**Компьютерная академия «ШАГ»**

**Днепропетровский филиал**

**Кафедра Разработки программного обеспечения**

**КУРСОВОЙ ПРОЕКТ**

**«Разработка Windows - приложений с использованием**

**Visual C++ и WinAPI»**

**Студента группы: ЕКО 15-П-1  
Ус Виталия Сергеевича**

**Научный руководитель:  
Шаптала Максим Валентинович**

(Днепропетровск) 2016

**Содержание**

Введение

1. Аннотация

2. Постановка задачи

3. Проектирование классов

4. Описание

5. Интерфейс программы

6. Руководство пользователя

Выводы

Список использованной литературы

**Введение**

1. **Аннотация**

Данная программа была написана с целью изучения, закрепления, углубления и расширения теоретических знаний, практических умений и навыков в соответствии с учебными планами. За время написания MP3 плеера мною были получены навыки самостоятельной работы с Windows приложением с использованием графики.

Для написания данной программы я выбрал библиотеку (BASS audio library).

**BASS (BASS audio library)** — это небольшая аудио-библиотека для использования в операционных системах Windows и Mac OS X, а также доступна для Linux, Win64, WinCE и iOS платформ.

Цель библиотеки заключается в том, чтобы предоставить разработчикам образец аудио-центра, в частности, обеспечить работу с потоками (MP3, MP2, MP1, OGG, WAV, AIFF), MOD файлами (XM, IT, S3M, MOD, MTM, UMX), MO3 файлами (mp3, ogg и сжатых MOD файлов), функциями записи, редактирования тегов аудиофайлов, конвертирования популярных форматов.

1. **Постановка задачи**
2. Обучение использования WinAPI.
3. Обучение использования библиотеки BASS.
4. Оптимизация интерфейса под пользователя.
5. Использование графики в проекте.
6. **Проектирование классов**

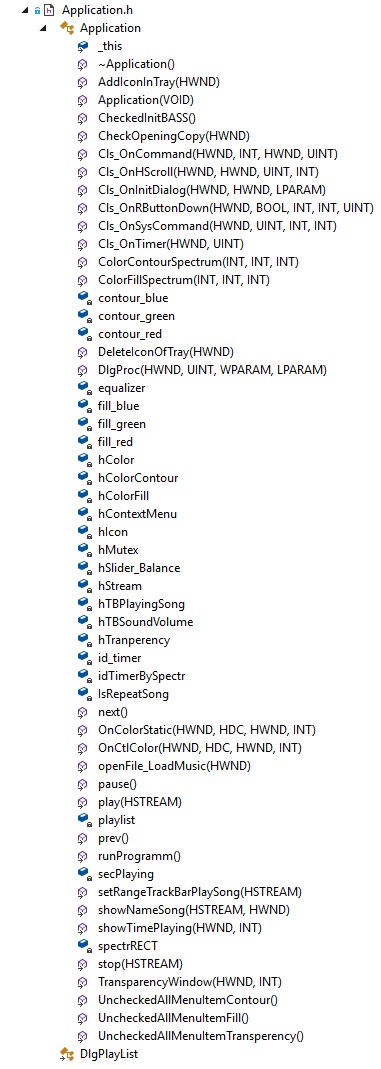
В данной программе использовалось 3 самоописанных класса: Application, DlgPlaylist, DlgEqualizer.

Application – основной класс, который управляет главным окном приложения.

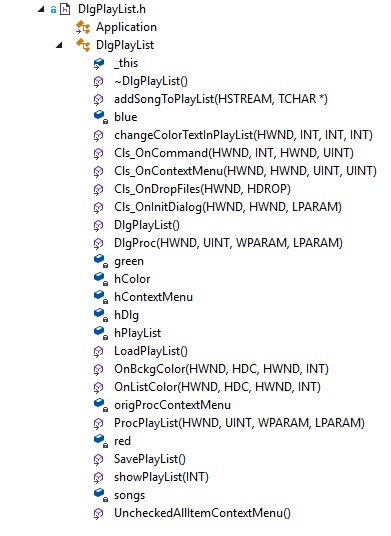
DlgPlaylist – данный класс отвечает за управление плейлистом.

