

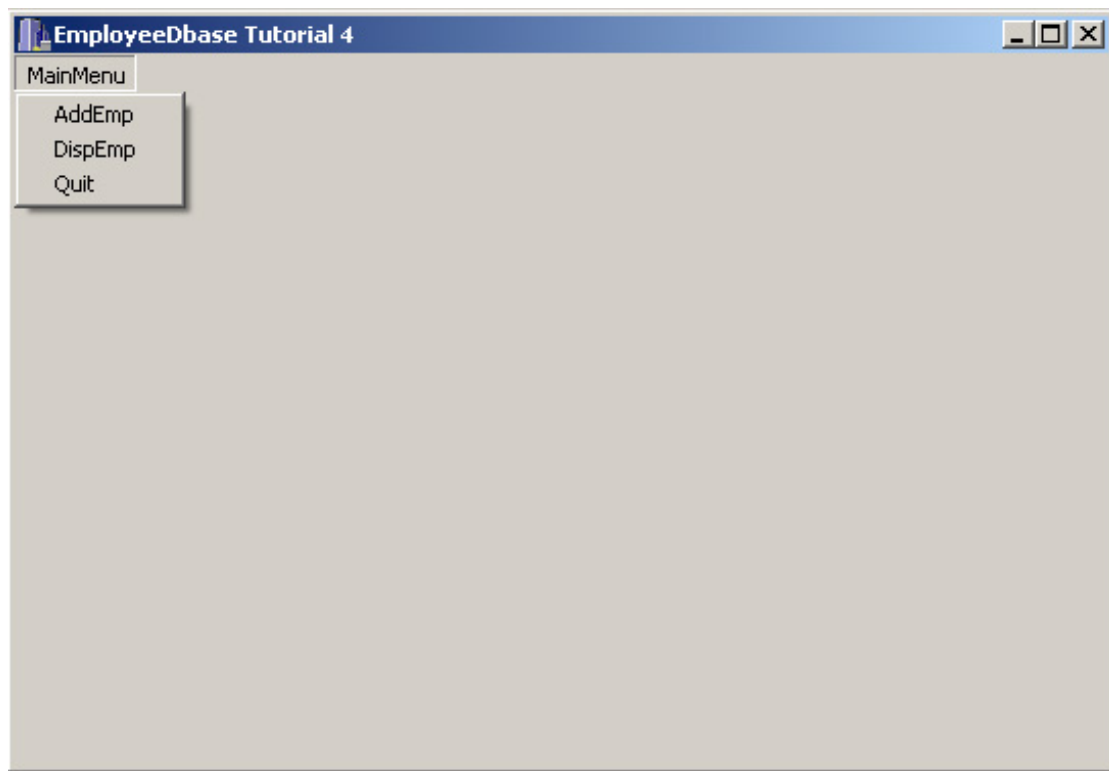
Tutorial 4 MT262

Tutor : Rifat Hamoudi
Staff No. : 00567451
Pager No. : 07669-801 509

I have put this tutorial on the web. This tutorial can be viewed and downloaded from <http://www.users.totalise.co.uk/~rifat> then selecting MT262 Tutorials then Tutorial 5.

Question 1

Write a menu driven software for the employee database and implement an event handler for quitting the software. Save the software as EmpDB and don't worry about implementing event handlers for adding and displaying employees for this tutorial. The graphical output should look similar to the figure below :



