

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Космических и информационных технологий
институт
Вычислительная техника
кафедра

УТВЕРЖДАЮ
Заведующий кафедрой
А.И. Легалов
подпись инициалы, фамилия
«__» 2016 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 – «Информатика и вычислительная техника»
код и наименование специальности

Система для изучения иностранного языка с использованием медиа заданий

тема

Пояснительная записка

Руководитель _____
подпись, дата _____ ст. преп. _____
должность, учёная степень _____ И.В.Матковский
инициалы, фамилия

Выпускник _____
подпись, дата _____ А.А.Котарев
инициалы, фамилия

Нормоконтролер _____
подпись, дата _____ В.И. Иванов
инициалы, фамилия

Красноярск 2016

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....
1 Анализ задания на проектирование.....
1.1 Цель задачи.....
1.2 Описание требований.....
1.3 Анализ аналогов.....
1.3.1 Веб-ресурсы.....
1.3.2 Приложения.....
1.4 Выбор типа программы.....
1.5 Вывод.....
2 Проектирование программного продукта.....
2.1 Общая идея.....
2.2 Выбор языка программирования.....
2.3 Выбор базы данных.....
2.4 Проектирование архитектуры программы.....
2.5 Вывод.....
3 Реализация программного продукта.....
3.1 Общая система.....
3.2 База данных.....
3.3 Модули программы.....
3.3.1 Основной модуль.....
3.3.2 Редактор.....
3.3.3 Модуль выбора заданий.....
3.3.4 Модуль заданий.....
3.3.5 Видеоплеер.....
3.3.6 Рабочая зона.....
3.3.7 Редактор.....
3.3.8 Модуль создания заданий.....
3.3.9 Модуль ожидания выполнения создания задания.....

3.3.10 Обработчик времени меток.....
3.3.11 Обработчик текста.....
3.3.12 Обработчик видео.....
3.4 Работа программы.....
3.4.1 Основные функции.....
3.4.2 Автоматическое создание заданий.....
3.5 Руководство пользователя.....
3.5.1 Создание заданий.....
3.6 Выводы.....
ЗАКЛЮЧЕНИЕ.....
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....
ПРИЛОЖЕНИЕ А.....

ВВЕДЕНИЕ

Серьезной проблемой при самостоятельном изучении иностранных языков является поиск учебных материалов, с помощью которых обучение происходит наиболее эффективно. В последнее время все больше людей предпочитают печатным материалам цифровые ресурсы из-за того, что они предоставляют больше возможностей как для тех, кто изучает язык, так и для тех, кто его преподает. В частности, использующиеся в электронных ресурсах мультимедиа, дают возможность развивать навыки говорения и слухового восприятия иностранной речи, что в гораздо меньшей степени реализовано в классических учебных курсах. Возможности взаимодействия между изучающим иностранный язык и программой позволяют гораздо более эффективно понимать грамматику и лексику языка.

Ключевой проблемой при выборе такого метода обучения остается поиск конкретного ресурса, который бы позволил наиболее продуктивно выстроить процесс обучения. Несмотря на большое разнообразие материалов, выбрать подходящий достаточно сложно ввиду того, что большинство не удовлетворяют всем требованиям тех, кто собирается изучать иностранный язык. Кроме того, такие требования сильно зависят от уровня владения языком и многих других параметров, что не всегда в полной мере учитывается составителями таких курсов. Таким образом, необходимо в первую очередь определить критерии, которым должна удовлетворять программа по обучению иностранным языкам, для чего нужно провести сравнение имеющихся на сегодняшний день аналогов.

Для исполнения принятого решения была поставлена цель и сформулированы задачи.

Цель данной выпускной квалификационной работы - разработка системы для изучения иностранных языков с использованием медиа заданий.

1 Анализ задания на проектирование

1.1 Цель задачи

Создать систему для эффективного и современного способа изучения иностранного языка. Система должна иметь возможность создавать и хранить медиа задания, воспроизводить их и иметь возможность вносить изменения. Пользователь должен иметь возможность выбрать задание из списка. Выполнение задания должно быть простым с точки зрения выполняемых действий. Система должна иметь возможность повторять медиа задание, ставить его на паузу и изменять громкость. Задание должно показывать пользователю фразу, состоящую из слов находящихся в случайном порядке. При прохождении задания, пользователю должно показываться, правильно ли он выбрал слово или нет. После верного составления фразы, пользователю должна быть показана фраза в ее изначальном, верном виде.

1.2 Описание требований

Требования к реализуемой программе:

- Возможность воспроизводить медиа файлы
- Автоматическое создание заданий
- Возможность ручного редактирования базы данных
- Возможность прохождения заданий путем составления фраз из предоставленных слов

1.3 Анализ аналогов

В настоящее время все программы по изучению иностранных языков можно разделить на web-ресурсы и приложения. Как первые, так и вторые

имеют свои преимущества и недостатки, поэтому однозначно определить, какой из вариантов наиболее приемлем, невозможно.

1.3.1 Веб-ресурсы

В сети можно найти огромное количество сайтов, так или иначе связанных с этой тематикой, однако выбрать из них подходящий сложно. В частности среди них можно найти те, которые практически полностью повторяют курсы печатных изданий [1, 2]. Такие порталы предоставляют возможность прочтения текстовых учебников и заданий, однако практически не используют мультимедийную составляющую. Такой вариант программы нецелесообразен в связи с тем, что не имеет преимуществ над классическими текстовыми учебниками и не представляет никакого интереса для реализации. В связи с наличием в свободном доступе огромного количества текстовых материалов, в данной работе не будет реализована эта составляющая, а основное внимание будет уделено обучению посредством работы с аудио и видео. Несмотря на то, что существует множество сайтов, предлагающих обучение с помощью аудио и видео, большинство из них реализовано в достаточно примитивном виде. Например, [3, 4, 5], на которых существует возможность просмотра видео уроков.

Более продвинутый функционал предоставляют сайты [6, 7, 8], на которых некоторые задания представлены в виде аудио и видео фрагментов, и пользователю предлагается выбрать один из нескольких вариантов ответа. Такой вариант так же плохо подходит для эффективного понимания разговорной речи и правил английского языка, так как ввиду ограниченного количества предлагаемых вариантов ответа пользователь, не пытается сам понять логику выстраивания предложений либо задания в целом, а лишь подбирает наиболее подходящий вариант. Альтернативой к этому методу построения заданий является способ, где пользователь не имеет ограниченного

выбора между вариантами, что приводит к необходимости самому находить правильный ответ.

Наиболее приближенным по функционалу к данной работе является сайт «ЛингваЛео», который является одним из наиболее популярных и функциональных сайтов в этой области в русскоязычном сегменте, но на бесплатной основе там доступны только самые простые части образовательного процесса [9].

Помимо этого можно отметить сайты, составленные носителями языка, и написанные полностью на английском языке. Такие ресурсы дают возможность людям, которые на достаточном уровне владеют данным языком улучшить свои навыки, однако для начинающих они не подходят. В качестве примера можно привести такие сайты, как [10, 11]. Главным преимуществом такого подхода является то, что такие сайты составлены носителями языка, а значит, они являются более грамотным средством для изучения английского языка. Однако такие ресурсы подходят лишь для тех пользователей, кто в достаточной мере знаком с изучаемым языком и может правильно понять смысл заданий, и, кроме того, такие сайты позволяют лишь улучшить свои навыки, и категорически не подходят для тех, кто только начинает осваивать язык.

