

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

КАФЕДРА ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

**Барсукова Наталья Владимировна**

**Выпускная квалификационная работа бакалавра**

**Система обучения английскому языку с  
применением аудио-визуальных стимулирующих  
воздействий**

Направление 010300

Фундаментальная информатика и информационные технологии

Научный руководитель,  
старший преподаватель  
Стученков А.Б.

Санкт-Петербург

2016

# Содержание

Введение .....	3
Постановка задачи .....	5
Обзор литературы .....	6
Глава 1. Теоретические основы .....	7
1.1. Анализ работы головного мозга .....	7
1.2. Обоснование применимости и описание методики аудио - визуальных воздействий .....	10
Глава 2. Программная реализация.....	13
2.1. Алгоритм работы .....	13
2.2. Разработка программного решения.....	15
2.2.1. Проектирование .....	15
2.2.2. Средства разработки.....	17
2.2.3. Архитектура программы .....	19
2.3. Интерфейс приложения .....	21
2.4. Тестирование работы приложения.....	25
Выводы .....	27
Заключение .....	28
Список литературы .....	29
Приложение 1 .....	31
Приложение 2 .....	34

## Введение

*В начале было Слово.*

*(Евангелие от Иоанна. Новый завет)*

Одним из главных факторов развития общества является способность человека к коммуникации. От общения жестами и звуками первобытных людей до передачи новостей с телеэкранов и видеоконференций в интернете. Дар речи помог построить современное общество, развить науку и технологии, и, чтобы работать сообща, нужно, как минимум, понимать друг друга, но многообразие рас, народов и языков усложняет задачу. Чтобы достичь невозможного, прикоснуться к тайнам вселенной, построить Вавилонскую башню люди должны понимать один и тот же язык. В настоящее время языком глобализации является английский, что объясняется колониальной политикой Британской империи в XIX веке и мировым влиянием Соединённых Штатов Америки сейчас. Во многих странах этот язык является родным для большинства жителей, например, в Великобритании, США, Ирландии, Канаде и Мальте. Так же этот язык считается официальным в Австралии, Новой Зеландии (наряду с маори), в некоторых государствах Азии (Индия, Пакистан и других) и Африки[9].

Сложно переоценить важность знания английского языка для современного человека: это возможность коммуникации с людьми практически в любой точке земного шара, чтения книг в оригинале для саморазвития, изучения новейших открытий в различных научных областях. Примеров множество и все они подводят нас к одному вопросу: как выучить язык на высоком уровне? Важно сделать это в короткие сроки, иметь большой словарный запас и не стесняться своего произношения.

В настоящее время этими проблемами озабочено огромное количество компаний, на рынке представлен широкий спектр услуг различных репетиторов, персональных курсов, тренингов и языковых школ. Развитие информационных технологий сыграло немаловажную роль: появилась

возможность устанавливать обучающие программы на планшеты и смартфоны; многочисленные сайты предоставляют доступ к видео с субтитрами, где в любой момент времени можно остановить просмотр и узнать интересующее Вас слово, просто наведя на него курсор; давно реализована возможность общения в реальном времени с англоговорящими людьми через Skype, Google+, Viber и прочее. Количество лингводидактического материала на полках книжных магазинов затмевает воображение. К сожалению, на все это затрачивается огромное количество времени, денег и нервов. Современный ритм жизни не всегда позволяет человеку оставить внутренние ресурсы на восприятие новой информации. Усталость и апатия не дают возможности заниматься саморазвитием, многим людям нужен способ обучения, который можно вписать в напряженный график. Эту проблему, от части, мы попытаемся решить в данной работе.

Доступны различные методики для обучения: коммуникативная, игровая, интервальные повторения[3]. Эти приёмы являются индивидуальными для каждого человека, но главным критерием обучения является количество слов, которыми можно оперировать во время разговора. Для этого нужно тренировать память, задействуя большие ресурсы мозга. Хотелось бы, чтобы впервые увиденное слово не забывалось через короткий промежуток времени, а откладывалось надолго. Чтобы улучшить скорость запоминания можно задействовать разные участки мозга в процессе усвоения информации. Если увидеть, услышать и проговорить слово, вероятность записи его в долговременную память увеличивается[13]. Таким образом, можно реализовать программу, которая будет подавать сигналы на различные органы чувств, чтобы задействовать большее количество нейронов и формировать новые связи для последующего использования. С её помощью человеку не нужно будет читать тонны литературы, ставить себе напоминания для повторения одних и тех же слов по сотни раз за день. Вы просто садитесь за монитор и «загружаете» в себя данные, почти как в культовом фильме «Матрица».

## **Постановка задачи**

**Целью** данной работы является:

*Создание программной системы обучения английскому языку с применением аудио-визуальных воздействий.*

Поставленная цель определяет следующие **задачи**:

- *анализ предметной области, выявление технических возможностей применения аудио-визуальных воздействий, основанных на рекомендациях ведущих психологов;*
- *проектирование программного решения;*
- *выбор используемых технологий и средств разработки;*
- *разработка продукта, реализующего поставленную цель;*
- *тестирование полученного решения.*

Учитывая то, что создаваемая система будет ориентирована в первую очередь на работу с обычным компьютерным оборудованием, применяемые воздействия должны быть направлены, в основном, на зрительную и слуховую системы человека.

В результате должна получиться функционирующая программа, которая будет особым образом (возможно, с применением стерео-эффекта) проигрывать слово на английском языке в соответствии с его транскрипцией. Слово с переводом должно быть представлено в активной области зрительного восприятия.

Предполагается также создание расширяемой базы слов с транскрипцией и переводом с возможностью добавления, изменения и удаления слов.

## Обзор литературы

В ходе написания данной работы были проанализированы научные и научно-популярные статьи о работе головного мозга человека.