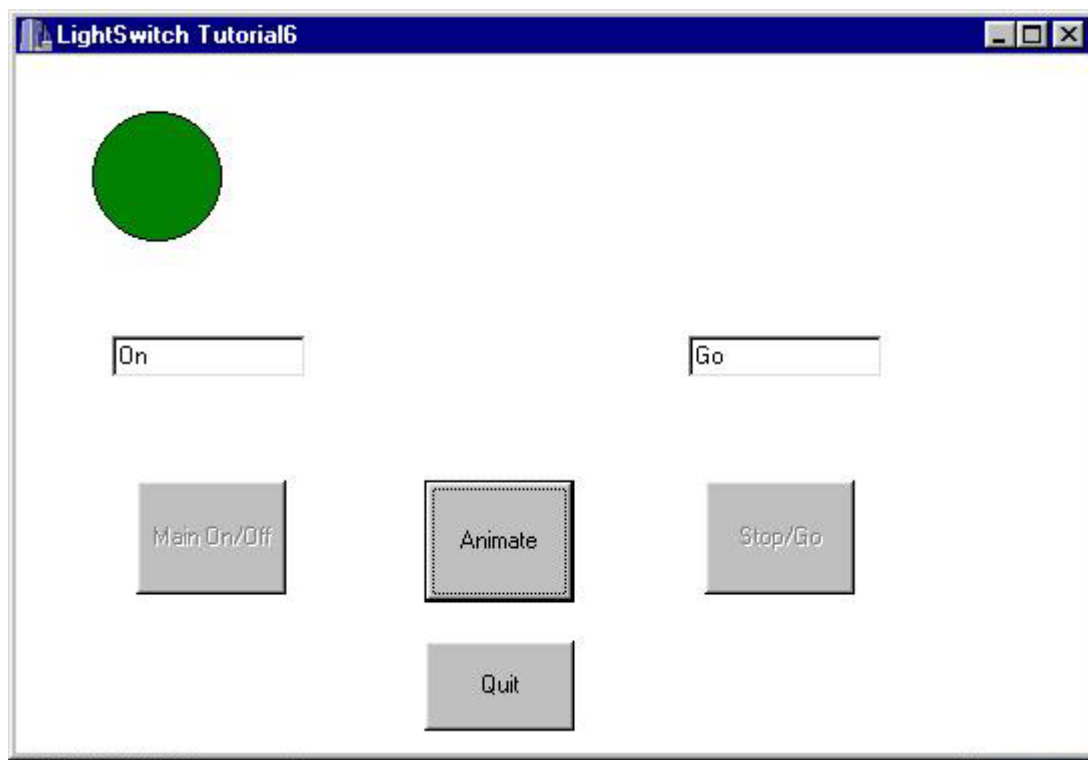
Question 2

Write an event driven software the has the following specifications :

The software should have four buttons :

- (a) Button called Main On/Off which when clicked will change the Form (background colour to black and display the word Off in the Edit box
- (b) Button called Stop/Go which when clicked will change the colour of the circle to red and display the Stop in the Edit box
- (c) Button called Animate which when clicked will move the Circle to the right of the form when it hits the boundary of the window it moves to the left, then right and so on until the user Quits the software
- (d) Button called Quit which when clicked will quit the software

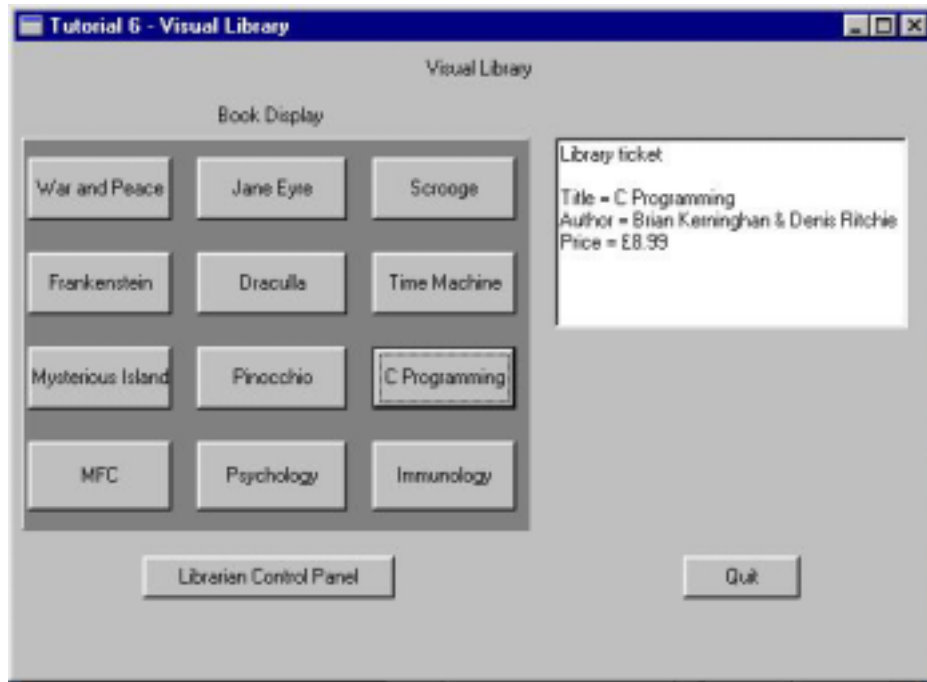
The graphical output should look similar to the figure below :