1.3.2 Приложения

Аналогично обстоит ситуация и с приложениями. Среди них так же можно выделить те, которые являются программами по запоминанию слов [12, 13, 14]. Некоторые из таких программ направлены на слова в конкретной области английского языка, например [15] для запоминания деловой лексики, или [16] для неправильных глаголов. Некоторые из таких программ подразумевают ассоциативное запоминание [17, 18], однако такие программы, по сути, ничем не отличаются от указанных выше аналогов. Так же можно отметить программы по изучению грамматики [19, 20, 21], которые, несмотря на преимущества над указанными выше аналогами, все же не используют в

полной мере возможности программирования и встраивания мультимедиа в процесс обучения, а просто повторяют структуру текстовых учебных пособий.

К более функциональным программам можно отнести те, которые содержат в себе тесты с вариантами ответа, либо задания в общем виде, которые предполагают какое-либо взаимодействие с пользователем. В качестве примера можно упомянуть [22, 23], либо еще более усовершенствованные варианты, такие как [24, 25], в которых пользователю предлагается самому написать вариант перевода, без выбора из заранее предложенных. Существует функциональная программа [26], которая создана для других задач, но с ее помощью, можно частично выполнить задачу, поставленную в данной квалификационной работе. Программа платная, версия «Student & Teacher Edition» стоит 149\$, что является существенным минусом. При более детальном рассмотрении удалось выяснить, что «Adobe Presenter» имеет очень небольшое количество русскоязычной документации, а в связи с огромным функционалом, свободное использование без обучающих материалов - крайне затруднительно. Большую часть функционала можно реализовать в «Adobe Presenter», но БД необходимо будет составлять вручную, слайд за слайдом. Сделав выводы, можно заключить следующее: «Adobe Presenter» – платный (минимум 149\$). Он не позволяет сделать задуманный функционал в точности, но повторить его можно, без автоматического составления заданий и немного в неудобном и менее функциональном виде.

На основе анализа имеющихся приложений и web-ресурсов можно сделать вывод, что последние, в настоящее время, развиваются более быстрыми темпами и значительно превосходят по функционалу скачиваемые программы в целом. Однако они требуют постоянного доступа в интернет, что не является необходимым в том случае, если программу можно заранее скачать на персональный компьютер. Кроме того, если web-ресурс осуществляет обучения посредством видео, для доступа к нему требуется высокоскоростное интернет соединение. В то же время, отличий в возможностях, которые можно реализовать в web-ресурсах и программах, практически нет, однако у

приложений существует преимущество в возможностях импортирования обучающих материалов, связанное с тем, что программа физически расположена на жестком диске пользователя и не требует соответствующих прав доступа к базе данных. В связи с этим в данной работе выбор между web-ресурсом и приложением делается в пользу последнего, ввиду указанных выше факторов и, в первую очередь, того, что такой выбор предпочтительнее для академической формы обучения, где преподаватель имеет возможность добавлять конкретные задания в программу.

1.4 Выбор типа программы

Таким образом, на основе произведенного анализа существующих в настоящее время ресурсов по изучению иностранного языка можно сделать вывод, что все доступные аналоги в той или иной мере содержат в себе нижеперечисленные свойства, которые отрицательно влияют на эффективность данных методов:

- Обучение на платной основе
- Не имеют возможность создания своих заданий
- Не имеют возможности импортирования сторонних заданий
- Нет персонализации и разделения по темам
- Тестирование с выбором из заранее предоставленных вариантов ответа
- Наиболее продвинутым функционалом, в большинстве своем, обладают только web-ресурсы

Целью данной работы является создание программы, которая будет в полной мере использовать мультимедийные возможности для изучения английского языка и будет удовлетворять необходимым требованиям, сформированным выше. Необходимость в этом обеспечена тем, что в настоящее время все существующие аналоги либо платные, либо реализованы в достаточно примитивном виде. Наиболее приемлемый, в данном контексте, вариант программы представляет собой задание, в котором пользователь

просматривает небольшой видеофрагмент на английском языке. Затем необходимо составить проговариваемую, в данном фрагменте, фразу из ряда предоставленных слов, аналогично пазлу. При правильном выполнении задания выстраивании подходящих слов в нужном порядке и соответствии произнесенной фразе - пользователь переходит к следующему фрагменту, если таковой имеется. Преимуществом такого способа является то, что хоть пользователю и предоставляются некоторые варианты, они не являются заранее подготовленными ответами, а лишь возможными частями их, что не позволяет выбирать наиболее подходящие, а создает необходимость логического согласования и грамматически правильного построения предложений самостоятельно. За счет того, что в данной работе разрабатывается приложение, значительно упрощена возможность персонализации заданий. Любой пользователь имеет возможность самостоятельно добавить видеофрагмент и составить к нему задание, заранее определив при этом тематику и уровень сложности, что дает возможность использования программы при изучении иностранных языков в учебных заведениях. Одним из основных плюсов является то, что данная программа является бесплатной в рамках выпускной квалификационной работы.

1.5 Вывод

В рамках анализа задания на проектирование были сформированы требования к программе, проведен анализ аналогов, определен тип реализуемой программы.

В ходе исследования актуальности и новизны выполняемой работы были сделаны выводы, что в данный момент аналогов полностью удовлетворяющих вышеописанному нет.

2 Проектирование программного продукта

2.1 Общая идея

Для реализации проекта были рассмотрены возможные пути, которые бы позволяли в полной мере выполнить поставленную задачу. В целом было принято решение о создании прикладной программы, которая будет использовать заранее заготовленные видео файлы, анализировать заранее подготовленный текст и составлять из него задание с записью необходимой информации в базу данных. Вид отображения списка видео файлов – список, с возможностью выбора конкретного видео для перехода к просмотру и дальнейшему выполнению задания. Было решено, что структура выполнения задания, будет представлять собой набор меток, при нажатии на которые, осуществляется проверка на соответствие правильному ответу. В результате, как только все метки будут выбраны верно, задание по видеофрагменту будет решено и будет осуществлен переход к следующему.

2.2 Выбор языка программирования

Поставленную цель было решено реализовать на языке C#.

C# отличается обширным набором стандартных библиотек, которые обеспечивают удобство для разработки оконных приложений. Является полностью объектно-ориентированном языком и включает инкапсуляцию, наследование и полиморфизм.

Решающим фактором выбора языка стало наличие опыта разработки программного обеспечения на C#.

2.3 Выбор базы данных

База данных была выбрана SQL Server 2012 Express LocalDB, так как имелся опыт разработки в этой среде. Express LocalDB представляет собой режим выполнения сервера SQL, при установке которого, устанавливается минимальный набор файлов для функционирования сервера. После установки к базе данных можно подключиться с помощью специальной строки подключения. При разработке программы учитывалось, что необходима легкая, быстрая, настроенная база данных, с быстрым доступом, который обеспечит эффективную работу при чтении/записи данных в уже существующие таблицы базы данных.

2.4 Проектирование архитектуры программы

При проектировании архитектуры программного обеспечения были учтены требования к заданию и на основе их разработана архитектура.

Приложение включает в себя базу данных.

База данных должна содержать данные о видео задании и тексте. Данные о видео задании должны содержать уникальный идентификатор, имя видеофайла и всех видео фрагментов, тип видео, который определяет является ли видео основным или видео является видео фрагментов, название папки с видео заданием. Данные о тексте задания так же должны содержать уникальный идентификатор, нумерацию слов для отображения, слова из предложений, порядковый номер слова во фразе, а так же уникальный идентификатор из данных о видео, который необходим для сопоставления.

Приложение так же включает в себя редактор базы данных.

Редактор базы данных позволяет просмотр данных вручную, а так же редактирование таблиц базы данных с возможностью автоматического обновления таблиц.

Следующий элемент приложения – модуль выбора задания.

Модуль выбора заданий позволяет осуществлять выбор задания из списка и переход к его выполнению.

Следующий элемент приложения – модуль задания.

Модуль задания связан с модулем выбора задания и на основании работы модуля выбора задания позволяет формировать визуальное отображение задания и его выполнение.

Последний элемент приложения – модуль создания заданий.

