

**МУНИЦИПАЛЬНОЕ БЮДЖЕТНОЕ
ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ГОРОДСКОГО ОКРУГА БАЛАШИХА
«ГИМНАЗИЯ №2»**

ПРОЕКТ

на тему «Идентификация пользователя. Разработка личного кабинета»

Шорина Евгения Максимовича,
ученика 9 «Г» класса

Руководитель проекта:

Терехова Елена Дмитриевна
учитель ИКТ и Информатики

2019 г

Введение

Мне еще в 8 классе было интересно как работает авторизация и как программы определяют правильности ввода данных. Я захотел написать свою программу, которая помогла бы мне разобраться в этом. Начал я с изучения самого популярного языка программирования C++, это было очень сложно, но я в нем разобрался и в этом проекте смогу рассказать немного о нем. Эту программу я писал несколько дней, я много ошибался, но в всё-таки достиг нужного результата. Я сумел написать программу, которая может запомнить уникальные данные пользователь (логин и пароль) и в дальнейшем пользователь сможет использовать эти данные для авторизации.

Проблема

Часто можно услышать выражение, что какой-то человек «авторизован» для выполнения данной операции. Но проблема заключается в том, что многие люди не знают, как работает авторизация и насколько это сложный процесс. Я решил в этом разобраться и написать программу, облегчающую этот процесс.

Цель проекта

Написать программу для авторизации пользователя.

Задачи проекта

- 1) Узнать немного о языке C++.
- 2) Узнать, что такое WINAPI.
- 3) Разработка программы.
- 4) Какие переменные существуют и что они означают.
- 5) Проверка работоспособности программы.
- 6) Сформулировать вывод.

Что такое C++

C++ - это язык программирования который сочетает в себе компиляцию и статическую типизацию общего назначения.

Компилируемый язык программирования – Это такие языки, в которых исходный код преобразуется в машинный код с помощью компилятора.

Статическая типизация – Это широко используемая функция для языков программирования, в котором переменная тесно связывается с типом и тип невозможно будет изменить позже.

Этот язык поддерживает многие парадигмы программирования, такие как:

- 1) Процедурное программирование.
- 2) Объектно-ориентированное программирование.
- 3) Обобщённое программирование.

Парадигмы программирования – Это объединение многих идей и понятий, которые определяют в каком стиле будет написана программа.

C++ сочетает в себе многие функции высокоуровневых и низкоуровневых языков. Он является одним из самых популярных языков программирования. С его помощью разрабатывают многие программы, операционные системы, драйвера, серверов и игр.

C++ существует в многих реализаций, например:

- 1) GCC.
- 2) Visual C++.
- 3) Intel C++.
- 4) C++ Builder.

Также C++ оказал значимое влияние на другие языки программирования, но в первую очереди на Java и C#

C++ Имеет два вида. Это Console и GUI, которые кардинально различаются, начиная с команд, а заканчивая дизайном.

Что такое WINAPI

WINAPI – Это наименование базовых функций для устройств на базе Windows от компании Microsoft. Это самый лёгкий способ взаимодействия пользователя с этой ОС. Для людей, которые любят создавать программы корпорация Microsoft создала PlatformSDK, которое содержит набор документации и утилит нужных для разработки.

WINAPI – Это самый близкий к ОС Windows способ взаимодействия с ней. А для облегчения переноса на другие ОС программ, написанных с помощью данной утилиты, используют библиотеку Wine.

Wine – Свободное ПО, которое помогает перенести программу с ОС Windows на другие ОС, например macOS.

Версии WINAPI:

- 1) Win16 – Самая первая версия WINAPI для x16 разрядных версий Windows.
- 2) Win32 – WINAPI для современных систем Windows, базируемых на x32 разрядной версии.
- 3) Win32s – Подтип относящийся к Win32. Которое дает возможность использовать некоторые функции Win32 для Win16.
- 4) Win64 – Одно из версий Win32, в которую добавили некоторые функции. Базируется на платформах Windows с x86-64 разрядной системы.

Ход работы

Для начала необходимо подключить библиотеку `Windows.h` (`#include <Windows.h>`), чтобы использовать WINAPI.