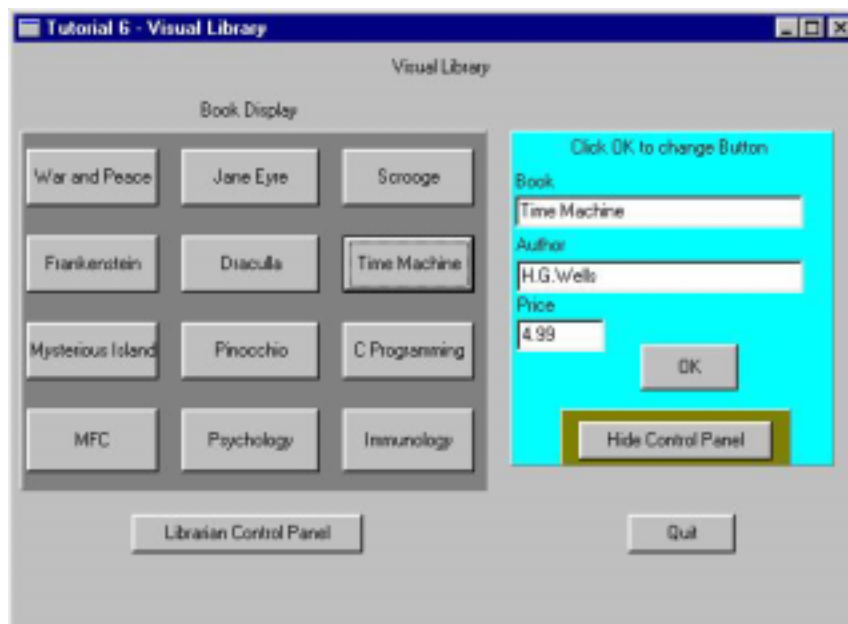
Question 3

Write a visual book library with upto 12 books (represented by clickable buttons) displayed on the shelves. The user should be able to view the Title, Author and Price of the book when then click on the respective button. Also there should be Librarian control panel that allows the Librarian to change the detail of the book.

The graphical output for the main program should look similar to the figure below :



The graphical output for the Librarian control panel should look similar to the figure below:



Answer to Question 1

To solve this do the following with C++ Builder :

- 1) Click on **File** then **New Application**
- 2) Save the project by clicking on **File** then **Save Project As**, call the .cpp file as EmpDBU and the project as EmpDB. **DO NOT** use the same name for both as this will crash the program and causes problems.
- 3) On the Form drag the menu button
- 4) Click on Form and change the **Caption** to EmployeeDatabase
- 5) Double click on the menu button on the Form and change the **Caption** to MainMenu
- 6) Press Enter and then change the Caption to **AddEmp** do the same for the rest
- 7) Click on Quit then on Events and double click on the right of **OnClick** this will take you to the event handler for what the program should do when the user presses the Quit button. Type Application->Terminate();
- 8) Save the program and run. This should give you your graphical database menu application.

Code for EmpDB software

EmpDB.cpp (generated by Builder)

```
//-----  
#include <vcl.h>  
#pragma hdrstop  
USERES("EmpDB.res");  
USEFORM("EmpDBU.cpp", MainForm);  
//-----  
WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)  
{  
    try  
    {  
        Application->Initialize();  
        Application->CreateForm(__classid(TMainForm), &MainForm);  
        Application->Run();  
    }  
    catch (Exception &exception)  
    {  
        Application->ShowException(&exception);  
    }  
    return 0;  
}  
//-----
```

EmpDBU.cpp

```
//-----  
#include <vcl.h>  
#pragma hdrstop  
  
#include "EmpDBU.h"  
//-----  
-----  
#pragma package(smart_init)  
#pragma resource "*.dfm"  
TMainForm *MainForm;  
//-----  
-----  
__fastcall TMainForm::TMainForm(TComponent* Owner)  
    : TForm(Owner)  
{  
}  
//-----  
-----  
  
void __fastcall TMainForm::QuitMClick(TObject *Sender)  
{  
    Application->Terminate();  
}  
//-----
```

EmpDBU.h

```
//-----  
#ifndef EmpDBUH  
#define EmpDBUH  
//-----  
#include <Classes.hpp>  
#include <Controls.hpp>  
#include <StdCtrls.hpp>  
#include <Forms.hpp>  
#include <Menus.hpp>  
//-----  
class TMainForm : public TForm  
{  
    __published:          // IDE-managed Components  
        TMainMenu *MainMenu1;  
        TMenuItem *MainMenu2;  
        TMenuItem *AddEmpM;  
        TMenuItem *DispEmpM;  
        TMenuItem *QuitM;  
        void __fastcall QuitMClick(TObject *Sender);  
private:                // User declarations  
public:                 // User declarations  
    __fastcall TMainForm(TComponent* Owner);  
};  
//-----  
extern PACKAGE TMainForm *MainForm;  
//-----  
#endif
```