Во-первых, ознакомление со статьёй «Память» в Википедии[16] и изучение некоторых источников, упомянутых в ней, например, работа Дж. Миллера о ёмкости кратковременной памяти[1] и статья о новых когнитивных исследованиях в нейробиологии[2]. Из видео лекции Джона Килстррома[12] были получены более детальные сведения о кратковременной и долговременной памяти. О способах улучшения способности запоминания автор узнала из подборки статей В.А. Казаренко[13]. Механизмы работы памяти доступно описаны в журнале «Знание-сила» [11], там также можно найти информацию о структурах кратковременной и долговременной памяти, об исследовании способности восстановления нервных клеток. Более подробные сведения о нейрогенезе были получены из статьи «Нейроны и нейросети»[14].

Информация о структуре, строении, работе и функциях гиппокампа была получена из работы о физиологии человека[17], из статьи «Морфофункциональные особенности гиппокампа»[18] и из исследования о закреплении новых знаний во время сна[15].

Рассмотрены приёмы обучения иностранному языку и методики преподавания английского языка в ВУЗах и школах[3], описанные Швейцером А. Д.

По работам Голуба Я. В. изучена методика светозвуковых стимуляций[10].

Для написания программной части работы были использованы: книга по программированию на C#[4], руководства пользования библиотекой BASS.NET[5] и NAudio[6]. По статьям в сети Интернет изучена структура WAV файлов[7][8].

# Глава 1

## Теоретические основы

### 1.1. Анализ работы головного мозга

Человеческий организм - это чрезвычайно сложная биологическая машина способная производить одновременно несколько действий и обрабатывать колоссальные объёмы данных, поступающих извне. Не секрет, что за функционирование этого автономного генератора жизни отвечает мозг, способный изменяться сам и настраивать организм в зависимости от ежедневных действий индивидуума, и от влияния окружающей среды. Мы воспринимаем намного больше информации, чем нам кажется, и способны влиять на развитие мозга с помощью актуализации внимания и специальных стимулирующих воздействий.

Существует несколько теоретических моделей памяти в психологии. Ее разделяют на первичную и вторичную; выделяют несколько структур: сенсорное хранилище, кратковременное хранилище и долговременное; так же существует модель уровней переработки и модель рабочей памяти[16]. Нейробиологические же исследования выделяют лишь два основных вида[12]: кратковременная и долговременная[2].

В памяти человека хранится множество различных данных: события; имена и телефоны; последовательности действий как сознательные (вождение автомобиля), так и бессознательные: отвечающие за дыхание, пищеварение, адаптацию к смене времен года и прочее. Почти всё, что ежедневно встречает человек, мимолетно задерживается в кратковременной памяти, позволяя производить простые расчеты, поверхностный анализ, а затем забывается. Объём кратковременной памяти является индивидуальной характеристикой и определяет механическую способность человека запоминать информацию, не используя специальных приемов. Но её ёмкость ограничена. Во время опытов Джордж Миллер получил результаты, в которых объём кратковременной памяти составляет  $7 \pm 2$  объекта[1].

Современные исследования оценивают её ёмкость даже ниже, всего 4-5 объектов[16], но при этом стало известно, что количество можно увеличить, используя метод группировки, называемый «Chunking». Так, например, мы объединяем цифры в номерах телефона в группы по 3 или 2.

В долговременную память может проникнуть лишь информация, побывавшая в кратковременной памяти, таким образом она является своеобразным буфером, пропускающим уже отобранные данные. Но не всё, что хранится в кратковременной памяти попадает в долговременную, информации ещё предстоит пройти «отбор» гиппокампа.

Гиппокамп - это маленькое образование, находящееся в глубине височных долей мозга, является частью лимбической системы и обладает многофункциональностью[17]. Существует теория, что во время бодрствования гиппокамп удерживает данные, а перевод их в долговременную память осуществляется во время сна[15]; он служит своеобразным командным центром для сортировки воспоминаний для дальнейшего хранения и для перегруппировки их по запросу. Так же гиппокамп влияет на процесс обучения, формирования эмоций, отвечает за пространственную ориентацию человека, обладая функцией запоминания и кодирования окружающей среды[18].

Механизм работы гиппокампа до конца не изучен, но существует гипотеза, что его решение о переносе сведений из кратковременной памяти в долговременную зависит как минимум от двух факторов. Первое, на чем основывается мозг, это эмоциональная реакция, имеет ли данная информация какую-либо значимость для индивидуума или является чем-то поразительным, необычным. Мозг конструирует мир в соответствии со своими интересами. Второй немаловажный фактор, это связь новых сведений с уже имеющимся банком данных[11].

Мозг не хранит информацию отдельно, как это делает жесткий диск компьютера, он все время стремится создавать ассоциативные связи. Не существует единой памяти в мозгу человека, есть несколько связанных



систем, каждая из которых выполняет свою роль. Одни нейроны запоминают цвет, другие вкус, третьи ответственны за звуковое восприятие и так далее. Используя различные нервы одновременно мозг соединяет их в единую систему, и, если повторять определенные действия несколько раз, связь укрепляется[11]. В итоге, воспоминание - это устоявшаяся сеть нервных клеток, в которой могут быть задействованы нейроны, находящиеся в различных участках головного мозга[14]. Таким образом новая информация усваивается легче, если имеется связь с прошлыми воспоминаниями.

Обладая нейропластичностью мозг может как объединять, так и разъединять неустойчивые связи, освобождая возможность клеткам устанавливать новые взаимодействия. Эмоциональные воздействия усиливают способность нейронов к прокладыванию новых путей, и это не только какие-либо агрессивные события, заставляющие организм мобилизоваться, но даже обычный смех способствует обучению. Мозг не теряет возможности продуцирования новых нейронов и образования связей даже в старости, поэтому развивать и поддерживать упорядоченность работы мозга можно в любом возрасте[14]. В течение всей жизни клетки мозга обновляются и создают новые сети. Обладая этим знанием, мы можем корректировать своё поведение, получать информацию, изменяющую наш мозг по желанию самого человека, выбирать другой образ мышления и восприятия для перестройки своего организма.