Модуль создания заданий позволяет в автоматическом режиме на основании имеющегося видео, файла с временными отрезками и текста создавать задание, которое записывается в базу данных и в дальнейшем может использоваться повторно. В результате работы модуля создания заданий, основное видео фрагментируется на отрезки, которые обозначены в файле с временными отрезками. Видео фрагменты помещаются в отдельную папку и хранятся там для повторного использования (рисунок 1).

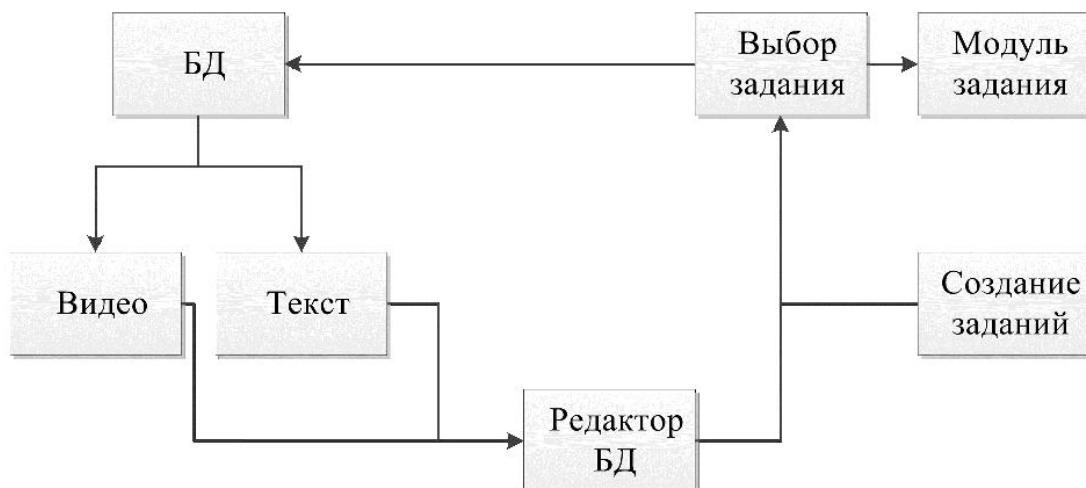


Рисунок 1 – Схема архитектуры приложения

2.5 Вывод

При проектировании программного продукта была определена общая идея программы и ее архитектура. В результате исследований и анализа были выбраны оптимальные средства для реализации программного продукта, которые бы в полной мере выполняли поставленную задачу.

3 Реализация программного продукта

3.1 Общая система

Общая система приложения включает в себя основной модуль, редактор, подразделяющийся на редактор текста и редактор видео, базу данных, модуль выбора задания, модуль задания, видеоплеер, рабочую зону, модуль создания задания, обработчик времени меток, обработчик текста, обработчик видео (рисунок 2).

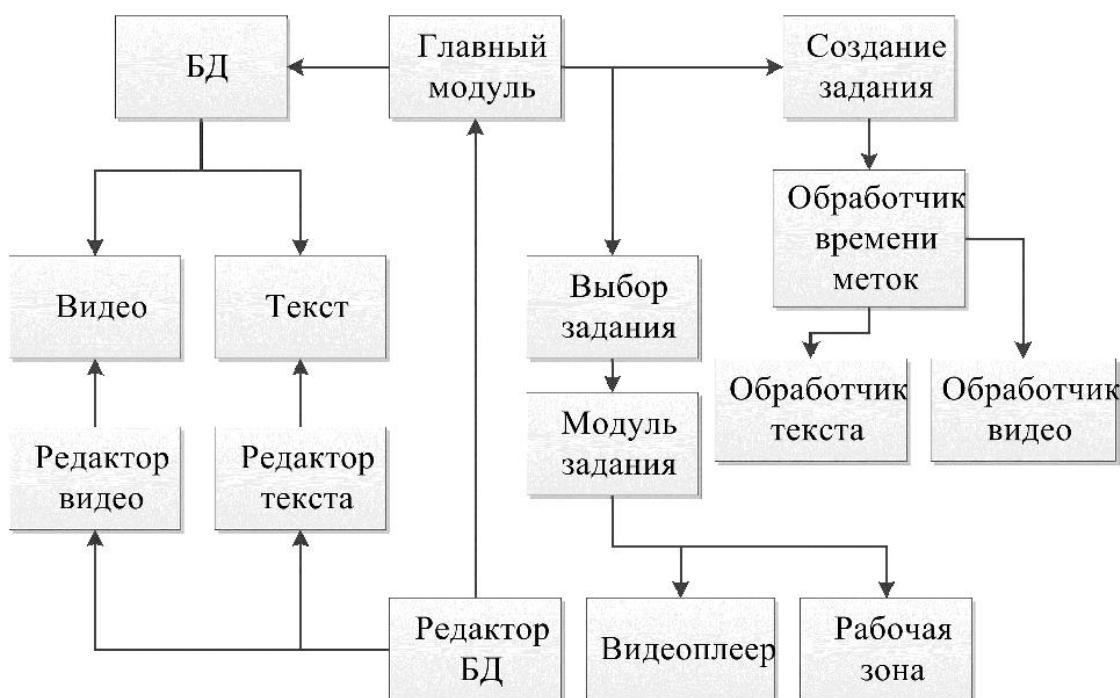


Рисунок 2 – Схема программного продукта

3.2 База данных

База данных реализуемого программного продукта состоит из двух таблиц.

Первая таблица «Video» имеет поля:

- «Id»
- «Video_name»
- «Video_type»
- «Video_path».

В поле «Id» таблицы «Video» вписывается название видео задания для основного видео и название видео задания с номером для видео фрагментов. В поле «Video_name» таблицы «Video» вписываются имена видео файлов. В поле «Video_type» таблицы «Video» вписывается параметр «MAIN» для главного видео и параметр «SECONDARY» для видео фрагментов. В поле «Video_path» таблицы «Video» вписываются названия папки с видео заданием.

Вторая таблица «Label» имеет поля:

- «Id»
- «Video_labelnumber»
- «Video_words»
- «Video_wordnumber»
- «Video_Id».

В поле «Id» таблицы «Label» вписывается название видео и его расширение для основного видео и название видео с инкрементом для видео фрагментов. В поле «Video_labelnumber» вписываются номера меток, которые будут активны для каждого видео фрагмента. В поле «Video_words» таблицы «Label» вписываются слова из фраз для видео фрагментов. В поле «Video_wordnumber» таблицы «Label» вписываются позиции слов из фраз для видео фрагментов. В поле «Video_Id» таблицы «Label» вписываются данные из поля «Id» таблицы «Video».

3.3 Модули программы

3.3.1 Основной модуль

Основной модуль является узловой точкой для модуля редактора, модуля выбора задания, модуля задания, видеоплеера, рабочей зоны, модуля создания задания, обработчика времени меток, обработчика текста, обработчика видео. Основной модуль изображен на рисунке 3.

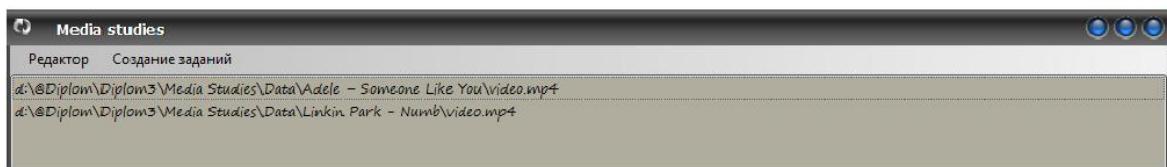


Рисунок 3 – Основной модуль

3.3.2 Редактор

В главном меню находится пункт «Редактор» показанный на рисунке 4. Он содержит подпункт «Видео», который вызывает редактор видео и содержит подпункт «Текст», который вызывает редактор текста. Редактор позволяет отображать данные из базы данных, а так же производить их редактирование.