Answer to Question 2

To solve this do the following with C++ Builder :

- 1) Click on **File** then **New Application**
- 2) Save the project by clicking on **File** then **Save Project As**, call the .cpp file as lightU and the project as light. **DO NOT** use the same name for both as this will crash the program and causes problems.
- 3) On the Form drag the button labelled OK
- 4) Click on Form and change the **Caption** according to specification
- 5) Double click on the relevant button on the Form and change the parameters according to specification
- 6) Click on relevant button in the component menu in object inspector then on events and double click on the right of **OnClick** this will take you to the event handler for what the program should do when the user presses the particular button. Type your event handler code there
- 7) Save the program and run. This should give you your graphical light switching and animation application.

Light code

Light.cpp

```
//-----  
#include <vcl.h>  
#pragma hdrstop  
USERES("light.res");  
USEFORM("lightU.cpp", LightForm);  
//-----  
WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)  
{  
    try  
    {  
        Application->Initialize();  
        Application->CreateForm(__classid(TLightForm), &LightForm);  
        Application->Run();  
    }  
    catch (Exception &exception)  
    {  
        Application->ShowException(&exception);  
    }  
    return 0;  
}  
//-----
```

LightU.cpp

```
//-----  
#include <vcl.h>  
#pragma hdrstop  
  
#include "lightU.h"  
//-----
```

```

#pragma package(smart_init)
#pragma resource "*.dfm"
TLightForm *LightForm;
//-----
__fastcall TLightForm::TLightForm(TComponent* Owner)
    : TForm(Owner)
{
    XMove = 5;
}
//-----
void __fastcall TLightForm::SwitchClick(TObject *Sender)
{
    if (Clock->Enabled)
        Clock->Enabled = false;

    if (LightForm->Color == clWhite)
    {
        LightForm->Color = clBlack;
        Status->Text = "Off";
    }
    else
    {
        LightForm->Color = clWhite;
        Status->Text = "On";
    }
}
//-----
void __fastcall TLightForm::TrafficLightClick(TObject *Sender)
{
    if (Clock->Enabled)
        Clock->Enabled = false;

    if (Circle->Brush->Color == clGreen)
    {
        Circle->Brush->Color = clRed;
        TrafficSign->Text = "Stop";
    }
    else
    {
        Circle->Brush->Color = clGreen;
        TrafficSign->Text = "Go";
    }
}
//-----
void __fastcall TLightForm::AnimateBClick(TObject *Sender)
{
    Clock->Enabled = true;
    Switch->Enabled = false;
    TrafficLight->Enabled = false;

    Circle->Brush->Color = clGreen;
    Circle->Left = Circle->Left + XMove;
    Clock->Interval = 50;

    if ((Circle->Left < 0) || (Circle->Left + Circle->Width) >
        (LightForm->ClientWidth - XMove))
        XMove = (-1) * XMove;
}
//-----

```

```

void __fastcall TLightForm::QuitBClick(TObject *Sender)
{
    Application->Terminate();
}
//-----

```

LightU.h

```

//-----
#ifndef lightUH
#define lightUH
//-----
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <ExtCtrls.hpp>
//-----
class TLightForm : public TForm
{
    __published:          // IDE-managed Components
        TButton *Switch;
        TEdit *Status;
        TShape *Circle;
        TButton *TrafficLight;
        TEdit *TrafficSign;
        TTimer *Clock;
        TButton *AnimateB;
        TButton *QuitB;
        void __fastcall SwitchClick(TObject *Sender);
        void __fastcall TrafficLightClick(TObject *Sender);

        void __fastcall AnimateBClick(TObject *Sender);
        void __fastcall QuitBClick(TObject *Sender);
private:                // User declarations
public:                 // User declarations
        int XMove;
        __fastcall TLightForm(TComponent* Owner);
};
//-----
extern PACKAGE TLightForm *LightForm;
//-----
#endif

```


Answer to Question 3

VisLib.cpp

```
//-----  
-  
#include <vcl.h>  
#pragma hdrstop  
USEFORM("VisLibU.cpp", Form1);  
USEUNIT("VLEng.cpp");  
//-----  
-  
WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)  
{  
    try  
    {  
        Application->Initialize();  
        Application->CreateForm(__classid(TForm1), &Form1);  
        Application->Run();  
    }  
    catch (Exception &exception)  
    {  
        Application->ShowException(&exception);  
    }  
    return 0;  
}  
//-----
```