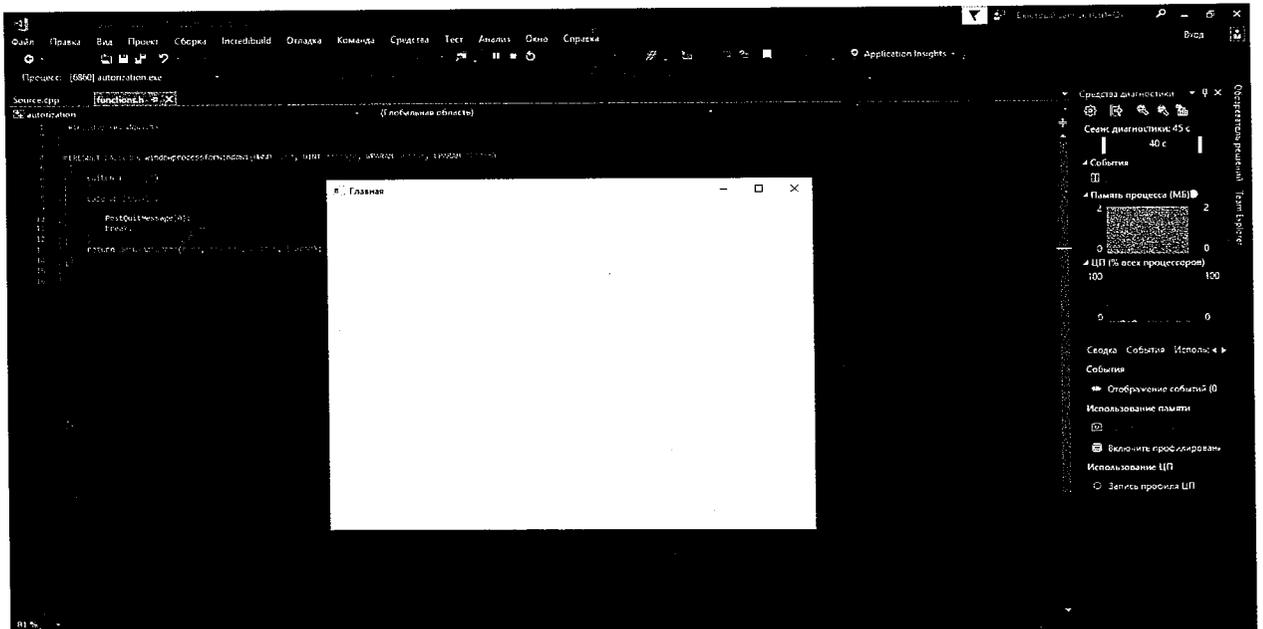
Дальше нам нужно создать и зарегистрировать новое окно

```
1 #include <windows.h>
2
3 int WINAPI WinMain(HINSTANCE hInst, HINSTANCE hPrevInst, LPSTR lpCmdLine, int nCmdShow)
4 {
5     MSG msg;
6     WNDCLASSEX windowclassforwindow1;
7     ZeroMemory(&windowclassforwindow1, sizeof(WNDCLASSEX));
8     windowclassforwindow1.cbClsExtra = NULL;
9     windowclassforwindow1.cbWndExtra = NULL;
10    windowclassforwindow1.cbSize = sizeof(WNDCLASSEX);
11    windowclassforwindow1.hbrBackground = (HBRUSH)COLOR_WINDOW;
12    windowclassforwindow1.hCursor = LoadCursor(NULL, IDC_ARROW);
13    windowclassforwindow1.hIcon = NULL;
14    windowclassforwindow1.hIconSm = NULL;
15    windowclassforwindow1.hInstance = hInst;
16    windowclassforwindow1.lpszClassName = "Таблица";
17    windowclassforwindow1.lpszMenuName = NULL;
18    windowclassforwindow1.lpfnWndProc = WndProc;
19    windowclassforwindow1.style = CS_HREDRAW | CS_VREDRAW;
20
21    if (!RegisterClassEx(&windowclassforwindow1))
22    {
23        int nResult = GetLastError();
24        MessageBox(NULL, "Не удалось создать окно", "Error", MB_ICONERROR);
25    }
26
27    int w = 640;
28    int h = 480;
29    int x = (GetSystemMetrics(SM_CXSCREEN) - w) / 2;
30    int y = (GetSystemMetrics(SM_CYSCREEN) - h) / 2;
31
32    HWND handleforwindow1 = CreateWindowEx(NULL, windowclassforwindow1.lpszClassName, "Таблица", WS_OVERLAPPEDWINDOW, x, y, w, h, NULL, NULL, hInst, NULL);
33
34    if (!handleforwindow1)
35    {
36        int nResult = GetLastError();
37        MessageBox(NULL, "Не удалось создать окно", "Error", MB_ICONERROR);
38    }
39
40    ShowWindow(handleforwindow1, nCmdShow);
41    while (GetMessage(&msg, NULL, NULL, NULL))
42    {
43        TranslateMessage(&msg);
44        DispatchMessage(&msg);
45    }
46
47    return msg.wParam;
48 }
```

