

МОСКОВСКИЙ ИНСТИТУТ РАДИОТЕХНИКИ
ЭЛЕКТРОНИКИ И АВТОМАТИКИ

Курсовая Работа

ПО

Прикладному Программному Обеспечению

Многофункциональный MP-3 плеер «Джим»

Выполнил: студент группы ИП-4-03

Верещагин Илья Геннадьевич

Принял:

Москва, 2006

«Джим»

*многофункциональный
MP3-плеер*

TDSV, Vereshchagin I.G.
© 2006, All Rights Reserved

Содержание

⌚	Соглашение	- - - - -	4
⌚	Описание продукта	- - - - -	5
⌚	История создания «Джима»	- - - - -	6
⌚	Техническое задание	- - - - -	8
⌚	Руководство Оператора	- - - - -	10
а)	Плеер	- - - - -	10
б)	Настройки	- - - - -	17
	в) Функции окон	- - - - -	21
	г) Статистика	- - - - -	22
⌚	Техническое описание функций	- - - - -	24
⌚	Файлы исходного пакета	- - - - -	25
⌚	Листинг кода	- - - - -	27
⌚	Прочее	- - - - -	130
⌚	Приложение: GNU GPL	- - - - -	131
⌚	Приложение: Понятия GPL.rus	- - - - -	141

Соглашение

Данный продукт проверен на наличие вирусов и возможных ошибок. В случае, если этот продукт вам оказался полезен и интересен и вы заинтересованы в его дальнейшем развитии и поддержке, разработчик всегда будет рад принять помощь – моральную и материальную.

Данная программа является свободным программным обеспечением. Вы вправе распространять ее и/или модифицировать в соответствии с условиями версии 2 либо по вашему выбору с условиями более поздней версии Стандартной Общественной Лицензии GNU, опубликованной Free Software Foundation.

Мы распространяем данную программу в надежде на то, что она будет вам полезной, однако НЕ ПРЕДОСТАВЛЯЕМ НА НЕЕ НИКАКИХ ГАРАНТИЙ, в том числе ГАРАНТИИ ТОВАРНОГО СОСТОЯНИЯ ПРИ ПРОДАЖЕ и ПРИГОДНОСТИ ДЛЯ ИСПОЛЬЗОВАНИЯ В КОНКРЕТНЫХ ЦЕЛЯХ. Для получения более подробной информации ознакомьтесь со Стандартной Общественной Лицензией GNU.

Описание продукта

Программа написана в среде интегрированной разработки Borland Builder 6.01 Enterprise на языке высокого уровня C++\Borland C\ANSI C++. Программа использует для работы дополнительную библиотеку – library.dll, входящую в комплект поставки. Все файлы, входящие в комплект поставки являются критически важными для стабильного функционирования программы.

«Джим» – не просто плеер. Плееров много – «Джим» один. Проигрывать MP3-файлы могут различные утилиты, стандартные программы, тот же MS WMP. В то же время, «Джим» – не просто «перепевка» WinAmp... Что отличает «Джима» от других плееров? «Джим» ориентирован только на работу с файлами типа *.mp3 (MpegLayer-3 (всех форматов)) и файлами плейлистов «Джима» *.jpl. В то же время, у Джима есть возможность просмотра графических изображений формата *.jpg (JPEG). Впрочем, отличия тем не заканчиваются. К возможностям «Джима» можно отнести:

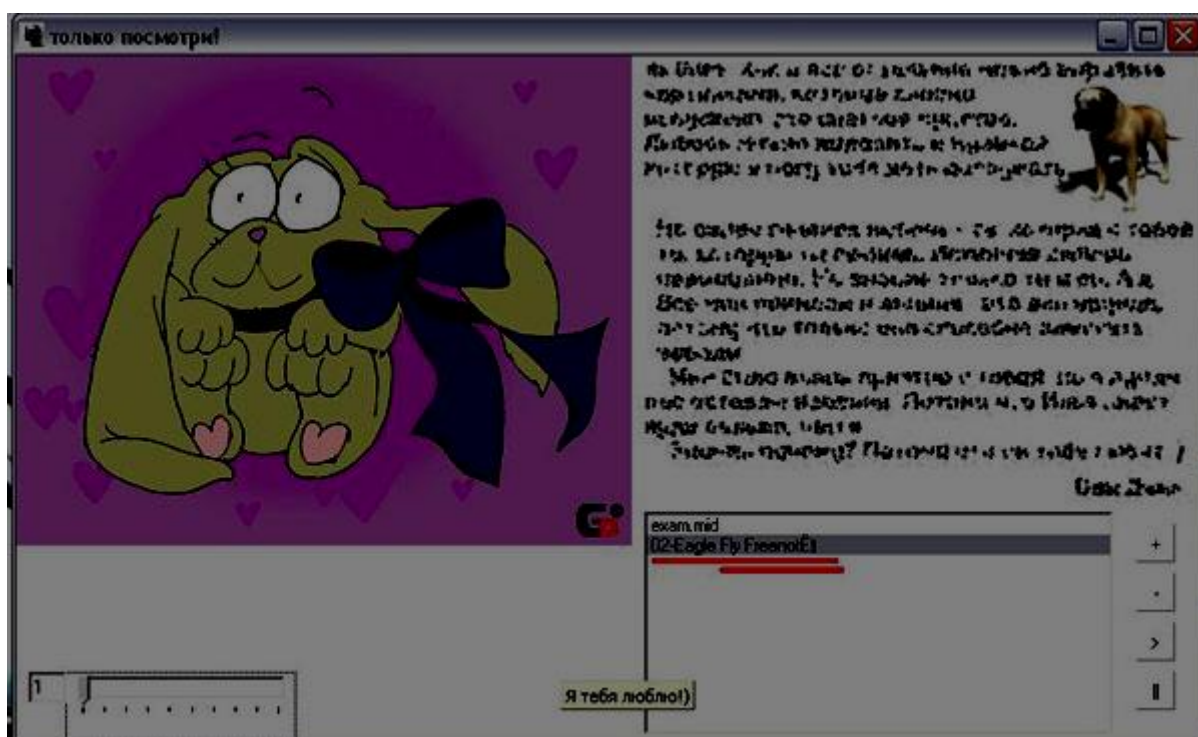
- « Проигрывание MP3-файлов
- « Просмотр JPEG-картинок
- « Добавление музыкальных\графических файлов по каталогам
- « Сортировка
- « Работа с плейлистами
- « Шкурки (skins)
- « Гибкая система настроек
- « «Умная» система обработки файлов и системы управления воспроизведением
- « Интернационализация (в версии 2.0 – Русский, Английский, Немецкий)

История создания

«Джима»

Текущая версия «Джима» - 2. Сегодня «Джим» представляет из себя многофункциональный плеер, с многими возможностями.

Изначально «Джим» задумывался в конце 2005 года (в середине ноября) как оригинальный подарок.



Таким образом, рано или поздно, меня завлекла идея сделать удобный, и в тоже время отличный от WinAmp'a плеер. Поставить себе задачу интересную и сложную, как профессиональному программисту. Да и, признаться, честно, одним из главных мотивов моего поступка стала политика NullSoft, еже с ним WinAmp, который день ото дня стали требовать с народа денег за полюбившийся продукт. Да и идея OpenSource, которая была в проекте «Джим-1.xx» не могла не нравиться.

Впрочем, от нее во второй части, ставшей уже общедоступной, пришлось отказаться вследствие сложных алгоритмов, и человеко-ресурсов в частности.

Интересно было бы вести чейнжлог «Джима», но по количеству новых идей его бы приходилось бы дополнять каждый день. Так, главными отличиями от «Джима-1» к «Джиму-2» стало добавление функций сортировки, отдельных плейлистов, скинов, интеллектуальной рандомизации и чтения имени файла (интеллектуальный ID3TAGV1.1), гибкая система настроек.

Но об отдельных функциях поговорим в следующем разделе.

Техническое Задание

на курсовой проект

Наименование работы:

“ Многофункциональный MP-3 плеер ”

Назначение разработки:

Воспроизведение Mp3 медиа\аудиофайлов в режиме realtime, воспроизведение и отображение JPEG-изображений в режиме слайдшоу, ведение статистики в БД, оптимизация пользовательского интерфейса.

Требования к системе:

Функциональные требования:

- Воспроизведение аудиофайлов формата *.mp3
- Просмотр графических изображений формата *.jpg
- Управление листом произведения (playlist): добавление, удаление, сортировка, сохранение листа, сохранение в HTML формате
- Управление слайдшоу: удаление и добавление изображений в слайдшоу
- Управление интерфейсом программы – системой шкурок (skins), языков

Требования к программному и техническому обеспечению:

IBM-совместимый компьютер, i586\i686 процессор (Intel Pentium III, AMD K6 и выше), 128Мб RAM, 64Mb DirectX-совместимая видеокарта, MS Windows XP/MCE, InterBase Borland-клиент (для

корректной работы статистики).

Требования к программной документации:

Программная документация должна включать:

- Техническое задание
- Техническое описание подсистемы
- Руководство оператора

Этапы разработки:

- Разработка теоретических алгоритмов 01.02-30.02
- Разработка приложения в среде Borland Builder 6 15.03-30.03
- Отладка программы 01.04-14.04
- Разработка программной документации 15.05-16.05
- Защита курсового проекта 17.05 - 31.05

Порядок приемки работы:

Исполнитель предъявляет:

- расчетно-пояснительную записку
- дискету, содержащую файл с приложением

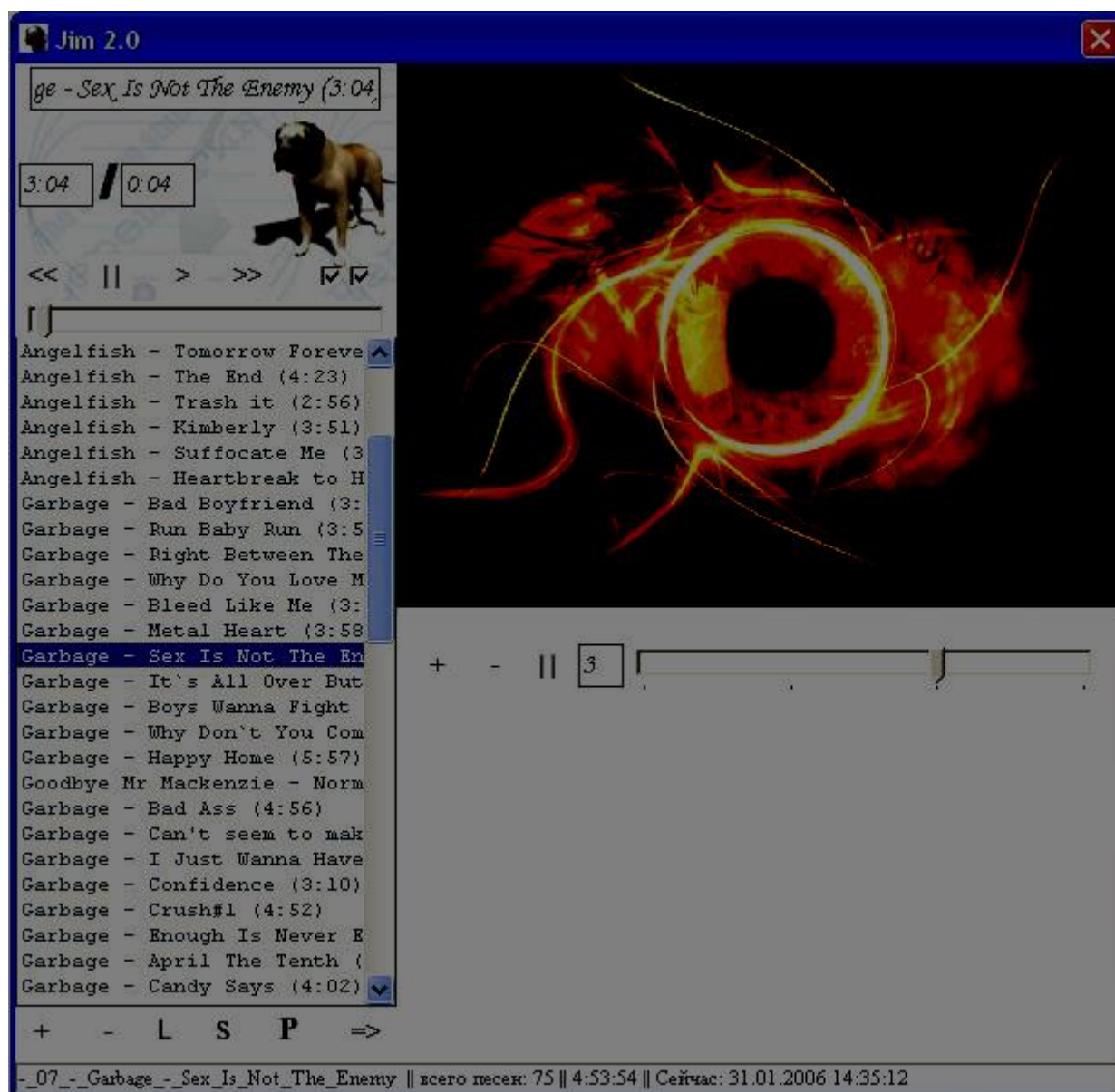
Исполнитель курсового проекта: Верещагин И.Г.

Дата: 01.02.06

Руководство Оператора

а) П л е е р

Внешний вид главного окна «Джима»:



Плеер выполнен в очень удобном и интуитивно понятном исполнении, является главной частью программы. К плееру относится и *окно просмотра картинок JPEG*. Под окошком изображений находятся:

- ⑩ **Окошко картинки**
- ⑩ **Окошко номера картинки** в списке просмотра
- ⑩ **Трекбар** (показывает текущее положение картинки, позволяет выбрать нужную картинку по номеру)

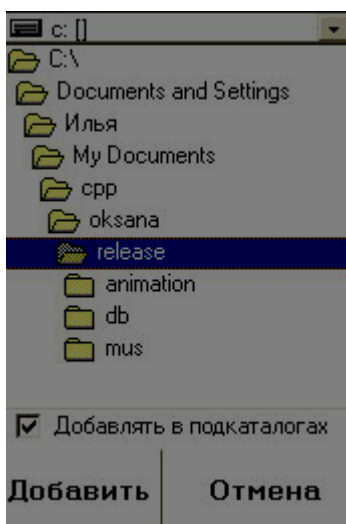
- ⑩ **Кнопка «Добавить»**. Добавляет файл картинки в список слайд шоу. При правом клике на кнопке возможно добавления всех картинок из выбранного каталога. Сохраняется и после закрытия программы.
- ⑩ **Кнопка «Удалить»**. Удаляет файл картинки из списка слайд шоу. При правом клике возможен выбор удаления всех картинок из списка. Сохраняется и после закрытия программы.
- ⑩ **Кнопка «Остановка\запуск слайд шоу»**.

Плеер состоит из:

- ⑩ **Окошко текущей песни**
- ⑩ **Окошко длины текущей песни**
- ⑩ **Окошко положения текущей песни** (при нажатии на него изменяется отсчет – от начала песни или до ее конца соответственно)
- ⑩ **СикерБар** – графическое положение позиции прослушивания песни. При паузе возможно менять положение песни (прокручивать песню) перетягиванием ползунка.
- ⑩ **Первая галочка** устанавливает режим прослушивания – повторный (если отмечено) и без повторения листа, если нет.
- ⑩ **Вторая галочка** устанавливает режим прослушивания – случайный (если отмечено) и по порядку, если нет.
- ⑩ **Кнопка «Назад»** – возвращает на песню назад (если не включен случайный режим). Если текущая песня была первой – переводит на последнюю (в режиме отсутствия случайного режима).
- ⑩ **Кнопка «Играть»** – начинает проигрывание файла в случае если есть выделенные

файлы в плейлисте, либо плеер был временно приостановлен.

- ❶ **Кнопка «Пауза»** – временно приостанавливает прослушивание.
- ❷ **Кнопка «Вперед»** – переводит на песню вперед (если не включен случайный режим). Если текущая песня была последней – переводит на первую (если включен повтор).
- ❸ **Окно плейлиста.** В нем находится список песен, готовых к прослушиванию. Через него проводится навигация. При двойном



щелчке на элементе начинается его немедленное воспроизведение (аналогично Enter, кнопке «Играть»). При нажатии клавиши *delete*, удаляются выбранные элементы (аналогично кнопке «Удалить»).

- ❹ **Кнопка «Добавить».** При щелчке левой кнопкой мыши открывает диалог добавления MP3 файла. При щелчке правой кнопкой мыши открывает всплывающее меню:

- а) Добавить файл. Аналогично щелчку левой кнопкой мыши
- б) Добавить каталог.

Открывает диалог добавления.

Пользователю предлагается выбрать нужный ему каталог *двойным щелчком* на нужной папке. Так же, если отмечена галочка «Добавлять в подкаталогах», добавляет файлы как из выбранного каталога, так и файлы из подкаталогов.

- ❺ **Кнопка удалить.** При щелчке левой кнопкой мыши удаляет выделенную в плейлисте песню. Если ничего не выделено ничего не удаляет. При щелчке правой кнопкой мыши открывает всплывающее меню:

- а) Удалить выделенное. Аналогично

щелчку левой кнопкой мыши.

б) Удалить всё. Удаляет все элементы из плейлиста.

в) удалить дубликаты – удаляет повторные песни в плейлисте

• **Кнопка «Сохранить плейлист».** Сохраняет все элементы в плейлисте в плейлист, загружающийся по умолчанию. При щелчке правой кнопкой мыши открывает всплывающее меню:

а) Сохранить по умолчанию.

Аналогично щелчку левой кнопкой мыши.

б) Сохранить как. Предлагается выбрать имя сохраняемого плейлиста, куда он будет сохранен для последующего вызова.

в) Загрузить. Выбирается нужный плейлист, загружает ранее сохраненный плейлист.

г) Сохранить HTML. Предлагает выбрать имя файла, куда будет сохранен текущий плейлист в формате HTML.

• **Кнопка «Сортировать».** При щелчке правой кнопкой мыши открывает всплывающее меню:

а) По имени файла. Производит сортировку плейлиста по имени файла и директории, где находится файл. При повторной сортировке (если лист был уже отсортирован этим методом) – сортирует обратным образом (А-Я или Я-А) .

б) По алфавиту. Сортирует лист по алфавиту: сначала группу, потом песню. При повторной сортировке (если лист был уже отсортирован этим методом) – сортирует обратным образом (А-Я или Я-А) .

в) По песне. Сортирует лист по

названиям песен. Исполнители игнорируются. При повторной сортировке (если лист был уже отсортирован этим методом) –

сортирует обратным образом (А-Я или Я-А) .

г) Перемешать. Случайно перемешивай лист воспроизведения.

Ⓢ **Кнопка «Настройки».** Вызывает окошко настроек.

Ⓢ **Кнопка «Открыть\Заккрыть просмотр картинок».** Открывает или сворачивает часть окна, где происходит слайд шоу.

Ⓢ **Строка состояния.** В ней отображается состояние Джим-Плеера. «Прослушивание» или «Приостановлено». В случае «Прослушивания» выводится информация о текущей песне, количестве песен в плейлисте. В том числе выводится информация о текущей дате и времени.

Замечания

При добавлении большого количества файлов в лист воспроизведения «Джиму» требуется значительное время для кэширования файлов. При добавлении 500+ файлов процесс может занять от 1-2 минут в зависимости от возможностей компьютера.

Сортировка при большом количестве файлов в плейлисте занимает тоже немалое время.

При любой сортировке удаляются дубликаты.

Загрузка плейлиста аналогична загрузке каталога.

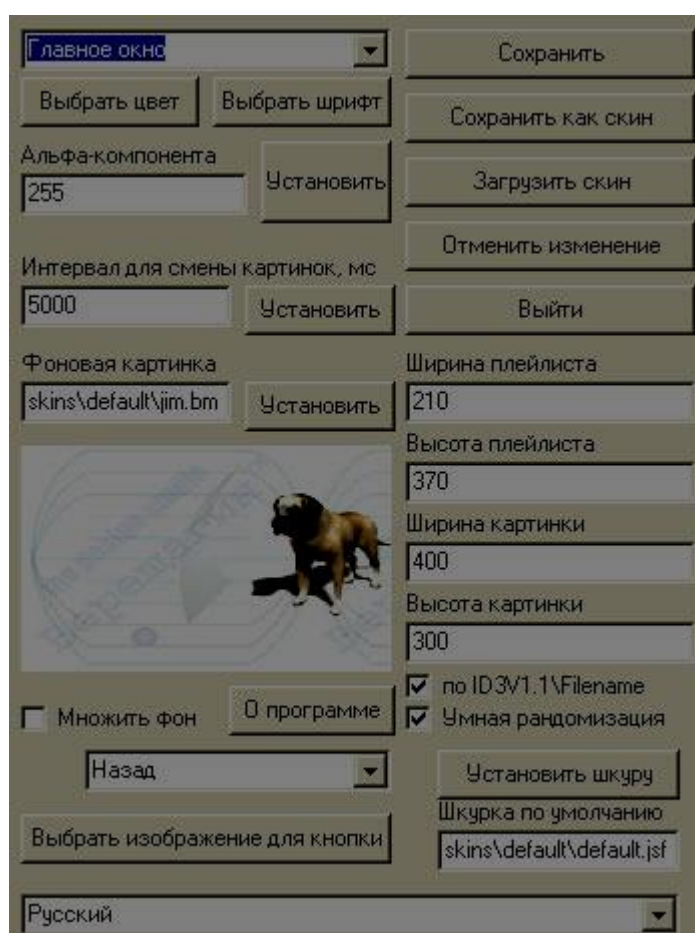
Настройки плеера - его расположения, выбора повторного и случайного воспроизведения сохраняются при выходе из плеера.

Не рекомендуется удалять Mp3-файлы, равно как и файлы изображений без предварительного удаления из Джима. Таким образом, при ошибке открытия файла Джим сгенерирует ошибку доступа.

Картинки, если их размер превосходит

установленного размера, автоматически масштабируются.

б) Настройки



Общий вид окна настроек. Данное окно блокирует обращения к плееру до того момента, пока не будет закрыто.

Настройки предназначены для оптимизации плеера под нужды и удобство каждого человека в отдельности. Они позволяют выбрать оптимальные алгоритмы работы «Джима» и его оформления, в том числе позволяют самому создать

собственные шкурки (скины).

- **Первый выпадающий список** позволяет выбрать тип изменяемого компонента: Индикаторы Времени, Плейлист, Главное Окно, Строка статуса.

- ⦿ Кнопка **«Выбрать цвет»**. Устанавливает желаемый пользователем цвет для выбранного компонента в первом выпадающем списке.
- ⦿ Кнопка **«Выбрать шрифт»**. Устанавливает желаемый пользователем шрифт и цвет шрифта для выбранного компонента в первом выпадающем списке.
- ⦿ **Альфа-компонента**. Устанавливает степень прозрачности окна плеера (от 255 (непрозрачный) до 0 (невидимый)). Для многих скинов не работает вследствие перерисовки картинки.
- ⦿ **Интервал для смены картинок**. Устанавливает время между сменой картинок.
- ⦿ **Фоновая картинка**. Устанавливает картинку шкурки. Только формат *.BMP. Рекомендуется использовать формат 210x150px 24bit, Windows.
- ⦿ **Изображение фоновой картинки**.
- ⦿ Галочка **«Множить фон»**. Устанавливает возможность размножения фоновой картинки. Внимание! Будет распространено на все окно! Не рекомендуется при частой смене картинок.
- ⦿ **Второй выпадающий список** позволяет выбирать тип изменяемого компонента (кнопок «Джима»). Рядом отображается выбранная картинка для кнопки.
- ⦿ Кнопка **«Выбрать изображение для кнопки»**. Позволяет пользователю выбрать нужное изображение для кнопки для выбранного компонента во втором выпадающем списке. Рекомендуется использовать картинки формата 25x25px BMP или JPEG.
- ⦿ Кнопка **«Сохранить»**. Сохраняет текущие настройки шкурки в скин по умолчанию. Будьте внимательны! Кнопка «сохранить» затирает все настройки по умолчанию!
- ⦿ Кнопка **«Сохранить как скин»**. Позволяет пользователю выбрать имя и место расположения сохраняемого скина.
- ⦿ Кнопка **«Загрузить скин»**. Позволяет пользователю

загрузить заранее сохраненный скин.

- Ⓢ Кнопка **«Отменить изменения»**. Позволяет загрузить настройки скина по умолчанию.
- Ⓢ Кнопка **«Выйти»**. Закрывает окно настроек и передает управление плееру.
- Ⓢ **Ширина плейлиста**. Устанавливает ширину плейлиста. Минимальное значение – 210пкс. Не рекомендуются большие значения.
- Ⓢ **Высота плейлиста**. Устанавливает ширину плейлиста. Не рекомендуются малые значения. Также, для корректности работы программы, рекомендуется устанавливать высоту плейлиста большей или равной высоте картинки.
- Ⓢ **Ширина картинки**. Устанавливает ширину картинки.
- Ⓢ **Высота картинки**. Устанавливает высоту картинки.
- Ⓢ Галочка **«по ID3V1.1\FileName»**. При отмеченной галочке плеер инициализирует имена в плейлисте по ID3-тэгу, содержащемуся в MP-3 файлах. При отсутствии галочки – будет выдано короткое имя файла.
- Ⓢ Галочка **«Умная рандомизация»**. Устанавливает тип рандомизации. При отмеченной галочке случайное проигрывание происходит с запоминанием проигранных песен, с наиболее успешным случайным моментом. При отсутствии галочки – «Джим» при переключении песни не запоминает предыдущую песню, каждый раз выбирается новая песня. Режим «Умной рандомизации» не рекомендован на слабых компьютерах при большом числе файлов в плейлисте.
- Ⓢ **«Установить шкурку по умолчанию»**. Выбирает шкурку, которая будет использована как шкурка по умолчанию, а так же она является шкуркой, грузящейся при запуске «Джима».
- Ⓢ **Третий выпадающий список, список языка**. Устанавливает язык «Джима»: Русский, Английский, Немецкий. Вводится немедленно.
- Ⓢ Кнопка **«О программе»**. Вызывает краткую

информацию о программе.

- «**Статистика**» – вызывает управление статистикой Джима
- Галочка «**Вести статистику**» – включает\выключает статистический механизм «Джима» при проигрывании песен.
- **Выбор языка** «Джима» – устанавливает язык интерфейса плеера: русский\английский\немецкий.

Замечания

Полностью шкурки инициализируются только при:

- Запуске «Джима»
- При нажатии на кнопке «Загрузить Скин»

При создании собственного скина старайтесь все изображения собрать в одной папке, а папку разместить в каталоге «Джима» – skins.

Все параметры, кроме интернационализации и загружаемого скина по умолчанию хранятся в файлах шкурки.

Не вносите вручную изменения в файлы шкурок!

в) Ф у н к ц и и о к о н

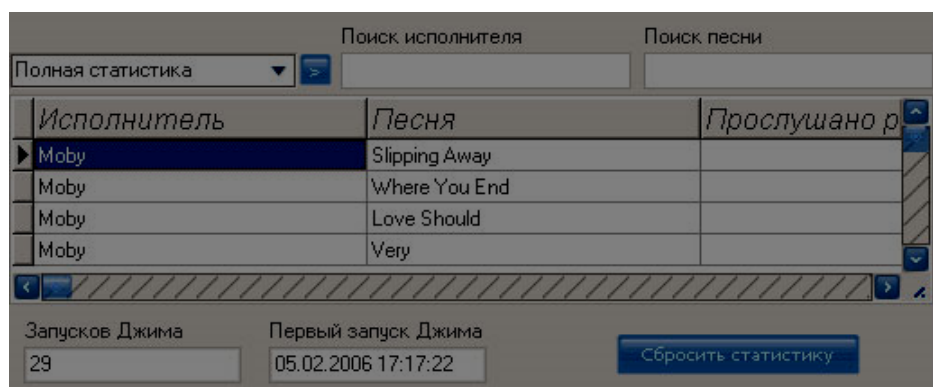
При запуске «Джима» автоматически добавляется значок в иконки (systray). Это дает возможность убрать «Джима» в невидимое состояние щелчком на панели запущенных программ. Когда джим убран в невидимое состояние его можно развернуть двойным щелчком на иконке в systray. Так же на значке systray выводится информация о текущем состоянии Джим-Плеера.

Положение «Джима» запоминается при выходе.

г) С т а т и с т и к а

«Джим» позволяет вести статистику прослушанных песен, если при работе с плеером отмечена галочка «Вести статистику». Статистика ведется при начале прослушивания песни и записывается в статическую базу данных в виде следующей таблицы:

<i>Исполнитель</i>	<i>Песня</i>	<i>Прослушано Раз</i>	<i>Первое прослушиван ие</i>	<i>Последнее прослушиван ие</i>	<i>Длительност ь</i>
...
...



Окно
управления
статистикой
состоит из:

- ⊗ **Статистическая таблица**, в которую выводятся те или иные сведения, согласно запросам, выбранным пользователем.
- ⊗ **Выпадающий список запросов**, который позволяет при выборе нужного пункта выдавать требуемую информацию в статистическую таблицу. Содержит запросы:
 - ⊗ *Полная статистика* – отображает все ранее прослушанные песни с сопутствующими данными.
 - ⊗ *Поиск Исполнителя* – производит поиск по исполнителю (или части его имени) и выдает соответствующий результат.
 - ⊗ *Поиск Песни* – производит поиск по песне (или части его имени) и выдает соответствующий

результат.

- Ⓢ *Любимый исполнитель* – показывает любимого исполнителя пользователя за все время, т.е. Того, чьи песни были прослушаны наибольшее количество раз.
- Ⓢ *Любимая песня* – показывает песню, прослушанную наибольшее количество раз.
- Ⓢ *Наибольшее количество песен* – показывает исполнителя, у которого было прослушано наибольшее количество различных песен.
- Ⓢ *Последняя прослушанная песня* – выдает последнюю прослушанную песню.
- Ⓢ *Первая прослушанная песня* – выдает первую прослушанную песню.
- Ⓢ *Самая длинная песня* – выдает самую длинную прослушанную песню (по времени).
- Ⓢ *Самая короткая песня* – выдает самую короткую прослушанную песню (по времени).
- Ⓢ *Прослушано всего песен* – выдает сообщение о количестве прослушанных песен.
- Ⓢ *Длительность песен всего* – выдает сообщение о суммарной длительности всех прослушанных песен.
- Ⓢ *Прослушано всего время* – выдает сообщение о времени, потраченном на прослушивание песен всего.
- Ⓢ **Кнопка >** – повторяет последний запрос.
- Ⓢ **Поле ввода «Поиск Исполнителя» и Поле ввода «Поиск Песни»** – используются для ввода данных для сопутствующих запросов.
- Ⓢ **Поле «Запусков Джима»** – отображает количество раз, когда был запущен плеер «Джим».
- Ⓢ **Поле «Первый запуск Джима»** – отображает дату и время первого запуска «Джима».
- Ⓢ **Кнопка «Сбросить статистику»** – сбрасывает всю статистику по Джиму.

Техническое описание функций

В программе используются системные таймеры для анимации Джима. Любое нарушение целостности программы вызывает ее немедленное прекращение. При отсутствии файла инициализации «jim.ini», он будет создан заново, однако программа будет неработоспособна до ее закрытия.

Кроме того, для инициализации программы используются файлы баз данных. В случае нарушения целостности программы (отсутствия доступа к базе данных) вызывает ее немедленное прекращение.

Плеер использует множество библиотечных функций. Программа просматривает до 10и вложенных каталогов.

Файлы исходного пакета

Исходный пакет файлов Джима выглядит следующим образом:

• /animation

содержит графические элементы «Джима»

- split.bmp /критически важный файл, разделитель времени/

файлы картинок для листа картин по умолчанию в случае удаления картин не из «Джима», прямым путём или запрета доступа к ним будет нарушена целостность программы. В случае нарушения целостности программа работать не будет!

- 1.jpg
- 2.jpg
- 3.jpg
- 4.jpg
- 5.jpg

• /mus

содержит музыкальные файлы «Джима» по умолчанию. Удаление\изменение не критично для программы.

- hello.mp3
- Levan_Polka.mp3
- Eartha_Kitt_-_I_want_to_be_evil.mp3

• /db

содержит критически важные для программы инициализационные базы данных. В случае нарушения целостности программа работать не будет!

