

# Разработка прикладного приложения с помощью Windows API.

Автор: Бабич Ростислав Олегович 11-Б

Руководитель: Муляшов Нил Дмитриевич (педагог дополнительного образования)

## Идея, проблема, актуальность.

- Идея: попробовать разработать приложение со взаимодействием с операционной системой (ОС).
- Актуальность: обычная панель управления в Windows не всегда удобна, довольно громоздка
- Проблема: использование панели управления может занять больше времени, чем нужно, и она не всегда вполне удобна.

## Что рассматривается в проекте.

- Объект: приложение под операционную систему Windows.
- Предмет: работа с интерфейсом и ОС с помощью Windows API и библиотек языка программирования (ЯП) C++.

## Цель.

- Цель: сократить среднее время, потраченное на настройку системы Windows, и добавить по возможности свои функции, отсутствующие в панели управления.

# Задачи.

1. Изучение работы Windows API и библиотек ЯП
  - a) Работа с интерфейсом (Windows API)
  - b) Работа с ОС (библиотеки)
2. Создание самого приложения
  - a) Создание интерфейса
  - b) Создание работы настроек

# Ход работы.

- Из поставленных задач выполнено:
  1. Изучена основная структура Windows API

```
int WINAPI WinMain(HINSTANCE hInst, HINSTANCE hPrevInst, LPSTR args, int nCmdShow)
{
    WNDCLASS MainClass = CreateWindowClass((HBRUSH)COLOR_WINDOW, LoadCursor(NULL, IDC_ARROW), hInst, LoadIcon(hInst, MAKEINTRESOURCE(IDI_ICON1)),
        name:L"MainWindow", MainClassProcedure);

    if (!RegisterClass(&MainClass)) return -1;

    MSG MainMessage = { 0 };

    CreateWindow(MainClass.lpszClassName, L"Window", WS_OVERLAPPED | WS_CAPTION | WS_SYSMENU | WS_MINIMIZEBOX | WS_VISIBLE, X, Y, WIDTH, HEIGHT, NULL,
        NULL, NULL, NULL);

    while (GetMessage(&MainMessage, NULL, NULL, NULL))
    {
        TranslateMessage(&MainMessage);
        DispatchMessageW(&MainMessage);
    }

    return 0;
}
```

# Ход работы.

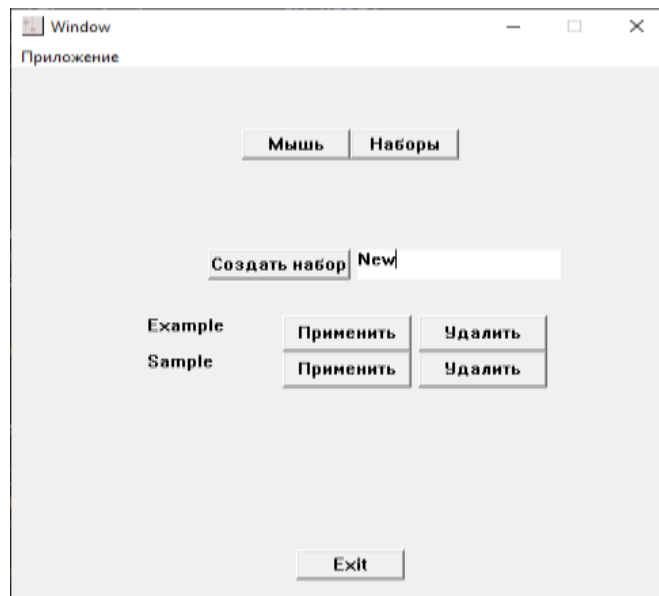
- Из поставленных задач выполнено:

2. Созданы основы для приложения:  
работа с мышью и система «наборов»

```
73     }
74     break;
75     case WM_CREATE:
76         AddMainWindowMenu(hwnd);
77         AddMainWindowWidgets(hwnd);
78         break;
79     case WM_DESTROY:
80         PostQuitMessage(0);
81         break;
82     default: return DefWindowProc(hwnd, msg, wParam, lParam);
83     }
84 }
85
86 Elvoid AddMainWindowMenu(HWND hwnd)
87 {
88     HMENU root_menu = CreateMenu();
89     HMENU exit_app = CreateMenu();
90
91     AppendMenuW(hwnd, exit_app, WM_STRING, WM_STRING, MenuExit, L"Выйти");
92     AppendMenuW(hwnd, root_menu, WM_STRING, WM_STRING, (UINT_PTR)exit_app, L"Приложение");
93
94     SetMenu(hwnd, Menu: root_menu);
95 }
96
97 Elvoid AddMainWindowWidgets(HWND hwnd)
98 {
99
100     CreateWindowA("button", "Exit", WS_VISIBLE | WS_CHILD, (WIDTH - 80)/ 2, HEIGHT - 100, 80, 25, hwnd, (HMENU)MenuExit, NULL, NULL);
101     CreateWindowA("button", "Menu", WS_VISIBLE | WS_CHILD, (WIDTH - 80)/ 2, 50, 80, 25, hwnd, (HMENU)OnMouseClicked, NULL, NULL);
102     textTesting = CreateWindowA("static", "a", WS_CHILD | WS_VISIBLE, 100, 100, 150, 25, hwnd, NULL, NULL, NULL);
```

## Ход работы.

- 3. Создана новая идея: т.н. наборы





При создании были использованы

- Официальная документация Microsoft
- Библиотека для работы с json nlohmann-json