Если мы попытаемся так запустить нашу программу, то у нас ничего не получится. Потому что мы не написали процессы данного окна.

```
1 #include <windows.h>
2
3
4 HRESULT CALLBACK windowprocessforwindow1(HWND hwnd, UINT message, WPARAM wParam, LPARAM lParam)
5 {
6     switch (message)
7     {
8         case WM_DESTROY:
9
10         PostQuitMessage(0);
11         break;
12     }
13     return DefWindowProc(hwnd, message, wParam, lParam);
14 }
```

Теперь проверяем.



Окно создалось, значит можно работать дальше.

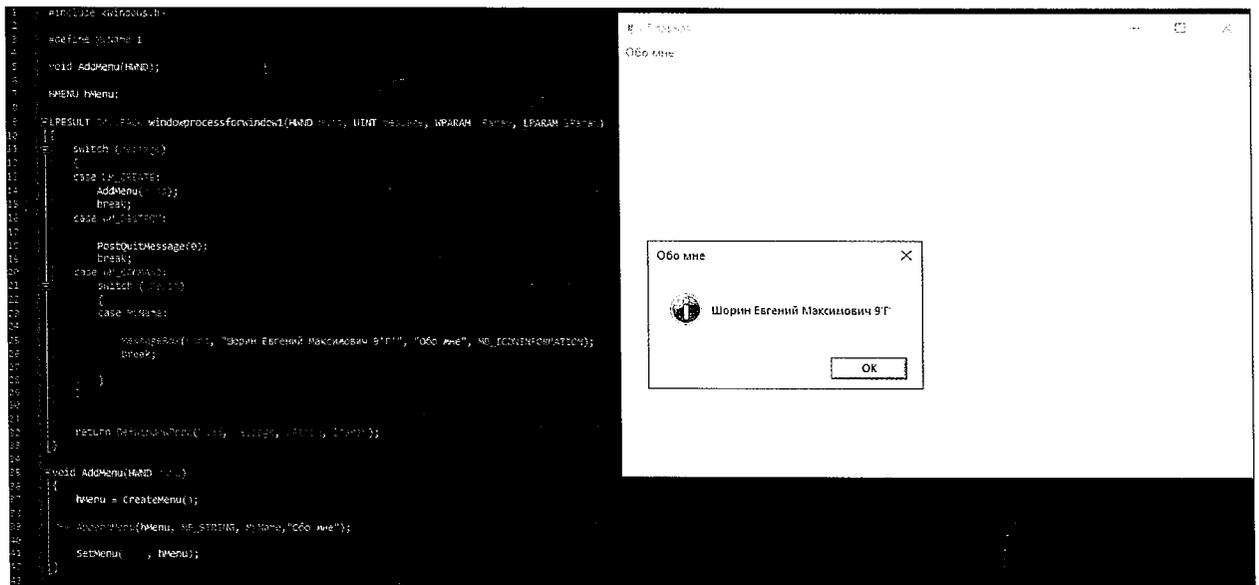
Теперь я добавлю в программу меню.

Для удобства я создам файл заголовка (functions.h)

```
functions.h
```

И подключу его к основному файлу (#include "functions.h")

В этот файл я буду записывать все функции и процессы, чтобы не было слишком много строк в основном файле (Source.cpp)



И еще несколько строк. Только осталось их практически так же зарегистрировать как и основное, но через void.