Предполагается, что активация биоэлектрической активности головного мозга, за счет синхронного использования зрительного, слухового и двигательного информационных каналов, может значительно повысить эффективность запоминания информации, благодаря формированию нового ядра памяти[10]. Использование данных воздействий на фоне музыкального сопровождения также может усилить эффект, действуя благоприятные положительные эмоции. С помощью данной методики можно качественно повысить эффективность освоения иностранных языков в части накопления словарного запаса и навыков разговорной речи.

## **1.2. Обоснование применимости методики аудио-визуальных воздействий.**

Технология сенсорной нейрокоррекции головного мозга – это физиологически корректный способ воздействия на человека. Используется подача привычных, безвредных и адекватных по интенсивности и частоте прерывистых световых и звуковых сигналов. В зависимости от режимов такая стимуляция позволяет достичь определенных эффектов, таких как: достижение глубокой релаксации; снижение мышечного напряжения; нормализация сна; отвлечение внимания от болезненных и неприятных ощущений; визуализация, развитие образного мышления; устранение психологических зажимов; повышение стрессоустойчивости. Светозвуковая стимуляция усиливает эффект визуализации мыслеобразов, активизирует правое полушарие, обеспечивая положительную эмоциональную окраску сеансов психотренинга. При этом увеличивается работоспособность, мысли становятся ясными и четкими. Испытывая положительное влияние на психоэмоциональное состояние человек непроизвольно старается поддерживать его во время всего сеанса[10].

Для возрастания эффективности занятий на 30-40% следует использовать музыкальное сопровождение, согласуемое с индивидуальным восприятием человека. Музыка задействует большее количество нейронных сетей мозга и побуждает к эмоциональному реагированию. Информация, приобретенная на фоне определенных эмоций, лучше откладывается в памяти, поэтому этот приём следует активно применять в процессе занятия, при наработке словарного запаса.

Необходимо также учитывать индивидуальные противопоказания к применению аудио-визуальных воздействий: использование мерцающего света и пульсирующего звука субъективно могут восприниматься как негативные. Вместе с тем стимуляция не должна использоваться людьми с эпилепсией, психическими расстройствами, страдающими серьезными

заболеваниями сердечно-сосудистой системы, находящимися под влиянием алкоголя, психоактивных веществ и прочее[10].

Практически эта технология реализуется следующим образом:

- предварительно проводится выбор подходящего режима: самостоятельно по инструкции или с помощью оператора;
- надеваются специальные очки и наушники;
- активируется программа, и нейрокоррекция проводится в автоматическом режиме.

При этом индивид должен повторять услышанные слова и визуализировать полученную информацию в перерыве между словами.

Тренажер мозга ЛИНГВОСТИМ разработан к.м.н. Голубом Я.В. на базе научно-производственного предприятия «МедПАССС», с разрешения которого это устройство было использовано при разработке настоящей технологии.

Для реализации алгоритма вывода зрительной информации экран дисплея функционально разделен на 3 зоны:

- *По центру экрана*, разделенного пополам по вертикали, представлено слово на английском языке в левой части, и зеркально отраженное слово в правой части.
- *В верхней части экрана* накопительными ёмкостями (многострочными текстовыми областями) выводится слово, предложение или набор слов на английском языке, каждое из которых в дальнейшем будет подано на центральную часть экрана и воспроизведено.
- *В нижней части экрана* накопительными ёмкостями выводится перевод слова, предложения или набора слов, которые использует индивид для изучения.

Для реализации алгоритма загрузки слухового канала предусмотрен параллельный вывод в наушники следующей информации:

- В *правое ухо* подается слово, выбранное из базы данных для изучения, с использованием транскрипции и заранее записанного звучания транскрипционных знаков.
- В *левое ухо* подается звучание того же слова, но закодированное азбукой Морзе.
- Также используется тематическое фоновое музыкальное сопровождение во время всего периода использования программы. Это может быть как классическая инструментальная музыка, так и песни на языке оригинала.

Рассматривая предложенную технологию с позиции теории деятельности, можно выделить несколько приоритетных видов деятельности: аудирование, говорение, речевую компрессию с преобразованием звуков в слова и зрительную деятельность, предполагающую визуализацию и осознание написанного иноязычного текста, что, фактически, ассоциируется с функцией перевода.

Следует отметить, что методический потенциал современных информационно-коммуникационных технологий дает ряд дополнительных возможностей, позволяя:

- подобрать текстовый, графический, фото-, аудио- и видеоматериал по изучаемым темам;
- организовать в группах и целом классе обсуждение культурных и социальных проблем;
- провести лингвистический анализ устной и письменной речи носителей языка.

## Глава 2

### Программная реализация

Задача ВКР состоит в разработке системы обучения английскому языку с применением аудио-визуальных стимулирующих воздействий, основанной на методике, описанной в разделе 1.2.

С точки зрения обучаемого данная методика заключается в следующем:

- Человек видит перед собой экран, разделенный на две части (для восприятия информации отдельно каждым глазом используется специальная перегородка, разграничивающая экран монитора) и наушники.
- После старта работы программы он видит различную информацию (но связанную с одним и тем же объектом): слово на иностранном языке и перевод; в это же время на каждый наушник подается своя звуковая дорожка: произношение слова и его код в азбуке Морзе.
- Также выполняется световая стимуляция в виде мигающих точек по центру, в нижней и в верхней частях экрана.