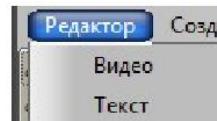


Рисунок 4 – Пункт «Редактор» главного меню

3.3.3 Модуль выбора заданий

Модуль выбора заданий осуществляет выбор заданий из списка доступных заданий, находящихся в папке, после выбора задания он передает информацию о выбранном задании в модуль заданий.

3.3.4 Модуль заданий

После получения информации о выбранном задании от модуля выбора заданий, модуль заданий осуществляет загрузку данных о задании из базы

данных, подготавливает загруженные данные и осуществляет их передачу в видеоплеер и рабочую зону.

3.3.5 Видеоплеер

Видеоплеер предназначен для воспроизведения медиа файлов с возможностью проигрывания, паузы, остановки, перемотки и изменения громкости звука.

3.3.6 Рабочая зона

Рабочая зона представляет собой пространство для начала выполнения задания, а так же отображения слов из фразы и возможности выбора ответов с переходом к следующему заданию.

В рабочей зоне имеется кнопка «Приступить к выполнению задания», после просмотра ролика, нажав на кнопку, будет осуществлен переход к выполнению задания (рисунок 5).

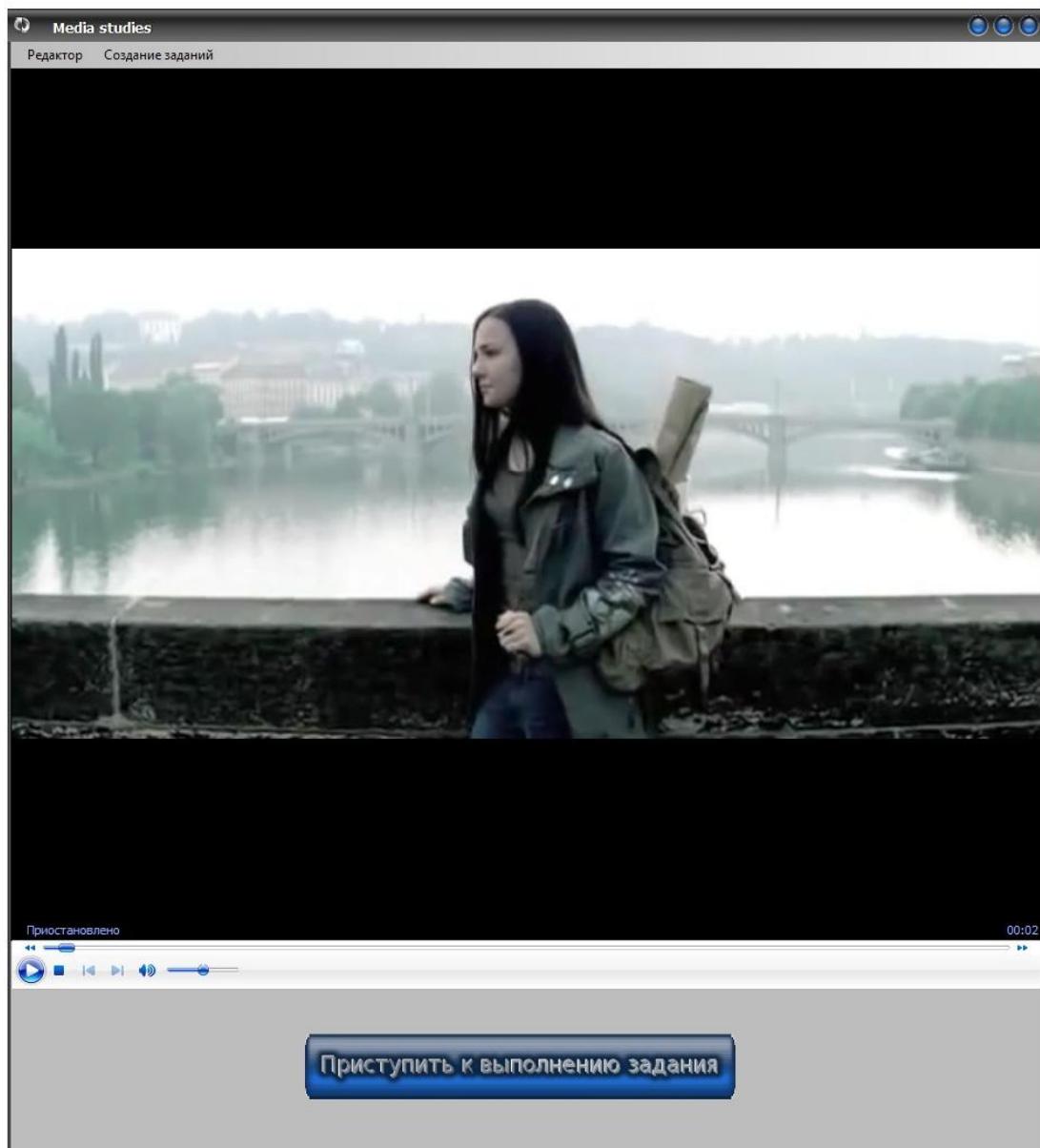


Рисунок 5 – Главная форма, после выбора задания

В рабочей зоне становятся видимыми элементы Labels, их количество зависит от количества слов во фразе, по архитектуре программы их может быть от двух до десяти. Пользователю необходимо нажимать на слова, чтобы составить верную фразу и перейти к следующему видео фрагменту. Фраза, которая должна быть составлена, воспроизводится в виде аудио непосредственно в самом видео фрагменте. При нажатии на слово, которое соответствует, верной позиции во фразе слово будет выделено зеленым цветом,

если пользователь при составлении фразы допускает ошибку в выборе верного порядка слов, слово будет выделено красным цветом означающее, что пользователь допустил ошибку (рисунок 6).

