

**ТЕХНИЧЕСКИ УНИВЕРСИТЕТ – ВАРНА**

**Курсов проект**

**по Програмни системи**

*Изготвили:*

*Карина Светлославова Щерева*

*6 група СИТ*

*18621824*

*Бурчин Рейханова Мустафова*

*6 група СИТ*

*18621810*

*Ива Илкова Николова*

*6 група СИТ*

*18621819*

**Задание:**

* Програма и класове, които да реализират прост диалог с потребителя от тип Ти си..., Аз съм..., напр.

Вход: Ти си прост компютър

Изход: Аз съм умен компютър или

Изход Ти си прост потребител

* Средства:

Windows

MS VS ++

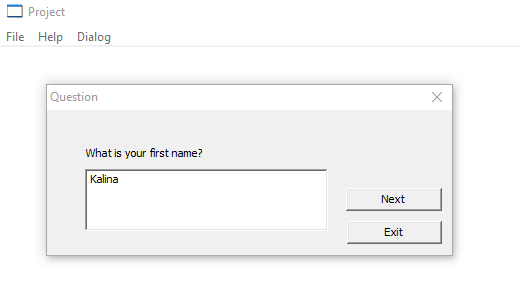
**Разделяне на заданието:**

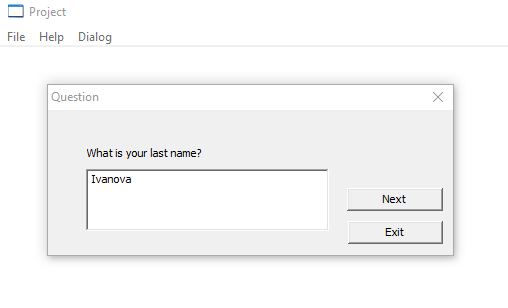
* Карина Щерева
* Dialog -> IDD\_QUESTION1
* WndProc -> case WM COMMAND -> case IDM\_DIALOG\_START
* Ива Николова
* void InitQuestion()
* void ProcessedQuestion()
* Бурчин Мустафова
* Start -> case WM COMMAND -> case IDNEXT1
* Start -> case WM COMMAND -> case IDEXIT1

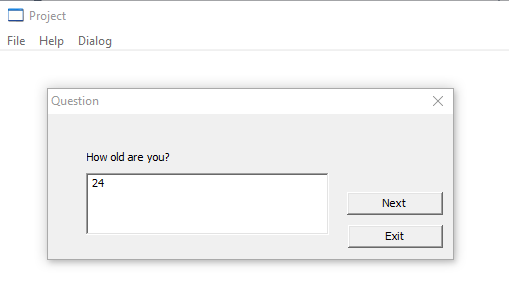
**Описание на заданието:**

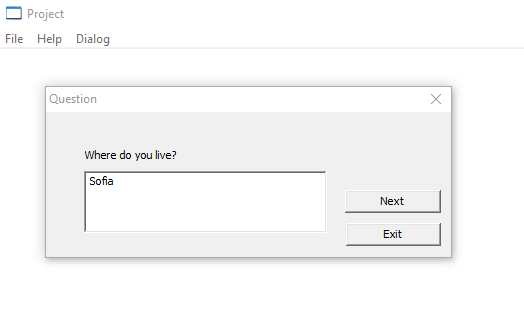
Идеята на програмата е копютърът да задава определени въпроси на потребителя, които са записани предварително в масив и се извикват един по един. След отговарянето им, комютъра си заделя информацията която му е нужда и връща в Message Box преработените данни.

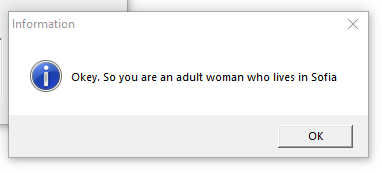
**Резултати:**

****

****

****

****

****

**Код:**

// Project.cpp : Defines the entry point for the application.