Пример: (2 окно)

```
80 void createwindow2(WNDCLASSEX& wc, HWND& hwnd, HINSTANCE hInst, int nWindowID) {
81     if (windowclass3registeredbefore == false) {
82         ZeroMemory(&wc, sizeof(WNDCLASSEX));
83         wc.cbClsExtra = NULL;
84         wc.cbSize = sizeof(WNDCLASSEX);
85         wc.cbWndExtra = NULL;
86         wc.hbrBackground = (HBRUSH)COLOR_WINDOW;
87         wc.hCursor = LoadCursor(NULL, IDC_ARROW);
88         wc.hIcon = NULL;
89         wc.hIconSm = NULL;
90         wc.hInstance = hInst;
91         wc.lpfnWndProc = (WNDPROC)windowprocessforwindow2;
92         wc.lpszClassName = "2 ОКНО";
93         wc.lpszMenuName = NULL;
94         wc.style = CS_HREDRAW | CS_VREDRAW;
95
96
97         if (!RegisterClassEx(&wc))
98         {
99             int nResult = GetLastError();
100             MessageBox(NULL, "Окно не было создано!", "ОШИБКА", MB_ICONERROR);
101         }
102         else
103             windowclass2registeredbefore = true;
104
105         int w = 450;
106         int h = 300;
107         int x = (GetSystemMetrics(SM_CXSCREEN) - w) / 2;
108         int y = (GetSystemMetrics(SM_CYSCREEN) - h) / 2;
109
110         hwnd = CreateWindowEx(NULL, wc.lpszClassName, "Меню", WS_OVERLAPPEDWINDOW, x, y, w, h, NULL, NULL, NULL, NULL);
111
112         if (!hwnd)
113         {
114             int nResult = GetLastError();
115             MessageBox(NULL, "Окно не было создано!", "ОШИБКА", MB_ICONERROR);
116         }
117
118         ShowWindow(hwnd, nWindowID);
119     }
120 }
```

Теперь добавим процессы этим окнам, пример 2 окно:

```
52 LRESULT CALLBACK windowprocessforwindow2(HWND hwnd, UINT message, WPARAM wParam, LPARAM lParam)
53 {
54     switch (message)
55     {
56     case WM_CREATE:
57         window2open = true;
58         break;
59
60     case WM_DESTROY:
61         window2open = false;
62         break;
63     }
64
65     switch (wParam)
66     {
67
68     case createwindowbuttoninwindow2:
69         DestroyWindow(hwnd);
70         window2openenum = window3;
71         break;
72     }
73
74     return DefWindowProc(hwnd, message, wParam, lParam);
75 }
```

Теперь можно начинать делать кнопки.

Путем добавления `createwindowbuttoninwindow1` и `createwindowbuttoninwindow2`, можно сделать функции для этих кнопок.

```
126 CreateWindow(L"Button", L"Авторизация", WS_VISIBLE | WS_CHILD | WS_BORDER, 340, 220, 140, 30, hwnd, (HMENU)createwindowbuttoninwindow1, NULL, NULL);
127 CreateWindow(L"Button", L"Регистрация", WS_VISIBLE | WS_CHILD | WS_BORDER, 170, 220, 140, 30, hwnd, (HMENU)createwindowbuttoninwindow2, NULL, NULL);
```

