втором классе, т.к. именно он отвечает за поведение диалогового окна. Работа программы основана на двух функциях: RemoveOneLineComments и RemoveMultiLineComments. Реализация второй является довольно простой. В нее передаются три параметра: исходный текст, символ открывающего комментария и символ закрывающего комментария. В тексте осуществляется поиск открывающих комментариев и парных им закрывающих, содержимое между ними удаляется.

Удаление однострочных комментариев более сложное из-за того, что в отдельных языках, например в Бейсике, строки могут переноситься и однострочный комментарий становится подобным многострочному. В функцию передаются четыре параметра: исходный текст, символ комментария, параметр, определяющий наличие переносов и сам символ переноса. Работа функции заключается в следующем. Исходный текст разбивается на строки, которые заносятся в массив. Затем производится поиск символа комментария. Если он найден, проверяется, есть ли в данной строке перенос. Строка или ее часть после комментария удаляется. При наличии переноса удаляется также и следующая строка.

Определение языка производится по расширению. Если расширение неизвестно, у пользователя запрашивается тип файла. Затем производится удаление комментариев и в случае успешного завершения выдается сообщение.

Результаты сохраняются в указанном файле.

3. Инструкция пользователя

Запуск программы осуществляется при открытии файла kurs.exe, который находится на дискете. При этом на экране появиться окно, в левой верхней части которого будет видна надпись "Удаление комментариев из текста программы " – это имя программы. Для работы необходимо указать исходный и конечный файлы, введя их пути в текстовые окна или выбрав их в диалоге выбора. Если файл имеет неизвестное программе расширение, необходимо дополнительно выбрать язык в диалоге, который автоматически появляется после выбора файла. Затем необходимо нажать кнопку "Удалить комментарии". Если текст программы не содержит некорректно расставленных комментариев, выдается сообщение об успешном выполнении операции.

Системные требования: Pentium 100 , 16 MB RAM, Windows 95/98/2000 NT/XP.

Заключение

В ходе выполнения данного курсового проекта были разработана программа на языке высокого программирования Visual C++. Также изучены графические возможности данного языка.

Систематизированы и закреплены практические навыки использования ЭВМ, программного обеспечения, существующих средств обслуживания системных программистов, а также теоретические знания по основным разделам курса "Программирование". Основное внимание уделено изучению современных операционных систем, способов проектирования приложений, объектно-ориентированному и системному программированию.

При выполнении курсового проекта произведено знакомство с реферативными журналами и другими информационными источниками по объектно-ориентированному и системному программированию с целью анализа состояния решаемой задачи.

Получены практические навыки работы в среде Visual C++ и в графической среде разработки.

Литература

1. Петзольд Ч. Программирование под Windows 95. В двух книгах: BHV – Санкт - Петербург, 2007, silt.

2. Ричард С.Линкер, Том Арчер. Программирование для Windows 7. Библия разработчика. "Диалектика " – Москва, 2009.-864 с.: ил.- Парал. тит. англ. Уч.пос.

3. Джесс Либерти. С++ за 21 день. "Вильямс" - Москва, 2000.-816 с.: ил. .- Парал.тит. англ.

Приложение

Файл kursDlg.cpp

// kursDlg.cpp : implementation file

//

#include "stdafx.h"

#include "kurs.h"

#include "kursDlg.h"

#include "SelectLangDlg.h"

#ifdef \_DEBUG

#define new DEBUG\_NEW

#undef THIS\_FILE

static char THIS\_FILE[] = \_\_FILE\_\_;

#endif

/////////////////////////////////////////////////////////////////////////////

#define LANG\_CPP 1

#define LANG\_PASCAL 2

#define LANG\_BASIC 3

#define LANG\_ASM 4

/////////////////////////////////////////////////////////////////////////////

// CKursDlg dialog

CKursDlg::CKursDlg(CWnd\* pParent /\*=NULL\*/)

: CDialog(CKursDlg::IDD, pParent)

{

//{{AFX\_DATA\_INIT(CKursDlg)

// NOTE: the ClassWizard will add member initialization here

//}}AFX\_DATA\_INIT

// Note that LoadIcon does not require a subsequent DestroyIcon in Win32

m\_hIcon = AfxGetApp()->LoadIcon(IDR\_MAINFRAME);

}

void CKursDlg::DoDataExchange(CDataExchange\* pDX)

{

CDialog::DoDataExchange(pDX);

//{{AFX\_DATA\_MAP(CKursDlg)

// NOTE: the ClassWizard will add DDX and DDV calls here

//}}AFX\_DATA\_MAP

}

BEGIN\_MESSAGE\_MAP(CKursDlg, CDialog)

//{{AFX\_MSG\_MAP(CKursDlg)

ON\_WM\_SYSCOMMAND()

ON\_WM\_PAINT()

ON\_WM\_QUERYDRAGICON()

ON\_BN\_CLICKED(IDC\_BTNSOURCE, OnBtnsource)