//

#include "stdafx.h"

#include "Project.h"

#include "commctrl.h"

#include <string>

#include <iostream>

#include "CommDlg.h"

#include "winuser.h"

using namespace std;

#define MAX\_LOADSTRING 100

// Global Variables:

HINSTANCE hInst; // current instance

TCHAR szTitle[MAX\_LOADSTRING]; // The title bar text

TCHAR szWindowClass[MAX\_LOADSTRING]; // the main window class name

// Forward declarations of functions included in this code module:

ATOM MyRegisterClass(HINSTANCE hInstance);

BOOL InitInstance(HINSTANCE, int);

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

INT\_PTR CALLBACK About(HWND, UINT, WPARAM, LPARAM);

INT\_PTR CALLBACK Start(HWND, UINT, WPARAM, LPARAM);

int APIENTRY \_tWinMain(HINSTANCE hInstance,

HINSTANCE hPrevInstance,

LPTSTR lpCmdLine,

int nCmdShow)

{

UNREFERENCED\_PARAMETER(hPrevInstance);

UNREFERENCED\_PARAMETER(lpCmdLine);

// TODO: Place code here.

MSG msg;

HACCEL hAccelTable;

// Initialize global strings

LoadString(hInstance, IDS\_APP\_TITLE, szTitle, MAX\_LOADSTRING);

LoadString(hInstance, IDC\_PROJECT, szWindowClass, MAX\_LOADSTRING);

MyRegisterClass(hInstance);

// Perform application initialization:

if (!InitInstance (hInstance, nCmdShow))

{

return FALSE;

}

hAccelTable = LoadAccelerators(hInstance, MAKEINTRESOURCE(IDC\_PROJECT));

// Main message loop:

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

{

if (!TranslateAccelerator(msg.hwnd, hAccelTable, &msg))

{

TranslateMessage(&msg);

DispatchMessage(&msg);

}

}

return (int) msg.wParam;

}

//

// FUNCTION: MyRegisterClass()

//

// PURPOSE: Registers the window class.

//

// COMMENTS:

//

// This function and its usage are only necessary if you want this code

// to be compatible with Win32 systems prior to the 'RegisterClassEx'

// function that was added to Windows 95. It is important to call this function

// so that the application will get 'well formed' small icons associated

// with it.

//

ATOM MyRegisterClass(HINSTANCE hInstance)

{

WNDCLASSEX wcex;

wcex.cbSize = sizeof(WNDCLASSEX);

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

wcex.lpfnWndProc = WndProc;

wcex.cbClsExtra = 0;

wcex.cbWndExtra = 0;

wcex.hInstance = hInstance;

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

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

wcex.hbrBackground = (HBRUSH)(COLOR\_WINDOW+1);

wcex.lpszMenuName = MAKEINTRESOURCE(IDC\_PROJECT);

wcex.lpszClassName = szWindowClass;

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

return RegisterClassEx(&wcex);

}

//

// FUNCTION: InitInstance(HINSTANCE, int)

//

// PURPOSE: Saves instance handle and creates main window

//

// COMMENTS:

//

// In this function, we save the instance handle in a global variable and

// create and display the main program window.

//

BOOL InitInstance(HINSTANCE hInstance, int nCmdShow)

{

HWND hWnd;

hInst = hInstance; // Store instance handle in our global variable

hWnd = CreateWindow(szWindowClass, szTitle, WS\_OVERLAPPEDWINDOW,

CW\_USEDEFAULT, 0, CW\_USEDEFAULT, 0, NULL, NULL, hInstance, NULL);

if (!hWnd)

{

return FALSE;

}

ShowWindow(hWnd, nCmdShow);

UpdateWindow(hWnd);

return TRUE;

}

//

// FUNCTION: WndProc(HWND, UINT, WPARAM, LPARAM)

//

// PURPOSE: Processes messages for the main window.