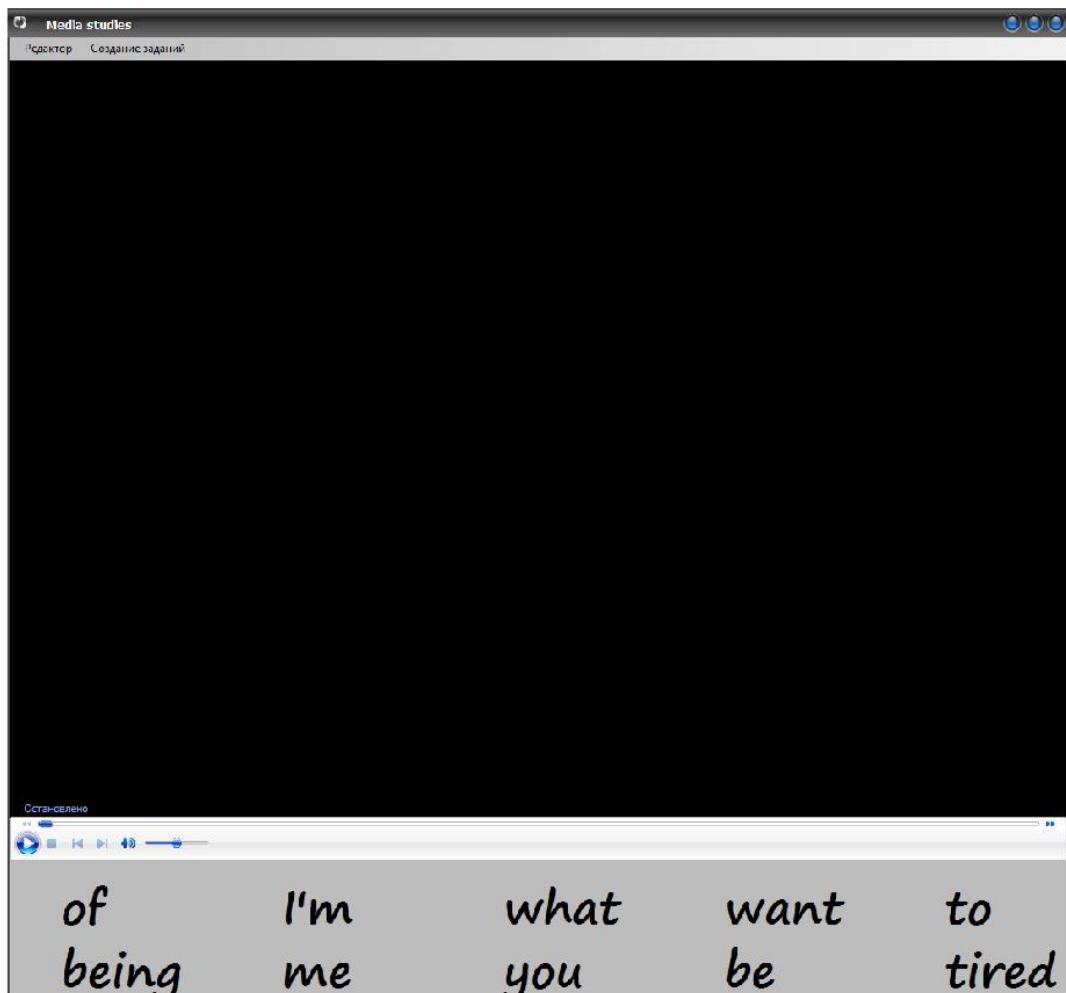


Рисунок 6 – Главная форма, после нажатия на кнопку «Приступить к выполнению задания»

3.3.7 Редактор

Редактор видео и редактор текста содержат таблицу для отображения данных, которая обеспечивает отображение данных из таблицы Видео и Текст базы данных, возможность редактирования, добавления и удаления данных (рисунок 7, 8).

VideoGridAdd

Id	Video_name	Video_type	Video_path
Numb	video.mp4	MAIN	Linkin Park - Numb
Numb_00	00.mp4	SECONDARY	Linkin Park - Numb
Numb_01	01.mp4	SECONDARY	Linkin Park - Numb
Numb_02	02.mp4	SECONDARY	Linkin Park - Numb
Numb_03	03.mp4	SECONDARY	Linkin Park - Numb
Numb_04	04.mp4	SECONDARY	Linkin Park - Numb
Numb_05	05.mp4	SECONDARY	Linkin Park - Numb
Numb_06	06.mp4	SECONDARY	Linkin Park - Numb
Numb_07	07.mp4	SECONDARY	Linkin Park - Numb
Numb_08	08.mp4	SECONDARY	Linkin Park - Numb
Numb_09	09.mp4	SECONDARY	Linkin Park - Numb
Numb_10	10.mp4	SECONDARY	Linkin Park - Numb

Рисунок 7 – Редактор базы данных видео заданий

LabelGridAdd

Id	Video_labelnumber	Video_words	Video_wordnumber	Video_id
Numb.mp4	0	0	0	Numb
Numb_00	1	I'm	1	Numb_00
Numb_00_3	2	tired	2	Numb_00
Numb_00_2	3	of	3	Numb_00
Numb_00_3	4	being	4	Numb_00
Numb_00_4	5	what	5	Numb_00
Numb_00_5	6	you	6	Numb_00
Numb_00_6	7	want	7	Numb_00
Numb_00_7	8	me	8	Numb_00
Numb_00_8	9	to	9	Numb_00
Numb_00_9	10	be	10	Numb_00

Рисунок 8 – Редактор базы данных текстовых заданий

На рисунке 9 показана панель управления DataGridView, она обеспечивает удобное и функциональное управление большой таблицей, которая содержит много данных.



Рисунок 9 – Панель управления DataGridView

Кнопка «Переместить в начало» обеспечивает возможность мгновенного перехода в начало элемента DataGridView (рисунок 10).



Рисунок 10 – Кнопка «Переместить в начало» панели управления DataGridView

Кнопка «Переместить назад» обеспечивает возможность перехода на одну строку назад в элементе DataGridView (рисунок 11).



Рисунок 11 – Кнопка «Переместить назад» панели управления DataGridView

Данное поле слева отображает «Текущее положение», т.е. выбранную строку в элементе DataGridView, поле справа «Общее число элементов» отображает общее количество строк в элементе DataGridView (рисунок 12).



Рисунок 12 – Текущее положение от общего числа элементов элемента

DataGridView

Кнопка «Переместить вперед» обеспечивает возможность перехода на одну строку вперед в элементе DataGridView (рисунок 13).



Рисунок 13 – Кнопка «Переместить вперед» панели управления DataGridView

Кнопка «Переместить в конец» обеспечивает возможность мгновенного перехода в конец элемента DataGridView (рисунок 14).



Рисунок 14 – Кнопка «Переместить в конец» панели управления DataGridView

Кнопка «Добавить» обеспечивает возможность добавления новой строки в элемент DataGridView (рисунок 15).



Рисунок 15 – Кнопка «Добавить» панели управления DataGridView

Кнопка «Удалить» обеспечивает возможность удаления строки из элемента DataGridView (рисунок 16).



Рисунок 16 – Кнопка «Удалить» панели управления DataGridView

Кнопка «Сохранить данные» обеспечивает возможность сохранения всех данных из элемента DataGridView в таблицу Video и Label существующей базы данных (рисунок 17).



Рисунок 17 – Кнопка «Сохранить данные» панели управления DataGridView

3.3.8 Модуль создания заданий

Модуль создания заданий осуществляет автоматическое создание заданий и запись их в базу данных, на основе заранее подготовленного видео, текста и тайм кодов для видеофрагментов.

На рисунке 18 показан пункт меню для выбора модуля создания заданий.

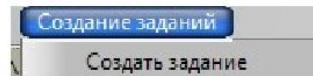


Рисунок 18 – Пункт «Создание заданий» главного меню

3.3.9 Модуль ожидания выполнения создания задания

Данный модуль предназначен для графического отображения выполнения автоматического создания задания (рисунок 19).



Рисунок 19 – Форма Waiter

Так как, создание задания с большим количеством видео фрагментов и большим количеством текста, занимает некоторое время, данный модуль уведомляет пользователя о том, что программа работает и необходимо немного

подождать, после того, как задание будет создано, данная форма будет закрыта и будет осуществлен переход к главной форме.

3.3.10 Обработчик времени меток

Обработчик времени меток позволяет считать данные из файла и на основе временных кодов фрагментировать основное видео на ряд видео фрагментов необходимых для задания.

3.3.11 Обработчик текста

Обработчик текста позволяет считать данные из файла и на основе считанных данных в автоматическом режиме заполнить базу данных в необходимой структуре, таким образом, создав часть задания, отвечающую за слова и фразы, которые необходимо составить.

3.3.12 Обработчик видео

Обработчик видео позволяет считать данные из файла и на основе считанных данных в автоматическом режиме заполнить базу данных в необходимой структуре, таким образом, создав часть задания, отвечающую за месторасположение основного видео и видео фрагментов, а так же их названий.

3.4 Работа программы

3.4.1 Основные функции

Инициализируется форма и при загрузке формы выполняется функция выбора задания, в которой предварительно очищается список заданий (чтобы избежать дописывание при повторном вызове), скрываются неиспользуемые в

данный момент элементы, в частности ряд меток, панелей и видеоплеер. Создается массив строк, в который записываются все пути до видеофайлов из папки Data данной программы, а в цикле «перебираются» и записываются в список, через свойство добавление элементов.

После проделанной работы, в главном окне имеется список видео заданий, в котором находится список из видеофайлов предварительно подготовленных и находящихся в папке Data (рисунок 20).

