概念题

1. 请简述图形用户接口(GUI)的概念，并说明GUI主要由哪些部分组成。

图形用户接口是人与计算机进行交互的一种方式。

由窗口、下拉菜单、对话框及其相应的控制机制构成。

用户通过鼠标、键盘等输入设备操纵屏幕上的图标和菜单向计算机发出指令；

计算机用图形输出来反馈操作的结果。

2. 打开对话框一般需要哪几个步骤？有哪些预定义的对话框类？作用分别是什么？

1) 首先要创建一个对话框类的对象；

然后通过与控件对应的成员变量为控件提供初始值；

调用对话框类的成员函数DoModal，显示对话框；

对话框关闭后，通过与控件对应的成员变量获取用户在控件中输入的内容。

2) CFileDialog：文件打开/保存对话框

CFontDialog：字体选择对话框

CColorDialog：颜色选择对话框

CPrintDialog：打印设置对话框

CFindReplaceDialog：查找/替换对话框

3. 请列举一些CView的派生类，并分别说明它们的功能。

CScrollView（带滚动功能的视）

CEditView（具有编辑功能的视）

CFormView（具有表格功能的视）

CHtmlView（具有Web浏览功能的视）

编程题

1. 创建一个MFC应用程序，可以选择文字属性并显示文本信息。

程序有“文件”和“编辑”菜单项，“编辑”菜单项有三个选项，“文件”菜单项有两个选项

// CDialogInput.h

class CDialogInput : public CDialogEx

{

……

protected:

virtual BOOL OnInitDialog();

public:

CString strText = "";

CEdit m\_edit;

afx\_msg void OnEnChangeEdit();

afx\_msg void OnBnClickedOk();

afx\_msg void OnBnClickedCancel();

};

// CDialogInput.cpp

BOOL CDialogInput::OnInitDialog()

{

CDialogEx::OnInitDialog();

m\_edit.SetWindowText(strText);

return TRUE;

}

void CDialogInput::OnEnChangeEdit()

{

CString strText;

m\_edit.GetWindowText(strText);

strText = strText;

}

void CDialogInput::OnBnClickedOk()

{

CDialogEx::OnOK();

GetDlgItem(IDC\_EDIT)->GetWindowText(strText);

}

void CDialogInput::OnBnClickedCancel()

{

CDialogEx::OnCancel();

}

// NotepadDlg.h

class CNotepadDlg : public CDialogEx

{

……

protected:

HICON m\_hIcon;

CFont m\_Font;

public:

CString strOld = "";

CString strNew = "";

CString strOpenPath = "";

int Save = 0;

CRichEditCtrl m\_FileText;

afx\_msg void OnSave();

afx\_msg void OnEnChangeFile();

afx\_msg void FileSave();

afx\_msg void OnOpen();

afx\_msg void OnFont();

afx\_msg void OnColor();

afx\_msg void OnText();

};

void CNotepadDlg::FileSave()

{

CString strText = "";

char write[10000];

if ((strOpenPath.Right(4) != ".TXT") && (strOpenPath.Right(4) != ".txt"))

strOpenPath += ".TXT";

CFile file(\_T(strOpenPath), CFile::modeCreate | CFile::modeWrite);

m\_FileText.GetWindowText(strText);

strcpy(write, strText);

file.Write(write, strText.GetLength());

strOld = strNew;

Save = 1;

file.Close();

}

// NotepadDlg.cpp

void CNotepadDlg::OnSave()

{

if (strOpenPath == "") {

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

OFN\_OVERWRITEPROMPT, "All Files(\*.TXT)|\*.TXT||", AfxGetMainWnd());

CString strPathAs, strText = "";

char write[10000];

if (dlg.DoModal() == IDOK) {

strPathAs = dlg.GetPathName();

strOpenPath = strPathAs;

if ((strPathAs.Right(4) != ".TXT") && (strPathAs.Right(4) != ".txt"))

strPathAs += ".TXT";