//

// WM\_COMMAND - process the application menu

// WM\_PAINT - Paint the main window

// WM\_DESTROY - post a quit message and return

//

//

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

{

int wmId, wmEvent;

PAINTSTRUCT ps;

HDC hdc;

switch (message)

{

case WM\_COMMAND:

wmId = LOWORD(wParam);

wmEvent = HIWORD(wParam);

// Parse the menu selections:

switch (wmId)

{

case IDM\_ABOUT:

DialogBox(hInst, MAKEINTRESOURCE(IDD\_ABOUTBOX), hWnd, About);

break;

case IDM\_EXIT:

DestroyWindow(hWnd);

break;

case IDM\_DIALOG\_START:

DialogBox(hInst, MAKEINTRESOURCE(IDD\_QUESTION1), hWnd, Start);

break;

default:

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

}

break;

case WM\_PAINT:

hdc = BeginPaint(hWnd, &ps);

// TODO: Add any drawing code here...

EndPaint(hWnd, &ps);

break;

case WM\_DESTROY:

PostQuitMessage(0);

break;

default:

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

}

return 0;

}

// Message handler for about box.

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

{

UNREFERENCED\_PARAMETER(lParam);

switch (message)

{

case WM\_INITDIALOG:

return (INT\_PTR)TRUE;

case WM\_COMMAND:

if (LOWORD(wParam) == IDOK || LOWORD(wParam) == IDCANCEL)

{

EndDialog(hDlg, LOWORD(wParam));

return (INT\_PTR)TRUE;

}

break;

}

return (INT\_PTR)FALSE;

}

#define MAX\_QUESTION 4

char szQuestions[MAX\_QUESTION][50] = {"What is your first name?", "What is your last name?",

"How old are you?","Where do you live?"};

int iNum = 0;

#define MAX\_ANSWER 4

char szАnswers[MAX\_ANSWER][30] = {};

void InitQuestion(HWND hDlg,int i){

SetDlgItemText(hDlg,IDC\_Q1,szQuestions[i]);

SetDlgItemText(hDlg,IDC\_A1,"");

}

void ProcessedQuestion(HWND hDlg,int i){

char buf[60];

int len = 0;

int age = 0;

len = GetDlgItemText(hDlg, IDC\_A1, buf, 60);

len = strlen(buf);

switch(i){

case 0:

strcpy(szАnswers[0], buf);

break;

case 1:

if(buf[len-1] == 'a') {

strcpy(szАnswers[1], "woman");

} else if(buf[len-1] == 'v'){

strcpy(szАnswers[1], "man");

} else

strcpy(szАnswers[1], "...");

break;

case 2:

age = atoi(buf);

if(age >= 18){

strcpy(szАnswers[2], "an adult");

}else if((age < 18 ) && (age > 0)){

strcpy(szАnswers[2], "a minor");

}else

strcpy(szАnswers[2], "...");

break;

case 3:

strcpy(szАnswers[3], buf);

break;

}

}

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

{

int mID=LOWORD(wParam);

char szResult[256];

#define SW\_SHOW = DialogBox(hInst, MAKEINTRESOURCE(IDD\_QUESTION2), hWnd, Start);

UNREFERENCED\_PARAMETER(lParam);

switch (message)

{

case WM\_INITDIALOG:

InitQuestion(hDlg,iNum);

return (INT\_PTR)TRUE;

break;

case WM\_COMMAND:

switch (mID)

{

case IDNEXT1:

if(iNum < (MAX\_QUESTION)){

ProcessedQuestion(hDlg,iNum);

iNum++;

InitQuestion(hDlg,iNum);

}

if((iNum + 1) > MAX\_QUESTION){

sprintf(szResult,"Okey. So you are %s %s who lives in %s",szАnswers[2],szАnswers[1],szАnswers[3]);

MessageBox(hDlg, szResult,\_T("Information"), MB\_OK | MB\_ICONINFORMATION);

}

break;

case IDEXIT1:

PostQuitMessage(0);

break;

default:

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

}

break;

}

return (INT\_PTR)FALSE;

}