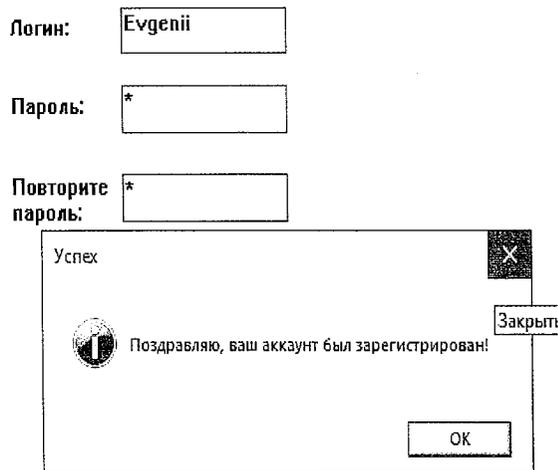
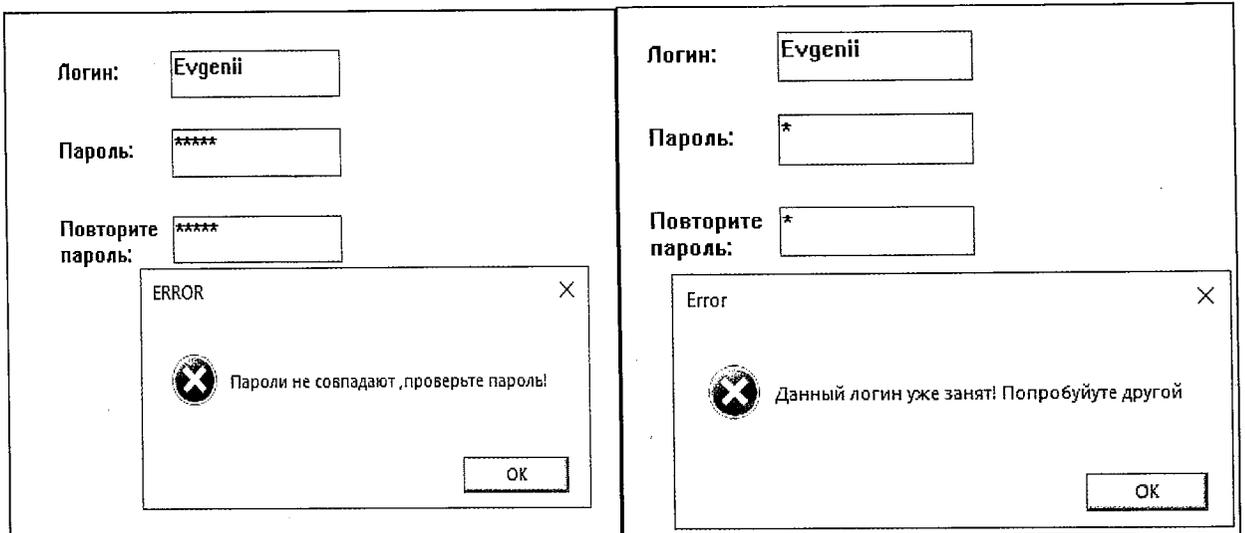
Для начала надо сделать окно регистрации.

```
void vRegistration(HWND hwnd);
```

```
void vRegistration(HWND hwnd)
{
    CreateWindow(L"static", L"Логин: ", WS_VISIBLE | WS_CHILD, 100, 115, 90, 30, hwnd, NULL, NULL, NULL);
    hLogin = CreateWindow(L"edit", L"", WS_VISIBLE | WS_CHILD | WS_BORDER | WS_TABSTOP, 260, 120, 90, 30, hwnd, NULL, NULL, NULL);
    CreateWindow(L"static", L"Пароль: ", WS_VISIBLE | WS_CHILD, 100, 165, 90, 30, hwnd, NULL, NULL, NULL);
    hPass = CreateWindow(L"edit", L"", WS_VISIBLE | WS_CHILD | WS_BORDER | ES_PASSWORD | WS_TABSTOP, 260, 170, 90, 30, hwnd, NULL, NULL, NULL);
    CreateWindow(L"static", L"Повторите пароль: ", WS_VISIBLE | WS_CHILD, 100, 215, 90, 30, hwnd, NULL, NULL, NULL);
    hPassv2 = CreateWindow(L"edit", L"", WS_VISIBLE | WS_CHILD | WS_BORDER | ES_PASSWORD | WS_TABSTOP, 260, 220, 90, 30, hwnd, NULL, NULL, NULL);
    CreateWindow(L"Button", L"Зарегистрироваться", WS_VISIBLE | WS_CHILD | WS_BORDER, 260, 310, 160, 30, hwnd, (HMENU)Registration, NULL, NULL);
}
```

```
#define Registration 2
121     case Registration:
122         char login[50], password[50], password2[50], buflog[50], bufpswd[50];
123
124
125         GetWindowText(hLogin, login, 50);
126         GetWindowText(hPass, password, 50);
127         GetWindowText(hPassv2, password2, 50);
128         ifstream fout;
129
130         fout.open("login.dll", ios_base::app);
131         for (i = 1; i < 1000; i++)
132         {
133             fout.getline(buflog, 50, ';');
134
135             fout.getline(bufpswd, 50, '\n');
136             if (strcmp(login, buflog) == 0)
137                 goto fail;
138         }
139         goto right;
140         goto right;
141
142     fail:MessageBox(hwnd, "Данный логин уже занят! Попробуйте другой", "Error", MB_ICONINFORMATION);
143         goto skip;
144     right:
145         if (strcmp(password2, password) == 0)
146         {
147             GetWindowText(hLogin, login, 50);
148             GetWindowText(hPass, password, 50);
149             ofstream fout;
150             fout.open("login.dll", ios_base::app);
151             fout << login << ';';
152             fout << password << '\n';
153             fout.close();
154             MessageBox(hwnd, "Поздравляю, ваш аккаунт был зарегистрирован!", "Успех", MB_ICONINFORMATION);
155             hParent = createwindowbuttoninwindow1;
156
157
158         }
159         else MessageBox(hwnd, "Пароли не совпадают, проверьте пароль!", "Error", MB_ICONERROR);
160         break;
```