ON\_BN\_CLICKED(IDC\_BTNMAKE, OnBtnmake)

ON\_BN\_CLICKED(IDC\_BTNDEST, OnBtndest)

//}}AFX\_MSG\_MAP

END\_MESSAGE\_MAP()

/////////////////////////////////////////////////////////////////////////////

// CKursDlg message handlers

BOOL CKursDlg::OnInitDialog()

{

CDialog::OnInitDialog();

/ Set the icon for this dialog. The framework does this automatically

// when the application's main window is not a dialog

SetIcon(m\_hIcon, TRUE);// Set big icon

SetIcon(m\_hIcon, FALSE);// Set small icon

m\_lang = LANG\_CPP;

m\_name = "";

// TODO: Add extra initialization here

return TRUE; // return TRUE unless you set the focus to a control

}

void CKursDlg::OnSysCommand(UINT nID, LPARAM lParam)

{

CDialog::OnSysCommand(nID, lParam);

}

// If you add a minimize button to your dialog, you will need the code below

// to draw the icon. For MFC applications using the document/view model,

// this is automatically done for you by the framework.

void CKursDlg::OnPaint()

{

if (IsIconic())

{

CPaintDC dc(this); // device context for painting

SendMessage(WM\_ICONERASEBKGND, (WPARAM) dc.GetSafeHdc(), 0);

// Center icon in client rectangle

int cxIcon = GetSystemMetrics(SM\_CXICON);

int cyIcon = GetSystemMetrics(SM\_CYICON);

CRect rect;

GetClientRect(&rect);

int x = (rect.Width() - cxIcon + 1) / 2;

int y = (rect.Height() - cyIcon + 1) / 2;

// Draw the icon

dc.DrawIcon(x, y, m\_hIcon);

}

else

{

CDialog::OnPaint();

}

}

// The system calls this to obtain the cursor to display while the user drags

// the minimized window.

HCURSOR CKursDlg::OnQueryDragIcon()

{

return (HCURSOR) m\_hIcon;

}

void CKursDlg::OnBtnsource()

{

CFileDialog dlg(TRUE, NULL, NULL, 0,

"C++ and C files (\*.cpp; \*.c; \*.h; \*.hpp)|\*.cpp;\*.c;\*.h;\*.hpp|Pascal files (\*.pas)|\*.pas|Basic files (\*.bas)|\*.bas|Assembler files (\*.asm)|\*.asm|All files|\*.\*||", this);

CString lang;

CSelectLangDlg select;

if(dlg.DoModal() == IDOK)

{

m\_name = dlg.GetPathName();

SetDlgItemText(IDC\_SOURCE, m\_name);

int point\_pos = m\_name.ReverseFind('.');

if(point\_pos == -1)

{

if(select.DoModal())

m\_lang = select.m\_language + 1;

else

m\_lang = LANG\_CPP;

}

else

{

lang = m\_name.Right(m\_name.GetLength() - point\_pos - 1);

if((lang == "c") || (lang == "cpp") || (lang == "h") || (lang == "hpp"))

m\_lang = LANG\_CPP;

else if(lang == "pas")

m\_lang = LANG\_PASCAL;

else if(lang == "bas")

m\_lang = LANG\_BASIC;

else if(lang == "asm")

m\_lang = LANG\_ASM;

else

{

if(select.DoModal())

m\_lang = select.m\_language + 1;

else

m\_lang = LANG\_CPP;

}

}

point\_pos = m\_name.ReverseFind('\\');

m\_dest = m\_name.Left(point\_pos);

m\_dest += "\\output.";

m\_dest += lang;

SetDlgItemText(IDC\_DEST, m\_dest);

}

}

void CKursDlg::OnBtnmake()

{

GetDlgItemText(IDC\_SOURCE, m\_name);

GetDlgItemText(IDC\_DEST, m\_dest);

CFile source(m\_name, CFile::modeRead | CFile::shareDenyRead | CFile::shareDenyWrite | CFile::normal);

CString text;

char buffer;

UINT i;

for(i = 0; i < source.GetLength(); i++)

{

source.Read(&buffer, 1);

text += buffer;

}

text.GetBuffer(0)[source.GetLength()] = 0;

switch(m\_lang)

{

case LANG\_CPP:

{

RemoveOneLineComments(text, "//", FALSE, "");

if(RemoveMultiLineComments(text, "/\*", "\*/") == -1)

{

AfxMessageBox("Неправильные многострочные коментарии", MB\_ICONERROR);

return;

}

break;

}

case LANG\_PASCAL:

{

RemoveOneLineComments(text, "//", FALSE, "");

if(RemoveMultiLineComments(text, "{", "}") == -1)

{

AfxMessageBox("Неправильные многострочные коментарии", MB\_ICONERROR);

return;

}

break;

}

case LANG\_BASIC:

{

RemoveOneLineComments(text, "'", TRUE, "\_");