CFile file(\_T(strPathAs), CFile::modeCreate | CFile::modeWrite);

m\_FileText.GetWindowText(strText);

strcpy(write, strText);

file.Write(write, strText.GetLength());

strOld = strNew;

Save = 1;

file.Close();

}

else

Save = 0;

}

else

FileSave();

}

void CNotepadDlg::OnEnChangeFile()

{

CString strText;

m\_FileText.GetWindowText(strText);

strNew = strText;

}

void CNotepadDlg::OnOpen()

{

if (strOld != strNew) {

if (MessageBox("内容已改变要保存吗？", NULL, MB\_YESNO | MB\_ICONQUESTION) == IDYES) {

OnSave();

if (Save == 0) return;

}

}

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

"All Files(\*.TXT)|\*.TXT||", AfxGetMainWnd());

CString strText = "";

if (dlg.DoModal() == IDOK) {

strOpenPath = dlg.GetPathName();

CFile file(strOpenPath, CFile::modeRead);

char read[10000];

file.Read(read, 10000);

for (int i = 0; i < file.GetLength(); i++) {

strText += read[i];

}

strOld = strText;

strNew = strText;

file.Close();

m\_FileText.SetWindowText(strOld);

}

}

void CNotepadDlg::OnFont()

{

CHARFORMAT cf = { 0 };

cf.cbSize = sizeof(cf);

m\_FileText.GetSelectionCharFormat(cf);

CFontDialog dlg(cf);

if (dlg.DoModal() == IDOK) {

dlg.GetCharFormat(cf);

m\_FileText.SetSelectionCharFormat(cf);

}

}

void CNotepadDlg::OnColor()

{

CHARFORMAT cf = { 0 };

cf.cbSize = sizeof(cf);

cf.dwMask = CFM\_COLOR;

m\_FileText.GetSelectionCharFormat(cf);

CColorDialog dlg(cf.crTextColor);

if (dlg.DoModal() == IDOK) {

cf.crTextColor = dlg.GetColor();

m\_FileText.SetSelectionCharFormat(cf);

}

}

void CNotepadDlg::OnText()

{

CDialogInput dlg;

dlg.strText = strNew;