Теперь у меня есть регистрация. Программа создает файл и записывает в него логин и пароль, благодаря этому можно войти в аккаунт.

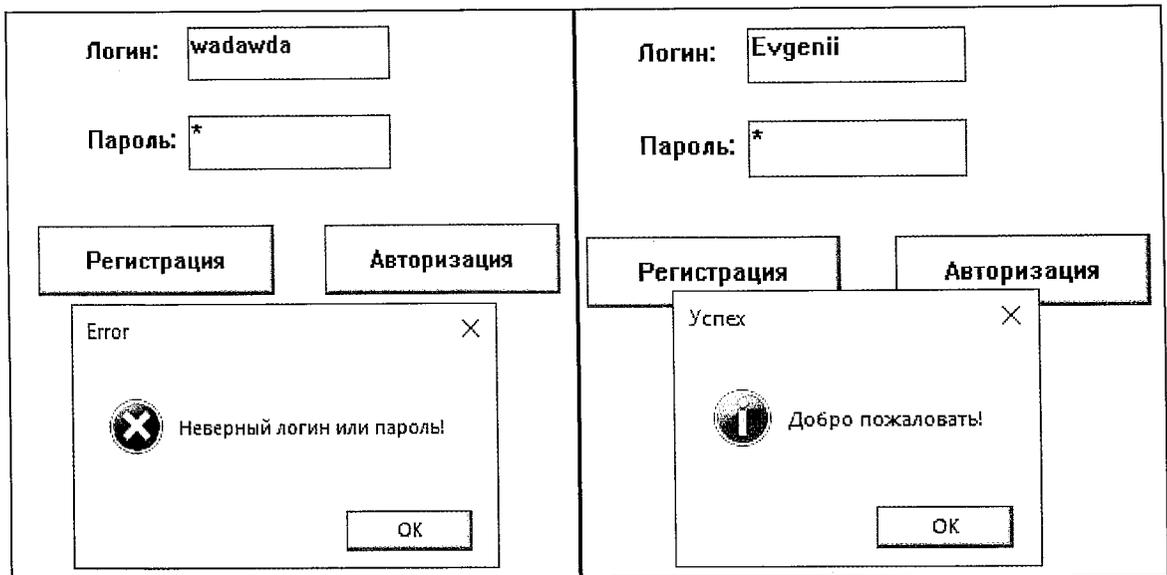


Теперь надо сделать авторизацию!

```

62 #define Buttonpass 4
63 case Buttonpass:
64     char bufpasswd[50];
65     char buflog[50];
66     char login[50];
67     char passwd[50];
68     ifstream fin("login.dll");
69     GetWindowText(hLogin, login, 50);
70     GetWindowText(hPass, passwd, 50);
71
72     for (i = 1; i < 1000; i++)
73     {
74         fin.getline(buflog, 50, ';');
75         fin.getline(bufpasswd, 50, '\n');
76         if (strcmp(login, buflog) == 0)
77             if (strcmp(passwd, bufpasswd) == 0)
78                 goto right;
79     }
80     goto error;
81 right:
82     MessageBox(hwnd, "Добро пожаловать!", "Успех", MB_ICONINFORMATION);
83     ..Param = createwindowbuttoninwindow;
84     break;
85 error:
86     MessageBox(hwnd, "Неверный логин или пароль!", "Error", MB_ICONERROR);
87     break;
88
89 CreateWindow(L"Button", L"Авторизация", WS_VISIBLE | WS_CHILD | WS_BORDER, 340, 220, 140, 38, hwnd, (HMENU)Buttonpass, NULL, NULL);

```



При авторизации программа находит в файле (созданный ранее при регистрации) логин и сверяет его с паролем, если пароль верный, тогда вы получаете доступ к аккаунту.