- Ⓢ mp3.db /содержит плейлист по умолчанию/
- Ⓢ mp3init.db /содержит пользовательский плейлист/
- Ⓢ pic.db /содержит список картинок/
- Ⓢ statistic.db /содержит статистику программы/
- Ⓢ statistic.VAL /validate-файл, сведения о статистической БД/

Ⓢ /skins

содержит шкурки программы

- Ⓢ /default /шкурка по умолчанию, Джим/
- Ⓢ /jim /шкурка Джим/
- Ⓢ /ro_angel /шкурка в аниме-стиле/
- Ⓢ /darkeyes /шкурка в Horror-стиле/
- Ⓢ /margo /шкурка в Dark-стиле/

Ⓢ /source

содержит исходные коды программы

- Ⓢ library.dll /библиотека функций/
- Ⓢ jim.exe /главный исполняемый файл программы/
- Ⓢ jim.ini / файл инициализации Джима/
- Ⓢ Джим.pdf /данный мануал/

Листинг кода

library.dll**library.h**

```
//-----
#ifndef __LIBRARY_H
#define __LIBRARY_H
#ifdef __DLL__
#define DLL_EI __declspec(dllexport)
#else
#define DLL_EI __declspec(dllimport)
#endif
extern int DLL_EI __fastcall getlenght(char* database);
extern char* DLL_EI __fastcall crofilename(WideString S);
#endif;
```

library.cpp

```
//-----

#include <vcl.h>
#include <windows.h>
#include "library.h"
#include <SysUtils.hpp>
#include <fstream.h>
#include <stdio.h>
#include <string.h>
#include <dos.h>
#include <dir.h>
#pragma hdrstop
//-----
// Important note about DLL memory management when your DLL uses the
// static version of the RunTime Library:
//
// If your DLL exports any functions that pass String objects (or structs/
// classes containing nested Strings) as parameter or function results,
// you will need to add the library MEMMGR.LIB to both the DLL project and
// any other projects that use the DLL. You will also need to use MEMMGR.LIB
// if any other projects which use the DLL will be performing new or delete
// operations on any non-TObject-derived classes which are exported from the
// DLL. Adding MEMMGR.LIB to your project will change the DLL and its calling
// EXE's to use the BORLNDMM.DLL as their memory manager. In these cases,
// the file BORLNDMM.DLL should be deployed along with your DLL.
//
// To avoid using BORLNDMM.DLL, pass string information using "char *" or
// ShortString parameters.
//
// If your DLL uses the dynamic version of the RTL, you do not need to
// explicitly add MEMMGR.LIB as this will be done implicitly for you
//-----
```

```
#pragma argsused
BOOL WINAPI DllMain(HINSTANCE hinstDLL, DWORD fdwReason, LPVOID lpvReserved)
{
    return 1;
}
//-----

int __fastcall getlength(char* database)
{
    int param=0;
    char temp[1024];
    fstream e;
    e.open(database);
    if (e)
    {
        //ShowMessage("Unable to open temp file "+AnsiString(database));
        while (!e.eof())
        {
            e.getline(temp,1024);
            param++;
        }
        e.close();
        return(param-1);
    }
    else
        return 0;
}

char* __fastcall crofilename(WideString S)
{
    int i=0;
    char tempword[1024];
    while (S[i]!='.')
        i++;
    while (S[i]!='\\')
        i--;
    i++;
    int j=0;
    for (;;)
    {
        tempword[j]=S[i];
        j++; i++;
        if ((S[i]== '.' && S[i+1]=='m' && S[i+2]=='p' && S[i+3]=='3'))
            break;
    }
    tempword[j]='\0';
    return tempword;
}
```

Главное Окно – окно плеера**Unit3.h**

```
//-----
#ifndef Unit3H
#define Unit3H
//-----
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <ComCtrls.hpp>
#include <ExtCtrls.hpp>
#include <jpeg.hpp>
#include <MPlayer.hpp>
#include <Dialogs.hpp>
#include <Menus.hpp>
#include <FileCtrl.hpp>
#include "trayicon.h"
#include <Graphics.hpp>
//-----
```

```
class TForm3 : public TForm
{
__published: // IDE-managed Components
    TImage *Image1;
    TTrackBar *TrackBar1;
    TEdit *Edit1;
    TTimer *Timer1;
    TMediaPlayer *MediaPlayer1;
    TListBox *ListBox1;
    TOpenDialog *OpenDialog1;
    TListBox *ListBox2;
    TPopupMenu *PopupMenu1;
    TMenuItem *N1;
    TMenuItem *N2;
    TTimer *Timer2;
    TMemo *Memo1;
    TOpenDialog *OpenDialog2;
    TTimer *Timer3;
    TEdit *Edit2;
    TEdit *Edit3;
    TTrackBar *TrackBar2;
    TPopupMenu *PopupMenu2;
    TMenuItem *MenuItem1;
    TMenuItem *MenuItem2;
    TCheckBox *CheckBox1;
```

```

TTrayIcon *TrayIcon1;
TStatusBar *StatusBar1;
TMediaPlayer *MediaPlayer2;
TImage *Image3;
TPopupMenu *PopupMenu3;
TMenuItem *MenuItem3;
TMenuItem *MenuItem4;
TEdit *Edit4;
TImage *Image2;
TImage *Image4;
TImage *Image5;
TImage *Image6;
TImage *Image7;
TImage *Image8;
TImage *Image9;
TImage *Image10;
TImage *Image11;
TImage *Image12;
TImage *Image13;
TImage *Image14;
TImage *Image16;
TPopupMenu *PopupMenu4;
TMenuItem *MenuItem5;
TMenuItem *MenuItem6;
TMenuItem *N3;
TSaveDialog *SaveDialog1;
TOpenDialog *OpenDialog3;
TEdit *Edit5;
TCheckBox *CheckBox2;
TMenuItem *HTML1;
TMemo *Memo2;
TSaveDialog *SaveDialog2;
TImage *Image15;
TPopupMenu *PopupMenu5;
TMenuItem *MenuItem7;
TMenuItem *MenuItem8;
TMenuItem *N4;
TMenuItem *N5;
TMenuItem *N6;
TPopupMenu *PopupMenu6;
TMenuItem *MenuItem9;
TMenuItem *MenuItem10;
TTimer *Timer4;
TEdit *Edit6;
TEdit *Edit7;
void __fastcall TrackBar1Change(TObject *Sender);
void __fastcall Timer1Timer(TObject *Sender);
void __fastcall Button2MouseDown(TObject *Sender,
    TMouseButton Button, TShiftState Shift, int X, int Y);
void __fastcall Panel1Click(TObject *Sender);

```

```

void __fastcall Panel3Click(TObject *Sender);
void __fastcall Panel2Click(TObject *Sender);
void __fastcall Panel4Click(TObject *Sender);
void __fastcall FormClose(TObject *Sender, TCloseAction &Action);
void __fastcall FormShow(TObject *Sender);
void __fastcall ListBox2DbClick(TObject *Sender);
void __fastcall N1Click(TObject *Sender);
void __fastcall N2Click(TObject *Sender);
void __fastcall Timer2Timer(TObject *Sender);
void __fastcall Panel5Click(TObject *Sender);
void __fastcall Panel6Click(TObject *Sender);
void __fastcall Panel8Click(TObject *Sender);
void __fastcall Panel9Click(TObject *Sender);
void __fastcall Timer3Timer(TObject *Sender);
void __fastcall Edit3Click(TObject *Sender);
void __fastcall Panel7Click(TObject *Sender);
void __fastcall Panel10Click(TObject *Sender);
void __fastcall MenuItem1Click(TObject *Sender);
void __fastcall MenuItem2Click(TObject *Sender);
void __fastcall ListBox2KeyDown(TObject *Sender, WORD &Key,
    TShiftState Shift);
void __fastcall MenuItem3Click(TObject *Sender);
void __fastcall MenuItem4Click(TObject *Sender);
void __fastcall FormPaint(TObject *Sender);
void __fastcall Image11Click(TObject *Sender);
void __fastcall Image7Click(TObject *Sender);
void __fastcall MenuItem5Click(TObject *Sender);
void __fastcall MenuItem6Click(TObject *Sender);
void __fastcall N3Click(TObject *Sender);
void __fastcall HTML1Click(TObject *Sender);
void __fastcall MenuItem7Click(TObject *Sender);
void __fastcall MenuItem8Click(TObject *Sender);
void __fastcall N4Click(TObject *Sender);
void __fastcall N5Click(TObject *Sender);
void __fastcall N6Click(TObject *Sender);
void __fastcall MenuItem10Click(TObject *Sender);
void __fastcall MenuItem9Click(TObject *Sender);
void __fastcall Timer4Timer(TObject *Sender);
void __fastcall Button1Click(TObject *Sender);
private:    // User declarations
//    friend void __fastcall initLang();
public:    // User declarations
    __fastcall TForm3(TComponent* Owner);
    double timetoplay; // время проигрывания
    int musstat; // позиция проигрывания
    int multiplybkg; // размножить фон
    AnsiString BOO;
    bool cashstat;
    int itemscount;
    WideString exec; // строка запуска

```

```
};
//-----
extern PACKAGE TForm3 *Form3;
//-----
#endif
```

Unit3.cpp

```
//-----

#include <vcl.h>
#pragma hdrstop

#include "Unit6.h"
#include "Unit5.h"
#include "Unit4.h"
#include "Unit3.h"
#include "Unit8.h"
#include <Math.hpp>
#include "library.h"
#include <string.h>
#include "filectrl.hpp"
//-----

#pragma package(smart_init)
#pragma link "trayicon"
#pragma resource "*.dfm"
TForm3 *Form3;
AnsiString ModeStr[7]= {"Не готово", "Приостановлено", "Прослушивание", "Запись",
"Поиск", "Пауза", "Открыто"};
int reloadstat=0;
int reloadstat2=0;
int reloadmax=0;
bool timestat=0; // отсчет времени с начала или до конца
bool picstat=false; // состояние открытия окна просмотра картинок
int scrolledit=0,scrollstat=0; // состояние скроллера, бегущих строк
AnsiString E4T,SBT; // строка скролла
int randstat=0; // позиция рандомизации
int *randommus; // массив рандомизации

int sortstat=0; /* статус сортировки
0 - несортировано
1 - сортировка по имени файла А-Я
2 - сортировка по имени файла Я-А
3 - сортировка по алфавиту А-Я
4 - сортировка по алфавиту Я-А
5 - сортировка по песне А-Я
6 - сортировка по песне Я-А */
```



```
void __fastcall DelDupe();
void __fastcall MiscList();
void __fastcall SortByFile();
void __fastcall SortByGroup();
void __fastcall SortBySong();
void __fastcall initLang();
void __fastcall nullRemoval();
void __fastcall setStatistic(AnsiString Group, AnsiString Song, AnsiString AccessTime,
AnsiString Length, AnsiString FileName);
void __fastcall GetStatPrepare();
```

```
struct TAG // объявляем структуру ID3 тега ver. 1.1
{
unsigned char tag [ 03 ]; // сигнатура 'TAG' 3 байта
unsigned char title [ 30 ]; // название 30 байт
unsigned char artist [ 30 ]; // исполнитель 30 байт
unsigned char album [ 30 ]; // альбом 30 байт
unsigned char year [ 04 ]; // год 4 байта
unsigned char comment [ 28 ]; // комментарий 28 байт
unsigned char sep [ 01 ]; // нулевой байт
unsigned char track [ 01 ]; // номер трека 1 байт
unsigned char genre [ 01 ]; // код стиля 1 байт
};
```

```
// Функция для чтения ID3 тегов ( параметр - имя файла )
AnsiString __fastcall GetMp3Tag ( AnsiString fName , int callparam )
```

```
{
    HANDLE hFile;
    DWORD dwFileSize;
    DWORD dwBytesRead;
    DWORD dwTagSize;
    DWORD dwStartPos;
    TAG tMp3;
    char c[31];

    hFile = CreateFile ( fName.c_str (), GENERIC_READ, 0, NULL, OPEN_EXISTING,
FILE_ATTRIBUTE_NORMAL, NULL );
    if ( hFile != INVALID_HANDLE_VALUE )
    {
        dwFileSize = GetFileSize ( hFile, NULL );
        dwTagSize = sizeof ( TAG );
        dwStartPos = dwFileSize - dwTagSize;
        SetFilePointer ( hFile, dwStartPos, NULL, FILE_BEGIN );
        ReadFile ( hFile, &tMp3, dwTagSize, &dwBytesRead, NULL );
    }
    CloseHandle ( hFile );
    memmove( c, tMp3.tag, 30 ); c[30] = 0;
    // MessageBox( NULL, c, "Title", NULL );
    if (callparam==0)
    return c;
```

```

    memmove( c, tMp3.title, 30 ); c[30] = 0;
//    MessageBox( NULL, c, "Title", NULL );
    if (callparam==1)
        return c;
    memmove( c, tMp3.artist, 30 ); c[30] = 0;
//    MessageBox( NULL, c, "Artist", NULL );
    if (callparam==2)
        return c;
    memmove( c, tMp3.album, 30 ); c[30] = 0;
//    MessageBox( NULL, c, "Album", NULL );
    if (callparam==3)
        return c;
    memmove( c, tMp3.year, 4 ); c[4] = 0;
//    MessageBox( NULL, c, "Year", NULL );
    if (callparam==4)
        return c;
    memmove( c, tMp3.comment, 28 ); c[28] = 0;
//    MessageBox( NULL, c, "Comment", NULL );
    if (callparam==5)
        return c;
}

```

```

void __fastcall GetStatPrepare()
{
    if (!Form3->cashstat)
    {
        bool grrerror=true;
        AnsiString RS=Form3->ListBox2->Items->Strings[Form3->musstat], SGroup=Form3->ListBox2->Items->Strings[Form3->musstat],
        SSong=Form3->ListBox2->Items->Strings[Form3->musstat],
        SLength=Form3->ListBox2->Items->Strings[Form3->musstat], SFileName=crofilename(Form3->ListBox1->Items->Strings[Form3->musstat]);
        int i=1;
        while (i<=RS.Length())
        {
            if (i<=RS.Length()-3)
            {
                SGroup[i]=RS[i];
                if (RS[i]==' ' && RS[i+1]=='-' && RS[i+2]==' ')
                {
                    grrerror=false;
                    SGroup[i]='\0';
                    break;
                }
            }
            i++;
        }
        if (grrerror)
        {
            SGroup="Unknown";

```

```

i=1;
}
else i=i+3;
int j=1;
if (i+1>=RS.Length())
ShowMessage("fuck!");
bool fatal_err=false;
for (;;)
{
if (RS[i]=='(' && (RS[i+2]=='\:' || RS[i+3]=='\:') && (RS[i+5]=='\" || RS[i+6]=='\"'))
{
fatal_err=true;
i--;j--;
break;
}
SSong[j]=RS[i];
i++; j++;
if (RS[i]=='(' && (RS[i+2]=='\:' || RS[i+3]=='\:') && (RS[i+5]=='\" || RS[i+6]=='\"'))
break;
}
if (!fatal_err)
SSong[j-1]='\0';
else { SSong="NoName"; i++; }
i++; j=1;
while (RS[i]!='\')
{
SLength[j]=RS[i];
i++; j++;
}
SLength[j]='\0';
setStatistic(SGroup,SSong,Now(),SLength,SFileName);
}
}

```

```

void __fastcall setStatistic(AnsiString Group, AnsiString Song, AnsiString AccessTime, AnsiString
Length, AnsiString FileName)
{
AnsiString SG1,SS1,SF1,SG2,SS2,SF2;
bool existrecord=false;
Form8->Table1->First();
while (!Form8->Table1->Eof)
{
int correct=0;
Form3->Edit6->Text=Group;
Form3->Edit7->Text=Form8->Table1->FieldByName("Group_c")->AsString;
if (Form3->Edit6->Text==Form3->Edit7->Text)
correct++;
Form3->Edit6->Text=Song;
Form3->Edit7->Text=Form8->Table1->FieldByName("Song_c")->AsString;
if (Form3->Edit6->Text==Form3->Edit7->Text)

```

```

correct++;
Form3->Edit6->Text=FileName;
Form3->Edit7->Text=Form8->Table1->FieldByName("FileName_c")->AsString;
if (Form3->Edit6->Text==Form3->Edit7->Text)
correct++;
if (correct==3)
{
existrecord=true;
Form8->Table1->Edit();
Form8->Table1->FieldByName("Access_c")->AsInteger=Form8->Table1-
>FieldByName("Access_c")->AsInteger+1;
Form8->Table1->FieldByName("LastAccess_c")->AsString=Now();
break;
}
Form8->Table1->Next();
}
if (!existrecord)
{
Form8->Table1->Insert();
Form8->Table1->FieldByName("Group_c")->AsString=Group;
Form8->Table1->FieldByName("Song_c")->AsString=Song;
Form8->Table1->FieldByName("Access_c")->AsInteger=1;
Form8->Table1->FieldByName("FirstAccess_c")->AsString=AccessTime;
Form8->Table1->FieldByName("LastAccess_c")->AsString=AccessTime;
Form8->Table1->FieldByName("FileName_c")->AsString=FileName;
Form8->Table1->FieldByName("UserLength_c")->AsString=Length;
Form8->Table1->Post();
}
}

//-----
__fastcall TForm3::TForm3(TComponent* Owner)
: TForm(Owner)
{
}
//-----

void __fastcall TForm3::TrackBar1Change(TObject *Sender)
{
try
{
try
{
Edit1->Text=TrackBar1->Position;
reloadstat2=(StrToInt(Edit1->Text)-1);
try
{
Image1->Picture->LoadFromFile(Memo1->Lines->Strings[StrToInt(Edit1->Text)-1]);
}
}
}
}

```

```

catch (...) {}
}
catch ( EInvalidGraphic& )
{
    //
    Memo1->Lines->Delete(StrToInt(Edit1->Text)-1);
    Memo1->Lines->SaveToFile("db\\pic.db");
    TrackBar1->Max=getlenght("db\\pic.db"); } //extension
}
catch ( EOpenError& )
{
    Memo1->Lines->Delete(StrToInt(Edit1->Text)-1);
    Memo1->Lines->SaveToFile("db\\pic.db");
    TrackBar1->Max=getlenght("db\\pic.db"); } //extension //no such file
}
//-----

```

```

void __fastcall TForm3::Timer1Timer(TObject *Sender)
{
    int maxedit,maxstat;
    scrolledit++;scrollstat++;
    maxedit=E4T.Length();
    maxstat=SBT.Length();
    if (scrollstat==maxstat)
    {
        scrollstat=0;
        StatusBar1->SimpleText=SBT;
    }
    try
    {
        Edit4->Text="";
        for (int i=scrolledit;i<=maxedit;i++)
            Edit4->Text=Edit4->Text+E4T[i];
        if (scrolledit==maxedit)
            scrolledit=0;
    }
    catch ( ERangeError& ) { scrolledit=0; Edit4->Text=E4T; }
    try
    {
        StatusBar1->SimpleText="";
        for (int i=scrollstat;i<maxstat-1;i++)
            StatusBar1->SimpleText=StatusBar1->SimpleText+SBT[i];
        if (scrollstat==maxstat-1)
            scrollstat=0;
    }
    catch ( ERangeError& ) { scrollstat=0; StatusBar1->SimpleText=SBT; }
}
//-----

```

```

void __fastcall TForm3::Button2MouseDown(TObject *Sender,
    TMouseButton Button, TShiftState Shift, int X, int Y)
{
    Form3->Timer1->Enabled=false;
    if (OpenDialog1->Execute())
    {
        ListBox1->Items->Add(OpenDialog1->FileName);
        ListBox2->Items->Add(AnsiString(crofilename(OpenDialog1->FileName)));
    }
    Form3->Timer1->Enabled=true;
}
//-----
void __fastcall TForm3::Panel1Click(TObject *Sender)
{
    try
    {
        Form3->Timer1->Enabled=false;
        Form3->Timer2->Enabled=false;
    }
    catch (...) {}
    if (OpenDialog1->Execute())
    {
        ListBox1->Items->Add(OpenDialog1->FileName);
        if (!Form6->CheckBox1->Checked)
            ListBox2->Items->Add(AnsiString(crofilename(OpenDialog1->FileName)));
        else
        {
            AnsiString TTagMP3=GetMp3Tag(OpenDialog1->FileName,0);
            if ((TTagMP3[1]=='T') && (TTagMP3[2]=='A') && (TTagMP3[3]=='G'))
                ListBox2->Items->Add(AnsiString(GetMp3Tag(OpenDialog1->FileName,2))+
                    "+AnsiString(GetMp3Tag(OpenDialog1->FileName,1)));
            else ListBox2->Items->Add(AnsiString(crofilename(OpenDialog1->FileName)));
        }
        if (!cashstat)
        {
            AnsiString TR;
            TR=MediaPlayer1->FileName;
            try
            {
                MediaPlayer1->Stop();
            }
            catch (...) {}
            MediaPlayer2->FileName=OpenDialog1->FileName;
            MediaPlayer2->Open();
            timetoplay+=MediaPlayer2->Length;
            int         retvalue=0,retvalue2=0;         retvalue=Floor((((StrToFloat(Form3->MediaPlayer2-
                >Length))/1000)/60)-(StrToInt(Form3->MediaPlayer2->Length)/1000)/60)*60);
            AnsiString TIME0;
            if (retvalue>9)
                TIME0=FloatToStr((StrToInt(Form3->MediaPlayer2-

```

```

>Length)/1000)/60)+":"+FloatToStr(retvalue);
else
TIME0=FloatToStr((StrToInt(Form3->MediaPlayer2-
>Length)/1000)/60)+"0"+FloatToStr(retvalue);
Form3->ListBox2->Items->Strings[ListBox2->Items->Count-1]=Form3->ListBox2->Items-
>Strings[ListBox2->Items->Count-1]+" \"("+TIME0+"\" ";
MediaPlayer2->Stop();
MediaPlayer2->Close();
MediaPlayer2->FileName=NULL;
MediaPlayer1->FileName=TR;
try {
MediaPlayer1->Open();
MediaPlayer1->Position=TrackBar2->Position;
MediaPlayer1->Play();
int  retvalue=0,retvalue2=0;  retvalue=Floor((((StrToFloat(MediaPlayer1->Length))/1000)/60)-
(StrToInt(MediaPlayer1->Length)/1000)/60)*60);      if      (retvalue>9)      Edit2-
>Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":"+FloatToStr(retvalue);      else
Edit2->Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+"0"+FloatToStr(retvalue);
E4T="  ***  "+AnsiString(ListBox2->Items->Strings[musstat]) + "  ***  ";
}
catch ( EMCIDeviceError& ) { }
}
}
}
catch ( EMCIDeviceError& ) { }
Form3->Timer1->Enabled=true;
Form3->Timer2->Enabled=true;
try { delete randommus; } catch (...) { }
randommus = new int[ListBox2->Items->Count*10];
randomize();
for (int i=0; i<ListBox2->Items->Count*10; i++)
randommus[i]=random(ListBox2->Items->Count);
if (!cashstat)
{
//nullRemoval();
AnsiString TSTR=ListBox2->Items->Strings[ListBox2->Items->Count-1];
int errs=0;
for (int j=1;j<TSTR.Length();j++)
{
if (j+1!=TSTR.Length())
{
if (TSTR[j]==' ' && TSTR[j+1]==' ')
{
errs++;
int h=j;
for (h=j;h<TSTR.Length();h++)
{
if(h+1!=TSTR.Length())
TSTR[h]=TSTR[h+1];
}
}
}
}
}

```

```

        j--;
    }
} //
} // for j
if (errs!=0)
TSTR[TSTR.Length()-errs]='\0';
else TSTR[TSTR.Length()-errs]='\0';
ListBox2->Items->Strings[ListBox2->Items->Count-1]=TSTR;
//
}
}
catch (...) { }
}
//-----

void __fastcall TForm3::Panel3Click(TObject *Sender)
{
if (ListBox2->ItemIndex>= 0)
{
Edit4->Text="          ***          "+AnsiString(GetMp3Tag(ListBox1->Items->Strings[ListBox2-
>ItemIndex],2))+ "          -          "+AnsiString(GetMp3Tag(ListBox1->Items->Strings[ListBox2-
>ItemIndex],1))+ " (";
scrolledit=0;
MediaPlayer1->FileName=ListBox1->Items->Strings[ListBox2->ItemIndex];
MediaPlayer1->Open();
MediaPlayer1->Play();
if (Form6->CheckBox4->Checked) GetStatPrepare();
if (TrackBar2->Enabled)
TrackBar2->Enabled=false;
musstat=ListBox2->ItemIndex;
Timer3->Enabled=true;
int  retvalue=0,retvalue2=0;  retvalue=Floor((((StrToFloat(MediaPlayer1->Length))/1000)/60)-
(StrToInt(MediaPlayer1->Length)/1000)/60)*60);          if          (retvalue>9)          Edit2-
>Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":"+FloatToStr(retvalue);          else
Edit2->Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":0"+FloatToStr(retvalue);
E4T="  ***  "+AnsiString(ListBox2->Items->Strings[musstat])+ "  ***  ";
}
}
//-----

void __fastcall TForm3::Panel2Click(TObject *Sender)
{
if (ListBox2->ItemIndex>= 0)
{
while(ListBox2->SelCount)
{
ListBox2->ItemIndex=0;
while(!ListBox2->Selected[ListBox2->ItemIndex])
ListBox2->ItemIndex++;
if (MediaPlayer1->FileName!=ListBox1->Items->Strings[ListBox2->ItemIndex])

```



```

{
MediaPlayer2->FileName=ListBox1->Items->Strings[ListBox2->ItemIndex];
MediaPlayer2->Open();
timetoplay=MediaPlayer2->Length;
MediaPlayer2->Stop();
MediaPlayer2->Close();
MediaPlayer2->FileName=NULL;
}
else timetoplay=MediaPlayer1->Length;
ListBox1->Items->Delete(ListBox2->ItemIndex);
ListBox2->Items->Delete(ListBox2->ItemIndex);
}
try {
int  retvalue=0,retvalue2=0;  retvalue=Floor((((StrToFloat(MediaPlayer1->Length))/1000)/60)-
(StrToInt(MediaPlayer1->Length)/1000)/60)*60);          if          (retvalue>9)          Edit2-
>Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":"+FloatToStr(retvalue);          else
Edit2->Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":0"+FloatToStr(retvalue);
E4T="  ***  "+AnsiString(ListBox2->Items->Strings[musstat]) + "  ***  ";
} catch (...) {}
}
delete randommus;
randommus = new int[ListBox2->Items->Count*10];
randomize();
for (int i=0; i<ListBox2->Items->Count*10; i++)
randommus[i]=random(ListBox2->Items->Count);
}
//-----

void __fastcall TForm3::Panel4Click(TObject *Sender)
{
if (MediaPlayer1->Mode==2 || MediaPlayer1->Mode == 1)
{
Timer3->Enabled=!(Timer3->Enabled);
TrackBar2->Enabled=!(TrackBar2->Enabled);
MediaPlayer1->Position=TrackBar2->Position;
if (!TrackBar2->Enabled)
{
MediaPlayer1->Play();
if (Form6->CheckBox4->Checked) GetStatPrepare();
int  retvalue=0,retvalue2=0;  retvalue=Floor((((StrToFloat(MediaPlayer1->Length))/1000)/60)-
(StrToInt(MediaPlayer1->Length)/1000)/60)*60);          if          (retvalue>9)          Edit2-
>Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":"+FloatToStr(retvalue);          else
Edit2->Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":0"+FloatToStr(retvalue);
E4T="  ***  "+AnsiString(ListBox2->Items->Strings[musstat]) + "  ***  ";
}
}
else
{
MediaPlayer1->Pause();
StatusBar1->SimpleText="          "+ModeStr[MediaPlayer1->Mode] + " ||  Всего песен: " +
IntToStr(ListBox1->Items->Count) + " || Сейчас: " + Now() + " ";
}
}

```

```
SBT=StatusBar1->SimpleText;
TrayIcon1->Hint="Джим";
}
}
}
//-----
```

```
void __fastcall TForm3::FormClose(TObject *Sender, TCloseAction &Action)
{
Form6->Memo2->Clear();
Form6->Memo2->Lines->Add(Form6->LabeledEdit3->Text);
Form6->Memo2->Lines->Add(Form3->Left);
Form6->Memo2->Lines->Add(Form3->Top);
Form6->Memo2->Lines->Add(Form6->ComboBox3->ItemIndex);
int tempbool;
tempbool=Form3->CheckBox1->Checked;
Form6->Memo2->Lines->Add(IntToStr(tempbool));
tempbool=Form3->CheckBox2->Checked;
Form6->Memo2->Lines->Add(IntToStr(tempbool));
Form6->Memo2->Lines->Add(Form4->Label2->Caption);
Form6->Memo2->Lines->Add(Form5->Label2->Caption);
tempbool=Form6->CheckBox4->Checked;
Form6->Memo2->Lines->Add(IntToStr(tempbool));
Form6->Memo2->Lines->Add(Form8->LabeledEdit3->Text);
Form6->Memo2->Lines->Add(Form8->LabeledEdit5->Text);
Form6->Memo2->Lines->SaveToFile("jim.ini");
Application->Terminate();
}
//-----
```

```
void __fastcall TForm3::FormShow(TObject *Sender)
{
cashstat=0;
itemscount=0;
Memo1->Clear(); //
Memo1->Lines->LoadFromFile("jim.ini");
/* initialize init status */
Form6->Memo1->Lines->LoadFromFile(Memo1->Lines->Strings[0]);
Form6->LabeledEdit3->Text=Memo1->Lines->Strings[0];
Form6->Memo2->Lines->LoadFromFile("jim.ini");
Form3->Left=StrToInt(Form6->Memo2->Lines->Strings[1]);
Form3->Top=StrToInt(Form6->Memo2->Lines->Strings[2]);
Form6->ComboBox3->ItemIndex=StrToInt(Form6->Memo2->Lines->Strings[3]);
Form3->CheckBox1->Checked=StrToInt(Form6->Memo2->Lines->Strings[4]);
Form3->CheckBox2->Checked=StrToInt(Form6->Memo2->Lines->Strings[5]);
Form4->Label2->Caption=Form6->Memo2->Lines->Strings[6];
Form5->Label2->Caption=Form6->Memo2->Lines->Strings[7];
Form6->CheckBox4->Checked=StrToInt(Form6->Memo2->Lines->Strings[8]);
Form8->LabeledEdit3->Text=IntToStr(StrToInt(Form6->Memo2->Lines->Strings[9])+1);
if (Form6->Memo2->Lines->Strings[10]=="0")
```

```

Form8->LabeledEdit5->Text=Now();
else Form8->LabeledEdit5->Text=Form6->Memo2->Lines->Strings[10];
Memo1->Clear(); //
initLang();
Edit1->Color=StrToInt(Form6->Memo1->Lines->Strings[0]);
Edit2->Color=StrToInt(Form6->Memo1->Lines->Strings[0]);
Edit3->Color=StrToInt(Form6->Memo1->Lines->Strings[0]);
Edit4->Color=StrToInt(Form6->Memo1->Lines->Strings[0]);
Edit1->Font->Name=Form6->Memo1->Lines->Strings[1];
Edit1->Font->Color=StrToInt(Form6->Memo1->Lines->Strings[2]);
Edit1->Font->Charset=StrToInt(Form6->Memo1->Lines->Strings[3]);
Edit1->Font->Size=StrToInt(Form6->Memo1->Lines->Strings[4]);
Edit2->Font->Name=Form6->Memo1->Lines->Strings[1];
Edit2->Font->Color=StrToInt(Form6->Memo1->Lines->Strings[2]);
Edit2->Font->Charset=StrToInt(Form6->Memo1->Lines->Strings[3]);
Edit2->Font->Size=StrToInt(Form6->Memo1->Lines->Strings[4]);
Edit3->Font->Name=Form6->Memo1->Lines->Strings[1];
Edit3->Font->Color=StrToInt(Form6->Memo1->Lines->Strings[2]);
Edit3->Font->Charset=StrToInt(Form6->Memo1->Lines->Strings[3]);
Edit3->Font->Size=StrToInt(Form6->Memo1->Lines->Strings[4]);
Edit4->Font->Name=Form6->Memo1->Lines->Strings[1];
Edit4->Font->Color=StrToInt(Form6->Memo1->Lines->Strings[2]);
Edit4->Font->Charset=StrToInt(Form6->Memo1->Lines->Strings[3]);
Edit4->Font->Size=StrToInt(Form6->Memo1->Lines->Strings[4]);
ListBox2->Color=StrToInt(Form6->Memo1->Lines->Strings[5]);
ListBox2->Font->Name=Form6->Memo1->Lines->Strings[6];
ListBox2->Font->Color=StrToInt(Form6->Memo1->Lines->Strings[7]);
ListBox2->Font->Charset=StrToInt(Form6->Memo1->Lines->Strings[8]);
ListBox2->Font->Size=StrToInt(Form6->Memo1->Lines->Strings[9]);
Form3->Color=StrToInt(Form6->Memo1->Lines->Strings[10]);
Form3->Font->Name=Form6->Memo1->Lines->Strings[11];
Form3->Font->Color=StrToInt(Form6->Memo1->Lines->Strings[12]);
Form3->Font->Charset=StrToInt(Form6->Memo1->Lines->Strings[13]);
Form3->Font->Size=StrToInt(Form6->Memo1->Lines->Strings[14]);
StatusBar1->Color=StrToInt(Form6->Memo1->Lines->Strings[15]);
StatusBar1->Font->Name=Form6->Memo1->Lines->Strings[16];
StatusBar1->Font->Color=StrToInt(Form6->Memo1->Lines->Strings[17]);
StatusBar1->Font->Charset=StrToInt(Form6->Memo1->Lines->Strings[18]);
StatusBar1->Font->Size=StrToInt(Form6->Memo1->Lines->Strings[19]);
Form3->AlphaBlendValue=StrToInt(Form6->Memo1->Lines->Strings[20]);
Form3->Timer2->Interval=StrToInt(Form6->Memo1->Lines->Strings[21]);
Image3->Picture->LoadFromFile(Form6->Memo1->Lines->Strings[22]);
Form6->Image1->Picture->LoadFromFile(Form6->Memo1->Lines->Strings[22]);
Form6->LabeledEdit2->Text=Form6->Memo1->Lines->Strings[22];
Image2->Picture->LoadFromFile(Form6->Memo1->Lines->Strings[23]);
Image10->Picture->LoadFromFile(Form6->Memo1->Lines->Strings[24]);
Image9->Picture->LoadFromFile(Form6->Memo1->Lines->Strings[25]);
Image8->Picture->LoadFromFile(Form6->Memo1->Lines->Strings[26]);
Image4->Picture->LoadFromFile(Form6->Memo1->Lines->Strings[27]);
Image5->Picture->LoadFromFile(Form6->Memo1->Lines->Strings[28]);