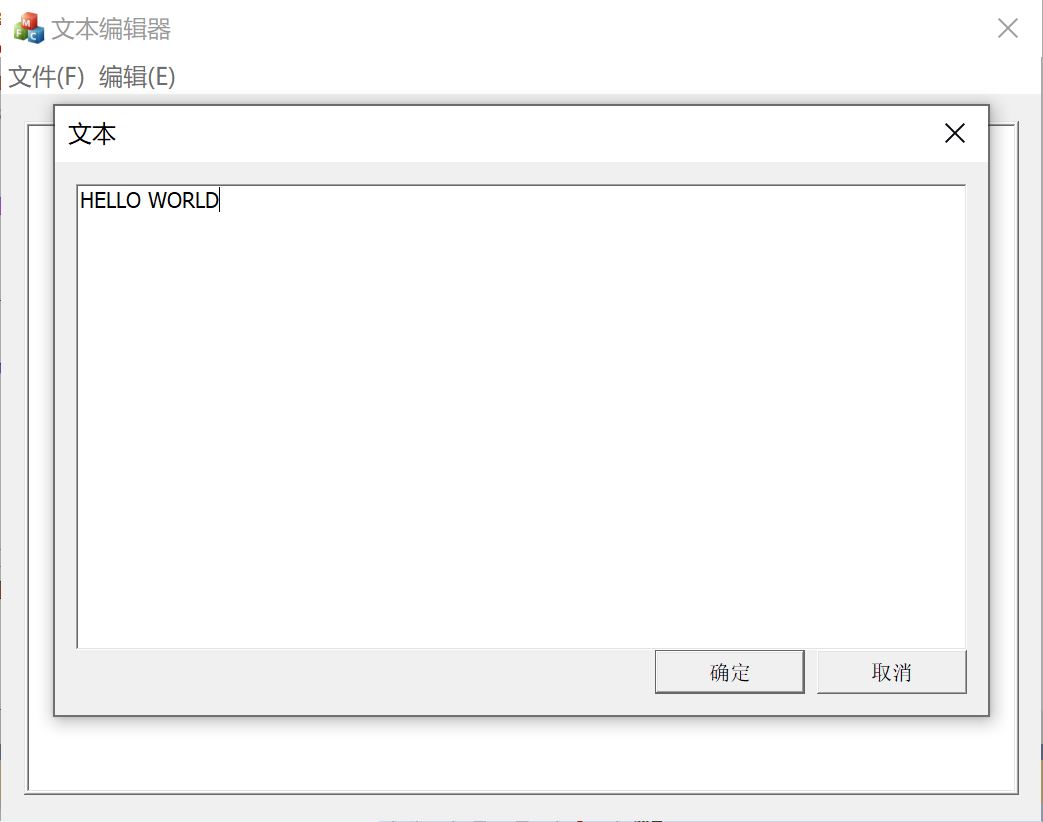
if (dlg.DoModal() == IDOK) {

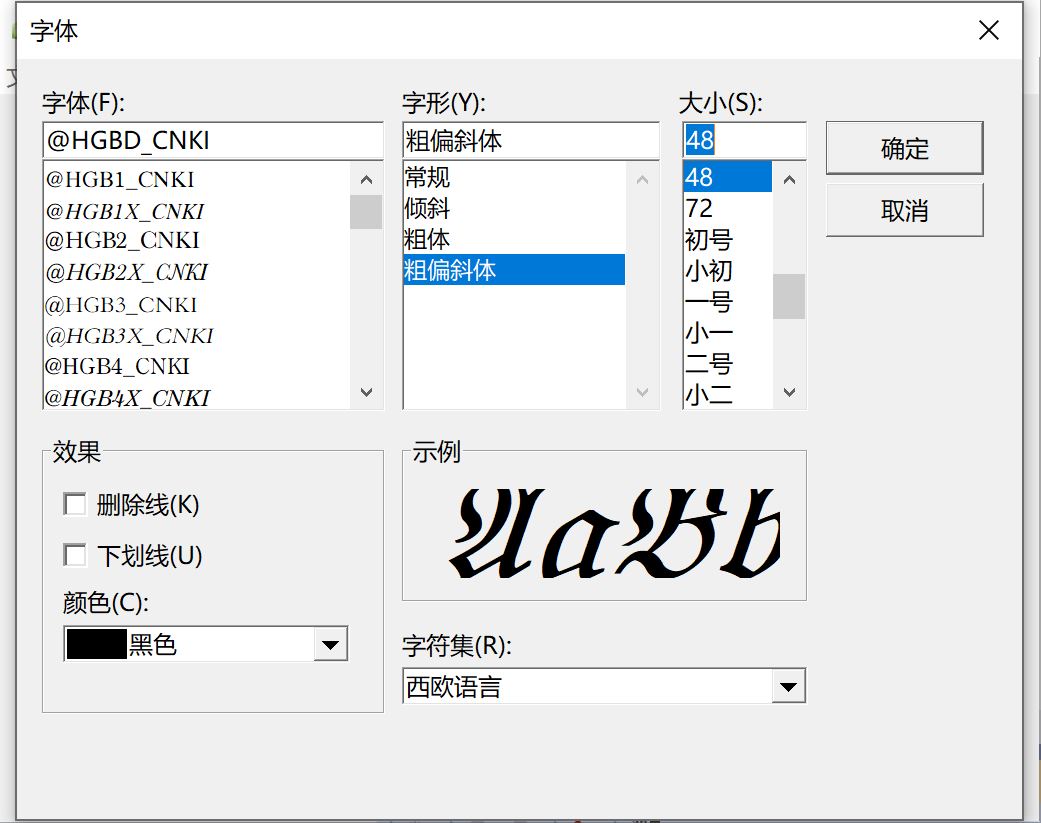
strNew = dlg.strText;

m\_FileText.SetWindowText(strNew);