DlgEqualizer – данный класс отвечает за управление эквалайзером.

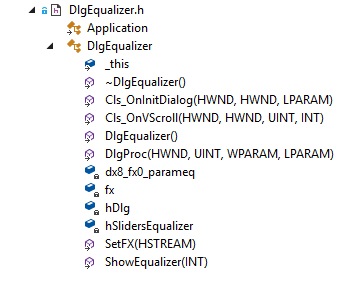
1. *Application*



1. *DlgPlaylist*



1. *DlgEqualizer*



1. **Описание**

*Класс Application:*

Поля:

static Application\* \_this – собственный указатель на класс

static INT\_PTR CALLBACK DlgProc(HWND hWnd, UINT uMsg, WPARAM wParam, LPARAM lParam) – диалоговая процедура обработки сообщения от окна

RECT spectrRECT – размер области спектра

HANDLE hMutex – мютекс, проверяющий открытие копии приложения

DlgPlayList playlist – объект класса Playlist

DlgEqualizer equalizer – объект класса Equalizer

HWND hTBSoundVolume – Slider – громкость

HWND hTBPlayingSong – Slider – полоса прокрутки

HWND hSlider\_Balance – Slider баланса

UINT id\_timer – ID таймера, который отвечает за полосу перемотки

UINT idTimerBySpectr – ID таймера, который отвечает за спектр

INT secPlaying – Количество секунд проигранных текущей песней

HICON hIcon – иконка приложения

BOOL IsRepeatSong – повтор песни

HSTREAM hStream – поток воспроизведения песни

INT contour\_red, contour\_green, contour\_blue – RGB смешевание контура спектра

INT fill\_red, fill\_green, fill\_blue – RGB смешевание заливки

HMENU hContextMenu, hColor, hColorContour, hColorFill, hTranperency

Методы:

VOID Cls\_OnCommand(HWND hwnd, INT id, HWND hwndCtl, UINT codeNotify) - WM\_COMMAND

BOOL Cls\_OnInitDialog(HWND hwnd, HWND hwndFocus, LPARAM lParam) - WM\_INITDIALOG

VOID Cls\_OnHScroll(HWND hwnd, HWND hwndCtl, UINT code, INT pos) - WM\_HSCROLL

VOID Cls\_OnTimer(HWND hwnd, UINT id) - WM\_TIMER

VOID Cls\_OnSysCommand(HWND hwnd, UINT cmd, INT x, INT y) - WM\_SYSCOMMAND

VOID Cls\_OnRButtonDown(HWND hwnd, BOOL fDoubleClick, INT x, INT y, UINT keyFlags) - WM\_RBUTTONDOWN

HBRUSH OnColorStatic(HWND hwnd, HDC hdc, HWND hwndChild, INT type) - WM\_CTLCOLORSTATIC

HBRUSH OnCtlColor(HWND hwnd, HDC hdc, HWND hwndChild, INT type) – WM\_CTLCOLORDLG

INT runProgramm() – запуск окна программы

INT CheckedInitBASS() – проверка и инициализация версии BASS

BOOL openFile\_LoadMusic(HWND hWnd) – открытие файла(песни) с компьютера

VOID setRangeTrackBarPlaySong(HSTREAM stream) – установка диапазона полосы прокрутки(0...по количество секунд)

VOID stop(HSTREAM stream) – останавливает поток

VOID play(HSTREAM stream) – воспроизводит песню из потока

VOID pause() – приостанавливает поток

VOID next() – переключает на следующую песню

VOID prev() – переключает на предыдущую песню

VOID DeleteIconOfTray(HWND hWnd) – удаляет иконку приложения из трея

VOID AddIconInTray(HWND hWnd) – добавляет иконку приложения в трей

VOID CheckOpeningCopy(HWND hwnd) – проверяет на открытие повторных копий приложения