```

```

Image6->Picture->LoadFromFile(Form6->Memo1->Lines->Strings[29]);
Form3->Image7->Picture->LoadFromFile(Form6->Memo1->Lines->Strings[30]);
Form3->Image11->Picture->LoadFromFile(Form6->Memo1->Lines->Strings[31]);
Form3->Image12->Picture->LoadFromFile(Form6->Memo1->Lines->Strings[32]);
Form3->Image13->Picture->LoadFromFile(Form6->Memo1->Lines->Strings[33]);
Form3->Image14->Picture->LoadFromFile(Form6->Memo1->Lines->Strings[34]);
Form6->ListBox1->Items->Strings[0]=Form6->Memo1->Lines->Strings[23];
Form6->ListBox1->Items->Strings[1]=Form6->Memo1->Lines->Strings[24];
Form6->ListBox1->Items->Strings[2]=Form6->Memo1->Lines->Strings[25];
Form6->ListBox1->Items->Strings[3]=Form6->Memo1->Lines->Strings[26];
Form6->ListBox1->Items->Strings[4]=Form6->Memo1->Lines->Strings[27];
Form6->ListBox1->Items->Strings[5]=Form6->Memo1->Lines->Strings[28];
Form6->ListBox1->Items->Strings[6]=Form6->Memo1->Lines->Strings[29];
Form6->ListBox1->Items->Strings[7]=Form6->Memo1->Lines->Strings[30];
Form6->ListBox1->Items->Strings[8]=Form6->Memo1->Lines->Strings[31];
Form6->ListBox1->Items->Strings[9]=Form6->Memo1->Lines->Strings[32];
Form6->ListBox1->Items->Strings[10]=Form6->Memo1->Lines->Strings[33];
Form6->ListBox1->Items->Strings[11]=Form6->Memo1->Lines->Strings[34];
Form6->ListBox1->Items->Strings[12]=Form6->Memo1->Lines->Strings[42];
Form3->multiplybkg=StrToInt(Form6->Memo1->Lines->Strings[35]);
Form6->CheckBox2->Checked=multiplybkg;
Form6->CheckBox1->Checked=StrToInt(Form6->Memo1->Lines->Strings[36]);
Form6->CheckBox3->Checked=StrToInt(Form6->Memo1->Lines->Strings[37]);
Form6->LabeledEdit4->Text=Form6->Memo1->Lines->Strings[38];
Form6->LabeledEdit5->Text=Form6->Memo1->Lines->Strings[39];
Form6->LabeledEdit6->Text=Form6->Memo1->Lines->Strings[40];
Form6->LabeledEdit7->Text=Form6->Memo1->Lines->Strings[41];
Image15->Picture->LoadFromFile(Form6->Memo1->Lines->Strings[42]);
Form3->ListBox2->Width=StrToInt(Form6->LabeledEdit4->Text);
Form3->ListBox2->Height=StrToInt(Form6->LabeledEdit5->Text);
Form3->Image1->Width=StrToInt(Form6->LabeledEdit6->Text);
Form3->Image1->Height=StrToInt(Form6->LabeledEdit7->Text);
Form3->Image1->Left=StrToInt(Form6->LabeledEdit4->Text);
Form3->Height=ListBox2->Top+ListBox2->Height+80;
Form3->Width=ListBox2->Width;
TrackBar2->Width=ListBox2->Width;
Image4->Top=ListBox2->Top+ListBox2->Height+2;
Image5->Top=ListBox2->Top+ListBox2->Height+2;
Image6->Top=ListBox2->Top+ListBox2->Height+2;
Image7->Top=ListBox2->Top+ListBox2->Height+2;
Image11->Top=ListBox2->Top+ListBox2->Height+2;
Image15->Top=ListBox2->Top+ListBox2->Height+2;
Image12->Left=ListBox2->Width+10;
Image13->Left=ListBox2->Width+Image12->Width+15;
Image14->Left=ListBox2->Width+Image12->Width+Image13->Width+20;
Image12->Top=Image1->Height+20;
Image13->Top=Image1->Height+20;
Image14->Top=Image1->Height+20;
TrackBar1->Top=Image1->Height+20;
Edit1->Top=Image1->Height+20;

```

```

Edit1->Left=ListBox2->Width+Image12->Width+Image13->Width+Image14->Width+25;
TrackBar1->Left=Edit1->Width+ListBox2->Width+Image12->Width+Image13->Width+Image14->Width+25;
Image16->Picture->LoadFromFile("animation\\split.bmp");
/* end of init*/
if (Form6->CheckBox2->Checked==false)
{
/* single block skin */
for (int i=0; i<=Image3->Picture->Width; i++)
{
for (int j=Image3->Picture->Height;j>=0;j--)
{
Form3->Canvas->Pixels[i][j]=Image3->Canvas->Pixels[i][j];
}
}
}
else
{
/* multiply block skin */
for (int i=0; i<=Form3->Width; i++)
{
for (int j=Form3->Height;j>=0;j--)
{
int Wcoeff,Hcoeff;
if (i>Image3->Picture->Width && j>Image3->Picture->Height)
{
Wcoeff=i/Image3->Picture->Width;
Hcoeff=j/Image3->Picture->Height;
Form3->Canvas->Pixels[i][j]=Image3->Canvas->Pixels[i-Wcoeff*Image3->Picture->Width][j-Hcoeff*Image3->Picture->Height];
}
else if (i>Image3->Picture->Width)
{
Wcoeff=i/Image3->Picture->Width;
Form3->Canvas->Pixels[i][j]=Image3->Canvas->Pixels[i-Wcoeff*Image3->Picture->Width][j];
}
else if (j>Image3->Picture->Height)
{
Hcoeff=j/Image3->Picture->Height;
Form3->Canvas->Pixels[i][j]=Image3->Canvas->Pixels[i][j-Hcoeff*Image3->Picture->Height];
}
else
Form3->Canvas->Pixels[i][j]=Image3->Canvas->Pixels[i][j];
}
}
}
musstat=0;
try
{
ListBox1->Items->LoadFromFile("db\\mp3.db");

```

```

ListBox2->Items->LoadFromFile("db\\mp3init.db");
TrackBar1->Max=getlenght("db\\pic.db");
Memo1->Lines->LoadFromFile("db\\pic.db");
}
catch ( EOpenError& ) {ShowMessage("Нарушение целостности программы. Программа будет
закрpppрыта."); Application->Terminate(); }
try
{
try
{
Form3->timetoplay=0;
for (int i=0; i<Form3->ListBox1->Items->Count; i++)
{
Form3->MediaPlayer2->FileName=Form3->ListBox1->Items->Strings[i];
Form3->MediaPlayer2->Open();
Form3->timetoplay+=Form3->MediaPlayer2->Length;
Form3->MediaPlayer2->Close();
}
if (ListBox1->Items->Count>0)
{
Edit4->Text="          ***          "+AnsiString(GetMp3Tag(ListBox1->Items->Strings[0],2))+ "   -
"+AnsiString(GetMp3Tag(ListBox1->Items->Strings[0],1))+ " (";
scrolledit=0;
MediaPlayer1->FileName=ListBox1->Items->Strings[0];
MediaPlayer1->Open();
MediaPlayer1->Play();
Timer3->Enabled=true;
int  retvalue=0,retvalue2=0;  retvalue=Floor((((StrToFloat(MediaPlayer1->Length))/1000)/60)-
(StrToInt(MediaPlayer1->Length)/1000)/60)*60);          if          (retvalue>9)          Edit2-
>Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":"+FloatToStr(retvalue);          else
Edit2->Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":0"+FloatToStr(retvalue);
E4T="  ***  "+AnsiString(ListBox2->Items->Strings[musstat]) + "  ***  ";
}
}
catch ( EStringListError& ) { }
musstat=0;
ListBox2->ItemIndex=musstat;
}
catch ( EMCIDeviceError& ) { Timer3->Enabled=false; musstat=0;}
randommus = new int[ListBox2->Items->Count*10];
randomize();
for (int i=0; i<ListBox2->Items->Count*10; i++)
randommus[i]=random(ListBox2->Items->Count);
}
//-----

void __fastcall TForm3::ListBox2DblClick(TObject *Sender)
{
if (ListBox2->ItemIndex>= 0)
{

```

```

Edit4->Text="          ***          "+AnsiString(GetMp3Tag(ListBox1->Items->Strings[ListBox2-
>ItemIndex],2))+ "          -          "+AnsiString(GetMp3Tag(ListBox1->Items->Strings[ListBox2-
>ItemIndex],1))+ " (";
scrolledit=0;
MediaPlayer1->FileName=ListBox1->Items->Strings[ListBox2->ItemIndex];
MediaPlayer1->Open();
MediaPlayer1->Play();
if (TrackBar2->Enabled)
TrackBar2->Enabled=false;
musstat=ListBox2->ItemIndex;
if (Form6->CheckBox4->Checked) GetStatPrepare();
int  retvalue=0,retvalue2=0;  retvalue=Floor((((StrToFloat(MediaPlayer1->Length))/1000)/60)-
(StrToInt(MediaPlayer1->Length)/1000)/60)*60);          if          (retvalue>9)          Edit2-
>Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":"+FloatToStr(retvalue);          else
Edit2->Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":0"+FloatToStr(retvalue);
E4T="  ***  "+AnsiString(ListBox2->Items->Strings[musstat]) + "  ***  ";
Timer3->Enabled=true;
}
}
//-----

```

```

void __fastcall TForm3::N1Click(TObject *Sender)
{
try {
if (ListBox2->ItemIndex>= 0)
{
while(ListBox2->SelCount)
{
ListBox2->ItemIndex=0;
while(!ListBox2->Selected[ListBox2->ItemIndex])
ListBox2->ItemIndex++;
if (MediaPlayer1->FileName!=ListBox1->Items->Strings[ListBox2->ItemIndex])
{
MediaPlayer2->FileName=ListBox1->Items->Strings[ListBox2->ItemIndex];
MediaPlayer2->Open();
timetoplay-=MediaPlayer2->Length;
MediaPlayer2->Stop();
MediaPlayer2->Close();
MediaPlayer2->FileName=NULL;
}
else timetoplay-=MediaPlayer1->Length;
ListBox1->Items->Delete(ListBox2->ItemIndex);
ListBox2->Items->Delete(ListBox2->ItemIndex);
}
}
try {
int  retvalue=0,retvalue2=0;  retvalue=Floor((((StrToFloat(MediaPlayer1->Length))/1000)/60)-
(StrToInt(MediaPlayer1->Length)/1000)/60)*60);          if          (retvalue>9)          Edit2-
>Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":"+FloatToStr(retvalue);          else
Edit2->Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":0"+FloatToStr(retvalue);
E4T="  ***  "+AnsiString(ListBox2->Items->Strings[musstat]) + "  ***  ";

```

```

} catch (...) {}
}
delete randommus;
randommus = new int[ListBox2->Items->Count*10];
randomize();
for (int i=0; i<ListBox2->Items->Count*10; i++)
randommus[i]=random(ListBox2->Items->Count);
} catch (...) {}
}
//-----

```

```

void __fastcall TForm3::N2Click(TObject *Sender)
{
try {
ListBox1->SelectAll();
ListBox2->SelectAll();
ListBox2->DeleteSelected();
ListBox1->DeleteSelected();
timetoplay=0;
try {
delete randommus;    // массив рандомизации
}
catch (...) {}
sortstat=0;
randstat=0;
cashstat=false;
itemscount=0;
Timer4->Enabled=false;
musstat=0;
} catch (...) {}
}
//-----

```

```

void __fastcall TForm3::Timer2Timer(TObject *Sender)
{
try
{
try
{
Memo1->Lines->Clear();
Memo1->Lines->LoadFromFile("db\\pic.db");
TrackBar1->Max=Memo1->Lines->Count;
if (reloadstat2<Memo1->Lines->Count-1)
reloadstat2++;
else { reloadstat2=0; }
Edit1->Text=reloadstat2+1;
try {
try {

```



```
Image1->Picture->LoadFromFile(Memo1->Lines->Strings[reloadstat2]);
}
catch (...) {reloadstat2=0; }
}
catch ( EInvalidGraphic& ) { }
TrackBar1->Position=StrToInt(Edit1->Text);
}
catch ( EOpenError& ) {ShowMessage("Нарушение целостности программы. Программа будет
закррррыта."); Application->Terminate(); }
}
catch (...) { }
}
//-----
```

```
void __fastcall TForm3::Panel5Click(TObject *Sender)
{
ListBox1->Items->SaveToFile("db\\mp3.db");
ListBox2->Items->SaveToFile("db\\mp3init.db");
ShowMessage("Плейлист сохранен");
}
//-----
```

```
void __fastcall TForm3::Panel6Click(TObject *Sender)
{
Form3->Timer1->Enabled=false;
Form3->Timer2->Enabled=false;
if (OpenDialog2->Execute())
Memo1->Lines->Add(OpenDialog2->FileName);
Memo1->Lines->SaveToFile("db\\pic.db");
Form3->Timer1->Enabled=true;
Form3->Timer2->Enabled=true;
TrackBar1->Max=getlenght("db\\pic.db");
}
//-----
```

```
void __fastcall TForm3::Panel8Click(TObject *Sender)
{
Memo1->Lines->Delete(TrackBar1->Position-1);
Memo1->Lines->SaveToFile("db\\pic.db");
TrackBar1->Max=getlenght("db\\pic.db");
}
//-----
```

```
void __fastcall TForm3::Panel9Click(TObject *Sender)
{
Timer2->Enabled=!(Timer2->Enabled);
}
//-----
```

```
void __fastcall TForm3::Timer3Timer(TObject *Sender)
```

```

{
try
{
int retvalue=0,retvalue2=0;
retvalue=Floor((((StrToFloat(MediaPlayer1->Length))/1000)/60)-(StrToInt(MediaPlayer1-
>Length)/1000)/60)*60);
if (retvalue>9)
Edit2->Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":"+FloatToStr(retvalue);
else
Edit2->Text=FloatToStr((StrToInt(MediaPlayer1-
>Length)/1000)/60)+":0"+FloatToStr(retvalue);
if (!timestat)
{

retvalue=Floor((((StrToFloat(MediaPlayer1->Position))/1000)/60)-(StrToInt(MediaPlayer1-
>Position)/1000)/60)*60);
if (retvalue>9)
Edit3->Text=FloatToStr((StrToInt(MediaPlayer1->Position)/1000)/60)+":"+FloatToStr(retvalue);
else
Edit3->Text=FloatToStr((StrToInt(MediaPlayer1->Position)/1000)/60)+":0"+FloatToStr(retvalue);
}
else
{
retvalue=Floor((((StrToFloat(MediaPlayer1->Length)-StrToFloat(MediaPlayer1-
>Position))/1000)/60)-((StrToInt(MediaPlayer1->Length)-StrToInt(MediaPlayer1-
>Position))/1000)/60)*60);
if (retvalue>9)
Edit3->Text=FloatToStr(((StrToInt(MediaPlayer1->Length)-StrToInt(MediaPlayer1-
>Position))/1000)/60)+":"+FloatToStr(retvalue);
else
Edit3->Text=FloatToStr(((StrToInt(MediaPlayer1->Length)-StrToInt(MediaPlayer1-
>Position))/1000)/60)+":0"+FloatToStr(retvalue);
}
TrackBar2->Max=MediaPlayer1->Length;
TrackBar2->Position=MediaPlayer1->Position;
if (MediaPlayer1->Position==MediaPlayer1->Length)
{
if (!CheckBox1->Checked)
{
if (ListBox2->Items->Count>0)
{
try
{
musstat++;
try
{
MediaPlayer1->FileName=ListBox1->Items->Strings[musstat];
int retvalue=0,retvalue2=0;
retvalue=Floor((((StrToFloat(MediaPlayer1->Length))/1000)/60)-(StrToInt(MediaPlayer1-
>Length)/1000)/60)*60);
if (retvalue>9)

```

```

Edit2->Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":"+FloatToStr(retvalue);
else
    Edit2->Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":0"+FloatToStr(retvalue);
scrolledit=0;
ListBox2->ItemIndex=musstat;
}
catch ( EStringListError& ) {
musstat=0;
MediaPlayer1->FileName=ListBox1->Items->Strings[musstat];
ListBox2->ItemIndex=musstat;
MediaPlayer1->Open();
MediaPlayer1->Play();
if (Form6->CheckBox4->Checked) GetStatPrepare();
int  retvalue=0,retvalue2=0;  retvalue=Floor((((StrToFloat(MediaPlayer1->Length))/1000)/60)-
(StrToInt(MediaPlayer1->Length)/1000)/60)*60);      if      (retvalue>9)      Edit2-
>Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":"+FloatToStr(retvalue);      else
Edit2->Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":0"+FloatToStr(retvalue);
E4T="  ***  "+AnsiString(ListBox2->Items->Strings[musstat]) + "  ***  ";
}
MediaPlayer1->Open();
MediaPlayer1->Play();
if (Form6->CheckBox4->Checked) GetStatPrepare();
int  retvalue=0,retvalue2=0;  retvalue=Floor((((StrToFloat(MediaPlayer1->Length))/1000)/60)-
(StrToInt(MediaPlayer1->Length)/1000)/60)*60);      if      (retvalue>9)      Edit2-
>Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":"+FloatToStr(retvalue);      else
Edit2->Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":0"+FloatToStr(retvalue);
E4T="  ***  "+AnsiString(ListBox2->Items->Strings[musstat]) + "  ***  ";
}
catch ( EMCIDeviceError& ) {
musstat++;
scrolledit=0;
MediaPlayer1->FileName=ListBox1->Items->Strings[musstat];
ListBox2->ItemIndex=musstat;
MediaPlayer1->Open();
MediaPlayer1->Play();
if (Form6->CheckBox4->Checked) GetStatPrepare();
int  retvalue=0,retvalue2=0;  retvalue=Floor((((StrToFloat(MediaPlayer1->Length))/1000)/60)-
(StrToInt(MediaPlayer1->Length)/1000)/60)*60);      if      (retvalue>9)      Edit2-
>Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":"+FloatToStr(retvalue);      else
Edit2->Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":0"+FloatToStr(retvalue);
E4T="  ***  "+AnsiString(ListBox2->Items->Strings[musstat]) + "  ***  ";
}
if (!CheckBox2->Checked && musstat==0)
MediaPlayer1->Stop();
}
}
else
{
if (ListBox2->Items->Count>0)
{

```

```

if (!Form6->CheckBox3->Checked)
{
int numelem;
if (ListBox1->Items->Count!=0)
numelem=ListBox1->Items->Count-1;
musstat=random(numelem);
scrolledit=0;
MediaPlayer1->FileName=ListBox1->Items->Strings[musstat];
ListBox2->ItemIndex=musstat;
MediaPlayer1->Open();
MediaPlayer1->Play();
if (Form6->CheckBox4->Checked) GetStatPrepare();
int retvalue=0,retvalue2=0; retvalue=Floor((((StrToFloat(MediaPlayer1->Length))/1000)/60)-
(StrToInt(MediaPlayer1->Length)/1000)/60)*60); if (retvalue>9) Edit2-
>Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":"+FloatToStr(retvalue); else
Edit2->Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":0"+FloatToStr(retvalue);
E4T=" *** "+AnsiString(ListBox2->Items->Strings[musstat])+" *** ";
}
else //если интеллектуальное
{
if (randstat<ListBox2->Items->Count*10)
randstat++;
else randstat=0;
musstat=randommus[randstat];
scrolledit=0;
MediaPlayer1->FileName=ListBox1->Items->Strings[musstat];
ListBox2->ItemIndex=musstat;
MediaPlayer1->Open();
MediaPlayer1->Play();
if (Form6->CheckBox4->Checked) GetStatPrepare();
int retvalue=0,retvalue2=0; retvalue=Floor((((StrToFloat(MediaPlayer1->Length))/1000)/60)-
(StrToInt(MediaPlayer1->Length)/1000)/60)*60); if (retvalue>9) Edit2-
>Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":"+FloatToStr(retvalue); else
Edit2->Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":0"+FloatToStr(retvalue);
E4T=" *** "+AnsiString(ListBox2->Items->Strings[musstat])+" *** ";
}
}
}
}
if (MediaPlayer1->Mode==2)
{
AnsiString retlength,TS;
TS=AnsiString(timetoplay);
retvalue=0;
retvalue2=0;
retvalue=Floor((((StrToFloat(TS))/1000)/60)-(StrToInt(TS)/1000)/60)*60);
retvalue2=Floor((((StrToFloat(TS))/1000)/60)/60)-((StrToInt(TS)/1000)/60)/60)*60);
if (retvalue>9 && retvalue2>9)
retlength=FloatToStr(((StrToInt(TS)/1000)/60)/60)+":"+FloatToStr(retvalue2)+":"+FloatToStr(retva
lue);
}

```

```

else if (retvalue<9 && retvalue2>9)
retlength=FloatToStr(((StrToInt(TS)/1000)/60)/60)+":"+FloatToStr(retvalue2)+":0"+FloatToStr(retv
alue);
else if (retvalue<9 && retvalue2<9)
retlength=FloatToStr(((StrToInt(TS)/1000)/60)/60)+":0"+FloatToStr(retvalue2)+":0"+FloatToStr(ret
value);
else
retlength=FloatToStr(((StrToInt(TS)/1000)/60)/60)+":0"+FloatToStr(retvalue2)+":"+FloatToStr(retv
alue);
if (Form6->ComboBox3->ItemIndex==0)
TrayIcon1->Hint="Джим играет " + AnsiString(crofilename(MediaPlayer1->FileName));
else if (Form6->ComboBox3->ItemIndex==1)
TrayIcon1->Hint="Jim playing " + AnsiString(crofilename(MediaPlayer1->FileName));
else if (Form6->ComboBox3->ItemIndex==2)
TrayIcon1->Hint="Jim spiel " + AnsiString(crofilename(MediaPlayer1->FileName));
if (Form6->ComboBox3->ItemIndex==0)
SBT=" "+ModeStr[MediaPlayer1->Mode]+ " песни " + AnsiString(crofilename(MediaPlayer1-
>FileName)) + " || всего песен: " + IntToStr(ListBox1->Items->Count) + " || " + retlength + " ||
Сейчас: " + Now() + " ";
if (Form6->ComboBox3->ItemIndex==1)
SBT=" Playing song " + AnsiString(crofilename(MediaPlayer1->FileName)) + " || Total: " +
IntToStr(ListBox1->Items->Count) + " || " + retlength + " || Now: " + Now() + " ";
if (Form6->ComboBox3->ItemIndex==2)
SBT=" Spiele singe " + AnsiString(crofilename(MediaPlayer1->FileName)) + " || Alle: " +
IntToStr(ListBox1->Items->Count) + " || " + retlength + " || Jetzt: " + Now() + " ";
}
}
catch ( EMCIDeviceError& ) { }
}
//-----

```

```

void __fastcall TForm3::Edit3Click(TObject *Sender)
{
timestat=!timestat;
}

```

```

//-----

```

```

void __fastcall TForm3::Panel7Click(TObject *Sender)
{
if (!CheckBox1->Checked)
{
if (ListBox2->Items->Count>0)
{
if (musstat!=0)
{
musstat--;
scrolledit=0;
MediaPlayer1->FileName=ListBox1->Items->Strings[musstat];
ListBox2->ItemIndex=musstat;
MediaPlayer1->Open();
MediaPlayer1->Play();
if (Form6->CheckBox4->Checked) GetStatPrepare();
int  retvalue=0,retvalue2=0;  retvalue=Floor((((StrToFloat(MediaPlayer1->Length))/1000)/60)-
(StrToInt(MediaPlayer1->Length)/1000)/60)*60);          if          (retvalue>9)          Edit2-
>Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":"+FloatToStr(retvalue);          else
Edit2->Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":0"+FloatToStr(retvalue);
E4T="  ***  "+AnsiString(ListBox2->Items->Strings[musstat]) + "  ***  ";
}
else
{
musstat=ListBox2->Items->Count-1;
scrolledit=0;
MediaPlayer1->FileName=ListBox1->Items->Strings[musstat];
ListBox2->ItemIndex=musstat;
MediaPlayer1->Open();
MediaPlayer1->Play();
if (Form6->CheckBox4->Checked) GetStatPrepare();
int  retvalue=0,retvalue2=0;  retvalue=Floor((((StrToFloat(MediaPlayer1->Length))/1000)/60)-
(StrToInt(MediaPlayer1->Length)/1000)/60)*60);          if          (retvalue>9)          Edit2-
>Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":"+FloatToStr(retvalue);          else
Edit2->Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":0"+FloatToStr(retvalue);
E4T="  ***  "+AnsiString(ListBox2->Items->Strings[musstat]) + "  ***  ";
}
}
}
else
{
if (!Form6->CheckBox3->Checked)
{
int numelem;
if (ListBox1->Items->Count!=0)
numelem=ListBox1->Items->Count-1;
musstat=random(numelem);
scrolledit=0;
MediaPlayer1->FileName=ListBox1->Items->Strings[musstat];
ListBox2->ItemIndex=musstat;
MediaPlayer1->Open();
}
}
}

```

```

MediaPlayer1->Play();
if (Form6->CheckBox4->Checked) GetStatPrepare();
int  retvalue=0,retvalue2=0;  retvalue=Floor((((StrToFloat(MediaPlayer1->Length))/1000)/60)-
(StrToInt(MediaPlayer1->Length)/1000)/60)*60);          if          (retvalue>9)          Edit2-
>Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":"+FloatToStr(retvalue);          else
Edit2->Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":0"+FloatToStr(retvalue);
E4T="  ***  "+AnsiString(ListBox2->Items->Strings[musstat]) + "  ***  ";
}
else //если интеллектуальное
{
if (ListBox1->Items->Count!=0)
{
if (randstat>0)
randstat--;
else randstat=randommus[ListBox2->Items->Count*10-1];
musstat=randommus[randstat];
scrolledit=0;
MediaPlayer1->FileName=ListBox1->Items->Strings[musstat];
ListBox2->ItemIndex=musstat;
MediaPlayer1->Open();
MediaPlayer1->Play();
if (Form6->CheckBox4->Checked) GetStatPrepare();
int  retvalue=0,retvalue2=0;  retvalue=Floor((((StrToFloat(MediaPlayer1->Length))/1000)/60)-
(StrToInt(MediaPlayer1->Length)/1000)/60)*60);          if          (retvalue>9)          Edit2-
>Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":"+FloatToStr(retvalue);          else
Edit2->Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":0"+FloatToStr(retvalue);
E4T="  ***  "+AnsiString(ListBox2->Items->Strings[musstat]) + "  ***  ";
}
else { randstat=0; musstat=0; MediaPlayer1->Stop(); }

}
}
}
//-----

```

```

void __fastcall TForm3::Panel10Click(TObject *Sender)
{
if (!CheckBox1->Checked)
{
if (ListBox2->Items->Count>0)
{
try
{
if (musstat+1<ListBox2->Items->Count)
musstat++;
else { musstat=0;}
}
try
{

```

```

int  retvalue=0,retvalue2=0;  retvalue=Floor((((StrToFloat(MediaPlayer1->Length))/1000)/60)-
(StrToInt(MediaPlayer1->Length)/1000)/60)*60);          if          (retvalue>9)          Edit2-
>Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":"+FloatToStr(retvalue);          else
Edit2->Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":0"+FloatToStr(retvalue);
E4T="  ***  "+AnsiString(ListBox2->Items->Strings[musstat]) + "  ***  ";
scrolledit=0;
}
catch ( EStringListError& ) {
musstat=0;
scrolledit=0;
MediaPlayer1->FileName=ListBox1->Items->Strings[musstat];
MediaPlayer1->Open();
MediaPlayer1->Play();
if (Form6->CheckBox4->Checked) GetStatPrepare();
int  retvalue=0,retvalue2=0;  retvalue=Floor((((StrToFloat(MediaPlayer1->Length))/1000)/60)-
(StrToInt(MediaPlayer1->Length)/1000)/60)*60);          if          (retvalue>9)          Edit2-
>Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":"+FloatToStr(retvalue);          else
Edit2->Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":0"+FloatToStr(retvalue);
E4T="  ***  "+AnsiString(ListBox2->Items->Strings[musstat]) + "  ***  ";
}
MediaPlayer1->FileName=ListBox1->Items->Strings[musstat];
ListBox2->ItemIndex=musstat;
}
catch ( EStringListError& ) {
musstat=0;
scrolledit=0;
MediaPlayer1->FileName=ListBox1->Items->Strings[musstat];
MediaPlayer1->Open();
MediaPlayer1->Play();
if (Form6->CheckBox4->Checked) GetStatPrepare();
int  retvalue=0,retvalue2=0;  retvalue=Floor((((StrToFloat(MediaPlayer1->Length))/1000)/60)-
(StrToInt(MediaPlayer1->Length)/1000)/60)*60);          if          (retvalue>9)          Edit2-
>Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":"+FloatToStr(retvalue);          else
Edit2->Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":0"+FloatToStr(retvalue);
E4T="  ***  "+AnsiString(ListBox2->Items->Strings[musstat]) + "  ***  ";
}
MediaPlayer1->Open();
MediaPlayer1->Play();
if (Form6->CheckBox4->Checked) GetStatPrepare();
int  retvalue=0,retvalue2=0;  retvalue=Floor((((StrToFloat(MediaPlayer1->Length))/1000)/60)-
(StrToInt(MediaPlayer1->Length)/1000)/60)*60);          if          (retvalue>9)          Edit2-
>Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":"+FloatToStr(retvalue);          else
Edit2->Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":0"+FloatToStr(retvalue);
E4T="  ***  "+AnsiString(ListBox2->Items->Strings[musstat]) + "  ***  ";
}
catch ( EMCIDeviceError& ) {
musstat++;
scrolledit=0;
MediaPlayer1->FileName=ListBox1->Items->Strings[musstat];
MediaPlayer1->Open();

```



```

MediaPlayer1->Play();
if (Form6->CheckBox4->Checked) GetStatPrepare();
int  retvalue=0,retvalue2=0;  retvalue=Floor((((StrToFloat(MediaPlayer1->Length))/1000)/60)-
(StrToInt(MediaPlayer1->Length)/1000)/60)*60);          if      (retvalue>9)          Edit2-
>Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":"+FloatToStr(retvalue);      else
Edit2->Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":0"+FloatToStr(retvalue);
E4T="  ***  "+AnsiString(ListBox2->Items->Strings[musstat]) + "  ***  ";
}
if (!CheckBox2->Checked && musstat==0)
MediaPlayer1->Stop();
}
}
else
{
if (!Form6->CheckBox3->Checked)
{
int numelem;
if (ListBox1->Items->Count!=0)
numelem=ListBox1->Items->Count-1;
musstat=random(numelem);
scrolledit=0;
MediaPlayer1->FileName=ListBox1->Items->Strings[musstat];
ListBox2->ItemIndex=musstat;
MediaPlayer1->Open();
MediaPlayer1->Play();
if (Form6->CheckBox4->Checked) GetStatPrepare();
int  retvalue=0,retvalue2=0;  retvalue=Floor((((StrToFloat(MediaPlayer1->Length))/1000)/60)-
(StrToInt(MediaPlayer1->Length)/1000)/60)*60);          if      (retvalue>9)          Edit2-
>Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":"+FloatToStr(retvalue);      else
Edit2->Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":0"+FloatToStr(retvalue);
E4T="  ***  "+AnsiString(ListBox2->Items->Strings[musstat]) + "  ***  ";
}
else //если интеллектуальное
{
if (ListBox1->Items->Count!=0)
{
if (randstat<ListBox2->Items->Count*10)
randstat++;
else randstat=0;
musstat=randommus[randstat];
scrolledit=0;
MediaPlayer1->FileName=ListBox1->Items->Strings[musstat];
ListBox2->ItemIndex=musstat;
MediaPlayer1->Open();
MediaPlayer1->Play();
if (Form6->CheckBox4->Checked) GetStatPrepare();
int  retvalue=0,retvalue2=0;  retvalue=Floor((((StrToFloat(MediaPlayer1->Length))/1000)/60)-
(StrToInt(MediaPlayer1->Length)/1000)/60)*60);          if      (retvalue>9)          Edit2-
>Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":"+FloatToStr(retvalue);      else
Edit2->Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":0"+FloatToStr(retvalue);