VisLibU.h

```
//-----  
#ifndef VLUH  
#define VLUH  
//-----  
#include <Classes.hpp>  
#include <Controls.hpp>  
#include <StdCtrls.hpp>  
#include <Forms.hpp>  
#include <ExtCtrls.hpp>  
#include "VLEng.h"  
//-----  
class TForm1 : public TForm  
{  
    __published:          // IDE-managed Components  
        TPanel *Panel;  
        TButton *B0;  
        TButton *B1;  
        TButton *B2;  
        TButton *B3;  
        TButton *B4;  
        TButton *B5;  
        TButton *B6;  
        TButton *B7;  
        TButton *B8;  
        TButton *B9;  
        TButton *B10;  
        TButton *B11;  
        TButton *Button13;  
        TPanel *ControlPanel;  
        TEdit *DescBox;
```

```

    TEdit *CostBox;
    TLabel *Label1;
    TLabel *Label2;
    TLabel *Label3;
    TButton *OKButton;
    TMemo *Display;
    TPanel *Panel1;
    TButton *Button1;
    TEdit *IdEdit;
    TLabel *Label4;
    TLabel *Label5;
    TLabel *bookdis;
    TButton *QuitButton;
    void __fastcall B0Click(TObject *Sender);
    void __fastcall FormCreate(TObject *Sender);
    void __fastcall Button13Click(TObject *Sender);
    void __fastcall OKButtonClick(TObject *Sender);
    void __fastcall FormActivate(TObject *Sender);
    void __fastcall QuitButtonClick(TObject *Sender);
private:    // User declarations
    VLType MyVL;
    void ProcessButton(int ButtonId);
    void SetDisplay(int ButtonId);
    AnsiString SetCaption(int ButtonId);
    int CurrentTag;
public:    // User declarations
    __fastcall TForm1(TComponent* Owner);
};
//-----
extern PACKAGE TForm1 *Form1;
//-----
#endif

```

VisLib.cpp

```

//-----
#include <vcl.h>
#pragma hdrstop

#include "VisLibU.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
AnsiString TForm1::SetCaption(int ButtonId)
{
    MyVL.SetSelection(ButtonId);
    return MyVL.GetTitle();
}

void TForm1::ProcessButton(int ButtonId)
{
    CurrentTag = ButtonId;
    MyVL.SetSelection(ButtonId);
}

```

```

    Display->Clear();
    Display->Lines->Add("Library ticket");
    Display->Lines->Add("");
    Display->Lines->Add("Title = " + MyVL.GetTitle());
    Display->Lines->Add("Author = " + MyVL.GetAuthor());
    Display->Lines->Add("Price = £" + MyVL.GetBookCost());
};

void TForm1::SetDisplay(int ButtonId)
{
    CurrentTag = ButtonId;
    MyVL.SetSelection(ButtonId);
    IdEdit->Text = MyVL.GetTitle();
    DescBox->Text = MyVL.GetAuthor();
    CostBox->Text = MyVL.GetBookCost();
}
//-----

void __fastcall TForm1::FormCreate(TObject *Sender)
{
    MyVL.Initialise("VisLibData.dat");
}
//-----

void __fastcall TForm1::B0Click(TObject *Sender)
{
    if (!ControlPanel->Visible)
        ProcessButton(((TButton*)Sender)->Tag);
    else
        SetDisplay(((TButton*)Sender)->Tag);
    ((TButton*)Sender)->Caption = MyVL.GetTitle();
}

void __fastcall TForm1::Button13Click(TObject *Sender)
{
    ControlPanel->Visible = !(ControlPanel->Visible) ;
    Display->Clear();
    Display->Visible = !(Display->Visible);
    MyVL.SaveData("VLData.dat");
}
//-----

void __fastcall TForm1::OKButtonClick(TObject *Sender)
{
    MyVL.SetTitle(IdEdit->Text);
    MyVL.SetAuthor(DescBox->Text);
    MyVL.SetBookCost(float(CostBox->Text.ToDouble()));
    switch (CurrentTag)
    {
        case 0: B0->Caption = IdEdit->Text;
                break;
        case 1: B1->Caption = IdEdit->Text;
                break;
        case 2: B2->Caption = IdEdit->Text;
                break;
        case 3: B3->Caption = IdEdit->Text;
                break;
        case 4: B4->Caption = IdEdit->Text;
                break;
        case 5: B5->Caption = IdEdit->Text;
                break;
        case 6: B6->Caption = IdEdit->Text;
    }
}