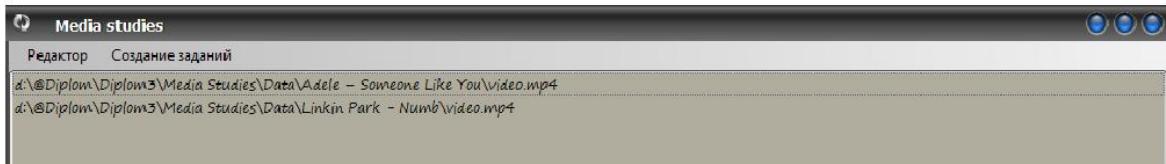


Рисунок 20 – Главное окно программы

Затем, когда пользователь выбирает необходимый ему элемент (видеофайл) из списка, в переменную записываются данные о выбранном элементе списка и переменная используется, чтобы присвоить значение ссылки видео файла для медиа плеера, затем используется свойство инициирующее проигрывание файла, чтобы начать воспроизведение файла.

Скрывается ненужный в данный момент список, а так же устанавливается свойство видимости нужным в данный момент элементам, а именно панелей, поверх которых находится медиа плеер и сам медиа плеер, в последнюю очередь устанавливается свойство медиа плееру, чтобы изображение было на весь медиа плеер.

После вышеописанных действий, пользователю отображается WindowsMediaPlayer с выбранным видеофайлом из списка и начато проигрывание видеофайла, чтобы пользователь мог ознакомиться с видеоклипом, для дальнейшего выполнения задания.

После просмотра, на нижней панели, пользователю необходимо нажать на кнопку «Приступить к выполнению видео задания», после нажатия на

кнопку происходит событие нажатия на кнопку, в результате которого выполняется предварительная очистка списков.

Устанавливается свойство, которое назначает «пустой» цвет для каждой из меток. Устанавливается свойство кнопки, которое скрывает ее, так как на данном этапе она больше не требуется, и устанавливается свойство видимости всех меток, которое обеспечивает их дальнейшее отображение.

Создаются два экземпляра видео форма и текстовая форма. Инкрементируется переменная, которая отвечает за номер задания

Далее с помощью операций над строковыми переменными определяется название папки с видео заданием.

Выполняется функция видео формы, в которой во временную переменную помещается значение переменной хранящей номер задания.

Объявляется переменная, в которой будет храниться количество строк таблицы для отображения данных видео формы.

Далее в цикле определяется строка в таблице для отображения данных, имеющая параметр «MAIN» для выбранного видео задания и запоминается номер этой строки, записав значение в соответствующую переменную.

Далее выполняется еще один цикл, в котором определяется ID найденной строки с видео заданием и путь до первого видео фрагмента для выбранного ранее видео задания, а так же переменной, которая отвечает за окончание задания присваивается единица, что означает, что определение нужных параметров окончено и происходит выход из функции.

После выхода из функции имеется условие, при котором, если это повторный запуск, необходимо снова выполнить функцию выбора задания и установить параметр в значение 0, что будет означать, что запуска не было и можно выполнить задание снова.

Так как, выполнение запущено в первый раз, функция выбора задания выполнена не будет и будет осуществлен переход к следующему действию, а именно установке свойства назначения ссылки для медиа плеера используя путь до папки Data в папке с данной программой и название первого видео

фрагмента, который был получен при вызове функции нажатия кнопки видео формы.

После выполняется объявление переменной, которой присваивается значение количества строк в таблицу для отображения данных текстовой формы.

Выполняется цикл, в котором проверяется условие, если ID видео задания из текстовой таблицы для отображения данных совпадает с ID задания, которое выполняется в данный момент, то в ранее созданные списки записываются данные, в один список записывается слово, а в другой список записывается позиция слова во фразе.

На рисунке 21 можно увидеть, что в первую итерацию, будут взяты слова, начиная с ID «Numb_00» и их позиции, затем будут выполнены дальнейшие действия и после ответа на эту фразу, будут взяты слова, начиная с ID «Numb_01», пока задание не будет завершено.



The screenshot shows a Microsoft Access form window titled "LabelGridAdd". The window has a toolbar with buttons for back, forward, search, and other operations. The main area contains a grid table with the following columns: Id, Video_labelnumber, Video_words, Video_wordnumber, and Video_id. The data in the grid is as follows:

	Id	Video_labelnumber	Video_words	Video_wordnumber	Video_id
▷	Numb.mp4	0	0	0	Numb
	Numb_00	3	I'm	3	Numb_00
	Numb_00_3	2	tired	2	Numb_00
	Numb_00_2	3	of	3	Numb_00
	Numb_00_3	4	being	4	Numb_00
	Numb_00_4	5	what	5	Numb_00
	Numb_00_5	6	you	6	Numb_00
	Numb_00_6	7	want	7	Numb_00
	Numb_00_7	8	me	8	Numb_00
	Numb_00_8	9	to	9	Numb_00
	Numb_00_9	30	be	30	Numb_00

Рисунок 21 – Фрагмент формы LabelGrid

На рисунке 22 можно увидеть, что выполнение будет продолжаться, пока не будет встречено следующее задание в БД.

	Id	Video_labelnumber	Video_words	Video_wordnumber	Video_id
	Numb_33_8	9	there	9	Numb_33
	Numb_34	3	I'm	3	Numb_34
	Numb_34_3	2	tired	2	Numb_34
	Numb_34_2	3	of	3	Numb_34
	Numb_34_3	4	being	4	Numb_34
	Numb_34_4	5	what	5	Numb_34
	Numb_34_5	6	you	6	Numb_34
	Numb_34_6	7	want	7	Numb_34
	Numb_34_7	8	me	8	Numb_34
	Numb_34_8	9	to	9	Numb_34
	Numb_34_9	10	be	10	Numb_34
	Someone like you.mp4	0	0	0	Someone like you

Рисунок 22 – Фрагмент формы LabelGrid

После заполнения списков объявляется переменная равная нулю, которая будет отвечать за количество необходимых меток (количество необходимых меток соответствует количеству слов во фразе).

Далее выполняется цикл «Рандомизации» слов фразы по меткам. Расставляются слова по элементам в случайном порядке. Таким образом, каждую итерацию цикла, имеется некая случайная позиция слова, далее идут два условия, первое условие происходит тогда, когда происходит первая итерация цикла, позиция записывается в список и переменная, отвечающая, за количество меток увеличивается на единицу.

Иначе, проверяется список, не содержит ли он уже записанное число, если не содержит, записывается позиция и переменная, отвечающая, за количество меток увеличивается на единицу, иначе переход к следующей итерации цикла.

После того, как все позиции заполнены, выполняется цикл по массиву «контролов», который в зависимости от количества слов делает видимыми определенное количество меток, а так же записываются уже сгенерированные позиции слов из списка.

Как видно на рисунке 23, на нижней панели отображены элементы labels, в которые выводятся слова из фразы, в случайном порядке.