```

```
E4T=" *** "+AnsiString(ListBox2->Items->Strings[musstat])+" *** ";
}
else { randstat=0; musstat=0; MediaPlayer1->Stop(); }
}
}
}
//-----
```

```
void __fastcall TForm3::MenuItem1Click(TObject *Sender)
{
try
{
Form3->Timer1->Enabled=false;
Form3->Timer2->Enabled=false;
try
{
if (OpenDialog1->Execute())
{
ListBox1->Items->Add(OpenDialog1->FileName);
if (!Form6->CheckBox1->Checked)
ListBox2->Items->Add(AnsiString(crofilename(OpenDialog1->FileName)));
else
{
AnsiString TTagMP3=GetMp3Tag(OpenDialog1->FileName,0);
if ((TTagMP3[1]=='T') && (TTagMP3[2]=='A') && (TTagMP3[3]=='G'))
ListBox2->Items->Add(AnsiString(GetMp3Tag(OpenDialog1->FileName,2))+
"+AnsiString(GetMp3Tag(OpenDialog1->FileName,1)));
else ListBox2->Items->Add(AnsiString(crofilename(OpenDialog1->FileName)));
}
if (!cashstat)
{
AnsiString TR;
TR=MediaPlayer1->FileName;
try
{
MediaPlayer1->Stop();
}
catch (...) { }
MediaPlayer2->FileName=OpenDialog1->FileName;
MediaPlayer2->Open();
timetoplay+=MediaPlayer2->Length;
int retvalue=0,retvalue2=0; retvalue=Floor((((StrToFloat(Form3->MediaPlayer2-
>Length))/1000)/60)-(StrToInt(Form3->MediaPlayer2->Length)/1000)/60)*60);
AnsiString TIME0;
if (retvalue>9)
TIME0=FloatToStr((StrToInt(Form3->MediaPlayer2-
>Length)/1000)/60)+":"+FloatToStr(retvalue);
else
TIME0=FloatToStr((StrToInt(Form3->MediaPlayer2-
```

```

>Length)/1000)/60)+":0"+FloatToStr(retvalue);
Form3->ListBox2->Items->Strings[ListBox2->Items->Count-1]=Form3->ListBox2->Items-
>Strings[ListBox2->Items->Count-1]+" \("+TIME0+"\) ";
MediaPlayer2->Stop();
MediaPlayer2->Close();
MediaPlayer2->FileName=NULL;
MediaPlayer1->FileName=TR;
try {
MediaPlayer1->Open();
MediaPlayer1->Position=TrackBar2->Position;
MediaPlayer1->Play();
int  retvalue=0,retvalue2=0;  retvalue=Floor((((StrToFloat(MediaPlayer1->Length))/1000)/60)-
(StrToInt(MediaPlayer1->Length)/1000)/60)*60);          if      (retvalue>9)          Edit2-
>Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+": "+FloatToStr(retvalue);      else
Edit2->Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":0"+FloatToStr(retvalue);
E4T="  ***  "+AnsiString(ListBox2->Items->Strings[musstat]) + "  ***  ";
}
catch ( EMCIDeviceError& ) { }
}
}
}
catch ( EMCIDeviceError& ) { }
Form3->Timer1->Enabled=true;
Form3->Timer2->Enabled=true;
try { delete randommus; } catch (...) { }
randommus = new int[ListBox2->Items->Count*10];
randomize();
for (int i=0; i<ListBox2->Items->Count*10; i++)
randommus[i]=random(ListBox2->Items->Count);
if (!cashstat)
{
//nullRemoval();
AnsiString TSTR=ListBox2->Items->Strings[ListBox2->Items->Count-1];
int errs=0;
for (int j=1;j<TSTR.Length();j++)
{
if (j+1!=TSTR.Length())
{
if (TSTR[j]==' ' && TSTR[j+1]==' ')
{
errs++;
int h=j;
for (h=j;h<TSTR.Length();h++)
{
if(h+1!=TSTR.Length())
TSTR[h]=TSTR[h+1];
}
j--;
}
} //

```

```

} // for j
if (errs!=0)
TSTR[TSTR.Length()-errs]='\0';
else TSTR[TSTR.Length()-errs]='\0';
ListBox2->Items->Strings[ListBox2->Items->Count-1]=TSTR;
//
}
}
catch (...) {}
}
//-----

```

```

void __fastcall TForm3::MenuItem2Click(TObject *Sender)
{
try {
try {
Form3->Timer1->Enabled=false;
Form3->Timer2->Enabled=false;
Form3->Timer3->Enabled=false;
Form4->ShowModal();
try
{
delete randommus;
}
catch (...) {}
randommus = new int[ListBox2->Items->Count*10];
randomize();
for (int i=0; i<ListBox2->Items->Count*10; i++)
randommus[i]=random(ListBox2->Items->Count);
}
catch ( EMCIDeviceError& ) {}
//nullRemoval();
timetoplay=0;
Form3->Timer1->Enabled=true;
Form3->Timer2->Enabled=true;
Form3->Timer3->Enabled=true;
}
catch (...) {}
}
//-----

```

```

void __fastcall TForm3::ListBox2KeyDown(TObject *Sender, WORD &Key,
TShiftState Shift)

```

```

{
if (Key==VK_DELETE)
{
if (ListBox2->ItemIndex>= 0)
{
while(ListBox2->SelCount)
{
ListBox2->ItemIndex=0;
while(!ListBox2->Selected[ListBox2->ItemIndex])
ListBox2->ItemIndex++;
if (MediaPlayer1->FileName!=ListBox1->Items->Strings[ListBox2->ItemIndex])
{
MediaPlayer2->FileName=ListBox1->Items->Strings[ListBox2->ItemIndex];
MediaPlayer2->Open();
timetoplay=MediaPlayer2->Length;
MediaPlayer2->Stop();
MediaPlayer2->Close();
MediaPlayer2->FileName=NULL;
}
else timetoplay=MediaPlayer1->Length;
ListBox1->Items->Delete(ListBox2->ItemIndex);
ListBox2->Items->Delete(ListBox2->ItemIndex);
}
try {
int  retvalue=0,retvalue2=0;  retvalue=Floor((((StrToFloat(MediaPlayer1->Length))/1000)/60)-
(StrToInt(MediaPlayer1->Length)/1000)/60)*60);          if      (retvalue>9)          Edit2-
>Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":"+FloatToStr(retvalue);          else
Edit2->Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":0"+FloatToStr(retvalue);
E4T="  ***  "+AnsiString(ListBox2->Items->Strings[musstat]) + "  ***  ";
} catch (...) {}
}
delete randommus;
randommus = new int[ListBox2->Items->Count*10];
randomize();
for (int i=0; i<ListBox2->Items->Count*10; i++)
randommus[i]=random(ListBox2->Items->Count);
}
else if (Key==VK_RETURN)
if (ListBox2->ItemIndex>= 0)
{
MediaPlayer1->FileName=ListBox1->Items->Strings[ListBox2->ItemIndex];
MediaPlayer1->Open();
MediaPlayer1->Play();
int  retvalue=0,retvalue2=0;  retvalue=Floor((((StrToFloat(MediaPlayer1->Length))/1000)/60)-
(StrToInt(MediaPlayer1->Length)/1000)/60)*60);          if      (retvalue>9)          Edit2-
>Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":"+FloatToStr(retvalue);          else
Edit2->Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":0"+FloatToStr(retvalue);
E4T="  ***  "+AnsiString(ListBox2->Items->Strings[musstat]) + "  ***  ";
if (TrackBar2->Enabled)
TrackBar2->Enabled=false;
}
}

```

```

musstat=ListBox2->ItemIndex;
if (Form6->CheckBox4->Checked) GetStatPrepare();
Timer3->Enabled=true;
}
}
//-----

void __fastcall TForm3::MenuItem3Click(TObject *Sender)
{
Form3->Timer1->Enabled=false;
Form3->Timer2->Enabled=false;
if (OpenDialog2->Execute())
Memo1->Lines->Add(OpenDialog2->FileName);
Memo1->Lines->SaveToFile("db\\pic.db");
Form3->Timer1->Enabled=true;
Form3->Timer2->Enabled=true;
TrackBar1->Max=getlenght("db\\pic.db");
}
//-----

void __fastcall TForm3::MenuItem4Click(TObject *Sender)
{
Form3->Timer1->Enabled=false;
Form3->Timer2->Enabled=false;
Form5->ShowModal();
}
//-----

void __fastcall TForm3::FormPaint(TObject *Sender)
{
if (Form6->CheckBox2->Checked==false)
{
// single block skin
for (int i=0; i<=Image3->Picture->Width; i++)
{
for (int j=Image3->Picture->Height;j>=0;j--)
{
Form3->Canvas->Pixels[i][j]=Image3->Canvas->Pixels[i][j];
}
}
}
else
{
// multiply block skin
for (int i=0; i<=Form3->Width; i++)
{
for (int j=Form3->Height;j>=0;j--)
{
int Wcoeff,Hcoeff;

```

```

if (i>Image3->Picture->Width && j>Image3->Picture->Height)
{
Wcoeff=i/Image3->Picture->Width;
Hcoeff=j/Image3->Picture->Height;
Form3->Canvas->Pixels[i][j]=Image3->Canvas->Pixels[i-Wcoeff*Image3->Picture->Width][j-
Hcoeff*Image3->Picture->Height];
}
else if (i>Image3->Picture->Width)
{
Wcoeff=i/Image3->Picture->Width;
Form3->Canvas->Pixels[i][j]=Image3->Canvas->Pixels[i-Wcoeff*Image3->Picture->Width][j];
}
else if (j>Image3->Picture->Height)
{
Hcoeff=j/Image3->Picture->Height;
Form3->Canvas->Pixels[i][j]=Image3->Canvas->Pixels[i][j-Hcoeff*Image3->Picture->Height];
}
else
Form3->Canvas->Pixels[i][j]=Image3->Canvas->Pixels[i][j];
}
}
}
}
}

```

//-----

```

void __fastcall TForm3::Image11Click(TObject *Sender)
{
// открытие\свертка окна просмотра изображений
if (picstat)
{
picstat=false;
Form3->ClientWidth=StrToInt(Form6->LabeledEdit4->Text);
Timer2->Enabled=false;
}
else
{
picstat=true;
Form3->ClientWidth=StrToInt(Form6->LabeledEdit4->Text)+StrToInt(Form6->LabeledEdit6-
>Text);
Timer2->Enabled=true;
}
}
}
//-----

```

```

void __fastcall TForm3::Image7Click(TObject *Sender)
{
// вызов настроек
Form6->ShowModal();
}

```

```
//-----

void __fastcall TForm3::MenuItem5Click(TObject *Sender)
{
// сохранение плейлиста по умолчанию
ListBox1->Items->SaveToFile("db\\mp3.db");
ListBox2->Items->SaveToFile("db\\mp3init.db");
ShowMessage("Плейлист сохранен");
}
//-----

void __fastcall TForm3::MenuItem6Click(TObject *Sender)
{
// сохранение плейлиста
Timer1->Enabled=false;
Timer2->Enabled=false;
Timer3->Enabled=false;
if (SaveDialog1->Execute())
{
if (ExtractFileExt(SaveDialog1->FileName)!=".jpl"    &&    ExtractFileExt(SaveDialog1->
FileName)!=".JPL")
SaveDialog1->FileName=SaveDialog1->FileName+".jpl";
ListBox1->Items->SaveToFile(SaveDialog1->FileName);
}
Timer1->Enabled=true;
Timer2->Enabled=true;
Timer3->Enabled=true;
}
//-----

void __fastcall TForm3::N3Click(TObject *Sender)
{
// загрузка плейлиста
Timer1->Enabled=false;
Timer2->Enabled=false;
Timer3->Enabled=false;
if (OpenDialog3->Execute())
{
MediaPlayer1->Close();
ListBox1->SelectAll();
ListBox2->SelectAll();
ListBox2->DeleteSelected();
ListBox1->DeleteSelected();
timetoplay=0;
ListBox1->Items->LoadFromFile(OpenDialog3->FileName);

for (int i=0;i<ListBox1->Items->Count;i++)
{
if (!Form6->CheckBox1->Checked)
ListBox2->Items->Add(AnsiString(crofilename(ListBox1->Items->Strings[i])));
}
}
}
//-----
```



```

else
{
AnsiString TTagMP3=GetMp3Tag(ListBox1->Items->Strings[i],0);
if ((TTagMP3[1]=='T') && (TTagMP3[2]=='A') && (TTagMP3[3]=='G'))
ListBox2->Items->Add(AnsiString(GetMp3Tag(ListBox1->Items->Strings[i],2))+
"+AnsiString(GetMp3Tag(ListBox1->Items->Strings[i],1)));
else ListBox2->Items->Add(AnsiString(crofilename(ListBox1->Items->Strings[i])));
}
}
Form3->cashstat=true;
Form3->itemscount=0;
Form3->Timer4->Enabled=true; //!!!!
}
MediaPlayer1->FileName=ListBox1->Items->Strings[0];
MediaPlayer1->Open();
MediaPlayer1->Play();
if (Form6->CheckBox4->Checked) GetStatPrepare();
int retvalue=0,retvalue2=0; retvalue=Floor((((StrToFloat(MediaPlayer1->Length))/1000)/60)-
(StrToInt(MediaPlayer1->Length)/1000)/60)*60); if (retvalue>9) Edit2-
>Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":"+FloatToStr(retvalue); else
Edit2->Text=FloatToStr((StrToInt(MediaPlayer1->Length)/1000)/60)+":0"+FloatToStr(retvalue);
E4T=" *** "+AnsiString(ListBox2->Items->Strings[musstat])+" *** ";
Timer1->Enabled=true;
Timer2->Enabled=true;
Timer3->Enabled=true;
delete randommus;
randommus = new int[ListBox2->Items->Count*10];
randomize();
for (int i=0; i<ListBox2->Items->Count*10; i++)
randommus[i]=random(ListBox2->Items->Count);
}
//-----

void __fastcall TForm3::HTML1Click(TObject *Sender)
{
Timer2->Enabled=false;
// сохранение плейлиста в формате HTML
if (SaveDialog2->Execute())
{
AnsiString retlength,TS;
TS=AnsiString(timetoplay);
int retvalue=0;
int retvalue2=0;
retvalue=Floor((((StrToFloat(TS))/1000)/60)-(StrToInt(TS)/1000)/60)*60);
retvalue2=Floor((((StrToFloat(TS))/1000)/60)/60)-((StrToInt(TS)/1000)/60)/60)*60);
if (retvalue>9 && retvalue2>9)
retlength=FloatToStr(((StrToInt(TS)/1000)/60)/60)+":"+FloatToStr(retvalue2)+":"+FloatToStr(retva
lue);
else if (retvalue<9 && retvalue2>9)
retlength=FloatToStr(((StrToInt(TS)/1000)/60)/60)+":"+FloatToStr(retvalue2)+":0"+FloatToStr(retv

```

```

alue);
else if (retvalue<9 && retvalue2<9)
retlength=FloatToStr(((StrToInt(TS)/1000)/60)/60)+":0"+FloatToStr(retvalue2)+":0"+FloatToStr(ret
value);
else
retlength=FloatToStr(((StrToInt(TS)/1000)/60)/60)+":0"+FloatToStr(retvalue2)+":"+FloatToStr(retv
alue);
if (ExtractFileExt(SaveDialog2->FileName)!=".htm" && ExtractFileExt(SaveDialog2-
>FileName)!=".html")
SaveDialog2->FileName=SaveDialog2->FileName+".html";
Memo2->Clear();
Memo2->Lines->Add("<HTML>");
Memo2->Lines->Add("<HEAD>");
Memo2->Lines->Add("<TITLE>Jim Playlist, "+ Now()+"</TITLE>");
Memo2->Lines->Add("</HEAD>");
Memo2->Lines->Add("<BODY bgcolor=navy text=yellow link=yellow vlink=yellow
alink=yellow>");
Memo2->Lines->Add("<H1>Плейлист сгенерирован Джимом 2.0</H1><BR>");
for (int i=0;i<ListBox2->Items->Count;i++)
Memo2->Lines->Add(ListBox2->Items->Strings[i]+"<BR>");
Memo2->Lines->Add("<BR><BR>Всего в плейлисте песен: " + IntToStr(ListBox1->Items-
>Count) + "<BR>Общая продолжительность: " + retlength);
Memo2->Lines->Add("<BR><BR>Страница разработчика: <a
href='\"http://\\illusionist7.narod.ru/\">http://\\illusionist7.narod.ru</a>");
Memo2->Lines->Add("</BODY>");
Memo2->Lines->Add("</HTML>");
Memo2->Lines->SaveToFile(SaveDialog2->FileName);
}
Timer2->Enabled=true;
}
//-----
void __fastcall SortByGroup() // сортировка по группе (алфавиту)
{
if (sortstat!=3)
{
for (int j=0; j<Form3->ListBox2->Items->Count; j++)
{
for (int i=j; i<Form3->ListBox2->Items->Count; i++)
{
if (i!=j)
{
AnsiString STR1=Form3->ListBox2->Items->Strings[j],STR2=Form3->ListBox2->Items-
>Strings[i];
int maxlength;
if (STR1.Length()<STR2.Length())
maxlength=STR1.Length();
else maxlength=STR2.Length();
for (int k=1; k<maxlength; k++)
{
if (STR1[k]<STR2[k])

```

```

break;
if (STR2[k]<STR1[k])
{
    AnsiString tempname,tempname2;
    tempname=Form3->ListBox2->Items->Strings[j];
    tempname2=Form3->ListBox1->Items->Strings[j];
    Form3->ListBox2->Items->Strings[j]=Form3->ListBox2->Items->Strings[i];
    Form3->ListBox1->Items->Strings[j]=Form3->ListBox1->Items->Strings[i];
    Form3->ListBox2->Items->Strings[i]=tempname;
    Form3->ListBox1->Items->Strings[i]=tempname2;
    break;
}
}
}
}
sortstat=3;
}
else
{
    for (int j=0; j<Form3->ListBox2->Items->Count; j++)
    {
        for (int i=j; i<Form3->ListBox2->Items->Count; i++)
        {
            if (i!=j)
            {
                AnsiString      STR1=Form3->ListBox2->Items->Strings[j],STR2=Form3->ListBox2->Items-
>Strings[i];
                int maxlength;
                if (STR1.Length()<STR2.Length())
                    maxlength=STR1.Length();
                else maxlength=STR2.Length();
                for (int k=1; k<maxlength; k++)
                {
                    if (STR2[k]<STR1[k])
                        break;
                    if (STR1[k]<STR2[k])
                    {
                        AnsiString tempname,tempname2;
                        tempname=Form3->ListBox2->Items->Strings[j];
                        tempname2=Form3->ListBox1->Items->Strings[j];
                        Form3->ListBox2->Items->Strings[j]=Form3->ListBox2->Items->Strings[i];
                        Form3->ListBox1->Items->Strings[j]=Form3->ListBox1->Items->Strings[i];
                        Form3->ListBox2->Items->Strings[i]=tempname;
                        Form3->ListBox1->Items->Strings[i]=tempname2;
                        break;
                    }
                }
            }
        }
    }
}
}

```

```

}
sortstat=4;
}
}

void __fastcall SortBySong() // сортировка по песням
{
int k=1,k1=1,ks,ks1;
if (sortstat!=5)
{
for (int j=0; j<Form3->ListBox2->Items->Count; j++)
{
for (int i=j; i<Form3->ListBox2->Items->Count; i++)
{
int k=1,k1=1,ks=0,ks1=0;
if (i!=j)
{
AnsiString STR1=Form3->ListBox2->Items->Strings[j],STR2=Form3->ListBox2->Items-
>Strings[i];
for(;;)
{
if (STR1[k]==' ' && STR1[k+1]=='-' && STR1[k+2]==' ')
break;
k++;
}
for(;;)
{
if (STR2[k1]==' ' && STR2[k1+1]=='-' && STR2[k1+2]==' ')
break;
k1++;
}
ks=k+2;ks1=k1+2;
for (k=ks,k1=ks1; k<STR1.Length(),k1<STR2.Length(); k++,k1++)
{
if (STR1[k]<STR2[k1])
break;
if (STR2[k1]<STR1[k])
{
AnsiString tempname,tempname2;
tempname=Form3->ListBox2->Items->Strings[j];
tempname2=Form3->ListBox1->Items->Strings[j];
Form3->ListBox2->Items->Strings[j]=Form3->ListBox2->Items->Strings[i];
Form3->ListBox1->Items->Strings[j]=Form3->ListBox1->Items->Strings[i];
Form3->ListBox2->Items->Strings[i]=tempname;
Form3->ListBox1->Items->Strings[i]=tempname2;
break;
}
}
if (k==STR1.Length() || k1==STR2.Length())
break;
}
}

```

```

}
}
}
sortstat=5;
}
else
{
for (int j=0; j<Form3->ListBox2->Items->Count; j++)
{
for (int i=j; i<Form3->ListBox2->Items->Count; i++)
{
int k=1,k1=1,ks=0,ks1=0;
if (i!=j)
{
AnsiString          STR1=Form3->ListBox2->Items->Strings[j],STR2=Form3->ListBox2->Items-
>Strings[i];
for(;;)
{
if (STR1[k]==' ' && STR1[k+1]=='\' && STR1[k+2]==' ')
break;
k++;
}
for(;;)
{
if (STR2[k1]==' ' && STR2[k1+1]=='\' && STR2[k1+2]==' ')
break;
k1++;
}
ks=k+2;ks1=k1+2;
for (k=ks,k1=ks1; k<STR1.Length(),k1<STR2.Length(); k++,k1++)
{
if (STR1[k]>STR2[k1])
break;
if (STR2[k1]>STR1[k])
{
AnsiString tempname,tempname2;
tempname=Form3->ListBox2->Items->Strings[j];
tempname2=Form3->ListBox1->Items->Strings[j];
Form3->ListBox2->Items->Strings[j]=Form3->ListBox2->Items->Strings[i];
Form3->ListBox1->Items->Strings[j]=Form3->ListBox1->Items->Strings[i];
Form3->ListBox2->Items->Strings[i]=tempname;
Form3->ListBox1->Items->Strings[i]=tempname2;
break;
}
}
if (k==STR1.Length() || k1==STR2.Length())
break;
}
}
}
}
}

```

```

sortstat=6;
}
}

void __fastcall SortByFile() // сортировка по имени файла
{
if (sortstat!=1)
{
for (int j=0; j<Form3->ListBox2->Items->Count; j++)
{
for (int i=j; i<Form3->ListBox2->Items->Count; i++)
{
if (i!=j)
{
AnsiString          STR1=Form3->ListBox1->Items->Strings[j],STR2=Form3->ListBox1->Items-
>Strings[i];
int maxlength;
if (STR1.Length()<STR2.Length())
maxlength=STR1.Length();
else maxlength=STR2.Length();
for (int k=1; k<maxlength; k++)
{
if (STR1[k]<STR2[k])
break;
if (STR2[k]<STR1[k])
{
AnsiString tempname,tempname2;
tempname=Form3->ListBox2->Items->Strings[j];
tempname2=Form3->ListBox1->Items->Strings[j];
Form3->ListBox2->Items->Strings[j]=Form3->ListBox2->Items->Strings[i];
Form3->ListBox1->Items->Strings[j]=Form3->ListBox1->Items->Strings[i];
Form3->ListBox2->Items->Strings[i]=tempname;
Form3->ListBox1->Items->Strings[i]=tempname2;
break;
}
}
}
}
sortstat=1;
}
else
{
for (int j=0; j<Form3->ListBox2->Items->Count; j++)
{
for (int i=j; i<Form3->ListBox2->Items->Count; i++)
{
if (i!=j)
{
AnsiString          STR1=Form3->ListBox1->Items->Strings[j],STR2=Form3->ListBox1->Items-

```

```

>Strings[i];
int maxlength;
if (STR1.Length()<STR2.Length())
maxlength=STR1.Length();
else maxlength=STR2.Length();
for (int k=1; k<maxlength; k++)
{
if (STR1[k]>STR2[k])
break;
if (STR2[k]>STR1[k])
{
AnsiString tempname,tempname2;
tempname=Form3->ListBox2->Items->Strings[j];
tempname2=Form3->ListBox1->Items->Strings[j];
Form3->ListBox2->Items->Strings[j]=Form3->ListBox2->Items->Strings[i];
Form3->ListBox1->Items->Strings[j]=Form3->ListBox1->Items->Strings[i];
Form3->ListBox2->Items->Strings[i]=tempname;
Form3->ListBox1->Items->Strings[i]=tempname2;
break;
}
}
}
}
sortstat=2;
}
}

void __fastcall DelDupe() // удаление дубликатов
{
for (int j=0; j<Form3->ListBox2->Items->Count; j++)
{
for (int i=0; i<Form3->ListBox2->Items->Count; i++)
{
if (i!=j)
{
AnsiString STR1=Form3->ListBox1->Items->Strings[j],STR2=Form3->ListBox1->Items-
>Strings[i];
if (STR1==STR2)
{
AnsiString TR=Form3->MediaPlayer1->FileName;
Form3->MediaPlayer1->Stop();
Form3->MediaPlayer1->Close();
Form3->MediaPlayer2->FileName=Form3->ListBox1->Items->Strings[i];
Form3->MediaPlayer2->Open();
Form3->timetoplay-=Form3->MediaPlayer2->Length;
Form3->MediaPlayer2->Stop();
Form3->MediaPlayer2->Close();
Form3->MediaPlayer2->FileName=NULL;
Form3->MediaPlayer1->FileName=TR;

```

```
Form3->MediaPlayer1->Open();
Form3->MediaPlayer1->Position=Form3->TrackBar2->Position;
Form3->MediaPlayer1->Play();
Form3->ListBox1->Items->Delete(i);
Form3->ListBox2->Items->Delete(i);
    break;
}
}
}
}
}
```

```
void __fastcall MiscList() // перемешать лист
{
DelDupe();
AnsiString *misc;
AnsiString *misc2;
misc = new AnsiString[Form3->ListBox2->Items->Count];
misc2 = new AnsiString[Form3->ListBox2->Items->Count];
for (int i=0;i<Form3->ListBox2->Items->Count;i++)
{
misc[i]=0;
misc2[i]=0;
}
bool error=true;
int j;
for (int k=0; k<Form3->ListBox2->Items->Count;k++)
{
error=true;
randomize();
while (error)
{
j=random(Form3->ListBox2->Items->Count);
error=false;
for (int i=0;i<Form3->ListBox2->Items->Count;i++)
{ if (Form3->ListBox2->Items->Strings[j]==misc[i]) error=true; }
if (!error)
{
misc[k]=Form3->ListBox2->Items->Strings[j];
misc2[k]=Form3->ListBox1->Items->Strings[j];
}
}
}
for (int i=0;i<Form3->ListBox2->Items->Count;i++)
{
Form3->ListBox2->Items->Strings[i]=misc[i];
Form3->ListBox1->Items->Strings[i]=misc2[i];
}
}
```


//-----

```
void __fastcall TForm3::MenuItem7Click(TObject *Sender)
{
if (!cashstat)
SortByFile();
}
//-----
```

```
void __fastcall TForm3::MenuItem8Click(TObject *Sender)
{
if (!cashstat)
SortByGroup();
}
//-----
```

```
void __fastcall TForm3::N4Click(TObject *Sender)
{
if (!cashstat)
SortBySong();
}
//-----
```

```
void __fastcall TForm3::N5Click(TObject *Sender)
{
if (!cashstat)
MiscList();
}
//-----
```

```
void __fastcall TForm3::N6Click(TObject *Sender)
{
if (!cashstat)
DelDupe();
}
//-----
```

```
void __fastcall initLang()
{
if (Form6->ComboBox3->ItemIndex==0) // русский
{
Form6->Button1->Caption="Выбрать цвет";
Form6->Button2->Caption="Выбрать шрифт";
Form6->Button3->Caption="Сохранить";
Form6->Button4->Caption="Установить";
Form6->Button5->Caption="Установить";
Form6->Button6->Caption="Установить";
Form6->Button7->Caption="Выбрать изображение для кнопки";
}
}
```

```

Form6->Button8->Caption="Отменить изменение";
Form6->Button9->Caption="Сохранить как скин";
Form6->Button10->Caption="Загрузить скин";
Form6->Button11->Caption="Выйти";
Form6->Button12->Caption="Установить шкуру";
Form6->Label1->Caption="Альфа-компонента";
Form6->LabeledEdit1->EditLabel->Caption="Интервал для смены картинок, мс";
Form6->LabeledEdit2->EditLabel->Caption="Фоновая картинка";
Form6->LabeledEdit3->EditLabel->Caption="Шкурка по умолчанию";
Form6->LabeledEdit4->EditLabel->Caption="Ширина плейлиста";
Form6->LabeledEdit5->EditLabel->Caption="Высота плейлиста";
Form6->LabeledEdit6->EditLabel->Caption="Ширина картинки";
Form6->LabeledEdit7->EditLabel->Caption="Высота картинки";
Form6->CheckBox2->Caption="Множить фон";
Form6->CheckBox1->Caption="по ID3V1.1\\Filename";
Form6->CheckBox3->Caption="Умная рандомизация";
Form6->ComboBox1->Items->Strings[0]="Индикаторы времени";
Form6->ComboBox1->Items->Strings[1]="Плейлист";
Form6->ComboBox1->Items->Strings[2]="Главное окно";
Form6->ComboBox1->Items->Strings[3]="Строка статуса";
Form6->ComboBox2->Items->Strings[0]="Назад";
Form6->ComboBox2->Items->Strings[1]="Вперед";
Form6->ComboBox2->Items->Strings[2]="Играть";
Form6->ComboBox2->Items->Strings[3]="Пауза";
Form6->ComboBox2->Items->Strings[4]="Добавить";
Form6->ComboBox2->Items->Strings[5]="Удалить";
Form6->ComboBox2->Items->Strings[6]="Сохранить плейлист";
Form6->ComboBox2->Items->Strings[7]="Настройки";
Form6->ComboBox2->Items->Strings[8]="Открыть картинки";
Form6->ComboBox2->Items->Strings[9]="Добавить картинку";
Form6->ComboBox2->Items->Strings[10]="Удалить картинку";
Form6->ComboBox2->Items->Strings[11]="Остановить\\запустить слайдшоу";
Form6->ComboBox2->Items->Strings[12]="Сортировать";
Form3->Image2->Hint="предыдущая песня";
Form3->Image8->Hint="Пауза";
Form3->Image9->Hint="Играть";
Form3->Image10->Hint="следующая песня";
Form3->Image4->Hint="Добавить \\ Пр. клик - опции";
Form3->Image5->Hint="Удалить \\ Пр. клик - опции";
Form3->Image6->Hint="Сохранить плейлист \\ Пр. клик - опции";
Form3->Image15->Hint="Сортировать \\ Пр. клик - опции";
Form3->Image7->Hint="Настройки";
Form3->Image11->Hint="Открыть\\Заккрыть просмотр картинок";
Form3->Image12->Hint="Добавить картинку \\ Пр. клик - опции";
Form3->Image13->Hint="Удалить картинку \\ Пр. клик - опции";
Form3->Image14->Hint="Остановить\\Запустить слайдшоу";
Form4->Panel1->Caption="Добавить";
Form4->Panel2->Caption="Отмена";
Form5->Panel1->Caption="Добавить";
Form5->Panel2->Caption="Отмена";

```

```

Form4->CheckBox1->Caption="Добавлять в подкаталогах";
Form3->N1->Caption="Удалить выделенное";
Form3->N2->Caption="Удалить всё";
Form3->N6->Caption="Удалить дубликаты";
Form3->MenuItem1->Caption="Добавить файл";
Form3->MenuItem2->Caption="Добавить каталог";
Form3->MenuItem3->Caption="Добавить файл";
Form3->MenuItem4->Caption="Добавить каталог";
Form3->MenuItem5->Caption="Сохранить по умолчанию";
Form3->MenuItem6->Caption="Сохранить как";
Form3->N3->Caption="Загрузить";
Form3->HTML1->Caption="Сохранить HTML";
Form3->MenuItem7->Caption="По имени файла";
Form3->MenuItem8->Caption="По алфавиту";
Form3->N4->Caption="По песне";
Form3->N5->Caption="Перемешать";
Form3->MenuItem9->Caption="Удалить";
Form3->MenuItem10->Caption="Удалить всё";
Form6->Button13->Caption="О программе";
Form6->CheckBox4->Caption="Вести статистику";
Form6->Button14->Caption="Статистика";
}
if (Form6->ComboBox3->ItemIndex==1)
{
Form6->Button1->Caption="Choose the colour";
Form6->Button2->Caption="Choose the font";
Form6->Button3->Caption="Save";
Form6->Button4->Caption="Set";
Form6->Button5->Caption="Set";
Form6->Button6->Caption="Set";
Form6->Button7->Caption="Choose the image for the button";
Form6->Button8->Caption="Cancell the changes";
Form6->Button9->Caption="Save as a skin";
Form6->Button10->Caption="Load the skin";
Form6->Button11->Caption="Exit";
Form6->Button12->Caption="Set the skin";
Form6->Label1->Caption="Alfa-component";
Form6->LabeledEdit1->EditLabel->Caption="Interval for the picture change, ms";
Form6->LabeledEdit2->EditLabel->Caption="Background picture";
Form6->LabeledEdit3->EditLabel->Caption="Default skin";
Form6->LabeledEdit4->EditLabel->Caption="Playlist width";
Form6->LabeledEdit5->EditLabel->Caption="Playlist height";
Form6->LabeledEdit6->EditLabel->Caption="Picture width";
Form6->LabeledEdit7->EditLabel->Caption="Picture height";
Form6->CheckBox2->Caption="Multiply the background";
Form6->CheckBox1->Caption="by ID3V1.1\\Filename";
Form6->CheckBox3->Caption="Smart randomisation";
Form6->ComboBox1->Items->Strings[0]="Time indicator";
Form6->ComboBox1->Items->Strings[1]="Playlist";
Form6->ComboBox1->Items->Strings[2]="Main window";