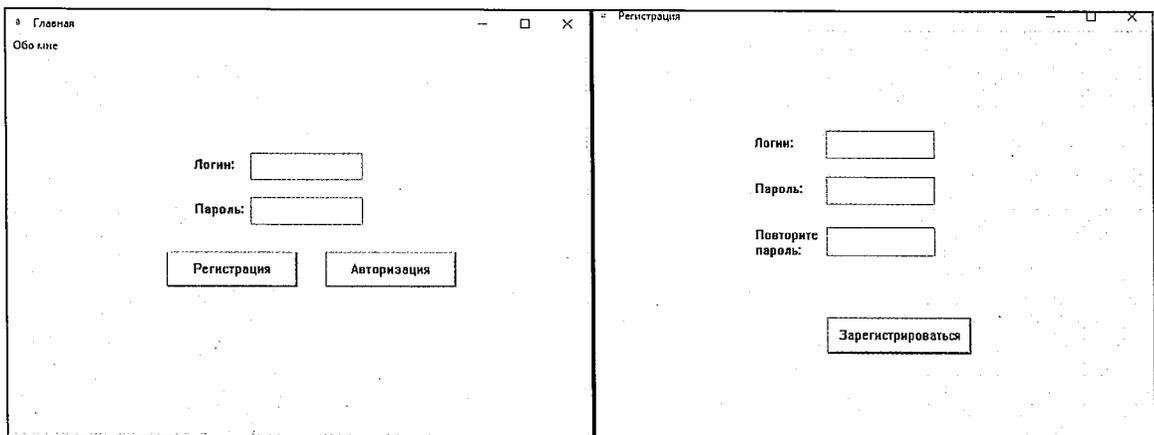
Теперь надо сделать красивый задний фон

```
hdc.hbrBackground = CreateSolidBrush( RGB(100, 200, 255));
```

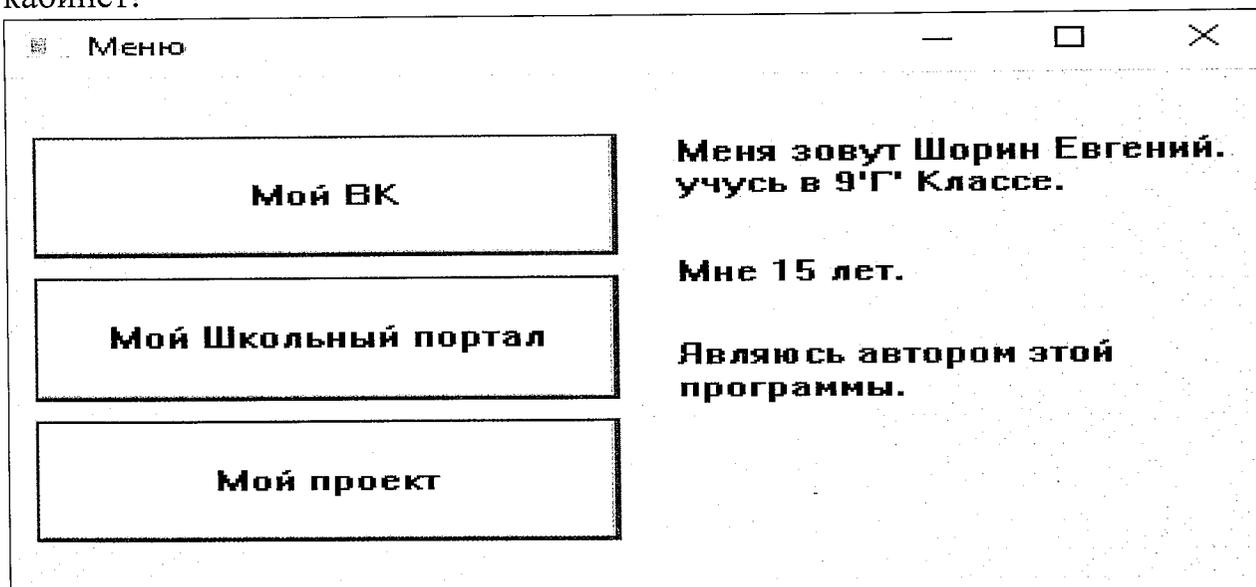
```
HDC hdcStatic;
```

```
case WM_CTLCOLORSTATIC:
    hdcStatic = (HDC)wParam;
    SetTextColor(hdcStatic, RGB(0, 0, 0));
    SetBkMode(hdcStatic, TRANSPARENT);

    return (LRESULT)GetStockObject(NULL_BRUSH);
```



Остается доделать личный кабинет!