VOID ColorFillSpectrum(INT r, INT g, INT b) – задает цвет заливки спектра

VOID ColorContourSpectrum(INT r, INT g, INT b) – задает цвет контура спектра

VOID TransparencyWindow(HWND hWnd, INT value) – устанавливает прозрачность окна

VOID showTimePlaying(HWND hWnd, INT secPlaying) – выводит в статик текущее время играющей песни

VOID showNameSong(HSTREAM stream, HWND hWnd) – задает заголовку имя текущей играющей песни

VOID UncheckedAllMenuItemContour() – снимает галочки со всех элементов контекстного меню Контур

VOID UncheckedAllMenuItemFill() – снимает галочки со всех элементов контекстного меню Заливка

VOID UncheckedAllMenuItemTransperency() – снимает галочки со всех элементов меню Прозрачность

*Класс DlgPlaylist:*

Поля:

HMENU hContextMenu – контекстное меню

WNDPROC origProcContextMenu – для подвязки дополнительной диалоговой процедуры обработки сообщения от ListBox

HWND hDlg – дескриптор основного окна приложения

HWND hPlayList – дескриптор плейлиста

std::vector<infoSong> songs – вектор хранящий в себе песни находящиеся в плейлисте

INT red, green, blue – RGB смешевание цвета текста в плейлисте

static DlgPlayList\* \_this – собственный указатель на класс

Методы:

VOID Cls\_OnCommand(HWND hwnd, INT id, HWND hwndCtl, UINT codeNotify) – WM\_COMMAND

BOOL Cls\_OnInitDialog(HWND hwnd, HWND hwndFocus, LPARAM lParam) – WM\_INITDIALOG

VOID Cls\_OnContextMenu(HWND hwnd, HWND hwndContext, UINT xPos, UINT yPos) – WM\_CONTEXTMENU

VOID Cls\_OnDropFiles(HWND hwnd, HDROP hdrop) – WM\_DROPFILES

VOID addSongToPlayList(HSTREAM stream, TCHAR\*path) – добавление песни в плейлист

VOID showPlayList(INT isShow) – отображение плейлиста

VOID changeColorTextInPlayList(HWND hWnd, INT r, INT g, INT b) – RGB смешевание цвета текста в плейлисте

HBRUSH OnListColor(HWND hwnd, HDC hdc, HWND hwndChild, INT type) – WM\_CTLCOLORLISTBOX

HBRUSH OnBckgColor(HWND hwnd, HDC hdc, HWND hwndChild, INT type) – WM\_CTLCOLORDLG

VOID UncheckedAllItemContextMenu() – снимает галочки со всех элементов меню Цвет текста

*Класс DlgEqualizer:*

Поля:

HWND hDlg – дескриптор окна

BASS\_DX8\_PARAMEQ dx8\_fx0\_parameq – структура определяющая данные для эквалайзера

HFX fx[10] – массив значений, позволяющих влиять на звучание канала (Эквалайзер)

HWND hSlidersEqualizer[10] – массив дескрипторов слайдеров управляющих эквалайзером

static DlgEqualizer\* \_this – собственный указатель на класс

Методы:

static INT\_PTR WINAPI DlgProc(HWND hWnd, UINT uMsg, WPARAM wParam, LPARAM lParam) – диалоговая процедура обрабатывающая сообщения от окна

VOID ShowEqualizer(INT isShow) – отображает эквалайзер

VOID SetFX(HSTREAM stream) – задает для каждого HFX позволение изменять звучание

VOID Cls\_OnVScroll(HWND hwnd, HWND hwndCtl, UINT code, INT pos) – WM\_VSCROLL

BOOL Cls\_OnInitDialog(HWND hwnd, HWND hwndFocus, LPARAM lParam) – WM\_INITDIALOG