```

```

Form6->ComboBox1->Items->Strings[3]="Status line";
Form6->ComboBox2->Items->Strings[0]="Back";
Form6->ComboBox2->Items->Strings[1]="Forward";
Form6->ComboBox2->Items->Strings[2]="Play";
Form6->ComboBox2->Items->Strings[3]="Pause";
Form6->ComboBox2->Items->Strings[4]="Add";
Form6->ComboBox2->Items->Strings[5]="Delete";
Form6->ComboBox2->Items->Strings[6]="Save playlist";
Form6->ComboBox2->Items->Strings[7]="Options";
Form6->ComboBox2->Items->Strings[8]="Show Picture";
Form6->ComboBox2->Items->Strings[9]="Add a picture";
Form6->ComboBox2->Items->Strings[10]="Delete a picture";
Form6->ComboBox2->Items->Strings[11]="Stop\\launch the slideshow";
Form6->ComboBox2->Items->Strings[12]="Sort";
Form3->Image2->Hint="Previous song";
Form3->Image8->Hint="Pause";
Form3->Image9->Hint="Play";
Form3->Image10->Hint="Next song";
Form3->Image4->Hint="Add \\ right click - options";
Form3->Image5->Hint="Delete \\ right click - options";
Form3->Image6->Hint="Save playlist \\ right click - options";
Form3->Image15->Hint="Sort \\ right click - options";
Form3->Image7->Hint="Options";
Form3->Image11->Hint="Open\\Close picture viewing";
Form3->Image12->Hint="Add a picture \\ right click - options";
Form3->Image13->Hint="Delete a picture \\ right click - options";
Form3->Image14->Hint="Stop\\launch the slideshow";
Form4->Panel1->Caption="Add";
Form4->Panel2->Caption="Cancel";
Form5->Panel1->Caption="Add";
Form5->Panel2->Caption="Cancel";
Form4->CheckBox1->Caption="Add in the subcatalogues";
Form3->N1->Caption="Delete the marked";
Form3->N2->Caption="Delete all";
Form3->N6->Caption="Delete the duplicates";
Form3->MenuItem1->Caption="Add a file";
Form3->MenuItem2->Caption="Add a catalogue";
Form3->MenuItem3->Caption="Add a file";
Form3->MenuItem4->Caption="Add a catalogue";
Form3->MenuItem5->Caption="Save as default";
Form3->MenuItem6->Caption="Save as";
Form3->N3->Caption="Load";
Form3->HTML1->Caption="Save HTML";
Form3->MenuItem7->Caption="By the filename";
Form3->MenuItem8->Caption="By the alphabet";
Form3->N4->Caption="By the song";
Form3->N5->Caption="Shuffle";
Form3->MenuItem9->Caption="Delete";
Form3->MenuItem10->Caption="Delete all";
Form6->Button13->Caption="About Programm";

```

```

Form6->CheckBox4->Caption="Statistic on\off";
Form6->Button14->Caption="Statistic";
}
if (Form6->ComboBox3->ItemIndex==2)
{
Form6->Button1->Caption="Wahle Farbe";
Form6->Button2->Caption="Wahle Schrift";
Form6->Button3->Caption="Erhalten";
Form6->Button4->Caption="Wahl";
Form6->Button5->Caption="Wahl";
Form6->Button6->Caption="Wahl";
Form6->Button7->Caption="Erhalten the image fur the button";
Form6->Button8->Caption="Abschaffung";
Form6->Button9->Caption="Erhalten Fell";
Form6->Button10->Caption="Wahl Fell";
Form6->Button11->Caption="Exit";
Form6->Button12->Caption="Wahl als Fell";
Form6->Label1->Caption="Alfa-component";
Form6->LabeledEdit1->EditLabel->Caption="Intervale fur Abbildung, ms";
Form6->LabeledEdit2->EditLabel->Caption="Hinterground Abbildung";
Form6->LabeledEdit3->EditLabel->Caption="Default Fell";
Form6->LabeledEdit4->EditLabel->Caption="Abhorenlist Breite";
Form6->LabeledEdit5->EditLabel->Caption="Abhorenlist Hohe";
Form6->LabeledEdit6->EditLabel->Caption="Abbildung Breite";
Form6->LabeledEdit7->EditLabel->Caption="Abbildung Hohe";
Form6->CheckBox2->Caption="Vervielfultigen Hintergrund";
Form6->CheckBox1->Caption="Nach ID3V1.1\\Filename";
Form6->CheckBox3->Caption="Kluge Zufall";
Form6->ComboBox1->Items->Strings[0]="Zeiteindikator";
Form6->ComboBox1->Items->Strings[1]="Abhorenlist";
Form6->ComboBox1->Items->Strings[2]="Wichtigste Fenster";
Form6->ComboBox1->Items->Strings[3]="Zeilezustand";
Form6->ComboBox2->Items->Strings[0]="Zuruck";
Form6->ComboBox2->Items->Strings[1]="Voran";
Form6->ComboBox2->Items->Strings[2]="Spiele";
Form6->ComboBox2->Items->Strings[3]="Pause";
Form6->ComboBox2->Items->Strings[4]="Hinzufugen";
Form6->ComboBox2->Items->Strings[5]="Fortschaffen";
Form6->ComboBox2->Items->Strings[6]="Erhalten Abhorenlist";
Form6->ComboBox2->Items->Strings[7]="Stimmen";
Form6->ComboBox2->Items->Strings[9]="Hinzufugen Abbildung";
Form6->ComboBox2->Items->Strings[8]="Hinzusetzen Abbildung";
Form6->ComboBox2->Items->Strings[10]="Entfernen Abbildung";
Form6->ComboBox2->Items->Strings[11]="Anhalten\\Starten Slideshow";
Form6->ComboBox2->Items->Strings[12]="Sortieren";
Form3->Image2->Hint="Zuruck";
Form3->Image8->Hint="Pause";
Form3->Image9->Hint="Spiele";
Form3->Image10->Hint="Voraus";
Form3->Image4->Hint="Hinzusetzen";

```

```

Form3->Image5->Hint="Entfernen";
Form3->Image6->Hint="Erhalten";
Form3->Image15->Hint="Sortieren";
Form3->Image7->Hint="Stimmen";
Form3->Image11->Hint="Abbildung offenen\\ zumachen";
Form3->Image12->Hint="Hinzusetzen Abbildung";
Form3->Image13->Hint="Entfernen Abbildung";
Form3->Image14->Hint="Stop\\Starten Slideshow";
Form4->Panel1->Caption="Hinzusetzen";
Form4->Panel2->Caption="Aufhebung";
Form5->Panel1->Caption="Hinzusetzen";
Form5->Panel2->Caption="Aufhebung";
Form4->CheckBox1->Caption="Hinzusetzen im Subcataloge";
Form3->N1->Caption="Entfernen";
Form3->N2->Caption="Entfernen Alles";
Form3->N6->Caption="Entfernen Duplicaten";
Form3->MenuItem1->Caption="Hinzusetzen File";
Form3->MenuItem2->Caption="Hinzusetzen Cataloge";
Form3->MenuItem3->Caption="Hinzusetzen File";
Form3->MenuItem4->Caption="Hinzusetzen Cataloge";
Form3->MenuItem5->Caption="Erhalten Verschweigen";
Form3->MenuItem6->Caption="Erhalten als";
Form3->N3->Caption="Beladen";
Form3->HTML1->Caption="Erhalten HTML";
Form3->MenuItem7->Caption="Als File";
Form3->MenuItem8->Caption="Als ABC";
Form3->N4->Caption="Als Singe";
Form3->N5->Caption="Verwirren";
Form3->MenuItem9->Caption="Entfernen";
Form3->MenuItem10->Caption="Entfernen Alles";
Form6->Button13->Caption="Uber Programme";
Form6->CheckBox4->Caption="Statistik";
Form6->Button14->Caption="Statistik";
}
}

```

```
void __fastcall TForm3::MenuItem10Click(TObject *Sender)
```

```

{
Memo1->Clear();
reloadstat2=0;
Memo1->Lines->SaveToFile("db\\pic.db");
TrackBar1->Max=getlenght("db\\pic.db");
}

```

```
//-----
```

```
void __fastcall TForm3::MenuItem9Click(TObject *Sender)
```

```

{
Memo1->Lines->Delete(TrackBar1->Position-1);
Memo1->Lines->SaveToFile("db\\pic.db");
TrackBar1->Max=getlenght("db\\pic.db");
}

```

```

}

void __fastcall nullRemoval()
{
for (int i=0;i<Form3->ListBox2->Items->Count;i++)
{
AnsiString TSTR=Form3->ListBox2->Items->Strings[i];
int errs=0;
for (int j=1;j<TSTR.Length();j++)
{
if (j+1!=TSTR.Length())
{
if (TSTR[j]==' ' && TSTR[j+1]==' ')
{
errs++;
int h=j;
for (h=j;h<TSTR.Length();h++)
{
if(h+1!=TSTR.Length())
TSTR[h]=TSTR[h+1];
}
j--;
}
}
}
if (errs!=0)
TSTR[TSTR.Length()-errs]='\0';
else TSTR[TSTR.Length()-errs]='\0';
Form3->ListBox2->Items->Strings[i]=TSTR;
}
}

//-----

```

```

void __fastcall TForm3::Timer4Timer(TObject *Sender)
{
if (itemscount<ListBox1->Items->Count)
{
MediaPlayer2->FileName=ListBox1->Items->Strings[itemscount];
if (MediaPlayer2->FileName!=MediaPlayer1->FileName)
{
MediaPlayer2->Open();
timetoplay+=MediaPlayer2->Length;
int retvalue=0,retvalue2=0; retvalue=Floor((((StrToFloat(Form3->MediaPlayer2-
>Length))/1000)/60)-(StrToInt(Form3->MediaPlayer2->Length)/1000)/60)*60);
AnsiString TIME0;
if (retvalue>9)
TIME0=FloatToStr((StrToInt(Form3->MediaPlayer2-
>Length)/1000)/60)+":"+FloatToStr(retvalue);
}
}

```

```

else
TIME0=FloatToStr((StrToInt(Form3->MediaPlayer2-
>Length)/1000)/60)+":0"+FloatToStr(retval);
Form3->ListBox2->Items->Strings[itemscount]=Form3->ListBox2->Items->Strings[itemscount]+"
\"+TIME0+"\" ";
MediaPlayer2->Close();
//nullRemoval();
AnsiString TSTR=ListBox2->Items->Strings[itemscount];
int errs=0;
for (int j=1;j<TSTR.Length();j++)
{
    if (j+1<TSTR.Length())
    {
        if (TSTR[j]==' ' && TSTR[j+1]==' ')
        {
            errs++;
            int h=j;
            for (h=j;h<TSTR.Length();h++)
            {
                if(h+1<TSTR.Length())
                TSTR[h]=TSTR[h+1];
            }
            j--;
        }
    } //
} // for j
if (errs!=0)
TSTR[TSTR.Length()-errs]="\0";
else TSTR[TSTR.Length()-errs]="\0";
ListBox2->Items->Strings[itemscount]=TSTR;
//
itemscount++;
}
else
{
    timetoplay+=MediaPlayer1->Length;
    int         retval=0,retval2=0;         retval=Floor((((StrToFloat(Form3->MediaPlayer1-
>Length))/1000)/60)-(StrToInt(Form3->MediaPlayer1->Length)/1000)/60)*60);
    AnsiString TIME0;
    if (retval>9)
    TIME0=FloatToStr((StrToInt(Form3->MediaPlayer1-
>Length)/1000)/60)+":"+FloatToStr(retval);
    else
    TIME0=FloatToStr((StrToInt(Form3->MediaPlayer1-
>Length)/1000)/60)+":0"+FloatToStr(retval);
    Form3->ListBox2->Items->Strings[itemscount]=Form3->ListBox2->Items->Strings[itemscount]+"
\"+TIME0+"\" ";
    //nullRemoval();
    AnsiString TSTR=ListBox2->Items->Strings[itemscount];
    int errs=0;

```



```

for (int j=1;j<TSTR.Length();j++)
{
    if (j<TSTR.Length())
    {
        if (TSTR[j]==' ' && TSTR[j+1]==' ')
        {
            errs++;
            int h=j;
            for (h=j;h<TSTR.Length();h++)
            {
                if(h+1<TSTR.Length())
                TSTR[h]=TSTR[h+1];
            }
            j--;
        }
    }
    // for j
    if (errs!=0)
    TSTR[TSTR.Length()-errs]='\0';
    else TSTR[TSTR.Length()-errs]='\0';
    ListBox2->Items->Strings[itemscount]=TSTR;
    //
    itemscount++;
}
}
else { itemscount=0; Timer4->Enabled=false; /* nullRemoval(); */ cashstat=false; }
}
//-----

```

```

void __fastcall TForm3::Button1Click(TObject *Sender)
{
    Form8->Show();
}
//-----

```

Окно-диалог добавления каталога песен

Unit4.h

```

//-----

#ifdef Unit4H
#define Unit4H
//-----

```

```
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <FileCtrl.hpp>
#include <ExtCtrls.hpp>
//-----
class TForm4 : public TForm
{
__published: // IDE-managed Components
    TDirectoryListBox *DirectoryListBox1;
    TDriveComboBox *DriveComboBox1;
    TPanel *Panel1;
    TPanel *Panel2;
    TFileListBox *FileListBox1;
    TLabel *Label1;
    TCheckBox *CheckBox1;
    TLabel *Label2;
    void __fastcall Panel2Click(TObject *Sender);
    void __fastcall Panel1Click(TObject *Sender);
    void __fastcall FormShow(TObject *Sender);
    void __fastcall FormCreate(TObject *Sender);
private: // User declarations
public: // User declarations
    __fastcall TForm4(TComponent* Owner);
};
//-----
extern PACKAGE TForm4 *Form4;
//-----
#endif
```

Unit4.cpp

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

#include "Unit4.h"
#include "Unit5.h"
#include "Unit3.h"
#include "Unit6.h"
#include "library.h"
#include <Math.hpp>
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
```

```

struct TAG // объявляем структуру ID3 тега ver. 1.1
{
unsigned char  tag   [ 03 ]; // сигнатура 'TAG' 3 байта
unsigned char  title [ 30 ]; // название 30 байт
unsigned char  artist [ 30 ]; // исполнитель 30 байт
unsigned char  album  [ 30 ]; // альбом 30 байт
unsigned char  year   [ 04 ]; // год 4 байта
unsigned char  comment [ 28 ]; // комментарий 28 байт
unsigned char  sep    [ 01 ]; // нулевой байт
unsigned char  track  [ 01 ]; // номер трека 1 байт
unsigned char  genre  [ 01 ]; // код стиля 1 байт
};

// Функция для чтения ID3 тегов ( параметр - имя файла )
AnsiString __fastcall GetMp3Tag ( AnsiString fName , int callparam )
{
    HANDLE      hFile;
    DWORD       dwFileSize;
    DWORD       dwBytesRead;
    DWORD       dwTagSize;
    DWORD       dwStartPos;
    TAG         tMp3;
    char        c[31];

    hFile = CreateFile ( fName.c_str (), GENERIC_READ, 0, NULL, OPEN_EXISTING,
        FILE_ATTRIBUTE_NORMAL, NULL );
    if ( hFile != INVALID_HANDLE_VALUE )
    {
        dwFileSize = GetFileSize ( hFile, NULL );
        dwTagSize = sizeof ( TAG );
        dwStartPos = dwFileSize - dwTagSize;
        SetFilePointer ( hFile, dwStartPos, NULL, FILE_BEGIN );
        ReadFile ( hFile, &tMp3, dwTagSize, &dwBytesRead, NULL );
    }
    CloseHandle ( hFile );
    memmove( c, tMp3.title, 30 ); c[30] = 0;
    //      MessageBox( NULL, c, "Title", NULL );
    if (callparam==1)
        return c;
    memmove( c, tMp3.artist, 30 ); c[30] = 0;
    //      MessageBox( NULL, c, "Artist", NULL );
    if (callparam==2)
        return c;
    memmove( c, tMp3.album, 30 ); c[30] = 0;
    //      MessageBox( NULL, c, "Album", NULL );
    if (callparam==3)
        return c;
    memmove( c, tMp3.year, 4 ); c[4] = 0;
    //      MessageBox( NULL, c, "Year", NULL );

```

```

if (callparam==4)
return c;
memmove( c, tMp3.comment, 28 ); c[28] = 0;
// MessageBox( NULL, c, "Comment", NULL );
if (callparam==5)
return c;
}

```

```

TForm4 *Form4;
//-----
__fastcall TForm4::TForm4(TComponent* Owner)
: TForm(Owner)
{

}
//-----

```

```

void __fastcall TForm4::Panel2Click(TObject *Sender)
{
DirectoryListBox1->Directory=Label1->Caption;
Form3->Timer1->Enabled=true;
Form3->Timer2->Enabled=true;
Form4->Close();
}
//-----
void __fastcall TForm4::Panel1Click(TObject *Sender)
{
try
{
Label2->Caption=DirectoryListBox1->Directory;
if (!CheckBox1->Checked)
{
for (int i=0; i<FileListBox1->Items->Count; i++)
{
FileListBox1->ItemIndex=i;
Form3->ListBox1->Items->Add(FileListBox1->FileName);
}
}
else {
Form4->Visible=false;
int
numsubs,curnum=0,numsubs2=0,curnum2=0,curnum3=0,curnum4=0,curnum5=0,curnum6=0,curnum7,curnum8,curnum9,numsubs3=0,numsubs4,numsubs5,numsubs6,numsubs7,numsubs8,numsubs9;
AnsiString topdir=DirectoryListBox1->Directory, topdir2, topdir3, topdir4, topdir5, topdir6, topdir7, topdir8, topdir9;
numsubs=DirectoryListBox1->Items->Count;
curnum=DirectoryListBox1->ItemIndex+1;
for (int i=0; i<FileListBox1->Items->Count; i++)

```

```

{
FileListBox1->ItemIndex=i;
Form3->ListBox1->Items->Add(FileListBox1->FileName);
}
while (curnum!=numsubs)
{
DirectoryListBox1->ItemIndex=curnum;
DirectoryListBox1->Directory=DirectoryListBox1->Items->Strings[DirectoryListBox1->ItemIndex];
if (curnum==numsubs)
{
for (int i=0; i<FileListBox1->Items->Count; i++)
{
FileListBox1->ItemIndex=i;
Form3->ListBox1->Items->Add(FileListBox1->FileName);
}
curnum++;
DirectoryListBox1->Directory=topdir;
}
else
{
for (int i=0; i<FileListBox1->Items->Count; i++)
{
FileListBox1->ItemIndex=i;
Form3->ListBox1->Items->Add(FileListBox1->FileName);
}
numsubs2=DirectoryListBox1->Items->Count;
curnum2=DirectoryListBox1->ItemIndex+1;
topdir2=DirectoryListBox1->Directory;
while (curnum2!=numsubs2)
{
DirectoryListBox1->ItemIndex=curnum2;
DirectoryListBox1->Directory=DirectoryListBox1->Items->Strings[DirectoryListBox1->ItemIndex];
if (curnum2==numsubs2)
{
for (int i=0; i<FileListBox1->Items->Count; i++)
{
FileListBox1->ItemIndex=i;
Form3->ListBox1->Items->Add(FileListBox1->FileName);
}
curnum2++;
DirectoryListBox1->Directory=topdir;
}
}
else
{
for (int i=0; i<FileListBox1->Items->Count; i++)
{
FileListBox1->ItemIndex=i;
Form3->ListBox1->Items->Add(FileListBox1->FileName);
}
}
}

```

```

}
numsubs3=DirectoryListBox1->Items->Count;
curnum3=DirectoryListBox1->ItemIndex+1;
topdir3=DirectoryListBox1->Directory;
while (curnum3!=numsubs3)
{
    DirectoryListBox1->ItemIndex=curnum3;
    DirectoryListBox1->Directory=DirectoryListBox1->Items->Strings[DirectoryListBox1->ItemIndex];
    if (curnum3==numsubs3)
    {
        for (int i=0; i<FileListBox1->Items->Count; i++)
        {
            FileListBox1->ItemIndex=i;
            Form3->ListBox1->Items->Add(FileListBox1->FileName);
        }
        curnum3++;
        DirectoryListBox1->Directory=topdir;
    }
else
{
    for (int i=0; i<FileListBox1->Items->Count; i++)
    {
        FileListBox1->ItemIndex=i;
        Form3->ListBox1->Items->Add(FileListBox1->FileName);
    }
    numsubs4=DirectoryListBox1->Items->Count;
    curnum4=DirectoryListBox1->ItemIndex+1;
    topdir4=DirectoryListBox1->Directory;
    while (curnum4!=numsubs4)
    {
        DirectoryListBox1->ItemIndex=curnum4;
        DirectoryListBox1->Directory=DirectoryListBox1->Items->Strings[DirectoryListBox1->ItemIndex];
        if (curnum4==numsubs4)
        {
            for (int i=0; i<FileListBox1->Items->Count; i++)
            {
                FileListBox1->ItemIndex=i;
                Form3->ListBox1->Items->Add(FileListBox1->FileName);
            }
            curnum4++;
            DirectoryListBox1->Directory=topdir;
        }
    }
    //если есть подкаталоги
else
{
    for (int i=0; i<FileListBox1->Items->Count; i++)
    {
        FileListBox1->ItemIndex=i;
    }
}

```

```

Form3->ListBox1->Items->Add(FileListBox1->FileName);
}
numsubs5=DirectoryListBox1->Items->Count;
curnum5=DirectoryListBox1->ItemIndex+1;
topdir5=DirectoryListBox1->Directory;
// ShowMessage("Есть подкаталог!" + DirectoryListBox1->Directory);
while (curnum5!=numsubs5)
{
    DirectoryListBox1->ItemIndex=curnum5;
    DirectoryListBox1->Directory=DirectoryListBox1->Items->Strings[DirectoryListBox1-
>ItemIndex];
    if (curnum5==numsubs5)
    {
        for (int i=0; i<FileListBox1->Items->Count; i++)
        {
            FileListBox1->ItemIndex=i;
            Form3->ListBox1->Items->Add(FileListBox1->FileName);
        }
        curnum5++;
        DirectoryListBox1->Directory=topdir;
    }
    else
    {
        for (int i=0; i<FileListBox1->Items->Count; i++)
        {
            FileListBox1->ItemIndex=i;
            Form3->ListBox1->Items->Add(FileListBox1->FileName);
        }
        numsubs6=DirectoryListBox1->Items->Count;
        curnum6=DirectoryListBox1->ItemIndex+1;
        topdir6=DirectoryListBox1->Directory;
        while (curnum6!=numsubs6)
        {
            DirectoryListBox1->ItemIndex=curnum6;
            DirectoryListBox1->Directory=DirectoryListBox1->Items->Strings[DirectoryListBox1-
>ItemIndex];
            if (curnum6==numsubs6)
            {
                for (int i=0; i<FileListBox1->Items->Count; i++)
                {
                    FileListBox1->ItemIndex=i;
                    Form3->ListBox1->Items->Add(FileListBox1->FileName);
                }
                curnum6++;
                DirectoryListBox1->Directory=topdir;
            }
        }
    }
    else
    {
        for (int i=0; i<FileListBox1->Items->Count; i++)
        {

```

```

FileListBox1->ItemIndex=i;
Form3->ListBox1->Items->Add(FileListBox1->FileName);
if (!Form6->CheckBox1->Checked)
    Form3->ListBox2->Items->Add(AnsiString(crofilename(FileListBox1->FileName)));
else
{
    AnsiString TTagMP3; try { TTagMP3=GetMp3Tag(FileListBox1->FileName,0); } catch (...)
{}
    if ((TTagMP3[1]=='T') && (TTagMP3[2]=='A') && (TTagMP3[3]=='G'))
        Form3->ListBox2->Items->Add(AnsiString(GetMp3Tag(FileListBox1->FileName,2))+
"+AnsiString(GetMp3Tag(FileListBox1->FileName,1)));
    else Form3->ListBox2->Items->Add(AnsiString(crofilename(FileListBox1->FileName)));
}
}
numsubs7=DirectoryListBox1->Items->Count;
curnum7=DirectoryListBox1->ItemIndex+1;
topdir7=DirectoryListBox1->Directory;
while (curnum7!=numsubs7)
{
    DirectoryListBox1->ItemIndex=curnum7;
    DirectoryListBox1->Directory=DirectoryListBox1->Items->Strings[DirectoryListBox1-
>ItemIndex];
    if (curnum7==numsubs7)
    {
        for (int i=0; i<FileListBox1->Items->Count; i++)
        {
            FileListBox1->ItemIndex=i;
            Form3->ListBox1->Items->Add(FileListBox1->FileName);
            if (!Form6->CheckBox1->Checked)
                Form3->ListBox2->Items->Add(AnsiString(crofilename(FileListBox1->FileName)));
            else
            {
                AnsiString TTagMP3; try { TTagMP3=GetMp3Tag(FileListBox1->FileName,0); } catch (...)
                {}
                if ((TTagMP3[1]=='T') && (TTagMP3[2]=='A') && (TTagMP3[3]=='G'))
                    Form3->ListBox2->Items->Add(AnsiString(GetMp3Tag(FileListBox1->FileName,2))+
"+AnsiString(GetMp3Tag(FileListBox1->FileName,1)));
                else Form3->ListBox2->Items->Add(AnsiString(crofilename(FileListBox1->FileName)));
            }
        }
        curnum7++;
        DirectoryListBox1->Directory=topdir;
    }
}
else
{
    for (int i=0; i<FileListBox1->Items->Count; i++)
    {
        FileListBox1->ItemIndex=i;
        Form3->ListBox1->Items->Add(FileListBox1->FileName);
    }
}

```



```

numsubs8=DirectoryListBox1->Items->Count;
curnum8=DirectoryListBox1->ItemIndex+1;
topdir8=DirectoryListBox1->Directory;
while (curnum8!=numsubs8)
{
    DirectoryListBox1->ItemIndex=curnum8;
    DirectoryListBox1->Directory=DirectoryListBox1->Items->Strings[DirectoryListBox1-
>ItemIndex];
    if (curnum8==numsubs8)
    {
        for (int i=0; i<FileListBox1->Items->Count; i++)
        {
            FileListBox1->ItemIndex=i;
            Form3->ListBox1->Items->Add(FileListBox1->FileName);
        }
        curnum8++;
        DirectoryListBox1->Directory=topdir;
    }
else
{
    for (int i=0; i<FileListBox1->Items->Count; i++)
    {
        FileListBox1->ItemIndex=i;
        Form3->ListBox1->Items->Add(FileListBox1->FileName);
    }
    numsubs9=DirectoryListBox1->Items->Count;
    curnum9=DirectoryListBox1->ItemIndex+1;
    topdir9=DirectoryListBox1->Directory;
    while (curnum9!=numsubs9)
    {
        DirectoryListBox1->ItemIndex=curnum9;
        DirectoryListBox1->Directory=DirectoryListBox1->Items->Strings[DirectoryListBox1-
>ItemIndex];
        for (int i=0; i<FileListBox1->Items->Count; i++)
        {
            FileListBox1->ItemIndex=i;
            Form3->ListBox1->Items->Add(FileListBox1->FileName);
        }
        curnum9++;
        DirectoryListBox1->Directory=topdir9;
    }
    curnum8++;
    DirectoryListBox1->Directory=topdir8;
}
curnum7++;
DirectoryListBox1->Directory=topdir7;
}
}

```

```

curnum6++;
DirectoryListBox1->Directory=topdir6;
}
}
curnum5++;
DirectoryListBox1->Directory=topdir5;
}
}
curnum4++;
DirectoryListBox1->Directory=topdir4;
}
}

curnum3++;
DirectoryListBox1->Directory=topdir3;
}
}
curnum2++;
DirectoryListBox1->Directory=topdir2;
}
curnum++;
DirectoryListBox1->Directory=topdir;
}
}
}
}
catch (...) {}
DirectoryListBox1->Directory=Label1->Caption;
Form3->timetoplay=0;
AnsiString TR;
Form3->ListBox2->SelectAll();
Form3->ListBox2->DeleteSelected();
AnsiString TTagMP3;
int cursong=-1;
for (int i=0; i<Form3->ListBox1->Items->Count;i++)
{
if (!Form6->CheckBox1->Checked)
Form3->ListBox2->Items->Add(AnsiString(crofilename(Form3->ListBox1->Items->Strings[i]))));
else
{
if (Form3->ListBox1->Items->Strings[i]!=Form3->MediaPlayer1->FileName)
{
TTagMP3=GetMp3Tag(Form3->ListBox1->Items->Strings[i],0);
AnsiString TT1,TT2,TT3;
int error=0;
try { TT1=TTagMP3[1]; } catch (...) {error++;}
try { TT2=TTagMP3[2]; } catch (...) {error++;}
try { TT3=TTagMP3[3]; } catch (...) {error++;}
if (!error)
{

```

```

if ((TT1=="T") && (TT2=="A") && (TT3=="G"))
    Form3->ListBox2->Items->Add(AnsiString(GetMp3Tag(Form3->ListBox1->Items-
>Strings[i],2))+ " - "+AnsiString(GetMp3Tag(Form3->ListBox1->Items->Strings[i],1)));
    else
        Form3->ListBox2->Items->Add(AnsiString(crofilename(Form3->ListBox1->Items-
>Strings[i])));
    }
    else
        Form3->ListBox2->Items->Add(AnsiString(crofilename(Form3->ListBox1->Items-
>Strings[i])));
    }
    else
    {
        cursong=i;
        Form3->ListBox2->Items->Add(AnsiString(crofilename(Form3->ListBox1->Items-
>Strings[i])));
    }
}
if (cursong>=0)
{
if (!Form6->CheckBox1->Checked)
    Form3->ListBox2->Items->Add(AnsiString(crofilename(Form3->ListBox1->Items-
>Strings[cursong])));
else
{
    Form3->MediaPlayer1->Stop();
    Form3->MediaPlayer1->Close();
    TTagMP3=GetMp3Tag(Form3->ListBox1->Items->Strings[cursong],0);
    AnsiString TT1,TT2,TT3;
    int error=0;
    try { TT1=TTagMP3[1]; } catch (...) {error++;}
    try { TT2=TTagMP3[2]; } catch (...) {error++;}
    try { TT3=TTagMP3[3]; } catch (...) {error++;}
    if (!error)
    {
        if ((TT1=="T") && (TT2=="A") && (TT3=="G"))
            Form3->ListBox2->Items->Strings[cursong]=(AnsiString(GetMp3Tag(Form3->ListBox1-
>Items->Strings[cursong],2))+ " - " +AnsiString(GetMp3Tag(Form3->ListBox1->Items-
>Strings[cursong],1)));
        else
            Form3->ListBox2->Items->Strings[cursong]=(AnsiString(crofilename(Form3->ListBox1-
>Items->Strings[cursong])));
    }
    else
        Form3->ListBox2->Items->Strings[cursong]=(AnsiString(crofilename(Form3->ListBox1-
>Items->Strings[cursong])));
    }
    Form3->MediaPlayer1->Open();
    Form3->MediaPlayer1->Position=Form3->TrackBar2->Position;
    Form3->MediaPlayer1->Play();
}
/*for (int i=0; i<Form3->ListBox1->Items->Count; i++)
{

```

```

if (Form3->ListBox1->Items->Strings[i]!=Form3->MediaPlayer1->FileName)
{
try
{
Form3->MediaPlayer2->FileName=Form3->ListBox1->Items->Strings[i];
try {
Form3->MediaPlayer2->Open();
}
catch ( EMCIDeviceError& ) { /}
Form3->timetoplay+=Form3->MediaPlayer2->Length;
int      retvalue=0,retvalue2=0;      retvalue=Floor((((StrToFloat(Form3->MediaPlayer2-
>Length))/1000)/60)-(StrToInt(Form3->MediaPlayer2->Length)/1000)/60)*60);
AnsiString TIME0;
if (retvalue>9)
TIME0=FloatToStr((StrToInt(Form3->MediaPlayer2-
>Length)/1000)/60)+":"+FloatToStr(retvalue);
else
TIME0=FloatToStr((StrToInt(Form3->MediaPlayer2-
>Length)/1000)/60)+":0"+FloatToStr(retvalue);
Form3->ListBox2->Items->Strings[i]=Form3->ListBox2->Items->Strings[i]+" \("+TIME0+"\) ";
Form3->MediaPlayer2->Close();
}
catch ( EMCIDeviceError& ) {}
}
else
{
Form3->timetoplay+=Form3->MediaPlayer1->Length;
int      retvalue=0,retvalue2=0;      retvalue=Floor((((StrToFloat(Form3->MediaPlayer1-
>Length))/1000)/60)-(StrToInt(Form3->MediaPlayer1->Length)/1000)/60)*60);
AnsiString TIME0;
if (retvalue>9)
TIME0=FloatToStr((StrToInt(Form3->MediaPlayer1-
>Length)/1000)/60)+":"+FloatToStr(retvalue);
else
TIME0=FloatToStr((StrToInt(Form3->MediaPlayer1-
>Length)/1000)/60)+":0"+FloatToStr(retvalue);
Form3->ListBox2->Items->Strings[i]=Form3->ListBox2->Items->Strings[i]+" \("+TIME0+"\) ";
}
} */
Form3->cashstat=true;
Form3->itemscount=0;
Form3->Timer4->Enabled=true; //!!!!
//Form3->MediaPlayer2->Stop();
//Form3->MediaPlayer2->Close();
Form3->MediaPlayer2->FileName=NULL;
Form3->Timer1->Enabled=true;
Form3->Timer2->Enabled=true;
Form3->TrackBar2->Enabled=false;
Form3->Timer3->Enabled=true;
Form4->Close();

```