#### 2.1 Алгоритм работы

Преподаватель выбирает профиль человека, который будет проходить обучение (далее обучаемый); если профиль отсутствует, то его можно зарегистрировать. После этого преподавателем делается выбор слова из словаря для изучения (если оно отсутствует, его можно добавить), нажимается кнопка «старт» и производится аудио-звуковая стимуляция. Возможен также режим работы без преподавателя, в этом случае выбор происходит автоматически.

Слово считывается из словаря программы вместе с транскрипцией и переводом и подается на вход в основной класс программы. Из этой связки

вычленяется само *слово*, которое отправляется в функцию преобразования его в код Морзе, *транскрипция*, которая отправляется в функцию, отвечающую за подбор звуковых файлов, и *перевод*, записываемый в отдельную переменную. На выходе из каждой функции мы имеем некоторую последовательность символов (для транскрипции – это последовательность имен аудиофайлов, а для слова – это последовательность из точек и тире). Далее обе последовательности отдаются механизму преобразования их в звуковые дорожки. Важной особенностью этого алгоритма является синхронизация получившихся аудиофайлов по времени звучания:

- По ряду точек и тире собирается один файл со всей последовательностью закодированного слова и оценивается его продолжительность звучания.
- По, известным алгоритму, длинам согласных звуков (продолжительность каждого из них одинаковая и в данной программе составляет 0.3 секунды), количеству гласных звуков (продолжительность каждого увеличена до 3 секунд, для реализации возможности растягивания длительности звучания слова) и временем проигрывания аудиодорожки с Морзе кодировкой определяется необходимая длина для каждого из гласных звуков.

$$\frac{LengthMorze - NumberOfConsonants * 0,3}{NumberOfVowels}$$

- Затем из набора транскрипционных звуков собирается один аудиофайл с озвученной транскрипцией слова, который по длине точно совпадает с длительностью звучания файла с кодом Морзе.

Далее программой выполняются операции воспроизведения и вывода. Обучаемый воспринимает конечный результат, а именно:

- Правым и левым глазом по отдельности он видит разделенный на 3 части экран:
  - *по центру экрана* выводится слово на английском языке в

- левой части, и зеркально отраженное слово в правой части;
- *в верхней части экрана* зафиксировано слово на английском языке;
  - *в нижней части экрана* представлен перевод слова.
- *Правым* ухом он слышит озвученную транскрипцию слова.
- *Левым* ухом он слышит слово, закодированное морзянкой.

## **2.2 Разработка программного решения.**

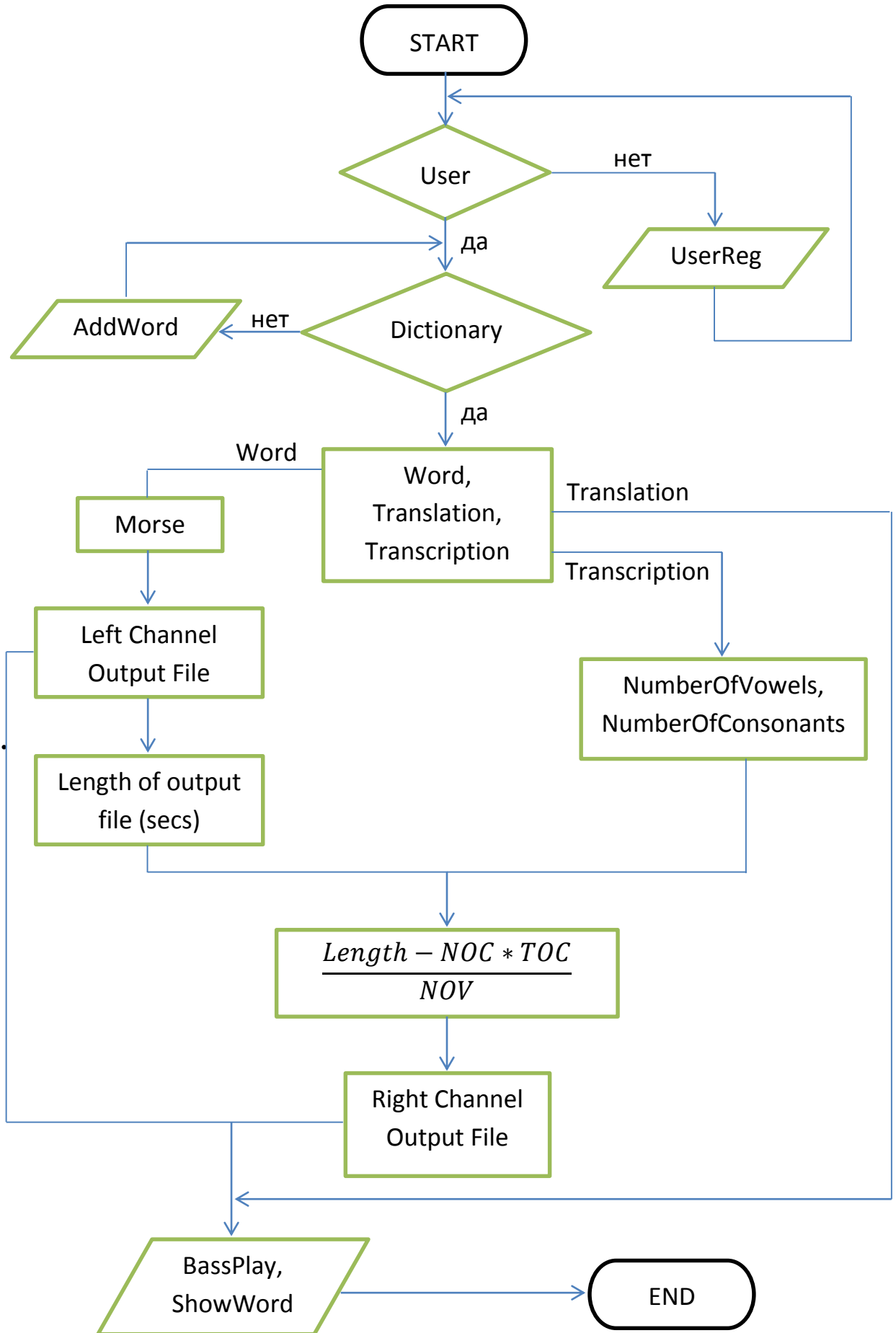
### **2.2.1 Проектирование.**

Задача проектирования для удобства реализации была разбита автором на следующие подзадачи, каждая из которых была реализована в процессе разработки:

1. Воспроизведение звукового файла.
2. Одновременное воспроизведение двух аудиофайлов с разделением по каналам.
3. Непрерывное воспроизведение двух цепочек аудиофайлов с разделением по каналам (эта подзадача в процессе разработки была модифицирована до вида, представленного объединением пунктов 2 и 4).
4. «Склейка» нескольких аудиофайлов в один.
5. Синхронизация времени воспроизведения двух аудиофайлов.
6. Создание словаря с возможностями хранения, добавления, удаления и редактирования троек: слово - транскрипция - перевод.
7. Вывод слова и перевода на экран.
8. Создание базы данных пользователей.
9. Разработка интерфейса программы.

Для реализации работы алгоритма на практике было спроектировано программное решение, представленное в этой главе и описанное следующей

блок-схемой:





## 2.2.2 Средства разработки

В качестве основной среды разработки была выбрана Microsoft Visual Studio 2013 ввиду своего удобства работы с кодом и поддержкой множества языков, таких как C#, C++ и других, а также легкостью работы с Windows Forms, что позволяет разрабатывать программы с поддержкой нескольких окон.

В процессе разработки программного решения автор столкнулась с рядом трудностей, повлиявших как на выбор средств разработки, так и на сам алгоритм работы программы. Так язык C# был выбран ввиду своего удобства работы с классами и наличием большого количества свободных библиотек[4].

Первая трудность возникла при решении подзадачи одновременного воспроизведения двух аудиофайлов с делением их по каналам. Автором работы были изучены и проанализированы различные библиотеки: NAudio[6], BASS[5], стандартные библиотеки C#[4]. Однако, каждая имела свои преимущества и недостатки, и, в результате, было принято решение о выборе BASS, позволяющей работать с потоками и каналами. Также на выбор повлияло наличие подробной инструкции использования и примеры с открытым исходным кодом.

Остановимся на этой библиотеке подробнее. BASS предоставляет разработчикам образец аудиоцентра, в частности, обеспечивает работу с потоками; функциями записи; позволяет редактировать теги аудиофайлов; конвертировать популярные форматы, задействуя при этом малое количество системных ресурсов. Весь набор возможностей предоставляется в виде динамической библиотеки, размером около 100 Кб.

Для работы BASS с ОС Windows требуется DirectX 3+, для воспроизведения и записи звука необходимы DirectSound и DirectSound3D. В Mac OS X BASS использует CoreAudio.

Библиотека распространяется под несколькими лицензиями[5]:

- Freeware: библиотека бесплатна для некоммерческого использования (именно этой версией пользовалась автор).
- Shareware: «условно бесплатная» лицензия. Даёт возможность использовать BASS в коммерческих целях и в неограниченных количествах shareware-продуктов, при условии, что – стоимость продукта не может превышать €40 за каждый. Подходит для индивидуальных лиц (не организаций).
- Единая коммерческая лицензия позволяет использовать BASS только в одном коммерческом продукте.
- Безлимитная коммерческая лицензия даёт возможность использовать BASS в неограниченном количестве коммерческих продуктов.

Для работы с библиотекой была использована .NET обертка, которая называется BASS.NET. Она позволяет легко использовать библиотеку для работы с .Net Framework и любым управляемым языком, таким как C#, Visual Basic, JScript, F# или C++.

В программном решении используются звуковые дорожки с расширением .WAV. WAVeform Audio File Format (WAVE, WAV) – формат файла-контейнера для хранения записи оцифрованного аудиопотока, подвид RIFF (Resource Interchange File Format)[7]. Файл RIFF начинается с заголовка, за которым следует последовательность фрагментов данных (они называются chunks)[8]. WAVE файл, помимо RIFF заголовка, содержит также:

- WAVE заголовок, который в свою очередь состоит из WAVE идентификатора и fmt идентификатора;
- fmt фрагмента данных, который содержит информацию о формате, такую как количество каналов, бит на семпл и другую;
- data (содержит сами данные аудиофайла и информацию об их размере).

Данная структура позволила довольно просто решить задачу с объединением файлов. Идея заключалась в том, чтобы создать единый для всех объединяемых файлов заголовок, а затем просто записывать сами данные в выходной файл.

Выбор в пользу .WAV был сделан ввиду простоты извлечения информации.

### 2.2.3 Архитектура программы

В этом параграфе подробно разберем каждую часть программного решения.

Класс *MainWindow* – основа всей программы. Здесь происходит взаимодействие большинства элементов: загрузка словаря (о нем – далее), выбор пользователя, объявление экземпляров многих классов (*Dictionary*, *Users*, *SoundManager*). Также он отвечает за отрисовку главного окна и взаимодействие всех частей его интерфейса, который продемонстрирован в следующем параграфе этой главы.

Класс *Dictionary* отвечает за хранение и извлечение троек слово – транскрипция – перевод и включает методы добавления, редактирования и удаления элементов из словаря. Поскольку в английском языке звуков больше чем букв (44 против 26), для хранения части из них используются буквы русского алфавита. При этом имеется второй массив, с помощью которого им в соответствие ставятся закодированные в Unicode символы транскрипций. Однако, учитывая схожесть некоторых звуков (например, краткий и длинный «о»), отдельные записи для них не требуются, так как согласно нашему алгоритму мы растягиваем гласные звуки для уравнивания длин аудиодорожек левого и правого канала.