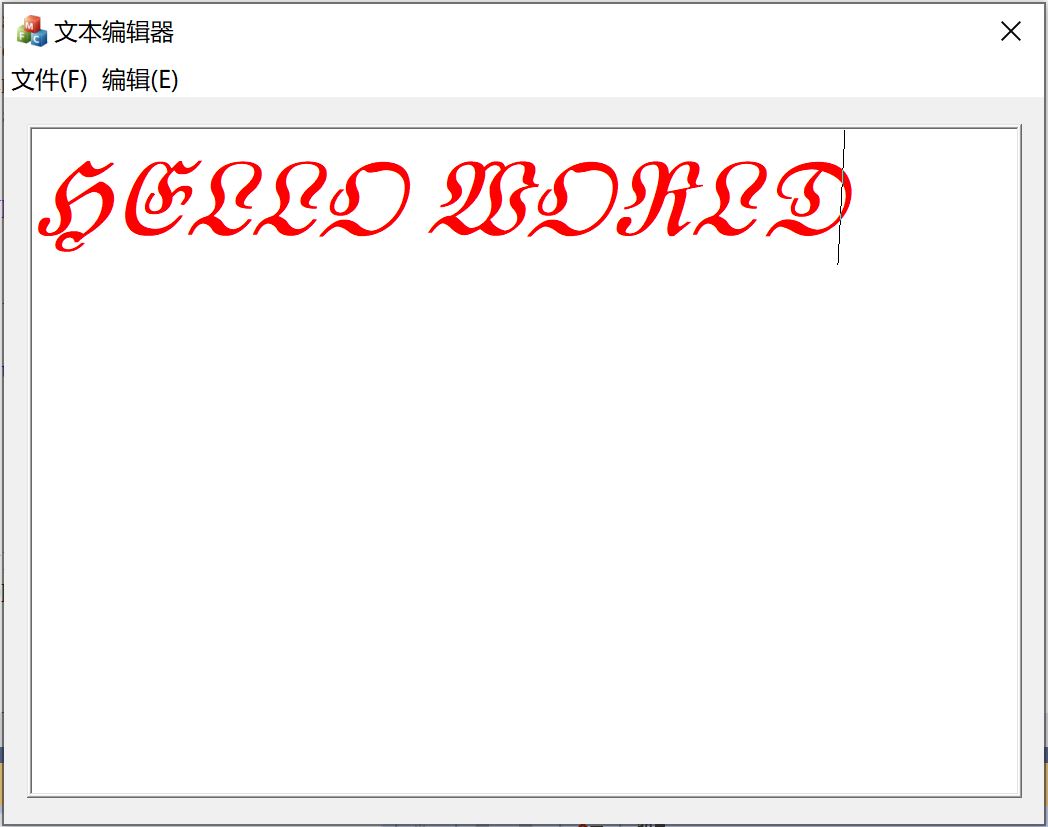
}

}









2. 请观看视频“图形用户接口设计(演示)”，完整实现学生管理系统的所有功能。

class student

{

public:

CString index;

CString name;

CString sex;

CString birthday;

CString hometown;

};

class Management

{

public:

CString add[100];

CString del[100];

CString sort[100];

};

void CStudentDlg::OnBnClickedMale()

{

m\_sex = 0;

}

void CStudentDlg::OnBnClickedFemale()

{

m\_sex = 1;

}

void CStudentDlg::OnBnClickedAdd()

{

UpdateData(true);

CString sex;

if (m\_sex == 0)

sex = "男";

else if (m\_sex == 1)

sex = "女";

m\_list.InsertItem(i, m\_index);

m\_list.SetItemText(i, 1, m\_name);

m\_list.SetItemText(i, 2, m\_sex);

m\_list.SetItemText(i, 3, m\_birthday);

m\_list.SetItemText(i, 4, m\_hometown);

MessageBox(\_T("添加成功！"));

UpdateData(false);

}

void CStudentDlg::OnBnClickedChange()

{

CListCtrl \*plist = (CListCtrl \*)GetDlgItem(IDC\_LIST);