*Структура infoSong:*

Данная структура содержит данные о загружаемой песне в MP3 плеер

HSTREAM hStream – поток песни

CHAR title[40] - Название

CHAR artist[30] - Исполнитель

CHAR album[30] - Альбом

CHAR year[10] - Год

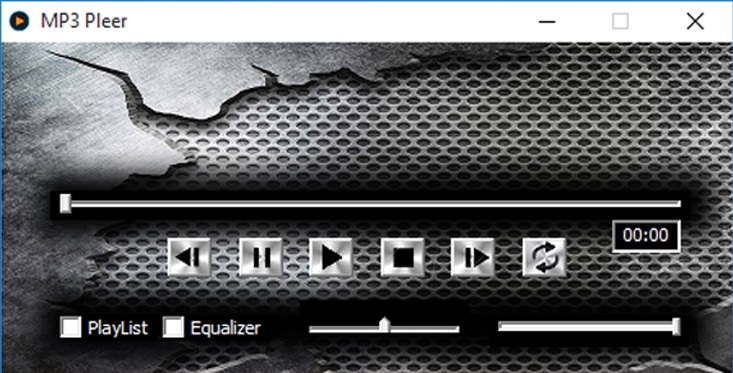
INT minutes - минут

INT seconds - секунд

TCHAR path[200] – путь к песне

1. **Интерфейс программы**

*Основное окно:*

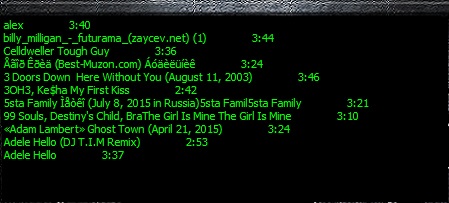


*Эквалайзер:*

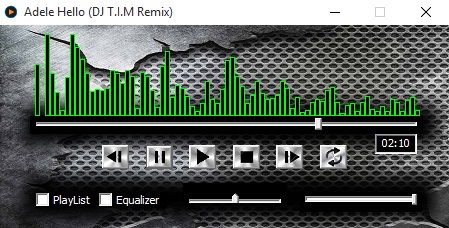
,



*Плейлист:*



*Вид при проигрывании:*



1. Руководство пользователя

*Добавление песни:*

Добавить песню можно четырьмя способами:

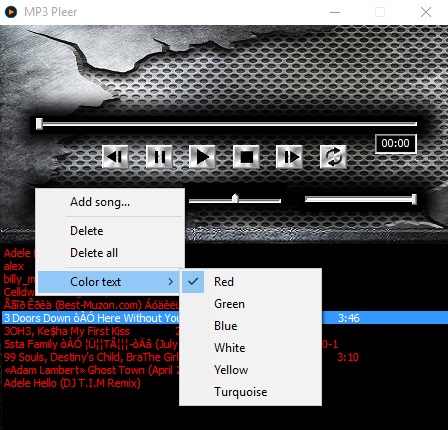
1. По нажатию на кнопку , если плейлист пуст.



1. Вызвав контекстное меню в плейлисте (При пустом плейлисте вызывается однострочное контекстное меню):