```

}

//
//-----

//-----

void __fastcall TForm4::FormShow(TObject *Sender)
{
Label1->Caption=DirectoryListBox1->Directory;
DirectoryListBox1->Directory=Label2->Caption;
}
//-----

void __fastcall TForm4::FormCreate(TObject *Sender)
{
Label2->Caption=DirectoryListBox1->Directory;
}
//-----

```

Окно-диалог добавления каталога изображений

Unit5.h

```

//-----

#ifndef Unit5H
#define Unit5H
//-----
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <ExtCtrls.hpp>
#include <FileCtrl.hpp>
//-----
class TForm5 : public TForm
{
__published: // IDE-managed Components
    TLabel *Label1;
    TLabel *Label2;
    TDirectoryListBox *DirectoryListBox1;
    TDriveComboBox *DriveComboBox1;
    TPanel *Panel1;
    TPanel *Panel2;
    TFileListBox *FileListBox1;
    void __fastcall Panel2Click(TObject *Sender);

```

```

void __fastcall Panel1Click(TObject *Sender);
void __fastcall FormCreate(TObject *Sender);
void __fastcall FormShow(TObject *Sender);
private:      // User declarations
public:       // User declarations
    __fastcall TForm5(TComponent* Owner);
};
//-----
extern PACKAGE TForm5 *Form5;
//-----
#endif

```

Unit5.cpp

```

//-----

#include <vcl.h>
#pragma hdrstop

#include "Unit5.h"
#include "Unit3.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm5 *Form5;
//-----
__fastcall TForm5::TForm5(TComponent* Owner)
    : TForm(Owner)
{
}
//-----

void __fastcall TForm5::Panel2Click(TObject *Sender)
{
    DirectoryListBox1->Directory=Label1->Caption;
    Form3->Timer1->Enabled=true;
    Form3->Timer2->Enabled=true;
    Form5->Close();
}
//-----

void __fastcall TForm5::Panel1Click(TObject *Sender)
{
    Label2->Caption=DirectoryListBox1->Directory;
    for (int i=0; i<FileListBox1->Items->Count; i++)
    {
        FileListBox1->ItemIndex=i;
        Form3->Memo1->Lines->Add(FileListBox1->FileName);
        Form3->Memo1->Lines->SaveToFile(Label1->Caption+"\\db\\pic.db");
    }
}

```

```

}
DirectoryListBox1->Directory=Label1->Caption;
Form3->Timer1->Enabled=true;
Form3->Timer2->Enabled=true;
Form5->Close();
}
//-----
void __fastcall TForm5::FormCreate(TObject *Sender)
{
Label2->Caption=DirectoryListBox1->Directory;
}
//-----
void __fastcall TForm5::FormShow(TObject *Sender)
{
Label1->Caption=DirectoryListBox1->Directory;
DirectoryListBox1->Directory=Label2->Caption;
}
//-----

```

Окно настроек программы

Unit6.h

```

//-----
#ifndef Unit6H
#define Unit6H
//-----
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <Dialogs.hpp>
#include <DBCtrls.hpp>
#include <ExtCtrls.hpp>
#include <ExtDlgs.hpp>
//-----
class TForm6 : public TForm
{
__published: // IDE-managed Components
    TFontDialog *FontDialog1;
    TColorDialog *ColorDialog1;
    TButton *Button1;
    TMemo *Memo1;
    TButton *Button2;
    TButton *Button3;
    TEdit *Edit1;
    TLabel *Label1;

```

```

TButton *Button4;
TComboBox *ComboBox1;
TLabelEdit *LabelEdit1;
TButton *Button5;
TLabelEdit *LabelEdit2;
TButton *Button6;
TOpenPictureDialog *OpenPictureDialog1;
TImage *Image1;
TComboBox *ComboBox2;
TImage *Image2;
TButton *Button7;
TCheckBox *CheckBox2;
TListBox *ListBox1;
TButton *Button8;
TButton *Button9;
TButton *Button10;
TButton *Button11;
TSaveDialog *SaveDialog1;
TOpenDialog *OpenDialog1;
TOpenPictureDialog *OpenPictureDialog2;
TLabelEdit *LabelEdit3;
TCheckBox *CheckBox1;
TCheckBox *CheckBox3;
TButton *Button12;
TMemo *Memo2;
TLabelEdit *LabelEdit4;
TLabelEdit *LabelEdit5;
TLabelEdit *LabelEdit6;
TLabelEdit *LabelEdit7;
TComboBox *ComboBox3;
TButton *Button13;
TCheckBox *CheckBox4;
TButton *Button14;
void __fastcall Button1Click(TObject *Sender);
void __fastcall Button2Click(TObject *Sender);
void __fastcall Button3Click(TObject *Sender);
void __fastcall Button4Click(TObject *Sender);
void __fastcall Button5Click(TObject *Sender);
void __fastcall Button6Click(TObject *Sender);
void __fastcall Button7Click(TObject *Sender);
void __fastcall CheckBox2Enter(TObject *Sender);
void __fastcall CheckBox2Exit(TObject *Sender);
void __fastcall Button9Click(TObject *Sender);
void __fastcall Button11Click(TObject *Sender);
void __fastcall Button10Click(TObject *Sender);
void __fastcall Button8Click(TObject *Sender);
void __fastcall Button12Click(TObject *Sender);
void __fastcall LabelEdit4Change(TObject *Sender);
void __fastcall Button13Click(TObject *Sender);
void __fastcall ComboBox3Change(TObject *Sender);

```



```

    void __fastcall Button14Click(TObject *Sender);
private:    // User declarations
public:    // User declarations
    __fastcall TForm6(TComponent* Owner);
};
//-----
extern PACKAGE TForm6 *Form6;
//-----
#endif

```

Unit4.cpp

```

//-----

#include <vcl.h>
#pragma hdrstop

#include "Unit6.h"
#include "Unit4.h"
#include "Unit5.h"
#include "Unit3.h"
#include "Unit7.h"
#include "Unit8.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm6 *Form6;
//-----
__fastcall TForm6::TForm6(TComponent* Owner)
    : TForm(Owner)
{

}
//-----

void __fastcall initLang()
{
if (Form6->ComboBox3->ItemIndex==0) // русский
{
Form6->Button1->Caption="Выбрать цвет";
Form6->Button2->Caption="Выбрать шрифт";
Form6->Button3->Caption="Сохранить";
Form6->Button4->Caption="Установить";
Form6->Button5->Caption="Установить";
Form6->Button6->Caption="Установить";
Form6->Button7->Caption="Выбрать изображение для кнопки";
Form6->Button8->Caption="Отменить изменение";
Form6->Button9->Caption="Сохранить как скин";
}
}

```

```

Form6->Button10->Caption="Загрузить скин";
Form6->Button11->Caption="Выйти";
Form6->Button12->Caption="Установить шкуру";
Form6->Label1->Caption="Альфа-компонента";
Form6->LabeledEdit1->EditLabel->Caption="Интервал для смены картинок, мс";
Form6->LabeledEdit2->EditLabel->Caption="Фоновая картинка";
Form6->LabeledEdit3->EditLabel->Caption="Шкурка по умолчанию";
Form6->LabeledEdit4->EditLabel->Caption="Ширина плейлиста";
Form6->LabeledEdit5->EditLabel->Caption="Высота плейлиста";
Form6->LabeledEdit6->EditLabel->Caption="Ширина картинки";
Form6->LabeledEdit7->EditLabel->Caption="Высота картинки";
Form6->CheckBox2->Caption="Множить фон";
Form6->CheckBox1->Caption="по ID3V1.1\\Filename";
Form6->CheckBox3->Caption="Умная рандомизация";
Form6->ComboBox1->Items->Strings[0]="Индикаторы времени";
Form6->ComboBox1->Items->Strings[1]="Плейлист";
Form6->ComboBox1->Items->Strings[2]="Главное окно";
Form6->ComboBox1->Items->Strings[3]="Строка статуса";
Form6->ComboBox2->Items->Strings[0]="Назад";
Form6->ComboBox2->Items->Strings[1]="Вперед";
Form6->ComboBox2->Items->Strings[2]="Играть";
Form6->ComboBox2->Items->Strings[3]="Пауза";
Form6->ComboBox2->Items->Strings[4]="Добавить";
Form6->ComboBox2->Items->Strings[5]="Удалить";
Form6->ComboBox2->Items->Strings[6]="Сохранить плейлист";
Form6->ComboBox2->Items->Strings[7]="Настройки";
Form6->ComboBox2->Items->Strings[8]="Открыть картинки";
Form6->ComboBox2->Items->Strings[9]="Добавить картинку";
Form6->ComboBox2->Items->Strings[10]="Удалить картинку";
Form6->ComboBox2->Items->Strings[11]="Остановить\\запустить слайдшоу";
Form6->ComboBox2->Items->Strings[12]="Сортировать";
Form3->Image2->Hint="предыдущая песня";
Form3->Image8->Hint="Пауза";
Form3->Image9->Hint="Играть";
Form3->Image10->Hint="следующая песня";
Form3->Image4->Hint="Добавить \\ Пр. клик - опции";
Form3->Image5->Hint="Удалить \\ Пр. клик - опции";
Form3->Image6->Hint="Сохранить плейлист \\ Пр. клик - опции";
Form3->Image15->Hint="Сортировать \\ Пр. клик - опции";
Form3->Image7->Hint="Настройки";
Form3->Image11->Hint="Открыть\\Заккрыть просмотр картинок";
Form3->Image12->Hint="Добавить картинку \\ Пр. клик - опции";
Form3->Image13->Hint="Удалить картинку \\ Пр. клик - опции";
Form3->Image14->Hint="Остановить\\Запустить слайдшоу";
Form4->Panel1->Caption="Добавить";
Form4->Panel2->Caption="Отмена";
Form5->Panel1->Caption="Добавить";
Form5->Panel2->Caption="Отмена";
Form4->CheckBox1->Caption="Добавлять в подкаталогах";
Form3->N1->Caption="Удалить выделенное";

```

```

Form3->N2->Caption="Удалить всё";
Form3->N6->Caption="Удалить дубликаты";
Form3->MenuItem1->Caption="Добавить файл";
Form3->MenuItem2->Caption="Добавить каталог";
Form3->MenuItem3->Caption="Добавить файл";
Form3->MenuItem4->Caption="Добавить каталог";
Form3->MenuItem5->Caption="Сохранить по умолчанию";
Form3->MenuItem6->Caption="Сохранить как";
Form3->N3->Caption="Загрузить";
Form3->HTML1->Caption="Сохранить HTML";
Form3->MenuItem7->Caption="По имени файла";
Form3->MenuItem8->Caption="По алфавиту";
Form3->N4->Caption="По песне";
Form3->N5->Caption="Перемешать";
Form3->MenuItem9->Caption="Удалить";
Form3->MenuItem10->Caption="Удалить всё";
Form6->Button13->Caption="О программе";
Form6->CheckBox4->Caption="Вести статистику";
Form6->Button14->Caption="Статистика";
}
if (Form6->ComboBox3->ItemIndex==1)
{
Form6->Button1->Caption="Choose the colour";
Form6->Button2->Caption="Choose the font";
Form6->Button3->Caption="Save";
Form6->Button4->Caption="Set";
Form6->Button5->Caption="Set";
Form6->Button6->Caption="Set";
Form6->Button7->Caption="Choose the image for the button";
Form6->Button8->Caption="Cancell the changes";
Form6->Button9->Caption="Save as a skin";
Form6->Button10->Caption="Load the skin";
Form6->Button11->Caption="Exit";
Form6->Button12->Caption="Set the skin";
Form6->Label1->Caption="Alfa-component";
Form6->LabeledEdit1->EditLabel->Caption="Interval for the picture change, ms";
Form6->LabeledEdit2->EditLabel->Caption="Background picture";
Form6->LabeledEdit3->EditLabel->Caption="Default skin";
Form6->LabeledEdit4->EditLabel->Caption="Playlist width";
Form6->LabeledEdit5->EditLabel->Caption="Playlist height";
Form6->LabeledEdit6->EditLabel->Caption="Picture width";
Form6->LabeledEdit7->EditLabel->Caption="Picture height";
Form6->CheckBox2->Caption="Multiply the background";
Form6->CheckBox1->Caption="by ID3V1.1\\Filename";
Form6->CheckBox3->Caption="Smart randomisation";
Form6->ComboBox1->Items->Strings[0]="Time indicator";
Form6->ComboBox1->Items->Strings[1]="Playlist";
Form6->ComboBox1->Items->Strings[2]="Main window";
Form6->ComboBox1->Items->Strings[3]="Status line";
Form6->ComboBox2->Items->Strings[0]="Back";

```

```

Form6->ComboBox2->Items->Strings[1]="Forward";
Form6->ComboBox2->Items->Strings[2]="Play";
Form6->ComboBox2->Items->Strings[3]="Pause";
Form6->ComboBox2->Items->Strings[4]="Add";
Form6->ComboBox2->Items->Strings[5]="Delete";
Form6->ComboBox2->Items->Strings[6]="Save playlist";
Form6->ComboBox2->Items->Strings[7]="Options";
Form6->ComboBox2->Items->Strings[8]="Show Picture";
Form6->ComboBox2->Items->Strings[9]="Add a picture";
Form6->ComboBox2->Items->Strings[10]="Delete a picture";
Form6->ComboBox2->Items->Strings[11]="Stop\\launch the slideshow";
Form6->ComboBox2->Items->Strings[12]="Sort";
Form3->Image2->Hint="Previous song";
Form3->Image8->Hint="Pause";
Form3->Image9->Hint="Play";
Form3->Image10->Hint="Next song";
Form3->Image4->Hint="Add \\ right click - options";
Form3->Image5->Hint="Delete \\ right click - options";
Form3->Image6->Hint="Save playlist \\ right click - options";
Form3->Image15->Hint="Sort \\ right click - options";
Form3->Image7->Hint="Options";
Form3->Image11->Hint="Open\\Close picture viewing";
Form3->Image12->Hint="Add a picture \\ right click - options";
Form3->Image13->Hint="Delete a picture \\ right click - options";
Form3->Image14->Hint="Stop\\launch the slideshow";
Form4->Panel1->Caption="Add";
Form4->Panel2->Caption="Cancell";
Form5->Panel1->Caption="Add";
Form5->Panel2->Caption="Cancell";
Form4->CheckBox1->Caption="Add in the subcatalogues";
Form3->N1->Caption="Delete the marked";
Form3->N2->Caption="Delete all";
Form3->N6->Caption="Delete the duplicates";
Form3->MenuItem1->Caption="Add a file";
Form3->MenuItem2->Caption="Add a catalogue";
Form3->MenuItem3->Caption="Add a file";
Form3->MenuItem4->Caption="Add a catalogue";
Form3->MenuItem5->Caption="Save as default";
Form3->MenuItem6->Caption="Save as";
Form3->N3->Caption="Load";
Form3->HTML1->Caption="Save HTML";
Form3->MenuItem7->Caption="By the filename";
Form3->MenuItem8->Caption="By the alphabet";
Form3->N4->Caption="By the song";
Form3->N5->Caption="Shuffle";
Form3->MenuItem9->Caption="Delete";
Form3->MenuItem10->Caption="Delete all";
Form6->Button13->Caption="About Programm";
Form6->CheckBox4->Caption="Statistic on\\off";
Form6->Button14->Caption="Statistic";

```

```

}
if (Form6->ComboBox3->ItemIndex==2)
{
Form6->Button1->Caption="Wahle Farbe";
Form6->Button2->Caption="Wahle Schrift";
Form6->Button3->Caption="Erhalten";
Form6->Button4->Caption="Wahl";
Form6->Button5->Caption="Wahl";
Form6->Button6->Caption="Wahl";
Form6->Button7->Caption="Erhalten the image fur the button";
Form6->Button8->Caption="Abschaffung";
Form6->Button9->Caption="Erhalten Fell";
Form6->Button10->Caption="Wahl Fell";
Form6->Button11->Caption="Exit";
Form6->Button12->Caption="Wahl als Fell";
Form6->Label1->Caption="Alfa-component";
Form6->LabeledEdit1->EditLabel->Caption="Intervale fur Abbildung, ms";
Form6->LabeledEdit2->EditLabel->Caption="Hinterground Abbildung";
Form6->LabeledEdit3->EditLabel->Caption="Default Fell";
Form6->LabeledEdit4->EditLabel->Caption="Abhorenlis Breite";
Form6->LabeledEdit5->EditLabel->Caption="Abhorenlis Hohe";
Form6->LabeledEdit6->EditLabel->Caption="Abbildung Breite";
Form6->LabeledEdit7->EditLabel->Caption="Abbildung Hohe";
Form6->CheckBox2->Caption="Vervielfultigen Hintergrund";
Form6->CheckBox1->Caption="Nach ID3V1.1\\Filename";
Form6->CheckBox3->Caption="Kluge Zufall";
Form6->ComboBox1->Items->Strings[0]="Zeiteindikator";
Form6->ComboBox1->Items->Strings[1]="Abhorenlis";
Form6->ComboBox1->Items->Strings[2]="Wichtigste Fenster";
Form6->ComboBox1->Items->Strings[3]="Zeilezustand";
Form6->ComboBox2->Items->Strings[0]="Zuruck";
Form6->ComboBox2->Items->Strings[1]="Vorant";
Form6->ComboBox2->Items->Strings[2]="Spiele";
Form6->ComboBox2->Items->Strings[3]="Pause";
Form6->ComboBox2->Items->Strings[4]="Hinzufugen";
Form6->ComboBox2->Items->Strings[5]="Fortschaffen";
Form6->ComboBox2->Items->Strings[6]="Erhalten Abhorenlis";
Form6->ComboBox2->Items->Strings[7]="Stimmen";
Form6->ComboBox2->Items->Strings[9]="Hinzufugen Abbildung";
Form6->ComboBox2->Items->Strings[8]="Hinzusetzen Abbildung";
Form6->ComboBox2->Items->Strings[10]="Entfernen Abbildung";
Form6->ComboBox2->Items->Strings[11]="Anhalten\\Starten Slideshow";
Form6->ComboBox2->Items->Strings[12]="Sortieren";
Form3->Image2->Hint="Zuruck";
Form3->Image8->Hint="Pause";
Form3->Image9->Hint="Spiele";
Form3->Image10->Hint="Voraus";
Form3->Image4->Hint="Hinzusetzen";
Form3->Image5->Hint="Entfernen";
Form3->Image6->Hint="Erhalten";

```

```

Form3->Image15->Hint="Sortieren";
Form3->Image7->Hint="Stimmen";
Form3->Image11->Hint="Abbildung offenen\\ zumachen";
Form3->Image12->Hint="Hinzusetzen Abbildung";
Form3->Image13->Hint="Entfernen Abbildung";
Form3->Image14->Hint="Stop\\Starten Slideshow";
Form4->Panel1->Caption="Hinzusetzen";
Form4->Panel2->Caption="Aufhebung";
Form5->Panel1->Caption="Hinzusetzen";
Form5->Panel2->Caption="Aufhebung";
Form4->CheckBox1->Caption="Hinzusetzen im Subcataloge";
Form3->N1->Caption="Entfernen";
Form3->N2->Caption="Entfernen Alles";
Form3->N6->Caption="Entfernen Duplicaten";
Form3->MenuItem1->Caption="Hinzusetzen File";
Form3->MenuItem2->Caption="Hinzusetzen Cataloge";
Form3->MenuItem3->Caption="Hinzusetzen File";
Form3->MenuItem4->Caption="Hinzusetzen Cataloge";
Form3->MenuItem5->Caption="Erhalten Verschweigen";
Form3->MenuItem6->Caption="Erhalten als";
Form3->N3->Caption="Beladen";
Form3->HTML1->Caption="Erhalten HTML";
Form3->MenuItem7->Caption="Als File";
Form3->MenuItem8->Caption="Als ABC";
Form3->N4->Caption="Als Singe";
Form3->N5->Caption="Verwirren";
Form3->MenuItem9->Caption="Entfernen";
Form3->MenuItem10->Caption="Entfernen Alles";
Form6->Button13->Caption="Uber Programme";
Form6->CheckBox4->Caption="Statistik";
Form6->Button14->Caption="Statistik";
}
}

```

```

void __fastcall TForm6::Button1Click(TObject *Sender)
{
if (ColorDialog1->Execute())
{
if (ComboBox1->ItemIndex==0)
{
Form3->Edit2->Color=ColorDialog1->Color;
Form3->Edit3->Color=ColorDialog1->Color;
}
if (ComboBox1->ItemIndex==1)
Form3->ListBox2->Color=ColorDialog1->Color;
if (ComboBox1->ItemIndex==2)
Form3->Color=ColorDialog1->Color;
if (ComboBox1->ItemIndex==3)
Form3->StatusBar1->Color=ColorDialog1->Color;
}
}

```

```

}
//-----
void __fastcall TForm6::Button2Click(TObject *Sender)
{
if (FontDialog1->Execute())
{
if (ComboBox1->ItemIndex==0)
{
Form3->Edit2->Font->Assign(FontDialog1->Font);
Form3->Edit3->Font->Assign(FontDialog1->Font);
}
if (ComboBox1->ItemIndex==1)
Form3->ListBox2->Font->Assign(FontDialog1->Font);
if (ComboBox1->ItemIndex==2)
Form3->Font->Assign(FontDialog1->Font);
if (ComboBox1->ItemIndex==3)
Form3->StatusBar1->Font->Assign(FontDialog1->Font);
}
}
//-----
void __fastcall TForm6::Button3Click(TObject *Sender)
{
Memo1->Lines->Clear();
/* данные об Edit'ax*/
Memo1->Lines->Add(Form3->Edit2->Color);
Memo1->Lines->Add(Form3->Edit2->Font->Name);
Memo1->Lines->Add(Form3->Edit2->Font->Color);
Memo1->Lines->Add(Form3->Edit2->Font->Charset);
Memo1->Lines->Add(Form3->Edit2->Font->Size);
/* данные об ListBox'e*/
Memo1->Lines->Add(Form3->ListBox2->Color);
Memo1->Lines->Add(Form3->ListBox2->Font->Name);
Memo1->Lines->Add(Form3->ListBox2->Font->Color);
Memo1->Lines->Add(Form3->ListBox2->Font->Charset);
Memo1->Lines->Add(Form3->ListBox2->Font->Size);
/* данные об Form'e*/
Memo1->Lines->Add(Form3->Color);
Memo1->Lines->Add(Form3->Font->Name);
Memo1->Lines->Add(Form3->Font->Color);
Memo1->Lines->Add(Form3->Font->Charset);
Memo1->Lines->Add(Form3->Font->Size);
/* данные об StatusBar'e*/
Memo1->Lines->Add(Form3->StatusBar1->Color);
Memo1->Lines->Add(Form3->StatusBar1->Font->Name);
Memo1->Lines->Add(Form3->StatusBar1->Font->Color);
Memo1->Lines->Add(Form3->StatusBar1->Font->Charset);
Memo1->Lines->Add(Form3->StatusBar1->Font->Size);
/* данные об альфа-компоненте */
Memo1->Lines->Add(Form3->AlphaBlendValue);
/* данные о таймере */

```

```

Memo1->Lines->Add(Form3->Timer2->Interval);
/* данные о картинке */
Memo1->Lines->Add(LabeledEdit2->Text);
/* данные о кнопках */
//for (int i=0; i<ListBox1->Items->Count-1;i++)
//Memo1->Lines->Add(ListBox1->Items->Strings[i]);
Memo1->Lines->Add(ListBox1->Items->Strings[0]);
Memo1->Lines->Add(ListBox1->Items->Strings[1]);
Memo1->Lines->Add(ListBox1->Items->Strings[2]);
Memo1->Lines->Add(ListBox1->Items->Strings[3]);
Memo1->Lines->Add(ListBox1->Items->Strings[4]);
Memo1->Lines->Add(ListBox1->Items->Strings[5]);
Memo1->Lines->Add(ListBox1->Items->Strings[6]);
Memo1->Lines->Add(ListBox1->Items->Strings[7]);
Memo1->Lines->Add(ListBox1->Items->Strings[8]);
Memo1->Lines->Add(ListBox1->Items->Strings[9]);
Memo1->Lines->Add(ListBox1->Items->Strings[10]);
Memo1->Lines->Add(ListBox1->Items->Strings[11]);

/* размножать ли? */
Memo1->Lines->Add(StrToInt(AnsiString(Form3->multiplybkg)));
int tempint;
tempint=CheckBox1->Checked;
Memo1->Lines->Add(tempint);
tempint=CheckBox3->Checked;
Memo1->Lines->Add(tempint);
Memo1->Lines->Add(LabeledEdit4->Text);
Memo1->Lines->Add(LabeledEdit5->Text);
Memo1->Lines->Add(LabeledEdit6->Text);
Memo1->Lines->Add(LabeledEdit7->Text);
Memo1->Lines->Add(ListBox1->Items->Strings[12]);
Memo1->Lines->SaveToFile(LabeledEdit3->Text);
}
//-----
void __fastcall TForm6::Button4Click(TObject *Sender)
{
try {
Form3->AlphaBlendValue=StrToInt(Edit1->Text);
}
catch ( EConvertError& ) { ShowMessage("Некорректный ввод. Введите целое число от 0 до 255.");}
}
//-----
void __fastcall TForm6::Button5Click(TObject *Sender)
{
try {
Form3->Timer2->Interval=StrToInt(LabeledEdit1->Text);
}
catch ( EConvertError& ) { ShowMessage("Некорректный ввод. Введите целое число.");}
}

```



```
//-----
void __fastcall TForm6::Button6Click(TObject *Sender)
{
Form3->Timer1->Enabled=false;
Form3->Timer2->Enabled=false;
Form3->Timer3->Enabled=false;
if (OpenPictureDialog1->Execute())
{
Image1->Picture->LoadFromFile(OpenPictureDialog1->FileName);
LabeledEdit2->Text=OpenPictureDialog1->FileName;
Form3->Image3->Picture->LoadFromFile(OpenPictureDialog1->FileName);
}
Form3->Timer1->Enabled=true;
Form3->Timer2->Enabled=true;
Form3->Timer3->Enabled=true;    //
}
//-----
void __fastcall TForm6::Button7Click(TObject *Sender)
{
Form3->Timer1->Enabled=false;
Form3->Timer2->Enabled=false;
Form3->Timer3->Enabled=false;
if (OpenPictureDialog2->Execute())
{
Image2->Picture->LoadFromFile(OpenPictureDialog2->FileName);
Image2->Picture->LoadFromFile(OpenPictureDialog2->FileName);
if (ComboBox2->ItemIndex==0)
ListBox1->Items->Strings[0]=OpenPictureDialog2->FileName;
if (ComboBox2->ItemIndex==1)
ListBox1->Items->Strings[1]=OpenPictureDialog2->FileName;
if (ComboBox2->ItemIndex==2)
ListBox1->Items->Strings[2]=OpenPictureDialog2->FileName;
if (ComboBox2->ItemIndex==3)
ListBox1->Items->Strings[3]=OpenPictureDialog2->FileName;
if (ComboBox2->ItemIndex==4)
ListBox1->Items->Strings[4]=OpenPictureDialog2->FileName;
if (ComboBox2->ItemIndex==5)
ListBox1->Items->Strings[5]=OpenPictureDialog2->FileName;
if (ComboBox2->ItemIndex==6)
ListBox1->Items->Strings[6]=OpenPictureDialog2->FileName;
if (ComboBox2->ItemIndex==7)
ListBox1->Items->Strings[7]=OpenPictureDialog2->FileName;
if (ComboBox2->ItemIndex==8)
ListBox1->Items->Strings[8]=OpenPictureDialog2->FileName;
if (ComboBox2->ItemIndex==9)
ListBox1->Items->Strings[9]=OpenPictureDialog2->FileName;
if (ComboBox2->ItemIndex==10)
ListBox1->Items->Strings[10]=OpenPictureDialog2->FileName;
if (ComboBox2->ItemIndex==11)
ListBox1->Items->Strings[11]=OpenPictureDialog2->FileName;
```

```

if (ComboBox2->ItemIndex==12)
ListBox1->Items->Strings[12]=OpenPictureDialog2->FileName;
}
Form3->Timer1->Enabled=true;
Form3->Timer2->Enabled=true;
Form3->Timer3->Enabled=true;
}      //
//-----
void __fastcall TForm6::CheckBox2Enter(TObject *Sender)
{
Form3->multiplybkg=CheckBox2->Checked;
}
//-----
void __fastcall TForm6::CheckBox2Exit(TObject *Sender)
{
Form3->multiplybkg=CheckBox2->Checked;
}
//-----
void __fastcall TForm6::Button9Click(TObject *Sender)
{
Form3->Timer1->Enabled=false;
Form3->Timer2->Enabled=false;
Form3->Timer3->Enabled=false;
if (SaveDialog1->Execute())
{
if      (ExtractFileExt(SaveDialog1->FileName)!=".jsf"      &&      ExtractFileExt(SaveDialog1->
>FileName)!=".JSF")
SaveDialog1->FileName=SaveDialog1->FileName+".jsf";
ListBox1->Items->SaveToFile(SaveDialog1->FileName);
Memo1->Lines->Clear();
// данные об Edit'ax
Memo1->Lines->Add(Form3->Edit2->Color);
Memo1->Lines->Add(Form3->Edit2->Font->Name);
Memo1->Lines->Add(Form3->Edit2->Font->Color);
Memo1->Lines->Add(Form3->Edit2->Font->Charset);
Memo1->Lines->Add(Form3->Edit2->Font->Size);
// данные об ListBox'e
Memo1->Lines->Add(Form3->ListBox2->Color);
Memo1->Lines->Add(Form3->ListBox2->Font->Name);
Memo1->Lines->Add(Form3->ListBox2->Font->Color);
Memo1->Lines->Add(Form3->ListBox2->Font->Charset);
Memo1->Lines->Add(Form3->ListBox2->Font->Size);
// данные об Form'e
Memo1->Lines->Add(Form3->Color);
Memo1->Lines->Add(Form3->Font->Name);
Memo1->Lines->Add(Form3->Font->Color);
Memo1->Lines->Add(Form3->Font->Charset);
Memo1->Lines->Add(Form3->Font->Size);
// данные об StatusBar'e
Memo1->Lines->Add(Form3->StatusBar1->Color);

```

```

Memo1->Lines->Add(Form3->StatusBar1->Font->Name);
Memo1->Lines->Add(Form3->StatusBar1->Font->Color);
Memo1->Lines->Add(Form3->StatusBar1->Font->Charset);
Memo1->Lines->Add(Form3->StatusBar1->Font->Size);
// данные об альфа-компоненте
Memo1->Lines->Add(Form3->AlphaBlendValue);
// данные о таймере
Memo1->Lines->Add(Form3->Timer2->Interval);
// данные о картинке
Memo1->Lines->Add(LabeledEdit2->Text);
// данные о кнопках
//for (int i=0; i<ListBox1->Items->Count-1;i++)
//Memo1->Lines->Add(ListBox1->Items->Strings[i]);
Memo1->Lines->Add(ListBox1->Items->Strings[0]);
Memo1->Lines->Add(ListBox1->Items->Strings[1]);
Memo1->Lines->Add(ListBox1->Items->Strings[2]);
Memo1->Lines->Add(ListBox1->Items->Strings[3]);
Memo1->Lines->Add(ListBox1->Items->Strings[4]);
Memo1->Lines->Add(ListBox1->Items->Strings[5]);
Memo1->Lines->Add(ListBox1->Items->Strings[6]);
Memo1->Lines->Add(ListBox1->Items->Strings[7]);
Memo1->Lines->Add(ListBox1->Items->Strings[8]);
Memo1->Lines->Add(ListBox1->Items->Strings[9]);
Memo1->Lines->Add(ListBox1->Items->Strings[10]);
Memo1->Lines->Add(ListBox1->Items->Strings[11]);
// размножать ли?
Memo1->Lines->Add(StrToInt(AnsiString(Form3->multiplybkg)));
int tempint;
tempint=CheckBox1->Checked;
Memo1->Lines->Add(tempint);
tempint=CheckBox3->Checked;
Memo1->Lines->Add(tempint);
Memo1->Lines->Add(LabeledEdit4->Text);
Memo1->Lines->Add(LabeledEdit5->Text);
Memo1->Lines->Add(LabeledEdit6->Text);
Memo1->Lines->Add(LabeledEdit7->Text);
Memo1->Lines->Add(ListBox1->Items->Strings[12]);
Memo1->Lines->SaveToFile(SaveDialog1->FileName);