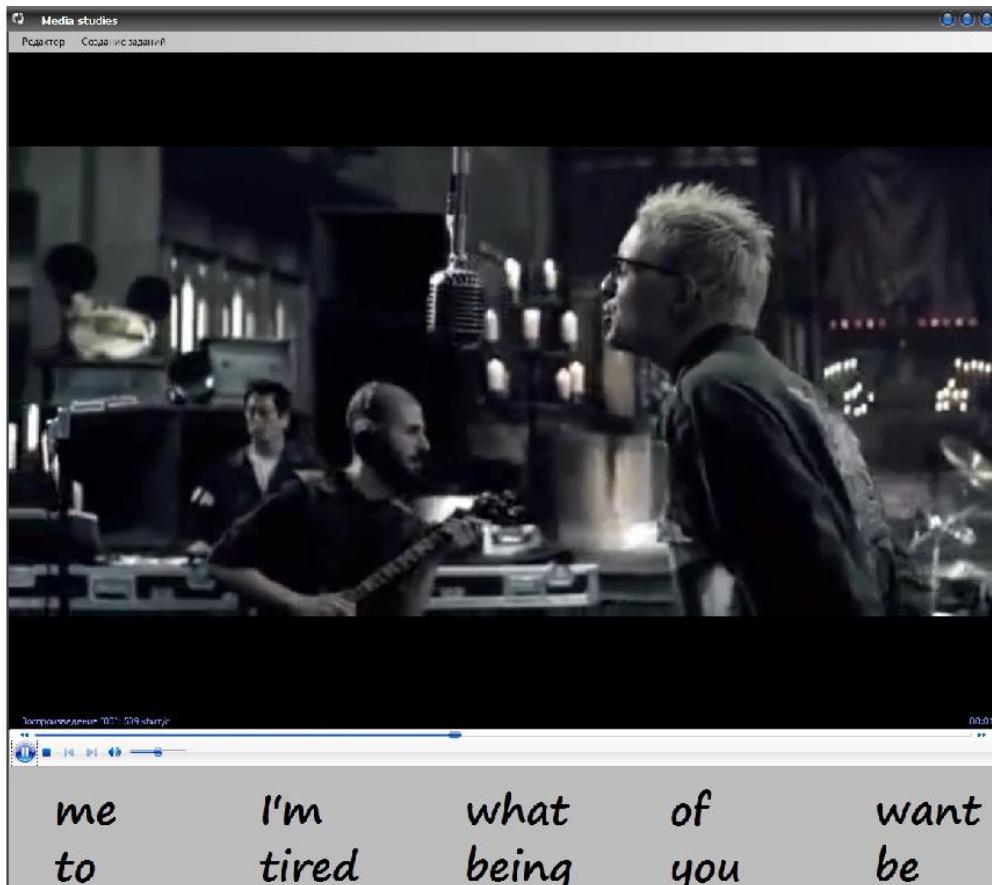


Рисунок 23 – Выполнение задания в главной форме

После нажатия на какую-либо метку происходит обработка события. В обработке события идет цикл с многочисленными проверками, на то, какая именно по счету метка была нажата, если, например, метка5 (слово want), было нажато первым, то проверяется цвет метки5, если он не зеленый (что соответствует правильному ответу), тогда проверяется, соответствует ли текст данной метки первому слову фразы. После проверки, если условие выполняется, проверяется, установлен ли счетчик неправильного слова и цвет метки не зеленый, тогда у метки убирается цвет.

Далее переменная, отвечающая за количество верных слов увеличивается на единицу, а счетчик неправильного слова устанавливается в 0, так как было найдено верное слово и метке устанавливается свойство зеленого цвета.

Если условие, не выполнено и в метке5 не содержится первое слово во фразе, то счетчик неправильного слова устанавливается в 1, по аналогии проверяется, если цвет не зеленый, то у метки убирается цвет (чтобы не было два неправильных слова подряд) и затем устанавливается конкретной метке, в данном случае метка5, свойство красного цвета.

Аналогично проверяются условия, если в этой метке, слово стоит не первым, только к условию добавляется проверка количества верных слов, что позволяет понять, какое именно слово по счету проверяется.

В конце всех проверок в метке, проверяется, окончена ли фраза, для этого переменной, отвечающей за количество верных слов присваивается значение 0 и проверяется, если данная метка видима, а так же, если метка имеет зеленый цвет, переменная, отвечающая за количество верных слов увеличивается на единицу и проверяется значение этой переменной на условие (переменная верных слов \geq количество слов), если условие выполняется, то фраза завершена, так как количество зеленых меток больше или равно количеству всех меток. И происходит вызов функции для обработки следующей фразы.

3.4.2 Автоматическое создание заданий

В главной форме, в главном меню находится кнопка «Создать задание», при нажатии на которую, срабатывает обработчик событий, который открывает диалоговое окно позволяющее пользователю выбрать файл с рядом параметров (папка инициализации, фильтр и т.д.), затем, если файл был выбран удачно, то объявляются строковые переменные текущая папка, текущее название видео, путь до текущего txt файла, итоговое название текущего txt файла и числовая переменная переменная хранящая длину имени txt файла.

После того, как объявлены все переменные, переменной текущей папки присваивается значение «Имя папки видеофайла».

Следующим шагом, переменной текущего названия видео присваивается значение «Имя видеофайла.расширение».

Далее вычисляется, количество символов в имени видеофайла и следующим шагом записывается в переменную путь до текущего txt файла, путь до папки с видео файлом исключив из него название видео файла, путем вычитания количества символов имени видео файла.

В итоге, в переменную итоговое название текущего txt файла записывается полный путь до txt файла с заданием.

Далее создаются два экземпляра видео форма и текстовая форма, построчно считывается заранее подготовленный текстовый файл с заданием в массив строк.

В набор данных видео формы добавляется строка вида – название видео клипа, название видео файла, параметр «MAIN» (который необходим по придуманному алгоритму работоспособности программы), название папки, в которой хранится видео файл.

Аналогичным образом в набор данных текстовой формы добавляется строка вида – название видео клипа + .mp4, 0, 0, 0, название видео клипа.

Объявляется строковая переменная и присваивается ей значение количества файлов фрагментов видео задания.

Создается экземпляр формы ожидания. Далее выполняется цикл на заполнение базы данных. Сначала объявляется массив строк, которому присваивается значение отдельных слов, в итоге в массиве строк хранятся отдельные слова.

Затем следует условие, если количество итераций цикла меньше десяти, то добавляется строка о файле в базу данных. Если количество итераций больше девяти и меньше значения общего количества видео фрагментов хранящегося в переменной, то в базу данных добавляется строка о файле.

Следом идет вложенный цикл, который отвечает за запись строк в текстовую таблицу.

Если итерация цикла первая и количество итераций прошло меньше десяти, то записывается строка о слове. Если количество итераций больше

девяти и меньше значения общего количества видео фрагментов хранящегося в переменной, то в базу данных добавляется строка о слове.

Если итерация цикла не первая и количество итераций прошло меньше десяти, то записывается строка о слове. Если количество итераций больше девяти и меньше значения общего количества видео фрагментов хранящегося в переменной, то в базу данных добавляется строка о слове.

Далее используется код, который сохраняет новую информацию в таблицу данных и закрывает экземпляр формы ожидания.

3.5 Руководство пользователя

3.5.1 Создание заданий

Для создания медиа задания необходимо скачать интересующее видео и поместить его в папку «Data», предварительно создав папку с названием видео задания, пример: «Data/Linkin Park – Numb». Далее по тому же пути необходимо добавить текстовый файл с текстом задания, вид файла: первая строка – название задания, последующие строки – фразы для отдельных видео фрагментов. Затем необходимо добавить текстовый файл с временными кодами для нарезки видео фрагментов, вид файла: первая строка – начальный момент с которого производить фрагментацию, вторая строка – какова будет длительность видеофрагмента, третья строка – пробел. Таким образом, файл должен быть наполнен для каждого видео фрагмента. После подготовки видео и двух текстовых файлов, в главном меню необходимо нажать на кнопку «Создать задание» показанную на рисунке 24.

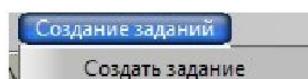


Рисунок 24 – Пункт «Создание заданий» главного меню

Далее в появившемся окне выбора файла, необходимо выбрать основной видео файл и ждать пока задание будет создано.

Чтобы выполнить задание, необходимо выбрать его из списка заданий в главном окне, прослушать появившийся медиа файл и нажать на кнопку «Приступить к заданию». Затем выполнять задание слушая видеофрагменты и нажимая на слова в верном порядке.