RemoveOneLineComments(text, "REM ", TRUE, "\_");

RemoveOneLineComments(text, "rem ", TRUE, "\_");

RemoveOneLineComments(text, "Rem ", TRUE, "\_");

break;

}

case LANG\_ASM:

{

RemoveOneLineComments(text, ";", FALSE, "");

break;

}

}

source.Close();

CFile dest(m\_dest, CFile::modeCreate | CFile::modeReadWrite | CFile::normal);

dest.Write(text.GetBuffer(0), text.GetLength());

AfxMessageBox("Комментарии удалены", MB\_ICONINFORMATION);

}

int CKursDlg::RemoveOneLineComments(CString& text, CString comment, BOOL bHasLineWraps, CString wrap)

{

CArray<CString, CString&> linelist;

CString line, full = text;

char \*buffer = full.LockBuffer();

char \*token = strtok(buffer, "\r\n");

while(token)

{

linelist.Add(CString(token));

token = strtok(NULL, "\r\n");

}

full.UnlockBuffer();

int i, find, count;

BOOL bWrap = FALSE;

for(i = 0; i < linelist.GetSize(); i++)

{

if(linelist[i].Find(comment, 0) == 0)

{

if(bHasLineWraps)

{

if(linelist[i].Find(wrap, 0) == linelist[i].GetLength() - wrap.GetLength())

bWrap = TRUE;

}

linelist.RemoveAt(i);

i--;

continue;

}

else if((find = linelist[i].Find(comment, 0)) > 0)

{

if(bHasLineWraps)

{

if(linelist[i].Find(wrap, 0) == linelist[i].GetLength() - wrap.GetLength())

bWrap = TRUE;

}

count = linelist[i].GetLength() - find;

linelist[i].Delete(find, count);

continue;

}

if(bHasLineWraps && bWrap)

{

if(linelist[i].Find(wrap, 0) == linelist[i].GetLength() - wrap.GetLength())

bWrap = TRUE;

else

bWrap = FALSE;

linelist.RemoveAt(i);

i--;

}

}

text = "";

for(i = 0; i < linelist.GetSize(); i++)

{

text += linelist[i];

text += "\r\n";

}

linelist.RemoveAll();

return 0;

}

int CKursDlg::RemoveMultiLineComments(CString &text, CString opencomment, CString closecomment)

{

int open, close;

while((open = text.Find(opencomment, 0)) != -1)

{

if((close = text.Find(closecomment, open)) != -1)

{

text.Delete(open, close - open + closecomment.GetLength());

}

else

return -1;

}

return 0;

}

void CKursDlg::OnBtndest()

{

CFileDialog dlg(FALSE, NULL, NULL, OFN\_HIDEREADONLY | OFN\_OVERWRITEPROMPT,

"C++ and C files (\*.cpp; \*.c; \*.h; \*.hpp)|\*.cpp;\*.c;\*.h;\*.hpp|Pascal files (\*.pas)|\*.pas|Basic files (\*.bas)|\*.bas|Assembler files (\*.asm)|\*.asm|All files|\*.\*||", this);

if(dlg.DoModal() == IDOK)

{

m\_dest = dlg.GetPathName();

SetDlgItemText(IDC\_DEST, m\_dest);

}

}

Файл SelectLangDlg;

// SelectLangDlg.cpp : implementation file

//

#include "stdafx.h"

#include "kurs.h"

#include "SelectLangDlg.h"

#ifdef \_DEBUG

#define new DEBUG\_NEW

#undef THIS\_FILE

static char THIS\_FILE[] = \_\_FILE\_\_;

#endif

/////////////////////////////////////////////////////////////////////////////

// CSelectLangDlg dialog

CSelectLangDlg::CSelectLangDlg(CWnd\* pParent /\*=NULL\*/)

: CDialog(CSelectLangDlg::IDD, pParent)

{

//{{AFX\_DATA\_INIT(CSelectLangDlg)

m\_language = 0;

//}}AFX\_DATA\_INIT

}

void CSelectLangDlg::DoDataExchange(CDataExchange\* pDX)

{

CDialog::DoDataExchange(pDX);

//{{AFX\_DATA\_MAP(CSelectLangDlg)

DDX\_Radio(pDX, IDC\_CPP, m\_language);

//}}AFX\_DATA\_MAP

}

BEGIN\_MESSAGE\_MAP(CSelectLangDlg, CDialog)

//{{AFX\_MSG\_MAP(CSelectLangDlg)

//}}AFX\_MSG\_MAP

END\_MESSAGE\_MAP()

/////////////////////////////////////////////////////////////////////////////

// CSelectLangDlg message handlers

void CSelectLangDlg::OnOK()

{

UpdateData();

CDialog::OnOK();

}

BOOL CSelectLangDlg::OnInitDialog()

{

CDialog::OnInitDialog();

return TRUE; // return TRUE unless you set the focus to a control

// EXCEPTION: OCX Property Pages should return FALSE

}