POSITION pos = plist->GetFirstSelectedItemPosition();

int nSel=plist->GetNextSelectedItem(pos);

if (nSel < 0) {

MessageBox(\_T("请选中要修改的项！"), MB\_OK);

}

else {

if (AfxMessageBox(\_T("确认修改?"), MB\_YESNO) == IDYES) {

CString str;

GetDlgItemText(IDC\_INDEX, str);

plist->SetItemText(nSel, 0, str);

GetDlgItemText(IDC\_NAME, str);

plist->SetItemText(nSel, 1, str);

if (m\_sex == 0) {

GetDlgItemText(IDC\_MALE, str);

plist->SetItemText(nSel, 2, str);

}

else if (m\_sex == 1) {

GetDlgItemText(IDC\_FEMALE, str);

plist->SetItemText(nSel, 2, str);

}

GetDlgItemText(IDC\_BIRTH, str);

plist->SetItemText(nSel, 3, str);

GetDlgItemText(IDC\_HOME, str);

plist->SetItemText(nSel, 4, str);

}

}

}

void CStudentDlg::OnBnClickedDel()

{

POSITION pos = m\_list.GetFirstSelectedItemPosition();

if (pos == NULL)

return;

else {

while(pos) {

int nItem = m\_list.GetNextSelectedItem(pos);

m\_list.DeleteItem(nItem);

}

}

}

void CStudentDlg::SavetoFile(CString & strFilePath)

{

USES\_CONVERSION;

CFile mytxtFile;

CString strCaption,strMsg;

if (!mytxtFile.Open(strFilePath, CFile::modeCreate | CFile::modeReadWrite)) {

MessageBox(\_T("打开失败"));

return 0;

}

int width[5] = {0};

int i=0, j=0, nLen=0;

int nCount = m\_list.GetItemCount();

char format[512] = {0};

char buf[1024] = {0};

CString str0, str1, str2, str3, str4;

str0.LoadString(IDC\_INDEX);

width[0] = strlen(T2A(str0));

str1.LoadString(IDC\_NAME);

width[1]=strlen(T2A(str1));

if (m\_sex == 0) {

str2.LoadString(IDC\_MALE);

width[2] = strlen(T2A(str2));

}

else if (m\_sex == 1) {

str2.LoadString(IDC\_FEMALE);

width[2] = strlen(T2A(str2));

}

str3.LoadString(IDC\_BIRTH);

width[3] = strlen(T2A(str3));

str4.LoadString(IDC\_HOME);

width[4] = strlen(T2A(str4));

sprintf\_s(format, "%%-%ds %%-%ds %%-%ds %%-%ds %%-%ds %%-%ds \r\n"

,width[0], width[1], width[2], width[3], width[4]);

mytxtFile.Write(buf, strlen(buf));

if (DoModal()==IDOK) {

strPath=pFile->GetPathName();

}

CStdioFile file;

CString strLine, temp;

CStringArray strFile;

int count = m\_list.GetItemCount();

setlocale(LC\_CTYPE,("chs"));

student stu;

if (!file.Open(strPath,CFile::modeRead))

{

temp.Format(\_T("%s文件不存在!"), strPath);

AfxMessageBox(temp);

}

while(file.ReadString(strLine))

{

AfxExtractSubString(stu.index, strLine, 0, '-');

AfxExtractSubString(stu.name, strLine, 1, '-');

AfxExtractSubString(stu.sex, strLine, 2, '-');

AfxExtractSubString(stu.birthday, 3, '-');

AfxExtractSubString(stu.hometown, strLine, 4, '-');

m\_list.InsertItem(count, stu.index);

m\_list.SetItemText(count, 1, stu.name);

m\_list.SetItemText(count, 2, stu.sex);

m\_list.SetItemText(count, 3, stu.birthday);

m\_list.SetItemText(count, 4, stu.hometown);

}

file.Close();