```

```

        break;
    case 7: B7->Caption = IdEdit->Text;
        break;
    case 8: B8->Caption = IdEdit->Text;
        break;
    case 9: B9->Caption = IdEdit->Text;
        break;
    case 10: B10->Caption = IdEdit->Text;
        break;
    case 11: B11->Caption = IdEdit->Text;
        break;
    }
}
//-----

void __fastcall TForm1::FormActivate(TObject *Sender)
{
    B0->Caption = SetCaption(B0->Tag);
    B1->Caption = SetCaption(B1->Tag);
    B2->Caption = SetCaption(B2->Tag);
    B3->Caption = SetCaption(B3->Tag);
    B4->Caption = SetCaption(B4->Tag);
    B5->Caption = SetCaption(B5->Tag);
    B6->Caption = SetCaption(B6->Tag);
    B7->Caption = SetCaption(B7->Tag);
    B8->Caption = SetCaption(B8->Tag);
    B9->Caption = SetCaption(B9->Tag);
    B10->Caption = SetCaption(B10->Tag);
    B11->Caption = SetCaption(B11->Tag);
    Display->Clear();
}
//-----

void __fastcall TForm1::QuitButtonClick(TObject *Sender)
{
    Application->Terminate();
}
//-----

```

VLEng.h

```

//-----
#ifndef VLEngH
#define VLEngH
//-----
class VLType
{
private:
    struct VLRecord
    {
        AnsiString Title; // Button label for book
        AnsiString Author;
        float Cost;
    };
    VLRecord List[20];
    int SelectedItem;

public:
    void Initialise(AnsiString FileName);
    void SetSelection(int SelectionNumber);
}

```

```

void SetTitle(AnsiString NewBook);
AnsiString GetTitle(void);
void SetAuthor(AnsiString NewAuth);
AnsiString GetAuthor(void);
void SetBookCost(float NewCost);
AnsiString GetBookCost(void);
AnsiString GetPrice(void);
void SaveData(AnsiString FileName);
};
#endif

```

VLEng.cpp

```

//-----
-----
#include <vcl.h>
#include <fstream.h>
#pragma hdrstop
#include "VLEng.h"

//-----
-----
#pragma package(smart_init)
void VLType::Initialise(AnsiString FileName)
{
int Index;
int NumItems;
TStringList *S1;
for (Index = 0; Index < 12; Index = Index + 1)
{
List[Index].Title = "";
List[Index].Author = "None";
List[Index].Cost = 0;
}
NumItems = 0;
if (FileExists(FileName))
{
S1 = new TStringList();
S1->LoadFromFile(FileName);
NumItems = S1->Count / 3;
if (NumItems > 20)
NumItems = 20;
if (NumItems > 0)
for (Index = 0; Index < NumItems; Index = Index + 1)
{
List[Index].Title = S1->Strings[3*Index];
List[Index].Author = S1->Strings[3*Index + 1];
List[Index].Cost = S1->Strings[3*Index + 2].ToDouble();
}
delete S1;
}
SelectedItem = 0;
}

void VLType::SetSelection(int SelectionNumber)
{
SelectedItem = SelectionNumber;
}

void VLType::SetTitle(AnsiString NewBook)

```

```

{
    List[SelectedItem].Title = NewBook;
}

AnsiString VLType::GetTitle(void)
{
    return List[SelectedItem].Title;
}

void VLType::SetAuthor(AnsiString NewAuth)
{
    List[SelectedItem].Author = NewAuth;
}

AnsiString VLType::GetAuthor(void)
{
    return(List[SelectedItem].Author);
}

void VLType::SetBookCost(float NewCost)
{
    List[SelectedItem].Cost = NewCost;
}

AnsiString VLType::GetBookCost(void)
{
    return(FormatFloat("0.00",List[SelectedItem].Cost));
}

void VLType::SaveData(AnsiString FileName)
{
    TStringList *S2;
    int Index;
    S2 = new TStringList();
    for(Index = 0; Index < 20; Index = Index + 1)
    {
        S2->Add(List[Index].Title);
        S2->Add(List[Index].Author);
        S2->Add(AnsiString(List[Index].Cost));
    }
    S2->SaveToFile(FileName);
    delete S2;
}

```