}
Form3->Timer1->Enabled=true;
Form3->Timer2->Enabled=true;
Form3->Timer3->Enabled=true;

}
//-----

```

```

void __fastcall TForm6::Button11Click(TObject *Sender)
{

```

```
Form6->Close();
}
//-----
```

```
void __fastcall TForm6::Button10Click(TObject *Sender)
{
Form3->Timer1->Enabled=false;
Form3->Timer2->Enabled=false;
Form3->Timer3->Enabled=false;
if (OpenDialog1->Execute())
{
Form6->Memo1->Lines->LoadFromFile(OpenDialog1->FileName);
/* initialize init status */
Form3->Edit1->Color=StrToInt(Form6->Memo1->Lines->Strings[0]);
Form3->Edit2->Color=StrToInt(Form6->Memo1->Lines->Strings[0]);
Form3->Edit3->Color=StrToInt(Form6->Memo1->Lines->Strings[0]);
Form3->Edit4->Color=StrToInt(Form6->Memo1->Lines->Strings[0]);
Form3->Edit1->Font->Name=Form6->Memo1->Lines->Strings[1];
Form3->Edit1->Font->Color=StrToInt(Form6->Memo1->Lines->Strings[2]);
Form3->Edit1->Font->Charset=StrToInt(Form6->Memo1->Lines->Strings[3]);
Form3->Edit1->Font->Size=StrToInt(Form6->Memo1->Lines->Strings[4]);
Form3->Edit2->Font->Name=Form6->Memo1->Lines->Strings[1];
Form3->Edit2->Font->Color=StrToInt(Form6->Memo1->Lines->Strings[2]);
Form3->Edit2->Font->Charset=StrToInt(Form6->Memo1->Lines->Strings[3]);
Form3->Edit2->Font->Size=StrToInt(Form6->Memo1->Lines->Strings[4]);
Form3->Edit3->Font->Name=Form6->Memo1->Lines->Strings[1];
Form3->Edit3->Font->Color=StrToInt(Form6->Memo1->Lines->Strings[2]);
Form3->Edit3->Font->Charset=StrToInt(Form6->Memo1->Lines->Strings[3]);
Form3->Edit3->Font->Size=StrToInt(Form6->Memo1->Lines->Strings[4]);
Form3->Edit4->Font->Name=Form6->Memo1->Lines->Strings[1];
Form3->Edit4->Font->Color=StrToInt(Form6->Memo1->Lines->Strings[2]);
Form3->Edit4->Font->Charset=StrToInt(Form6->Memo1->Lines->Strings[3]);
Form3->Edit4->Font->Size=StrToInt(Form6->Memo1->Lines->Strings[4]);
Form3->ListBox2->Color=StrToInt(Form6->Memo1->Lines->Strings[5]);
Form3->ListBox2->Font->Name=Form6->Memo1->Lines->Strings[6];
Form3->ListBox2->Font->Color=StrToInt(Form6->Memo1->Lines->Strings[7]);
Form3->ListBox2->Font->Charset=StrToInt(Form6->Memo1->Lines->Strings[8]);
Form3->ListBox2->Font->Size=StrToInt(Form6->Memo1->Lines->Strings[9]);
Form3->Color=StrToInt(Form6->Memo1->Lines->Strings[10]);
Form3->Font->Name=Form6->Memo1->Lines->Strings[11];
Form3->Font->Color=StrToInt(Form6->Memo1->Lines->Strings[12]);
Form3->Font->Charset=StrToInt(Form6->Memo1->Lines->Strings[13]);
Form3->Font->Size=StrToInt(Form6->Memo1->Lines->Strings[14]);
Form3->StatusBar1->Color=StrToInt(Form6->Memo1->Lines->Strings[15]);
Form3->StatusBar1->Font->Name=Form6->Memo1->Lines->Strings[16];
Form3->StatusBar1->Font->Color=StrToInt(Form6->Memo1->Lines->Strings[17]);
Form3->StatusBar1->Font->Charset=StrToInt(Form6->Memo1->Lines->Strings[18]);
Form3->StatusBar1->Font->Size=StrToInt(Form6->Memo1->Lines->Strings[19]);
Form3->AlphaBlendValue=StrToInt(Form6->Memo1->Lines->Strings[20]);
Form3->Timer2->Interval=StrToInt(Form6->Memo1->Lines->Strings[21]);
```

```

Form3->Image3->Picture->LoadFromFile(Form6->Memo1->Lines->Strings[22]);
Form6->Image1->Picture->LoadFromFile(Form6->Memo1->Lines->Strings[22]);
Form6->LabeledEdit2->Text=Form6->Memo1->Lines->Strings[22];
Form3->Image2->Picture->LoadFromFile(Form6->Memo1->Lines->Strings[23]);
Form3->Image10->Picture->LoadFromFile(Form6->Memo1->Lines->Strings[24]);
Form3->Image9->Picture->LoadFromFile(Form6->Memo1->Lines->Strings[25]);
Form3->Image8->Picture->LoadFromFile(Form6->Memo1->Lines->Strings[26]);
Form3->Image4->Picture->LoadFromFile(Form6->Memo1->Lines->Strings[27]);
Form3->Image5->Picture->LoadFromFile(Form6->Memo1->Lines->Strings[28]);
Form3->Image6->Picture->LoadFromFile(Form6->Memo1->Lines->Strings[29]);
Form3->Image7->Picture->LoadFromFile(Form6->Memo1->Lines->Strings[30]);
Form3->Image11->Picture->LoadFromFile(Form6->Memo1->Lines->Strings[31]);
Form3->Image12->Picture->LoadFromFile(Form6->Memo1->Lines->Strings[32]);
Form3->Image13->Picture->LoadFromFile(Form6->Memo1->Lines->Strings[33]);
Form3->Image14->Picture->LoadFromFile(Form6->Memo1->Lines->Strings[34]);
Form3->multiplybkg=StrToInt(Form6->Memo1->Lines->Strings[35]);
Form6->CheckBox2->Checked=Form3->multiplybkg;
Form6->CheckBox1->Checked=StrToInt(Form6->Memo1->Lines->Strings[36]);
Form6->CheckBox3->Checked=StrToInt(Form6->Memo1->Lines->Strings[37]);
Form6->LabeledEdit4->Text=Form6->Memo1->Lines->Strings[38];
Form6->LabeledEdit5->Text=Form6->Memo1->Lines->Strings[39];
Form6->LabeledEdit6->Text=Form6->Memo1->Lines->Strings[40];
Form6->LabeledEdit7->Text=Form6->Memo1->Lines->Strings[41];
Form3->ListBox2->Width=StrToInt(Form6->LabeledEdit4->Text);
Form3->ListBox2->Height=StrToInt(Form6->LabeledEdit5->Text);
Form3->Image1->Width=StrToInt(Form6->LabeledEdit6->Text);
Form3->Image1->Height=StrToInt(Form6->LabeledEdit7->Text);
Form3->Image1->Left=StrToInt(Form6->LabeledEdit4->Text);
Form3->Height=Form3->ListBox2->Top+Form3->ListBox2->Height+80;
Form3->Width=Form3->ListBox2->Width;
Form3->TrackBar2->Width=Form3->ListBox2->Width;
Form3->Image4->Top=Form3->ListBox2->Top+Form3->ListBox2->Height+2;
Form3->Image5->Top=Form3->ListBox2->Top+Form3->ListBox2->Height+2;
Form3->Image6->Top=Form3->ListBox2->Top+Form3->ListBox2->Height+2;
Form3->Image7->Top=Form3->ListBox2->Top+Form3->ListBox2->Height+2;
Form3->Image11->Top=Form3->ListBox2->Top+Form3->ListBox2->Height+2;
Form3->Image15->Top=Form3->ListBox2->Top+Form3->ListBox2->Height+2;
Form3->Image12->Left=Form3->ListBox2->Width+10;
Form3->Image13->Left=Form3->ListBox2->Width+Form3->Image12->Width+15;
Form3->Image14->Left=Form3->ListBox2->Width+Form3->Image12->Width+Form3->Image13-
>Width+20;
Form3->Image12->Top=Form3->Image1->Height+20;
Form3->Image13->Top=Form3->Image1->Height+20;
Form3->Image14->Top=Form3->Image1->Height+20;
Form3->TrackBar1->Top=Form3->Image1->Height+20;
Form3->Edit1->Top=Form3->Image1->Height+20;
Form3->Edit1->Left=Form3->ListBox2->Width+Form3->Image12->Width+Form3->Image13-
>Width+Form3->Image14->Width+25;
Form3->TrackBar1->Left=Form3->Edit1->Width+Form3->ListBox2->Width+Form3->Image12-
>Width+Form3->Image13->Width+Form3->Image14->Width+25;

```

```

Form3->Image16->Picture->LoadFromFile("animation\\split.bmp");
Form6->ListBox1->Items->Strings[0]=Form6->Memo1->Lines->Strings[23];
Form6->ListBox1->Items->Strings[1]=Form6->Memo1->Lines->Strings[24];
Form6->ListBox1->Items->Strings[2]=Form6->Memo1->Lines->Strings[25];
Form6->ListBox1->Items->Strings[3]=Form6->Memo1->Lines->Strings[26];
Form6->ListBox1->Items->Strings[4]=Form6->Memo1->Lines->Strings[27];
Form6->ListBox1->Items->Strings[5]=Form6->Memo1->Lines->Strings[28];
Form6->ListBox1->Items->Strings[6]=Form6->Memo1->Lines->Strings[29];
Form6->ListBox1->Items->Strings[7]=Form6->Memo1->Lines->Strings[30];
Form6->ListBox1->Items->Strings[8]=Form6->Memo1->Lines->Strings[31];
Form6->ListBox1->Items->Strings[9]=Form6->Memo1->Lines->Strings[32];
Form6->ListBox1->Items->Strings[10]=Form6->Memo1->Lines->Strings[33];
Form6->ListBox1->Items->Strings[11]=Form6->Memo1->Lines->Strings[34];
Form6->ListBox1->Items->Strings[12]=Form6->Memo1->Lines->Strings[42];
/* end of init*/
}
Form3->Timer1->Enabled=true;
Form3->Timer2->Enabled=true;
Form3->Timer3->Enabled=true;
}
//-----

void __fastcall TForm6::Button8Click(TObject *Sender)
{
Form6->Memo1->Clear();
Form6->Memo1->Lines->LoadFromFile(LabeledEdit3->Text);
/* initialize init status */
Form3->Edit1->Color=StrToInt(Form6->Memo1->Lines->Strings[0]);
Form3->Edit2->Color=StrToInt(Form6->Memo1->Lines->Strings[0]);
Form3->Edit3->Color=StrToInt(Form6->Memo1->Lines->Strings[0]);
Form3->Edit4->Color=StrToInt(Form6->Memo1->Lines->Strings[0]);
Form3->Edit1->Font->Name=Form6->Memo1->Lines->Strings[1];
Form3->Edit1->Font->Color=StrToInt(Form6->Memo1->Lines->Strings[2]);
Form3->Edit1->Font->Charset=StrToInt(Form6->Memo1->Lines->Strings[3]);
Form3->Edit1->Font->Size=StrToInt(Form6->Memo1->Lines->Strings[4]);
Form3->Edit2->Font->Name=Form6->Memo1->Lines->Strings[1];
Form3->Edit2->Font->Color=StrToInt(Form6->Memo1->Lines->Strings[2]);
Form3->Edit2->Font->Charset=StrToInt(Form6->Memo1->Lines->Strings[3]);
Form3->Edit2->Font->Size=StrToInt(Form6->Memo1->Lines->Strings[4]);
Form3->Edit3->Font->Name=Form6->Memo1->Lines->Strings[1];
Form3->Edit3->Font->Color=StrToInt(Form6->Memo1->Lines->Strings[2]);
Form3->Edit3->Font->Charset=StrToInt(Form6->Memo1->Lines->Strings[3]);
Form3->Edit3->Font->Size=StrToInt(Form6->Memo1->Lines->Strings[4]);
Form3->Edit4->Font->Name=Form6->Memo1->Lines->Strings[1];
Form3->Edit4->Font->Color=StrToInt(Form6->Memo1->Lines->Strings[2]);
Form3->Edit4->Font->Charset=StrToInt(Form6->Memo1->Lines->Strings[3]);
Form3->Edit4->Font->Size=StrToInt(Form6->Memo1->Lines->Strings[4]);
Form3->ListBox2->Color=StrToInt(Form6->Memo1->Lines->Strings[5]);
Form3->ListBox2->Font->Name=Form6->Memo1->Lines->Strings[6];
Form3->ListBox2->Font->Color=StrToInt(Form6->Memo1->Lines->Strings[7]);

```

```

Form3->ListBox2->Font->Charset=StrToInt(Form6->Memo1->Lines->Strings[8]);
Form3->ListBox2->Font->Size=StrToInt(Form6->Memo1->Lines->Strings[9]);
Form3->Color=StrToInt(Form6->Memo1->Lines->Strings[10]);
Form3->Font->Name=Form6->Memo1->Lines->Strings[11];
Form3->Font->Color=StrToInt(Form6->Memo1->Lines->Strings[12]);
Form3->Font->Charset=StrToInt(Form6->Memo1->Lines->Strings[13]);
Form3->Font->Size=StrToInt(Form6->Memo1->Lines->Strings[14]);
Form3->StatusBar1->Color=StrToInt(Form6->Memo1->Lines->Strings[15]);
Form3->StatusBar1->Font->Name=Form6->Memo1->Lines->Strings[16];
Form3->StatusBar1->Font->Color=StrToInt(Form6->Memo1->Lines->Strings[17]);
Form3->StatusBar1->Font->Charset=StrToInt(Form6->Memo1->Lines->Strings[18]);
Form3->StatusBar1->Font->Size=StrToInt(Form6->Memo1->Lines->Strings[19]);
Form3->AlphaBlendValue=StrToInt(Form6->Memo1->Lines->Strings[20]);
Form3->Timer2->Interval=StrToInt(Form6->Memo1->Lines->Strings[21]);
Form3->Image3->Picture->LoadFromFile(Form6->Memo1->Lines->Strings[22]);
Form6->Image1->Picture->LoadFromFile(Form6->Memo1->Lines->Strings[22]);
Form6->LabeledEdit2->Text=Form6->Memo1->Lines->Strings[22];
Form3->Image2->Picture->LoadFromFile(Form6->Memo1->Lines->Strings[23]);
Form3->Image10->Picture->LoadFromFile(Form6->Memo1->Lines->Strings[24]);
Form3->Image9->Picture->LoadFromFile(Form6->Memo1->Lines->Strings[25]);
Form3->Image8->Picture->LoadFromFile(Form6->Memo1->Lines->Strings[26]);
Form3->Image4->Picture->LoadFromFile(Form6->Memo1->Lines->Strings[27]);
Form3->Image5->Picture->LoadFromFile(Form6->Memo1->Lines->Strings[28]);
Form3->Image6->Picture->LoadFromFile(Form6->Memo1->Lines->Strings[29]);
Form3->Image7->Picture->LoadFromFile(Form6->Memo1->Lines->Strings[30]);
Form3->Image11->Picture->LoadFromFile(Form6->Memo1->Lines->Strings[31]);
Form3->Image12->Picture->LoadFromFile(Form6->Memo1->Lines->Strings[32]);
Form3->Image13->Picture->LoadFromFile(Form6->Memo1->Lines->Strings[33]);
Form3->Image14->Picture->LoadFromFile(Form6->Memo1->Lines->Strings[34]);
Form3->multiplybkg=StrToInt(Form6->Memo1->Lines->Strings[35]);
Form6->CheckBox2->Checked=Form3->multiplybkg;
Form6->CheckBox1->Checked=StrToInt(Form6->Memo1->Lines->Strings[36]);
Form6->CheckBox3->Checked=StrToInt(Form6->Memo1->Lines->Strings[37]);
Form6->LabeledEdit4->Text=Form6->Memo1->Lines->Strings[38];
Form6->LabeledEdit5->Text=Form6->Memo1->Lines->Strings[39];
Form6->LabeledEdit6->Text=Form6->Memo1->Lines->Strings[40];
Form6->LabeledEdit7->Text=Form6->Memo1->Lines->Strings[41];
Form3->Image15->Picture->LoadFromFile(Form6->Memo1->Lines->Strings[42]);
Form3->ListBox2->Width=StrToInt(Form6->LabeledEdit4->Text);
Form3->ListBox2->Height=StrToInt(Form6->LabeledEdit5->Text);
Form3->Image1->Width=StrToInt(Form6->LabeledEdit6->Text);
Form3->Image1->Height=StrToInt(Form6->LabeledEdit7->Text);
Form3->Image1->Left=StrToInt(Form6->LabeledEdit4->Text);
Form3->Height=Form3->ListBox2->Top+Form3->ListBox2->Height+80;
Form3->Width=Form3->ListBox2->Width;
Form3->TrackBar2->Width=Form3->ListBox2->Width;
Form3->Image4->Top=Form3->ListBox2->Top+Form3->ListBox2->Height+2;
Form3->Image5->Top=Form3->ListBox2->Top+Form3->ListBox2->Height+2;
Form3->Image6->Top=Form3->ListBox2->Top+Form3->ListBox2->Height+2;
Form3->Image7->Top=Form3->ListBox2->Top+Form3->ListBox2->Height+2;

```

```

Form3->Image11->Top=Form3->ListBox2->Top+Form3->ListBox2->Height+2;
Form3->Image15->Top=Form3->ListBox2->Top+Form3->ListBox2->Height+2;
Form3->Image12->Left=Form3->ListBox2->Width+10;
Form3->Image13->Left=Form3->ListBox2->Width+Form3->Image12->Width+15;
Form3->Image14->Left=Form3->ListBox2->Width+Form3->Image12->Width+Form3->Image13-
>Width+20;
Form3->Image12->Top=Form3->Image1->Height+20;
Form3->Image13->Top=Form3->Image1->Height+20;
Form3->Image14->Top=Form3->Image1->Height+20;
Form3->TrackBar1->Top=Form3->Image1->Height+20;
Form3->Edit1->Top=Form3->Image1->Height+20;
Form3->Edit1->Left=Form3->ListBox2->Width+Form3->Image12->Width+Form3->Image13-
>Width+Form3->Image14->Width+25;
Form3->TrackBar1->Left=Form3->Edit1->Width+Form3->ListBox2->Width+Form3->Image12-
>Width+Form3->Image13->Width+Form3->Image14->Width+25;
Form3->Image16->Picture->LoadFromFile("animation\\split.bmp");
Form6->ListBox1->Items->Strings[0]=Form6->Memo1->Lines->Strings[23];
Form6->ListBox1->Items->Strings[1]=Form6->Memo1->Lines->Strings[24];
Form6->ListBox1->Items->Strings[2]=Form6->Memo1->Lines->Strings[25];
Form6->ListBox1->Items->Strings[3]=Form6->Memo1->Lines->Strings[26];
Form6->ListBox1->Items->Strings[4]=Form6->Memo1->Lines->Strings[27];
Form6->ListBox1->Items->Strings[5]=Form6->Memo1->Lines->Strings[28];
Form6->ListBox1->Items->Strings[6]=Form6->Memo1->Lines->Strings[29];
Form6->ListBox1->Items->Strings[7]=Form6->Memo1->Lines->Strings[30];
Form6->ListBox1->Items->Strings[8]=Form6->Memo1->Lines->Strings[31];
Form6->ListBox1->Items->Strings[9]=Form6->Memo1->Lines->Strings[32];
Form6->ListBox1->Items->Strings[10]=Form6->Memo1->Lines->Strings[33];
Form6->ListBox1->Items->Strings[11]=Form6->Memo1->Lines->Strings[34];
Form6->ListBox1->Items->Strings[12]=Form6->Memo1->Lines->Strings[42];
Form3->Image16->Picture->LoadFromFile("animation\\split.bmp");
}
//-----

```

```

void __fastcall TForm6::Button12Click(TObject *Sender)
{
Form3->Timer1->Enabled=false;
Form3->Timer2->Enabled=false;
Form3->Timer3->Enabled=false;
if (OpenDialog1->Execute())
{
LabeledEdit3->Text=OpenDialog1->FileName;
Form6->Memo2->Clear();
Form6->Memo2->Lines->Add(Form6->LabeledEdit3->Text);
Form6->Memo2->Lines->Add(Form3->Left);
Form6->Memo2->Lines->Add(Form3->Top);
Form6->Memo2->Lines->SaveToFile("jim.ini");
}
Form3->Timer1->Enabled=true;
Form3->Timer2->Enabled=true;

```



```

Form3->Timer3->Enabled=true;
}
//-----

void __fastcall TForm6::LabeledEdit4Change(TObject *Sender)
{
try
{
if (StrToInt(LabeledEdit4->Text)<209)
{
ShowMessage("Значение должно быть больше или равно 210");
LabeledEdit4->Text=210;
}
}
catch ( EConvertError& ) { ShowMessage("Неверный ввод");}
}
//-----

void __fastcall TForm6::Button13Click(TObject *Sender)
{
Form7->ShowModal();
}
//-----

void __fastcall TForm6::ComboBox3Change(TObject *Sender)
{
initLang();
}
//-----

void __fastcall TForm6::Button14Click(TObject *Sender)
{
Form8->ShowModal();
}
//-----

```

Окно «Об авторах»

Unit7.h

```

//-----

#ifndef Unit7H
#define Unit7H
//-----

```

```
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <ExtCtrls.hpp>
#include <jpeg.hpp>
//-----
class TForm7 : public TForm
{
__published: // IDE-managed Components
    TMemo *Memo1;
    TImage *Image1;
    TImage *Image2;
    TPanel *Panel1;
    void __fastcall Panel1Click(TObject *Sender);
private: // User declarations
public: // User declarations
    __fastcall TForm7(TComponent* Owner);
};
//-----
extern PACKAGE TForm7 *Form7;
//-----
#endif
```

Unit7.cpp

```
//-----

#include <vcl.h>
#pragma hdrstop

#include "Unit7.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm7 *Form7;
//-----
__fastcall TForm7::TForm7(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TForm7::Panel1Click(TObject *Sender)
{
Form7->Close();
}
//-----
```

Окно статистики

Unit8.h

```
//-----

#ifndef Unit8H
#define Unit8H
//-----
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <DB.hpp>
#include <DBTables.hpp>
#include <DBGrids.hpp>
#include <Grids.hpp>
#include <DBCtrls.hpp>
#include <ExtCtrls.hpp>
#include <Mask.hpp>
//-----
class TForm8 : public TForm
{
__published: // IDE-managed Components
    TDataSource *DataSource1;
    TTable *Table1;
    TQuery *Query1;
    TDBGrid *DBGrid2;
    TDataSource *DataSource2;
    TComboBox *ComboBox1;
    TLabeledEdit *LabeledEdit1;
    TLabeledEdit *LabeledEdit2;
    TDBNavigator *DBNavigator1;
    TDBGrid *DBGrid1;
    TButton *Button1;
    TLabeledEdit *LabeledEdit3;
    TLabeledEdit *LabeledEdit5;
    TButton *Button2;
    TEdit *Edit1;
    void __fastcall Button1Click(TObject *Sender);
    void __fastcall ComboBox1Change(TObject *Sender);
    void __fastcall Button2Click(TObject *Sender);
private: // User declarations
public: // User declarations
    __fastcall TForm8(TComponent* Owner);
};
//-----
extern PACKAGE TForm8 *Form8;
```

```
//-----
#endif
```

Unit8.cpp

```
//-----
```

```
#include <vcl.h>
#pragma hdrstop
```

```
#include "Unit8.h"
```

```
//-----
```

```
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm8 *Form8;
```

```
//-----
```

```
__fastcall TForm8::TForm8(TComponent* Owner)
    : TForm(Owner)
```

```
{
}
```

```
//-----
```

```
void __fastcall TForm8::Button1Click(TObject *Sender)
```

```
{
if (ComboBox1->ItemIndex==0)
{
Query1->Close();
Query1->SQL->Clear();
Query1->SQL->Add("SELECT Group_c, Song_c, Access_c, FirstAccess_c, LastAccess_c,
UserLength_c FROM db\\statistic.db");
Query1->Open();
}
if (ComboBox1->ItemIndex==1)
{
Query1->Close();
Query1->SQL->Clear();
Query1->SQL->Add("SELECT Group_c, Song_c, Access_c, FirstAccess_c, LastAccess_c,
UserLength_c FROM db\\statistic.db WHERE Group_c LIKE '\\%"+LabeledEdit1->Text+"%\\"");
Query1->Open();
}
if (ComboBox1->ItemIndex==2)
{
Query1->Close();
Query1->SQL->Clear();
Query1->SQL->Add("SELECT Group_c, Song_c, Access_c, FirstAccess_c, LastAccess_c,
UserLength_c FROM db\\statistic.db WHERE Song_c LIKE '\\%"+LabeledEdit2->Text+"%\\"");
Query1->Open();
}
if (ComboBox1->ItemIndex==3)
{
```

```

Query1->Close();
Query1->SQL->Clear();
Query1->SQL->Add("SELECT Group_c, Song_c, Access_c, FirstAccess_c, LastAccess_c,
UserLength_c FROM db\\statistic.db WHERE Group_c LIKE \"%\" + LabeledEdit1->Text + \"%
and Song_c LIKE \"%\" + LabeledEdit2->Text + \"%\"");
Query1->Open();
}
if (ComboBox1->ItemIndex==4)
{
Query1->Close();
Query1->SQL->Clear();
Query1->SQL->Add("SELECT Group_c, sum(Access_c) FROM db\\statistic.db GROUP BY
Group_c");
Query1->Open();
AnsiString Group_b;
int maxcount=0;
Query1->First();
while (!Query1->Eof)
{
if (maxcount<Query1->FieldByName("SUM OF Access_c")->AsInteger)
{
Group_b=Query1->FieldByName("Group_c")->AsString;
maxcount=Query1->FieldByName("SUM OF Access_c")->AsInteger;
}
}
Query1->Next();
}
Query1->SQL->Clear();
Query1->SQL->Add("SELECT Group_c, Song_c, Access_c, FirstAccess_c, LastAccess_c,
UserLength_c FROM db\\statistic.db WHERE Group_c=\"\"+Group_b+"\"");
Query1->Open();
}
if (ComboBox1->ItemIndex==5)
{
Query1->Close();
Query1->SQL->Clear();
Query1->SQL->Add("SELECT Group_c, Song_c, Access_c, FirstAccess_c, LastAccess_c,
UserLength_c FROM db\\statistic.db WHERE Access_c=(SELECT max(Access_c) FROM
db\\statistic.db)");
Query1->Open();
}
if (ComboBox1->ItemIndex==6)
{
Query1->Close();
Query1->SQL->Clear();
Query1->SQL->Add("SELECT Group_c, count(Access_c) FROM db\\statistic.db GROUP BY
Group_c");
Query1->Open();
AnsiString Group_b;
int maxcount=0;
Query1->First();

```

```

while (!Query1->Eof)
{
if (maxcount<Query1->FieldByName("COUNT OF Access_c")->AsInteger)
{
Group_b=Query1->FieldByName("Group_c")->AsString;
maxcount=Query1->FieldByName("COUNT OF Access_c")->AsInteger;
}
Query1->Next();
}
Query1->SQL->Clear();
Query1->SQL->Add("SELECT Group_c, Song_c, Access_c, FirstAccess_c, LastAccess_c,
UserLength_c FROM db\\statistic.db WHERE Group_c='"+Group_b+"'");
Query1->Open();
}
if (ComboBox1->ItemIndex==7)
{
Query1->SQL->Clear();
Query1->SQL->Add("SELECT Group_c, Song_c, Access_c, FirstAccess_c, LastAccess_c,
UserLength_c FROM db\\statistic.db WHERE LastAccess_c=(SELECT max(LastAccess_c) FROM
db\\statistic.db)");
Query1->Open();
}
if (ComboBox1->ItemIndex==8)
{
Query1->SQL->Clear();
Query1->SQL->Add("SELECT Group_c, Song_c, Access_c, FirstAccess_c, LastAccess_c,
UserLength_c FROM db\\statistic.db WHERE FirstAccess_c=(SELECT min(FirstAccess_c) FROM
db\\statistic.db)");
Query1->Open();
}
if (ComboBox1->ItemIndex==9)
{
Query1->SQL->Clear();
Query1->SQL->Add("SELECT Group_c, Song_c, Access_c, FirstAccess_c, LastAccess_c,
UserLength_c FROM db\\statistic.db WHERE UserLength_c=(SELECT max(UserLength_c) FROM
db\\statistic.db)");
Query1->Open();
}
if (ComboBox1->ItemIndex==10)
{
Query1->SQL->Clear();
Query1->SQL->Add("SELECT Group_c, Song_c, Access_c, FirstAccess_c, LastAccess_c,
UserLength_c FROM db\\statistic.db WHERE UserLength_c=(SELECT min(UserLength_c) FROM
db\\statistic.db)");
Query1->Open();
}
if (ComboBox1->ItemIndex==11)
{
int i=0;
Table1->First();

```

```

while (!Table1->Eof)
{
i++;
//ShowMessage(Table1->FieldByName("Group_c")->AsString);
Table1->Next();
}
ShowMessage("Количество прослушанных песен: "+IntToStr(i));
}
if (ComboBox1->ItemIndex==12)
{
long int min=0,sec=0;
AnsiString min0,sec0,fullUL;
Table1->First();
while (!Table1->Eof)
{
fullUL=Table1->FieldByName("UserLength_c")->AsString;
int i=1;
min0=fullUL;
sec0=fullUL;
while(fullUL[i]!='\:')
{
min0[i]=fullUL[i];
i++;
}
min0[i]='\0';
i=i+1;
int j=1;
while (i<=fullUL.Length())
{
sec0[j]=fullUL[i];
i++; j++;
}
sec0[j]='\0';
min=StrToInt(min0);
sec+=StrToInt(sec0);
sec+=60*min;
Table1->Next();
}
int sectotal=(IntToStr(sec)-(IntToStr(sec)/60)*60);
int mintotal=IntToStr(sec)/60;
ShowMessage(" "+IntToStr(mintotal)+"\:"+IntToStr(sectotal)+" ");
}
if (ComboBox1->ItemIndex==13)
{
long int min=0,sec=0,sec1=0;
AnsiString min0,sec0,fullUL;
Table1->First();
while (!Table1->Eof)
{
fullUL=Table1->FieldByName("UserLength_c")->AsString;

```

```

int i=1;
min0=fullUL;
sec0=fullUL;
while(fullUL[i]!='\:')
{
min0[i]=fullUL[i];
i++;
}
min0[i]='\0';
i=i+1;
int j=1;
while (i<=fullUL.Length())
{
sec0[j]=fullUL[i];
i++; j++;
}
sec0[j]='\0';
min=StrToInt(min0);
sec1=StrToInt(sec0);
sec1+=60*min;
sec1*=Table1->FieldByName("Access_c")->AsInteger;
sec+=sec1;
Table1->Next();
}
int sectotal=(IntToStr(sec)-(IntToStr(sec)/60)*60);
int mintotal=IntToStr(sec)/60;
ShowMessage(" "+IntToStr(mintotal)+"\:"+IntToStr(sectotal)+" ");
}
}
//-----
void __fastcall TForm8::ComboBox1Change(TObject *Sender)
{
if (ComboBox1->ItemIndex==0)
{
Query1->Close();
Query1->SQL->Clear();
Query1->SQL->Add("SELECT  Group_c,  Song_c,  Access_c,  FirstAccess_c,  LastAccess_c,
UserLength_c FROM db\\statistic.db");
Query1->Open();
}
if (ComboBox1->ItemIndex==1)
{
Query1->Close();
Query1->SQL->Clear();
Query1->SQL->Add("SELECT  Group_c,  Song_c,  Access_c,  FirstAccess_c,  LastAccess_c,
UserLength_c FROM db\\statistic.db WHERE Group_c LIKE '\\%"+LabeledEdit1->Text+"%\\"");
Query1->Open();
}
if (ComboBox1->ItemIndex==2)
{

```



```

Query1->Close();
Query1->SQL->Clear();
Query1->SQL->Add("SELECT Group_c, Song_c, Access_c, FirstAccess_c, LastAccess_c,
UserLength_c FROM db\\statistic.db WHERE Song_c LIKE \"\\%\"+LabeledEdit2->Text+\"\\%\"");
Query1->Open();
}
if (ComboBox1->ItemIndex==3)
{
Query1->Close();
Query1->SQL->Clear();
Query1->SQL->Add("SELECT Group_c, Song_c, Access_c, FirstAccess_c, LastAccess_c,
UserLength_c FROM db\\statistic.db WHERE Group_c LIKE \"\\%\" + LabeledEdit1->Text + \"\\%
and Song_c LIKE \"\\%\" + LabeledEdit2->Text + \"\\%\"");
Query1->Open();
}
if (ComboBox1->ItemIndex==4)
{
Query1->Close();
Query1->SQL->Clear();
Query1->SQL->Add("SELECT Group_c, sum(Access_c) FROM db\\statistic.db GROUP BY
Group_c");
Query1->Open();
AnsiString Group_b;
int maxcount=0;
Query1->First();
while (!Query1->Eof)
{
if (maxcount<Query1->FieldByName("SUM OF Access_c")->AsInteger)
{
Group_b=Query1->FieldByName("Group_c")->AsString;
maxcount=Query1->FieldByName("SUM OF Access_c")->AsInteger;
}
}
Query1->Next();
}
Query1->SQL->Clear();
Query1->SQL->Add("SELECT Group_c, Song_c, Access_c, FirstAccess_c, LastAccess_c,
UserLength_c FROM db\\statistic.db WHERE Group_c=\"\"+Group_b+\"\"");
Query1->Open();
}
if (ComboBox1->ItemIndex==5)
{
Query1->Close();
Query1->SQL->Clear();
Query1->SQL->Add("SELECT Group_c, Song_c, Access_c, FirstAccess_c, LastAccess_c,
UserLength_c FROM db\\statistic.db WHERE Access_c=(SELECT max(Access_c) FROM
db\\statistic.db)");
Query1->Open();
}
if (ComboBox1->ItemIndex==6)
{

```