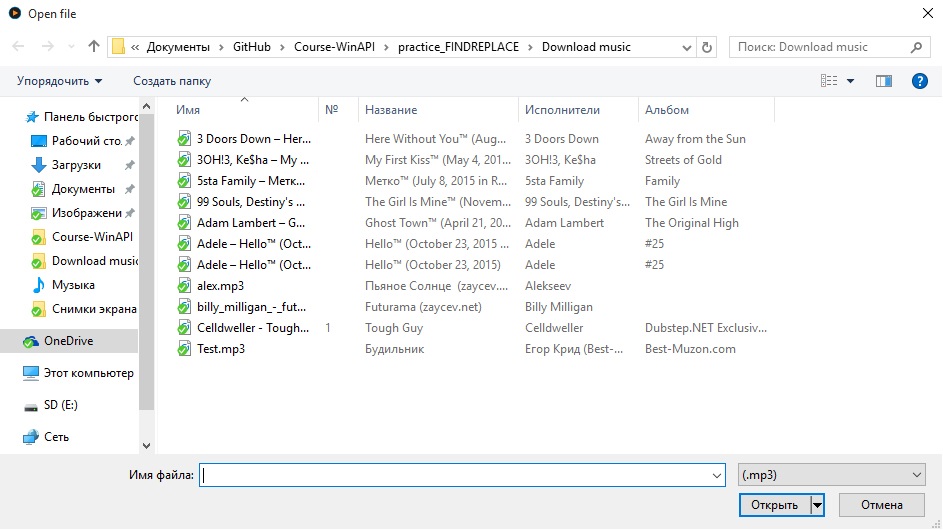


1. Вызвав контекстное меню в плейлисте при наличии в нем песен:



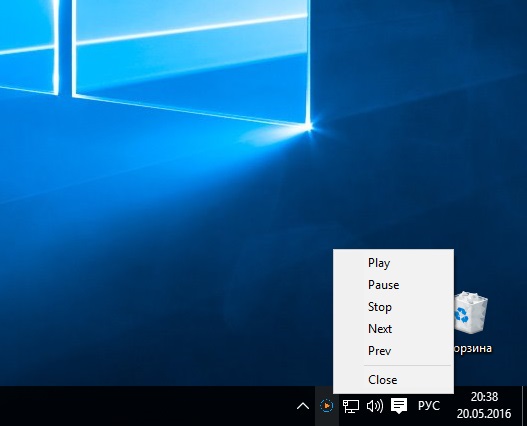
1. Добавлена возможность добавить любое количество песен в плейлист при помощи перетаскивания (Drag and Drop).

*При добавление песни открывается окно:*

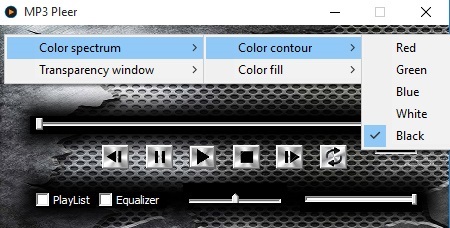


*Управление программой с трея:*

1. Play – воспроизвести
2. Pause – поставить на паузу
3. Stop – остановить
4. Next – воспроизвести следующую песню
5. Prev – воспроизвести предыдущую песню
6. Close – закрыть программу

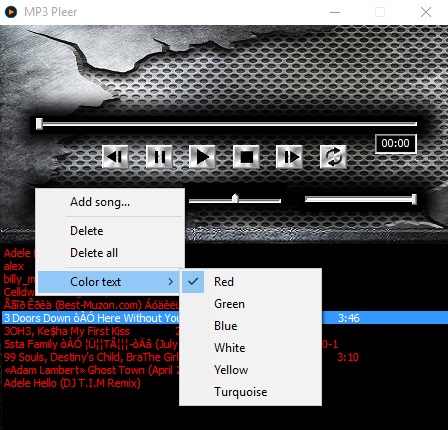


*Контекстное меню в основном окне:*



1. Color spectrum
   1. Color contour – позволяет пользователю установить цвет контура
   2. Color fill – позволяет пользователю установить цвет заливки
2. Transparency window – позволяет пользователю задать прозрачность окна от 10% до 100%

*Контекстное меню в плейлисте:*



1. Add song… - позволяет пользователю добавить песню
2. Delete – позволяет пользователю удалить песню
3. Delete all – позволяет пользователю очистить плейлист
4. Color text – позволяет пользователю задать цвет текста в плейлисте

**Выводы**

За время написания курсового проекта используя технологию Visual C++ WinAPI мною были получены навыки самостоятельной работы с графическим приложением Windows. Мною была выбрана тема, которая не относится к изучаемой программе. После выбора неизвестной мне библиотеки BASS и самостоятельного изучения документации, мною были получены большие навыки самостоятельной работы с неизвестной мне библиотекой.

**Список использованной литературы**

Единственный источник, который описывает функции библиотеки BASS:

1. http://www.un4seen.com/doc/ - документация на алглийском языке
2. <https://msdn.microsoft.com>