3.6 Выводы

Реализовано приложение, обладающее следующим функционалом

- Возможность автоматического создания заданий и хранение их в базе данных
- Возможность прослушивания и просмотра медиа файлов
- Возможность просмотра и/или редактирования базы данных
- Возможность быстрого и интуитивного прохождения заданий путем пошагового прослушивания медиа заданий и составления предложений из слов

Проведены тесты на соответствие установленным требованиям.

ЗАКЛЮЧЕНИЕ

В процессе выполнения бакалаврской работы были изучены основные характеристики различных сред программирования. По завершению анализа готовых сред, были сформированы требования к разрабатываемому продукту.

Для разработки интегрированной среды был выбран оптимальный для данного случая набор инструментов, в который вошли: язык программирования C# и набор библиотек .Net Framework. Это позволило в полной мере реализовать сформулированные требования к разрабатываемому продукту.

В процессе разработки были изучены ранее незнакомые возможности библиотек .Net, спроектирована архитектура разрабатываемого приложения и выполнена реализация приложения в виде программы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Сайт для изучения английского языка [Электронный ресурс] : грамматика английского языка. – Режим доступа: <http://catchenglish.ru/>
2. Сайт для изучения английского языка [Электронный ресурс] : грамматика английского языка. – Режим доступа: <http://www.homeenglish.ru/>
3. Сайт для изучения иностранных языков с нуля [Электронный ресурс] : изучение иностранных языков. – Режим доступа: <http://lingust.ru/>
4. Сайт для изучения английского языка [Электронный ресурс] : английский язык по красному Мёрфи. – Режим доступа: <http://english03.ru/>
5. Сайт для изучения английского языка [Электронный ресурс] : Английский язык для начинающих и не только. – Режим доступа: <http://list-english.ru/>
6. Сайт для изучения английского языка [Электронный ресурс] : The United Kingdom's international organization for cultural relations and educational opportunities. – Режим доступа: <http://britishcouncil.org>
7. Программа для проверки знаний [Электронный ресурс] : Net-testing - аттестация, тестирование, проведение экзаменов в учебных заведениях и на предприятиях. – Режим доступа: <http://www.net-testing.ru/>
8. Busuu [Электронный ресурс] : the language learning community. – Режим доступа: <https://www.busuu.com>
9. Lingualeo [Электронный ресурс] : мощный комплекс для изучения английского языка. – Режим доступа: <http://lingualeo.com/>
10. Manythings [Электронный ресурс] : is for people studying English as a Second Language (ESL) or English as a Foreign Language (EFL). – Режим доступа: <http://www.manythings.org/>
11. Сайт для изучения английского языка [Электронный ресурс] : This free website has helped students worldwide improve English grammar and vocabulary skills since 2003. – Режим доступа: <http://learnamericanenglishonline.com/>

12. Бесплатные программы для изучения английского языка [Электронный ресурс] : BX Language acquisition. – Режим доступа: <http://www.homeenglish.ru/Programms.htm>
13. Бесплатные программы для изучения английского языка [Электронный ресурс] : Запоминание слов. – Режим доступа: <http://www.homeenglish.ru/Programms.htm>
14. Программы для изучения английского языка [Электронный ресурс] : EZ Memo Booster. – Режим доступа: <http://www.correctenglish.ru/programs/>
15. Программы для изучения английского языка [Электронный ресурс] : Business Letter. – Режим доступа: <http://www.correctenglish.ru/programs/>
16. Программы для изучения английского языка [Электронный ресурс] : Irregular Verbs. – Режим доступа: <http://www.correctenglish.ru/programs/>
17. Бесплатные программы для изучения английского языка [Электронный ресурс] : Ассоциация 1.0. – Режим доступа: <http://www.homeenglish.ru/Programms.htm>
18. Программы для изучения английского языка [Электронный ресурс] : EngCards. – Режим доступа: <http://www.correctenglish.ru/programs/>
19. Бесплатные программы для изучения английского языка [Электронный ресурс] : Grammar. – Режим доступа: <http://www.homeenglish.ru/Programms.htm>
20. Бесплатные программы для изучения английского языка [Электронный ресурс] : Repeng. – Режим доступа: <http://www.homeenglish.ru/Programms.htm>
21. Программы для изучения английского языка [Электронный ресурс] : English Grammr in Use. – Режим доступа: <http://www.correctenglish.ru/programs/>
22. Бесплатные программы для изучения английского языка [Электронный ресурс] : FWords. – Режим доступа: <http://www.homeenglish.ru/Programms.htm>
23. Программы для изучения английского языка [Электронный ресурс] : EnglishCheck. – Режим доступа: <http://www.correctenglish.ru/programs/>

24. Бесплатные программы для изучения английского языка [Электронный ресурс] : ETrainer. – Режим доступа: <http://www.homeenglish.ru/Programms.htm>

25. Программы для изучения английского языка [Электронный ресурс] : SelfIn. – Режим доступа: <http://www.correctenglish.ru/programs/>

26. Transform slides into interactive eLearning [Электронный ресурс] : Adobe Presenter. – Режим доступа: <http://www.adobe.com/ru/products/presenter.html>

ПРИЛОЖЕНИЕ А

Листинг программы

```
private void button1_Click(object sender, EventArgs e)
{
    list_words.Clear();
    list_wordsnumber.Clear();
    list_wordsposition.Clear();
    label_counter = 0;
    red_counter = 0;
    labels_count = 0;
    labels_count_green = 0;
    for (int i = 0; i < maxLabelCount; i++)
    {
        labelArr[i].ForeColor = System.Drawing.Color.Empty;
        labelArr[i].Visible = true;
    }
    button1.Visible = false;
    VideoGridForm videoform = new VideoGridForm(this);
    videoform.Show();
    videoform.Visible = false;
    LabelGridForm labelform = new LabelGridForm(this);
    labelform.Show();
    labelform.Visible = false;
    Program.exercise_number++;
    curItem2 = curItem.Remove(0, curItem.LastIndexOf(@"\Data") + 6);
    curItem2 = curItem2.Remove(curItem2.IndexOf(@"\"), 
    curItem2.IndexOf(@".") - curItem2.IndexOf(@"\")); 
    curItem2 = curItem2.Replace(".mp4", "");
```

```

videoform.videolabel_Underline_name.Text = curItem2;
videoform.videobutton_Click(sender, e);
if (Program.exercise_finished == 1)
{
    Program.exercise_finished = 0;
    Choose_exercise(sender, e);
    return;
}
axWindowsMediaPlayer1.URL = Path.GetFullPath(@"Data") + @"\\" +
videoform.videolabel.Text;
labelform.labellabel_Id.Text = videoform.videolabel_Id.Text;
int rows_counter = labelform.labelDataGridView.RowCount;

for (int i = 0; i < rows_counter - 1; i++)
{
    if (labelform.labelDataGridView.Rows[i].Cells[4].Value.ToString() ==
    == labelform.labellabel_Id.Text)
    {
        list_words.Add(labelform.labelDataGridView.Rows[i].Cells[2].Va
lue.ToString());
        list_wordsnumber.Add(Convert.ToInt32(labelform.labelDataGridView
.Rows[i].Cells[3].Value));
    }
}
int labelscounter = 0;
for (int i = 0; labelscounter != list_words.Count; i++)
{
    Random rand = new Random();
    int random = rand.Next(0, list_words.Count);
    if (i == 0)

```

```

{
    list_wordsposition.Add(random);
    labelscounter++;
}
else
{
    if (!list_wordsposition.Contains(random))
    {
        list_wordsposition.Add(random);
        labelscounter++;
    }
}
labels_count = list_words.Count;
for (int i = 0; i < labels_count; i++)
{
    labelArr[i].Text = list_words[list_wordsposition[i]];
}
for (int i = labels_count; i < maxLabelCount; i++)
{
    labelArr[i].Visible = false;
}
}

```