```

Query1->Close();
Query1->SQL->Clear();
Query1->SQL->Add("SELECT Group_c, count(Access_c) FROM db\\statistic.db GROUP BY
Group_c");
Query1->Open();
AnsiString Group_b;
int maxcount=0;
Query1->First();
while (!Query1->Eof)
{
if (maxcount<Query1->FieldByName("COUNT OF Access_c")->AsInteger)
{
Group_b=Query1->FieldByName("Group_c")->AsString;
maxcount=Query1->FieldByName("COUNT OF Access_c")->AsInteger;
}
Query1->Next();
}
Query1->SQL->Clear();
Query1->SQL->Add("SELECT Group_c, Song_c, Access_c, FirstAccess_c, LastAccess_c,
UserLength_c FROM db\\statistic.db WHERE Group_c='"+Group_b+"'");
Query1->Open();
}
if (ComboBox1->ItemIndex==7)
{
Query1->SQL->Clear();
Query1->SQL->Add("SELECT Group_c, Song_c, Access_c, FirstAccess_c, LastAccess_c,
UserLength_c FROM db\\statistic.db WHERE LastAccess_c=(SELECT max(LastAccess_c) FROM
db\\statistic.db)");
Query1->Open();
}
if (ComboBox1->ItemIndex==8)
{
Query1->SQL->Clear();
Query1->SQL->Add("SELECT Group_c, Song_c, Access_c, FirstAccess_c, LastAccess_c,
UserLength_c FROM db\\statistic.db WHERE FirstAccess_c=(SELECT min(FirstAccess_c) FROM
db\\statistic.db)");
Query1->Open();
}
if (ComboBox1->ItemIndex==9)
{
Query1->SQL->Clear();
Query1->SQL->Add("SELECT Group_c, Song_c, Access_c, FirstAccess_c, LastAccess_c,
UserLength_c FROM db\\statistic.db WHERE UserLength_c=(SELECT max(UserLength_c) FROM
db\\statistic.db)");
Query1->Open();
}
if (ComboBox1->ItemIndex==10)
{
Query1->SQL->Clear();
Query1->SQL->Add("SELECT Group_c, Song_c, Access_c, FirstAccess_c, LastAccess_c,

```

```

UserLength_c FROM db\\statistic.db WHERE UserLength_c=(SELECT min(UserLength_c) FROM
db\\statistic.db));
Query1->Open();
}
if (ComboBox1->ItemIndex==11)
{
int i=0;
Table1->First();
while (!Table1->Eof)
{
i++;
//ShowMessage(Table1->FieldByName("Group_c")->AsString);
Table1->Next();
}
ShowMessage("Количество прослушанных песен: "+IntToStr(i));
}
if (ComboBox1->ItemIndex==12)
{
long int min=0,sec=0;
AnsiString min0,sec0,fullUL;
Table1->First();
while (!Table1->Eof)
{
fullUL=Table1->FieldByName("UserLength_c")->AsString;
int i=1;
min0=fullUL;
sec0=fullUL;
while(fullUL[i]!='\:')
{
min0[i]=fullUL[i];
i++;
}
min0[i]='\0';
i=i+1;
int j=1;
while (i<=fullUL.Length())
{
sec0[j]=fullUL[i];
i++; j++;
}
sec0[j]='\0';
min=StrToInt(min0);
sec+=StrToInt(sec0);
sec+=60*min;
Table1->Next();
}
int sectotal=(IntToStr(sec)-(IntToStr(sec)/60)*60);
int mintotal=IntToStr(sec)/60;
ShowMessage(" "+IntToStr(mintotal)+"\:"+IntToStr(sectotal)+" ");
}

```

```

if (ComboBox1->ItemIndex==13)
{
long int min=0,sec=0,sec1=0;
AnsiString min0,sec0,fullUL;
Table1->First();
while (!Table1->Eof)
{
fullUL=Table1->FieldByName("UserLength_c")->AsString;
int i=1;
min0=fullUL;
sec0=fullUL;
while(fullUL[i]!='\:')
{
min0[i]=fullUL[i];
i++;
}
min0[i]='\0';
i=i+1;
int j=1;
while (i<=fullUL.Length())
{
sec0[j]=fullUL[i];
i++; j++;
}
sec0[j]='\0';
min=StrToInt(min0);
sec1=StrToInt(sec0);
sec1+=60*min;
sec1*=Table1->FieldByName("Access_c")->AsInteger;
sec+=sec1;
Table1->Next();
}
int sectotal=(IntToStr(sec)-(IntToStr(sec)/60)*60);
int mintotal=IntToStr(sec)/60;
ShowMessage(" "+IntToStr(mintotal)+"\:"+IntToStr(sectotal)+" ");
}
} //
//-----

```

```

void __fastcall TForm8::Button2Click(TObject *Sender)
{
Table1->First();
while (!Table1->Eof)
{
Table1->Edit();
Table1->Delete();
Table1->Next();
}
}

```

```
Table1->Edit();
try {
Table1->Post();
}
catch (...) {}
}
LabeledEdit3->Text=0;
LabeledEdit5->Text=0;
}
//-----
```

Прочее

Программа разработана в среде *Borland Builder 6.01 Enterprise* на языке ANSI C++\Borland C\Objective C, в разработке использовались также:

- ⌚ Borland Image Editor
- ⌚ Adobe Photoshop 8 CS
- ⌚ Adobe ImageReady 8 CS
- ⌚ OpenOffice.org 2.0 Writer
- ⌚ Opera 8.5
- ⌚ Microsoft Notepad
- ⌚ Microsoft Calculator
- ⌚ ICQ 2003b pro
- ⌚ Mozilla ThunderBird 1.07
- ⌚ NullSoft WinAmp 5.06

Спасибо конечно всем исполнителям, песни которых были неотъемлемой частью жизни «Джима». Особо Г. Бачинскому, С. Стилавину не только за утреннее шоу, но и за их песни-субтитры, на которых производилось тестирование и настройка интеллектуального алгоритма ID3V1.1-распознавания. Спасибо замечательным художникам и умелым сканерам, благодаря которым на JPEG-картинах тестировалась графическая часть программы. Ну и конечно нашим родным пиратам и просто продавцам с Царицинского Радиорынка.

Особую благодарность хочу выразить своей вдохновительнице, музе, Оксане.

А так же специальное спасибо Пушки за бета-тестинг и английский перевод.

Верещагин И. Г.

GNU GENERAL PUBLIC LICENSE

Версия 2, июнь 1991г.

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Каждый вправе копировать и распространять экземпляры настоящей Лицензии без внесения изменений в ее текст.

Преамбула

Большинство лицензий на программное обеспечение лишает вас права распространять и вносить изменения в это программное обеспечение. Стандартная Общественная Лицензия GNU, напротив, разработана с целью гарантировать вам право совместно использовать и вносить изменения в свободное программное обеспечение, т.е. обеспечить свободный доступ к программному обеспечению для всех пользователей. Условия настоящей Стандартной Общественной Лицензии применяются к большей части программного обеспечения Free Software Foundation, а также к любому другому программному обеспечению по желанию его автора. (К некоторому программному обеспечению Free Software Foundation применяются условия Стандартной Общественной Лицензии GNU для Библиотек). Вы также можете применять Стандартную Общественную Лицензию к разработанному вами программному обеспечению.

Говоря о свободном программном обеспечении, мы имеем в виду свободу, а не безвозмездность. Настоящая Стандартная Общественная Лицензия разработана с целью гарантировать вам право распространять экземпляры свободного программного обеспечения (и при желании получать за это вознаграждение), право получать исходный текст программного обеспечения или иметь возможность его получить, право вносить изменения в программное обеспечение или использовать его части в новом свободном программном обеспечении, а также право знать, что вы имеете все вышеперечисленные права.

Чтобы защитить ваши права, мы вводим ряд ограничений с тем, чтобы никто не имел возможности лишить вас этих прав или обратиться к вам с предложением отказаться от этих прав. Данные ограничения налагают на вас определенные обязанности в случае, если вы распространяете экземпляры программного обеспечения или модифицируете программное обеспечение.

Например, если вы распространяете экземпляры такого программного обеспечения за плату или бесплатно, вы обязаны передать новым обладателям все права в том же объеме, в каком они принадлежат вам. Вы обязаны обеспечить получение новыми обладателями программы ее исходного текста или возможность его получить. Вы также обязаны

ознакомить их с условиями настоящей Лицензии.

Для защиты ваших прав мы: (1) оставляем за собой авторские права на программное обеспечение и (2) предлагаем вам использовать настоящую Лицензию, в соответствии с условиями которой вы вправе воспроизводить, распространять и/или модифицировать программное обеспечение.

Кроме того, для защиты как нашей репутации, так и репутации других авторов программного обеспечения, мы уведомляем всех пользователей, что на данное программное обеспечение никаких гарантий не предоставляется. Те, кто приобрел программное обеспечение, с внесенными в него третьими лицами изменениями, должны знать, что они получают не оригинал, в силу чего автор оригинала не несет ответственности за ошибки в работе программного обеспечения, допущенные третьими лицами при внесении изменений.

Наконец, программное обеспечение перестает быть свободным в случае, если лицо приобретает на него исключительные права [1]. Недопустимо, чтобы лица, распространяющие свободное программное обеспечение, могли приобрести исключительные права на использование данного программного обеспечения и зарегистрировать их в Патентном ведомстве. Чтобы избежать этого, мы заявляем, что обладатель исключительных прав обязан предоставить любому лицу права на использование программного обеспечения либо не приобретать исключительных прав вообще.

Ниже изложены условия воспроизведения, распространения и модификации программного обеспечения.

Условия воспроизведения, распространения и модификации

0. Условия настоящей Лицензии применяются ко всем видам программного обеспечения или любому иному произведению, которое содержит указание правообладателя на то, что данное произведение может распространяться на условиях Стандартной Общественной Лицензии. Под термином "Программа" далее понимается любое подобное программное обеспечение или иное произведение. Под термином "произведение, производное от Программы" понимается Программа или любое иное производное произведение в соответствии с законодательством об авторском праве [2], т.е. произведение, включающее в себя Программу или ее часть, как с внесенными в ее текст изменениями, так и без них и/или переведенную на другой язык. (Здесь и далее, понятие "модификация" включает в себя понятие перевода в самом широком смысле). Каждый приобретатель экземпляра Программы именуется в дальнейшем "Лицензиат".

Действие настоящей Лицензии не распространяется на осуществление иных прав, кроме воспроизведения, распространения и модификации программного обеспечения. Не устанавливается ограничений на запуск Программы. Условия Лицензии распространяются на выходные данные из Программы только в том случае, если их содержание составляет

произведение, производное от Программы (независимо от того, было ли такое произведение создано в результате запуска Программы). Это зависит от того, какие функции выполняет Программа.

1. Лицензиат вправе изготавливать и распространять экземпляры исходного текста Программы в том виде, в каком он его получил, без внесения в него изменений на любом носителе, при соблюдении следующих условий: на каждом экземпляре помещен знак охраны авторского права и уведомление об отсутствии гарантий; оставлены без изменений все уведомления, относящиеся к настоящей Лицензии и отсутствию гарантий; вместе с экземпляром Программы приобретателю передается копия настоящей Лицензии.

Лицензиат вправе взимать плату за передачу экземпляра Программы, а также вправе за плату оказывать услуги по гарантийной поддержке Программы.

2. Лицензиат вправе модифицировать свой экземпляр или экземпляры Программы полностью или любую ее часть. Данные действия Лицензиата влекут за собой создание произведения, производного от Программы. Лицензиат вправе изготавливать и распространять экземпляры такого произведения, производного от Программы, или собственно экземпляры изменений в соответствии с пунктом 1 настоящей Лицензии при соблюдении следующих условий:

а) файлы, измененные Лицензиатом, должны содержать хорошо заметную пометку, что они были изменены, а также дату внесения изменений;

б) при распространении или публикации Лицензиатом любого произведения, которое содержит Программу или ее часть или является производным от Программы или от ее части, Лицензиат обязан передавать права на использование данного произведения третьим лицам на условиях настоящей Лицензии, при этом Лицензиат не вправе требовать уплаты каких-либо лицензионных платежей. Распространяемое произведение лицензируется как одно целое;

с) если модифицированная Программа при запуске обычно читает команды в интерактивном режиме, Лицензиат обязан обеспечить вывод на экран дисплея или печатающее устройство сообщения, которое должно включать в себя:

знак охраны авторского права;

уведомление об отсутствии гарантий на Программу (или иное, если Лицензиат предоставляет гарантии);

указание на то, что пользователи вправе распространять экземпляры Программы в соответствии с условиями настоящей Лицензии, а также на то, каким образом пользователь может ознакомиться с текстом настоящей Лицензии. (Исключение: если оригинальная Программа является интерактивной, но не выводит в своем обычном режиме работы сообщение такого рода, то вывод подобного сообщения произведением, производным от Программы, в этом случае не обязателен).

Вышеуказанные условия применяются к модифицированному произведению, производному от Программы, в целом. В случае если отдельные части данного произведения не являются производными от Программы, являются результатом творческой деятельности и могут быть использованы как самостоятельное произведение, Лицензиат вправе распространять отдельно такое произведение на иных лицензионных условиях. В случае если Лицензиат распространяет вышеуказанные части в составе произведения, производного от Программы, то условия настоящей Лицензии применяются к произведению в целом, при этом права, приобретаемые сублицензиатами на основании Лицензии, передаются им в отношении всего произведения, включая все его части, независимо от того, кто является их авторами.

Целью настоящего пункта 2 не является заявление прав или оспаривание прав на произведение, созданное исключительно Лицензиатом. Целью настоящего пункта является обеспечение права контролировать распространение произведений, производных от Программы, и составных произведений, производных от Программы.

Размещение произведения, которое не является производным от Программы, на одном устройстве для хранения информации или носителе вместе с Программой или произведением, производным от Программы, не влечет за собой распространения условий настоящей Лицензии на такое произведение.

3. Лицензиат вправе воспроизводить и распространять экземпляры Программы или произведения, которое является производным от Программы, в соответствии с пунктом 2 настоящей Лицензии, в виде объектного кода или в исполняемой форме в соответствии с условиями п.п.1 и 2 настоящей Лицензии при соблюдении одного из перечисленных ниже условий:

а) к экземпляру должен прилагаться соответствующий полный исходный текст в машиночитаемой форме, который должен распространяться в соответствии с условиями п.п. 1 и 2 настоящей Лицензии на носителе, обычно используемом для передачи программного обеспечения, либо

б) к экземпляру должно прилагаться действительное в течение трех лет предложение в письменной форме к любому третьему лицу передать за плату, не превышающую стоимость осуществления собственно передачи, экземпляр соответствующего полного исходного текста в машиночитаемой форме в соответствии с условиями п.п. 1 и 2 настоящей Лицензии на носителе, обычно используемом для передачи программного обеспечения, либо

с) к экземпляру должна прилагаться полученная Лицензиатом информация о предложении, в соответствии с которым можно получить соответствующий исходный текст. (Данное положение применяется исключительно в том случае, если Лицензиат осуществляет некоммерческое распространение программы, при этом программа была получена самим Лицензиатом в виде объектного кода или в исполняемой

форме и сопровождалась предложением, соответствующим условиям пп.б п.3 настоящей Лицензии).

Под исходным текстом произведения понимается такая форма произведения, которая наиболее удобна для внесения изменений. Под полным исходным текстом исполняемого произведения понимается исходный текст всех составляющих произведение модулей, а также всех файлов, связанных с описанием интерфейса, и сценариев, предназначенных для управления компиляцией и установкой исполняемого произведения. Однако, в качестве особого исключения, распространяемый исходный текст может не включать того, что обычно распространяется (в виде исходного текста или в бинарной форме) с основными компонентами (компилятор, ядро и т.д.) операционной системы, в которой работает исполняемое произведение, за исключением случаев, когда исполняемое произведение сопровождается таким компонентом.

В случае если произведение в виде объектного кода или в исполняемой форме распространяется путем предоставления доступа для копирования его из определенного места, обеспечение равноценного доступа для копирования исходного текста из этого же места удовлетворяет требованиям распространения исходного текста, даже если третьи лица при этом не обязаны копировать исходный текст вместе с объектным кодом произведения.

4. Лицензиат вправе воспроизводить, модифицировать, распространять или передавать права на использование Программы только на условиях настоящей Лицензии. Любое воспроизведение, модификация, распространение или передача прав на иных условиях являются недействительными и автоматически ведут к расторжению настоящей Лицензии и прекращению всех прав Лицензиата, предоставленных ему настоящей Лицензией. При этом права третьих лиц, которым Лицензиат в соответствии с настоящей Лицензией передал экземпляры Программы или права на нее, сохраняются в силе при условии полного соблюдения ими настоящей Лицензии.

5. Лицензиат не обязан присоединяться к настоящей Лицензии, поскольку он ее не подписал. Однако только настоящая Лицензия предоставляет право распространять или модифицировать Программу или произведение, производное от Программы. Подобные действия нарушают действующее законодательство, если они не осуществляются в соответствии с настоящей Лицензией. Если Лицензиат внес изменения или осуществил распространение экземпляров Программы или произведения, производного от Программы, Лицензиат тем самым подтвердил свое присоединение к настоящей Лицензии в целом, включая условия, определяющие порядок воспроизведения, распространения или модификации Программы или произведения, производного от Программы.

6. При распространении экземпляров Программы или произведения, производного от Программы, первоначальный лицензиар автоматически передает приобретателю такого экземпляра право воспроизводить,

распространять и модифицировать Программу в соответствии с условиями настоящей Лицензии. Лицензиат не вправе ограничивать каким-либо способом осуществление приобретателями полученных ими прав. Лицензиат не несет ответственности за несоблюдение условий настоящей Лицензии третьими лицами.

7. Лицензиат не освобождается от исполнения обязательств в соответствии с настоящей Лицензией в случае, если в результате решения суда или заявления о нарушении исключительных прав или в связи с наступлением иных обстоятельств, не связанных непосредственно с нарушением исключительных прав, на Лицензиата на основании решения суда, договора или ином основании возложены обязательства, которые противоречат условиям настоящей Лицензии. В этом случае Лицензиат не вправе распространять экземпляры Программы, если он не может одновременно исполнить условия настоящей Лицензии и возложенные на него указанным выше способом обязательства. Например, если по условиям лицензионного соглашения сублицензиатам не может быть предоставлено право бесплатного распространения экземпляров Программы, которые они приобрели напрямую или через третьих лиц у Лицензиата, то в этом случае Лицензиат обязан отказаться от распространения экземпляров Программы.

Если любое положение настоящего пункта при наступлении конкретных обстоятельств будет признано недействительным или неприменимым, настоящий пункт применяется за исключением такого положения. Настоящий пункт применяется в целом при прекращении вышеуказанных обстоятельств или их отсутствии.

Целью данного пункта не является принуждение Лицензиата к нарушению патента или заявления на иные права собственности или к оспариванию действительности такого заявления. Единственной целью данного пункта является защита неприкосновенности системы распространения свободного программного обеспечения, которая обеспечивается за счет общественного лицензирования. Многие люди внесли свой щедрый вклад в создание большого количества программного обеспечения, которое распространяется через данную систему в надежде на ее длительное и последовательное применение. Лицензиат не вправе вынуждать автора распространять программное обеспечение через данную систему. Право выбора системы распространения программного обеспечения принадлежит исключительно его автору.

Настоящий пункт 7 имеет целью четко определить те цели, которые преследуют все остальные положения настоящей Лицензии.

8. В том случае если распространение и/или использование Программы в отдельных государствах ограничено соглашениями в области патентных или авторских прав, первоначальный правообладатель, распространяющий Программу на условиях настоящей Лицензии, вправе ограничить территорию распространения Программы, указав только те государства, на территории которых допускается распространение Программы без ограничений, обусловленных такими соглашениями. В этом случае такое

указание в отношении территорий определенных государств признается одним из условий настоящей Лицензии.

9. Free Software Foundation может публиковать исправленные и/или новые версии настоящей Стандартной Общественной Лицензии. Такие версии могут быть дополнены различными нормами, регулирующими правоотношения, которые возникли после опубликования предыдущих версий, однако в них будут сохранены основные принципы, закрепленные в настоящей версии.

Каждой версии присваивается свой собственный номер. Если указано, что Программа распространяется в соответствии с определенной версией, т.е. указан ее номер, или любой более поздней версией настоящей Лицензии, Лицензиат вправе присоединиться к любой из этих версий Лицензии, опубликованных Free Software Foundation. Если Программа не содержит такого указания на номер версии Лицензии Лицензиат вправе присоединиться к любой из версий Лицензии, опубликованных когда-либо Free Software Foundation.

10. В случае если Лицензиат намерен включить часть Программы в другое свободное программное обеспечение, которое распространяется на иных условиях, чем в настоящей Лицензии, ему следует испросить письменное разрешение на это у автора программного обеспечения. Разрешение в отношении программного обеспечения, права на которое принадлежат Free Software Foundation, следует испрашивать у Free Software Foundation. В некоторых случаях Free Software Foundation делает исключения. При принятии решения Free Software Foundation будет руководствоваться двумя целями: сохранение статуса свободного для любого произведения, производного от свободного программного обеспечения Free Software Foundation и обеспечение наиболее широкого совместного использования программного обеспечения.

ОТСУТСТВИЕ ГАРАНТИЙНЫХ ОБЯЗАТЕЛЬСТВ

11. ПОСКОЛЬКУ НАСТОЯЩАЯ ПРОГРАММА РАСПРОСТРАНЯЕТСЯ БЕСПЛАТНО, ГАРАНТИИ НА НЕЕ НЕ ПРЕДОСТАВЛЯЮТСЯ В ТОЙ СТЕПЕНИ, В КАКОЙ ЭТО ДОПУСКАЕТСЯ ПРИМЕНИМЫМ ПРАВОМ. НАСТОЯЩАЯ ПРОГРАММА ПОСТАВЛЯЕТСЯ НА УСЛОВИЯХ "КАК ЕСТЬ". ЕСЛИ ИНОЕ НЕ УКАЗАНО В ПИСЬМЕННОЙ ФОРМЕ, АВТОР И/ИЛИ ИНОЙ ПРАВООБЛАДАТЕЛЬ НЕ ПРИНИМАЕТ НА СЕБЯ НИКАКИХ ГАРАНТИЙНЫХ ОБЯЗАТЕЛЬСТВ, КАК ЯВНО ВЫРАЖЕННЫХ, ТАК И ПОДРАЗУМЕВАЕМЫХ, В ОТНОШЕНИИ ПРОГРАММЫ, В ТОМ ЧИСЛЕ ПОДРАЗУМЕВАЕМУЮ ГАРАНТИЮ ТОВАРНОГО СОСТОЯНИЯ ПРИ ПРОДАЖЕ И ПРИГОДНОСТИ ДЛЯ ИСПОЛЬЗОВАНИЯ В КОНКРЕТНЫХ ЦЕЛЯХ, А ТАКЖЕ ЛЮБЫЕ ИНЫЕ ГАРАНТИИ. ВСЕ РИСКИ, СВЯЗАННЫЕ С КАЧЕСТВОМ И ПРОИЗВОДИТЕЛЬНОСТЬЮ ПРОГРАММЫ, НЕСЕТ ЛИЦЕНЗИАТ. В СЛУЧАЕ ЕСЛИ В ПРОГРАММЕ БУДУТ ОБНАРУЖЕНЫ НЕДОСТАТКИ, ВСЕ РАСХОДЫ, СВЯЗАННЫЕ С ТЕХНИЧЕСКИМ ОБСЛУЖИВАНИЕМ, РЕМОНТОМ ИЛИ ИСПРАВЛЕНИЕМ ПРОГРАММЫ, НЕСЕТ ЛИЦЕНЗИАТ.

12. ЕСЛИ ИНОЕ НЕ ПРЕДУСМОТРЕНО ПРИМЕНЯЕМЫМ ПРАВОМ ИЛИ НЕ СОГЛАСОВАНО СТОРОНАМИ В ДОГОВОРЕ В ПИСЬМЕННОЙ ФОРМЕ, АВТОР И/ИЛИ ИНОЙ ПРАВООБЛАДАТЕЛЬ, КОТОРЫЙ МОДИФИЦИРУЕТ И/ИЛИ РАСПРОСТРАНЯЕТ ПРОГРАММУ

НА УСЛОВИЯХ НАСТОЯЩЕЙ ЛИЦЕНЗИИ, НЕ НЕСЕТ ОТВЕТСТВЕННОСТИ ПЕРЕД ЛИЦЕНЗИАТОМ ЗА УБЫТКИ, ВКЛЮЧАЯ ОБЩИЕ, РЕАЛЬНЫЕ, ПРЕДВИДИМЫЕ И КОСВЕННЫЕ УБЫТКИ (В ТОМ ЧИСЛЕ УТРАТУ ИЛИ ИСКАЖЕНИЕ ИНФОРМАЦИИ, УБЫТКИ, ПОНЕСЕННЫЕ ЛИЦЕНЗИАТОМ ИЛИ ТРЕТЬИМИ ЛИЦАМИ, НЕВОЗМОЖНОСТЬ РАБОТЫ ПРОГРАММЫ С ЛЮБОЙ ДРУГОЙ ПРОГРАММОЙ И ИНЫЕ УБЫТКИ). АВТОР И/ИЛИ ИНОЙ ПРАВООБЛАДАТЕЛЬ В СООТВЕТСТВИИ С НАСТОЯЩИМ ПУНКТОМ НЕ НЕСУТ ОТВЕТСТВЕННОСТИ ДАЖЕ В ТОМ СЛУЧАЕ, ЕСЛИ ОНИ БЫЛИ ПРЕДУПРЕЖДЕНЫ О ВОЗМОЖНОСТИ ВОЗНИКНОВЕНИЯ ТАКИХ УБЫТКОВ.

Порядок применения условий Стандартной Общественной Лицензии к созданной вами программе

Если вы создали новую программу и хотите, чтобы она принесла наибольшую пользу обществу, лучший способ достичь этого - сделать вашу программу свободной, когда каждый сможет распространять ее и вносить в нее изменения в соответствии с условиями настоящей Лицензии.

В этих целях Программа должна содержать приведенное ниже уведомление. Наиболее правильным будет поместить его в начале исходного текста каждого файла для максимально ясного указания на то, что гарантии на данную программу не предоставляются. Каждый файл в любом случае должен содержать знак охраны авторского права и пояснение, где можно ознакомиться с полным текстом уведомления.

[одна строка с наименованием Программы и кратким описанием ее назначения]

© имя (наименование) автора или иного правообладателя, год первого опубликования программы

Данная программа является свободным программным обеспечением. Вы вправе распространять ее и/или модифицировать в соответствии с условиями версии 2 либо по вашему выбору с условиями более поздней версии Стандартной Общественной Лицензии GNU, опубликованной Free Software Foundation.

Мы распространяем данную программу в надежде на то, что она будет вам полезной, однако НЕ ПРЕДОСТАВЛЯЕМ НА НЕЕ НИКАКИХ ГАРАНТИЙ, в том числе ГАРАНТИИ ТОВАРНОГО СОСТОЯНИЯ ПРИ ПРОДАЖЕ и ПРИГОДНОСТИ ДЛЯ ИСПОЛЬЗОВАНИЯ В КОНКРЕТНЫХ ЦЕЛЯХ. Для получения более подробной информации ознакомьтесь со Стандартной Общественной Лицензией GNU.

Вместе с данной программой вы должны были получить экземпляр Стандартной Общественной Лицензии GNU. Если вы его не получили, сообщите об этом в Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

Также укажите, как можно связаться с вами по электронной или обычной почте.

Если программа работает в интерактивном режиме, сделайте так, чтобы при запуске в интерактивном режиме выводилось короткое сообщение в

соответствии с образцом:

Gnomovision version 69, © имя автора, год первого опубликования программы

Gnomovision распространяется БЕЗ ВСЯКИХ ГАРАНТИЙ; чтобы ознакомиться с более подробной информацией, наберите "show w". Данная программа является свободным программным обеспечением и вы можете распространять ее в соответствии с условиями Стандартной Общественной Лицензии GNU. Для получения более подробной информации, наберите "show c".

При введении предлагаемых команд "show w" и "show c" на экран должны выводиться соответствующие пункты Стандартной Общественной Лицензии. Не обязательно использовать именно команды "show w" и "show c". В зависимости от функций программы, команды могут вызываться нажатием кнопки мыши или быть добавлены в меню программы.

Если вы создали программу в порядке выполнения служебных обязанностей или служебного задания работодателя либо для образовательного учреждения (школы, ВУЗа и т.д.) в период обучения или работы в данном образовательном учреждении, вам следует получить в случае необходимости письменный отказ от исключительных прав на использование данной программы [3]. Нижеприведенный текст вы можете использовать в качестве образца, заменив соответствующие имена и наименования:

ЗАО "АБВ" настоящим отказывается от всех исключительных прав на использование программы для ЭВМ "Gnomovision", автором которой является Иванов Алексей Петрович, и передает все исключительные права на использование указанной программы ее автору, Иванову Алексею Петровичу.

Подпись руководителя организации, печать, 1 января 2001г.
[Фамилия, Имя, Отчество], Генеральный директор

Стандартная Общественная Лицензия GNU запрещает включать вашу программу в программы, использование которых ограничено их правообладателями. Если ваша программа является библиотекой подпрограмм, вероятно, более полезным будет разрешить связывание программ, использование которых ограничено их правообладателями, с вашей библиотекой. В этом случае вам следует использовать Стандартную Общественную Лицензию GNU для Библиотек вместо настоящей Лицензии.

Понятия GPL.rus

В настоящее время на территории Российской Федерации порядок воспроизведения, распространения и модификации программного обеспечения регулируется Законом РФ "О правовой охране программ для ЭВМ и баз данных" от 23.09.1992г. N 3523-1 и Законом РФ "Об авторском праве и смежных правах" от 09.07.1993г. N 5351-1.

С целью наибольшего соответствия настоящего неофициального перевода Стандартной Общественной Лицензии GNU на русский язык нормам действующего законодательства РФ об авторском праве, ниже приводятся основные понятие, используемые в тексте перевода, и их определения в соответствии с указанными выше Законами РФ:

Программное обеспечение - данное понятие не применяется в указанных Законах, однако оно является наиболее общепринятым при обозначении программ для ЭВМ в переводах лицензионных соглашений, в частности Лицензионных соглашений с конечным пользователем (EULA), на русский язык. В силу этого понятие "Программное обеспечение" используется в тексте перевода для обозначения понятия "программа для ЭВМ". Под программой для ЭВМ Законе РФ понимается объективная форма представления совокупности данных и команд, предназначенных для функционирования электронных вычислительных машин (ЭВМ) и других компьютерных устройств с целью получения определенного результата, включая подготовительные материалы, полученные в ходе разработки программы для ЭВМ, и порождаемые ею аудиовизуальные отображения.

Исключительные права на использование произведения – означает право осуществлять или разрешать следующие действия: воспроизводить произведение (право на воспроизведение); распространять экземпляры произведения любым способом: продавать, сдавать в прокат и так далее (право на распространение); публично показывать произведение (право на публичный показ), переводить произведение (право на перевод); переделывать, аранжировать или другим образом перерабатывать произведение (право на переработку), а также иные права в соответствии с Законом РФ "Об авторском праве и смежных правах".

Исключительные (или имущественные) права на использование программы для ЭВМ – означает исключительное право осуществлять и (или) разрешать осуществление следующих действий: выпуск в свет программы для ЭВМ, воспроизведение программы для ЭВМ (полное или частичное) в любой форме, любыми способами, распространение программы для ЭВМ, модификацию программы для ЭВМ, в том числе перевод программы для ЭВМ с одного языка на другой, а также иное использование в соответствии с Законом РФ "О правовой охране программ для ЭВМ и баз данных".

Воспроизведение Программного Обеспечения – это изготовление одного или более экземпляров Программного обеспечения в любой материальной форме, а также его запись в память ЭВМ.

Модификация (переработка) Программного Обеспечения – любые его изменения, не являющиеся адаптацией.

Распространение Программного Обеспечения – это предоставление доступа для воспроизведения в любой материальной форме Программного Обеспечения, в том числе сетевыми и иными способами, а также путем продажи, проката, сдачи в наем, предоставление займы, включая импорт для любой из этих целей.