Вывод

За год, который я потратил на изучение C++ и создания своей программы. Я узнал много нового. Например, я понял, что такое C++ и WINAPI, также я усвоил и запомнил много новой информации и различные команды. Я научился создавать окна, печатать текст и делать кнопки, которые могли открыть новую вкладку в браузере с моей или другой информацией. Ещё я смог реализовать регистрацию и вход в моей программе, эти функции работают исправно и без ошибок. А для упрощения своей задачи, я сделал шаблоны нужных мне команд или их связки и часто их использую.

Список литературы

- 1) <https://ru.wikipedia.org/wiki/C%2B%2B>
- 2) <https://ru-wiki.ru/wiki/WinAPI>
- 3) <https://ru.wikipedia.org/wiki/Wine>
- 4) https://ru.wikipedia.org/wiki/Парадигма_программирования
- 5) https://ru.wikipedia.org/wiki/Статическая_типизация
- 6) https://ru.wikipedia.org/wiki/Компилируемый_язык_программирования
- 7) https://ru.wikipedia.org/wiki/Машинный_код
- 8) <https://ru.wikipedia.org/wiki/Компилятор>
- 9) https://ru.wikipedia.org/wiki/Язык_программирования

РЕЦЕНЗИЯ

на индивидуальный итоговый проект

тема проекта «Идентификация пользователя. Разработка личного кабинета»
выпускника 9Г класса МБОУ «Гимназия №2»
Шорина Евгения Максимовича

Данная работа соответствует требованиям ФГОС, предъявляемым к содержанию и оформлению индивидуального итогового проекта.

Тема работы сформулирована грамотно и отражает содержание проекта.

Структура проекта содержит в себе: титульный лист, содержание, проблему, цель, задачи, краткое описание хода проекта и его результаты, список литературы.

Проект начинается с проблемы, которая возникла у учащегося еще в 8 классе, что послужило толчком к изучению данной темы и разработки своей собственной программы.

Цель сформулирована и достигнута в результате выполнения проекта. Самостоятельно сформулированы конкретные задачи, которые необходимо было решить, чтобы достичь поставленной цели.

Выбор средств и методов осуществлен самостоятельно, осознанно и адекватен поставленным цели и задачам:

- изучены источники с информацией о языке программирования Си++,
- разработана программа по авторизации на языке Си++,
- протестирована на собственных страницах выхода на различные свои страницы социальных сетей.

Самостоятельно спланирована и определена последовательность работы над проектом. Ход проекта полностью осуществлялся посредством самоконтроля и самооценки.

Все это свидетельствует о **сформированности регулятивных УУ на повышенном уровне.**

Ход проекта содержит теоретический материал о языке программирования Си++ и его базовых функций. Этот материал самостоятельно обобщен учащимся. Также проект содержит последовательное описание шагов по созданию программы. Учащимся самостоятельно создана программа для упрощенной авторизации человека в любой программе или нескольких программах. Мысли излагаются логично и последовательно. Все это свидетельствует о **сформированности познавательных УУД.**

Перечень использованной литературы оформлен в соответствии с требованиями.

В работе прослеживается литературность языка. Письменная речь орфографически грамотная, пунктуация соответствует правилам.

В ходе проекта продемонстрированы навыки индивидуальной работы. Учащийся сумел организовать сотрудничество и совместную деятельность со мной, как с руководителем в короткий промежуток времени, на этапе оформления и рецензирования самостоятельно выполненного проекта.

Все это свидетельствует о **сформированности коммуникативных УУД.**

Работа выполнена аккуратно, содержит наглядный материал (скриншоты экрана). Проверена на плагиат (<https://text.ru/antiplagiat>). Уникальность текста составляет 70%. Продемонстрировано умение свободно и самостоятельно решать учебно-практическую задачу, основанную на материале по информатике. Это свидетельствует о **сформированности предметных знаний, умений и навыков.**

Особое мнение: данный проект представляет высокую социальную значимость для активных пользователей сетей интернета.

Руководитель проекта



Терехова Е.Д.

«___» _____ 2019г.