}

3. 有一个MFC应用程序，运行过程中可以随时通过鼠标画不同颜色大小的圆形。

每个圆形以与水平成某角度做匀速直线运动，当遇到视图边界时反弹并继续运动。

// CBallView.h

#include <vector>

using namespace std;

const double walk = 5.0;

const double dirX[] = { walk, -walk, -walk, walk };

const double dirY[] = { -walk, -walk, walk, walk };

struct circle {

double x, y, r, r1, g1, b1, r2, g2, b2;

int dir;

circle(double x, double y, double r) {

this->x = x; this->y = y; this->r = r;

this->r1 = rand() % 256;

this->r2 = rand() % 256;

this->g1 = rand() % 256;

this->g2 = rand() % 256;

this->b1 = rand() % 256;

this->b2 = rand() % 256;

dir = 0;

}

};

extern vector<circle> circles;

class CBallView : public CView

{

……

public:

afx\_msg void OnLButtonDown(UINT nFlags, CPoint point);

afx\_msg void OnLButtonUp(UINT nFlags, CPoint point);

virtual void OnInitialUpdate();

afx\_msg void OnTimer(UINT\_PTR nIDEvent);

};

// CBallView.cpp

vector<circle> circles;

CBallView::CBallView() noexcept

{

srand(time(0));

}

void CBallView::OnDraw(CDC\* /\*pDC\*/)

{

CDC\* pDC = GetDC();

pDC->SelectStockObject(NULL\_BRUSH);

pDC->SetROP2(R2\_XORPEN);

for (int i = 0; i < circles.size(); i++) {

circle& c = circles[i];

CBrush fillbrush;

pDC->SelectObject(CPen(0, 1, RGB(c.r2, c.g2, c.b2)));

pDC->Ellipse(c.x - c.r, c.y - c.r, c.x + c.r, c.y + c.r);

fillbrush.CreateSolidBrush(RGB(c.r1, c.g1, c.b1));

pDC->SelectObject(fillbrush);

const int len = 5;

pDC->Ellipse(c.x - c.r + len, c.y - c.r + len, c.x + c.r - len, c.y + c.r - len);

}

ReleaseDC(pDC);

}

void CBallView::OnLButtonDown(UINT nFlags, CPoint point)

{

flag\_LBTNDown = true;

m\_pStart = point;

CView::OnLButtonDown(nFlags, point);

}

void CBallView::OnLButtonUp(UINT nFlags, CPoint point)

{

if (flag\_LBTNDown) {

CPoint center;

double nRadius;

center.x = (float(m\_pStart.x + point.x)) / 2;

center.y = (float(m\_pStart.y + point.y)) / 2;

nRadius = sqrt((double)(point.y - m\_pStart.y) \* (point.y - m\_pStart.y) +

(point.x - m\_pStart.x) \* (point.x - m\_pStart.x)) / 2;

circles.push\_back(circle(center.x, center.y, nRadius));

flag\_LBTNDown = false;

}

CView::OnMouseMove(nFlags, point);

}

void CBallView::OnInitialUpdate()

{

CView::OnInitialUpdate();

SetTimer(1, 10, NULL);

}

void CBallView::OnTimer(UINT\_PTR nIDEvent)

{

if (nIDEvent == 1) {

CRect rcClient;

GetClientRect(rcClient);

for (int i = 0; i < circles.size(); i++) {

circle& c = circles[i];

c.x += dirX[c.dir]; c.y += dirY[c.dir];

double left = c.x - c.r, right = c.x + c.r;

double up = c.y - c.r, down = c.y + c.r;

if (left <= 0) {

if (c.dir == 1) { c.dir = 0; continue; }

if (c.dir == 2) { c.dir = 3; continue; }

}

if (right >= rcClient.Width()) {

if (c.dir == 0) { c.dir = 1; continue; }

if (c.dir == 3) { c.dir = 2; continue; }

}

if (up <= 0) {

if (c.dir == 0) { c.dir = 3; continue; }

if (c.dir == 1) { c.dir = 2; continue; }

}

if (down >= rcClient.Height()) {

if (c.dir == 2) { c.dir = 1; continue; }

if (c.dir == 3) { c.dir = 0; continue; }

}

}

Invalidate();

}

CView::OnTimer(nIDEvent);

}