Класс *AddWord* является вспомогательным для класса *Dictionary* и отвечает за отрисовку формы добавления слова в словарь. Этот функционал был выделен в отдельный класс, поскольку требует для себя отдельного окна в интерфейсе программы.

Класс *Users* отвечает за хранение информации о пользователях и включает методы добавления, редактирования и удаления сведений. В дальнейшем функциональные возможности этого класса можно расширить до базы данных, что позволит также получать информацию не только о человеке, но и обо всем пройденном им материале, а также добавлять слова для следующего занятия, чтобы обучаемый мог заниматься без помощи преподавателя.

Класс *Morse* отвечает за преобразование слова в двоичную кодировку азбуки Морзе (точки и тире). Для этого в классе хранятся два массива *Literals* и *LiteralCodes* с типами `string`: один – с буквами английского алфавита, а второй – с соответствующими им кодами. Алгоритм шифровки получает на вход переменную типа `string` (это может быть как символ, так и слово) и по соответствиям элементов двух массивов строит результирующую последовательность из точек и тире.

Класс *Concatenator* используется для объединения нескольких файлов в один. Идея его работы довольно проста: поскольку WAV файлы состоят из заголовка и блока данных, мы создаем единый заголовок и добавляем новый набор информации в результирующий аудиофайл. Объекты соединяются только строго по 2 (поскольку для создания заголовка необходимо знать размер каждого из конкатенируемых файлов). Наиболее важные фрагменты кода класса *Concatenator* представлены в Приложении 2.

В классе *SoundManager* описаны методы образования звуковых дорожек из последовательностей звуков для правого (*RightStream*) и левого (*LeftStream*) каналов, их воспроизведения с помощью библиотеки BASS (*Play*). Ниже приведены алгоритмы работы каждого из них.

Метод *RightStream*, подобно классу *Morse*, хранит 4 массива типа `string` (по 2 для гласных (сюда же входят и дифтонги) и согласных: *Vowels*, *VowelSounds*, *Consonants*, *ConsonantSounds*), с помощью которых определяется соответствие звука названию аудиофайла с его произношением. Получая информацию о длительности дорожки с кодом Морзе,

рассчитывается необходимое время воспроизведения каждого из гласных звуков (как уже было отмечено в параграфе 1 этой главы, длительность согласных звуков фиксирована и равна 300 миллисекундам). После чего создается экземпляр класса *Concatenator* и с использованием его метода *Concatenate* формируется файл с результатом.

Метод *LeftStream* получает на вход переменную *Word* с типом *string*, хранящую в себе выбранное слово. В первую очередь создается экземпляр класса *Morse*, с помощью которого происходит кодирование слова, и количество символов записывается в переменную *SignsInMorse*, имеющую тип *int*. Далее запускается цикл, внутри которого с помощью оператора *switch*, выбирается необходимый звук (“dot.wav” или “line.wav” (время воспроизведения первого 125 миллисекунд, второго – 325 миллисекунд)). После чего создается экземпляр класса *Concatenator* и с использованием его метода *Concatenate* формируется файл с результатом. Код методов *RightStream* и *LeftStream* представлен в Приложении 1.

Метод *Play* начинается с объявления двух переменных с типом *int*: *RStream*, *LStream*. С точки зрения библиотеки BASS они являются потоками, по данным из которых можно проиграть трек. Далее, с использованием стандартного метода из библиотеки *BASS\_StreamCreateFile*, в них записывается необходимая для воспроизведения информация из аудиофайлов, полученных двумя предыдущими методами. С помощью *BASS\_ChannelSetAttribute* двум полученным потокам присваиваются параметры воспроизведения (в данном случае задается правый или левый канал наушников), после чего вызывается функция *BASS\_ChannelPlay*, отвечающая за проигрывание.

Метод *ShowWord* отвечает за вывод слова и перевода в различные блоки главного окна.

Ниже приведена иерархия классов программного решения:

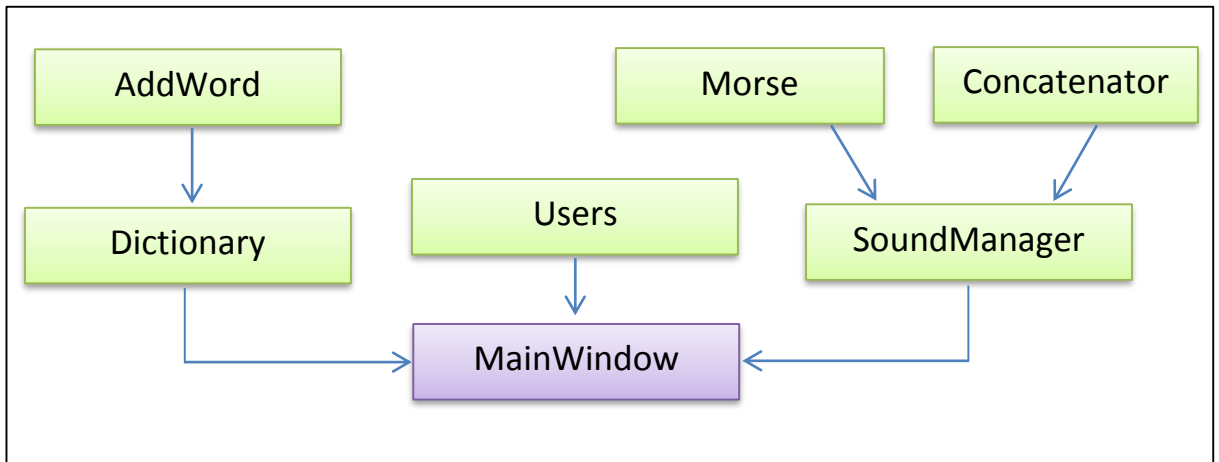


Рис. 2.2.3.1 Иерархия классов

## 2.3 Интерфейс приложения

Перейдем к рассмотрению работы программы с точки зрения пользователя. После запуска открывается главное окно:

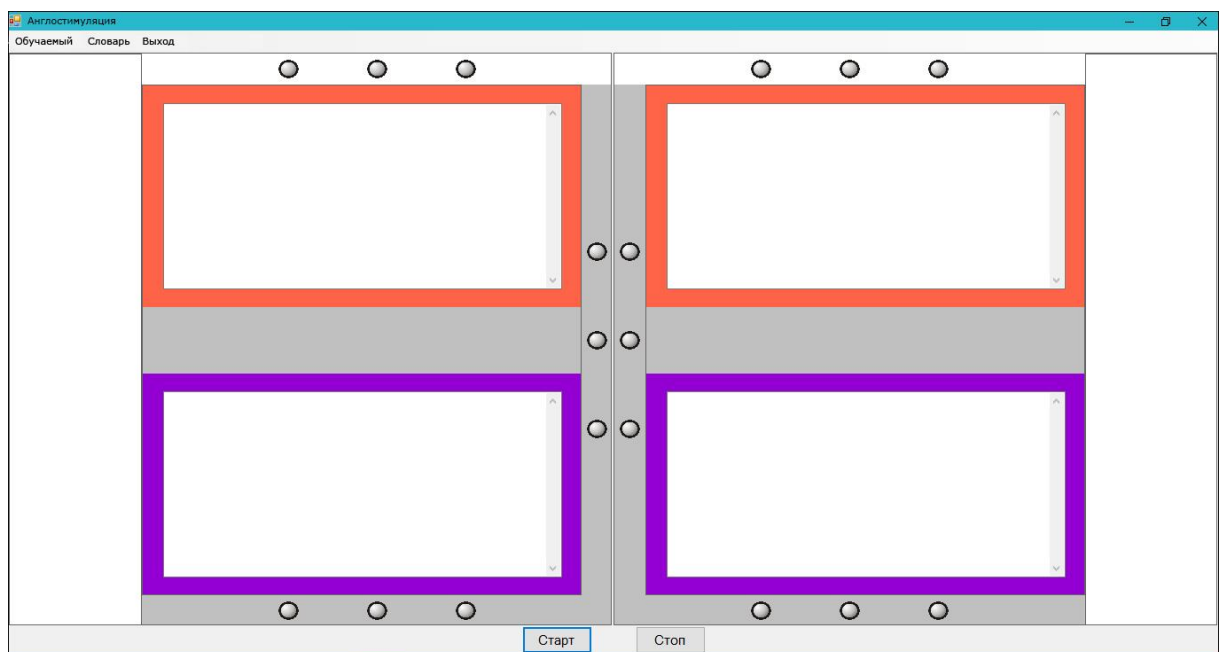


Рис. 2.3.1 Главное окно

Первое, что необходимо сделать для работы – это выбрать обучаемого из выпадающего списка.

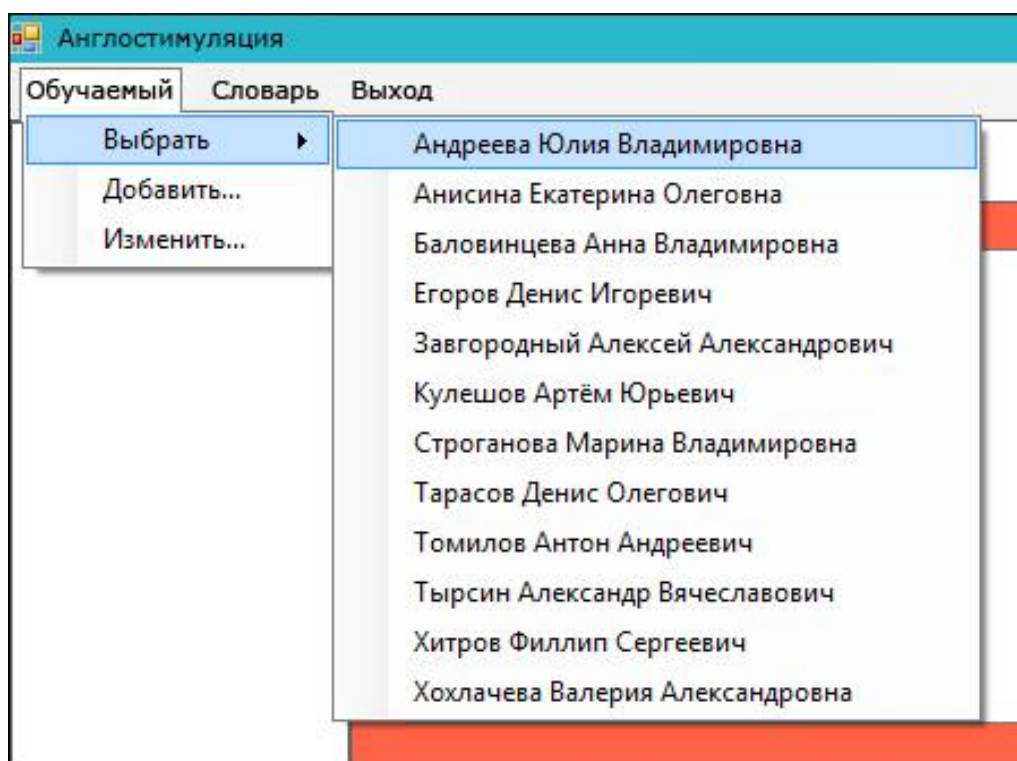


Рис. 2.3.2 Выбор обучаемого

В случае отсутствия нужного профиля предусмотрена функция добавления. После нажатия кнопки *Добавить*, всплывет новое окно:

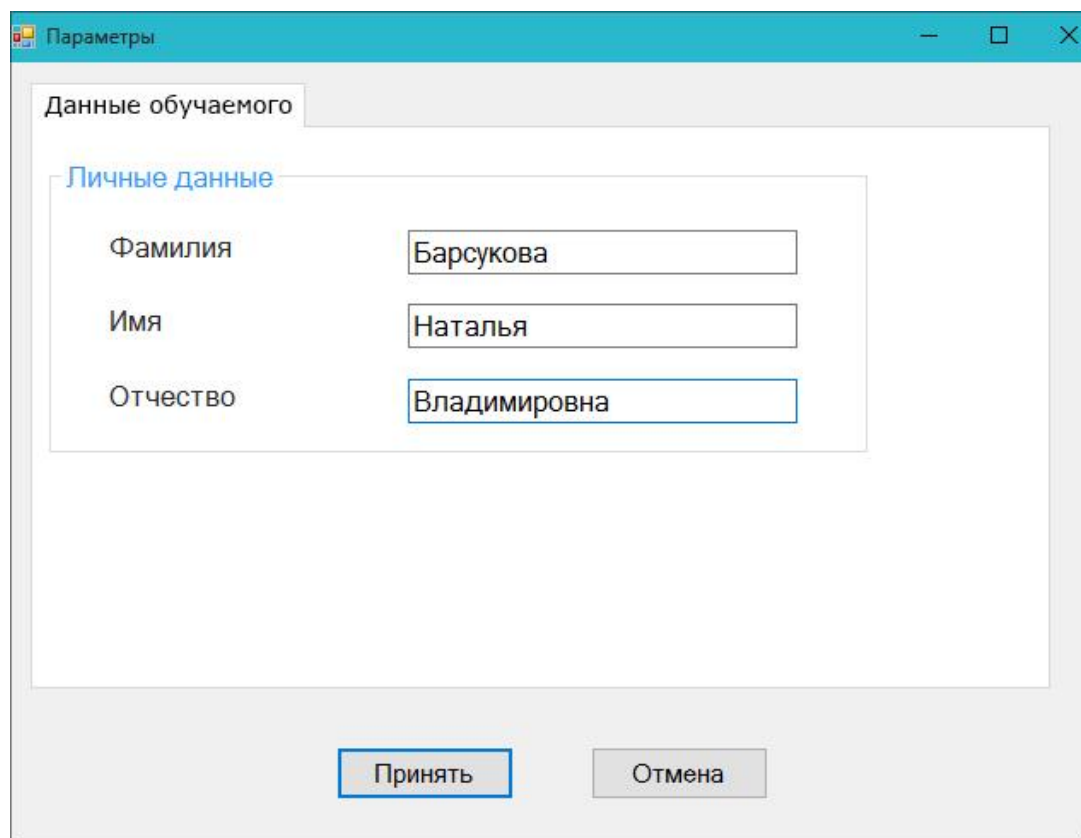


Рис. 2.3.3 Добавление обучаемого

Следующее действие – это выбор слова или его добавление в словарь:

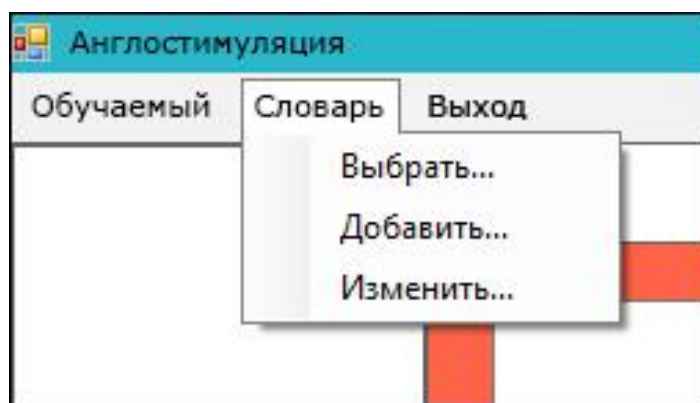


Рис. 2.3.4 Элементы управления словарём

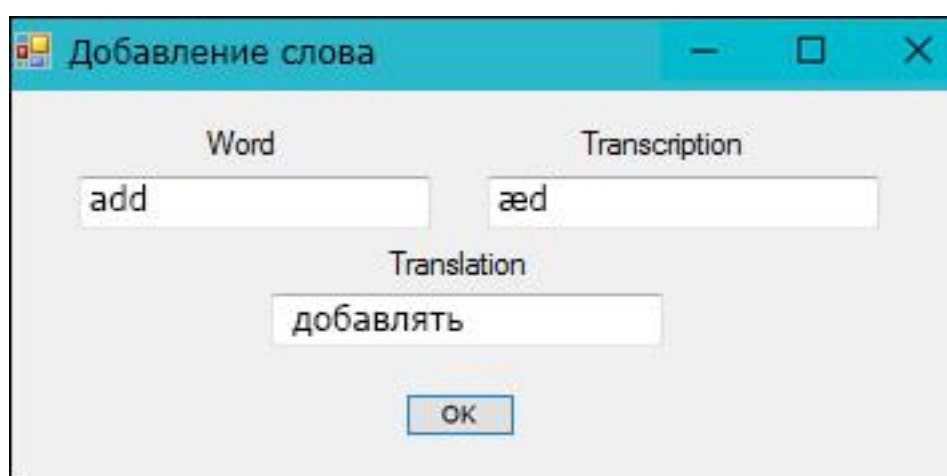


Рис. 2.3.5 Добавление слова в словарь

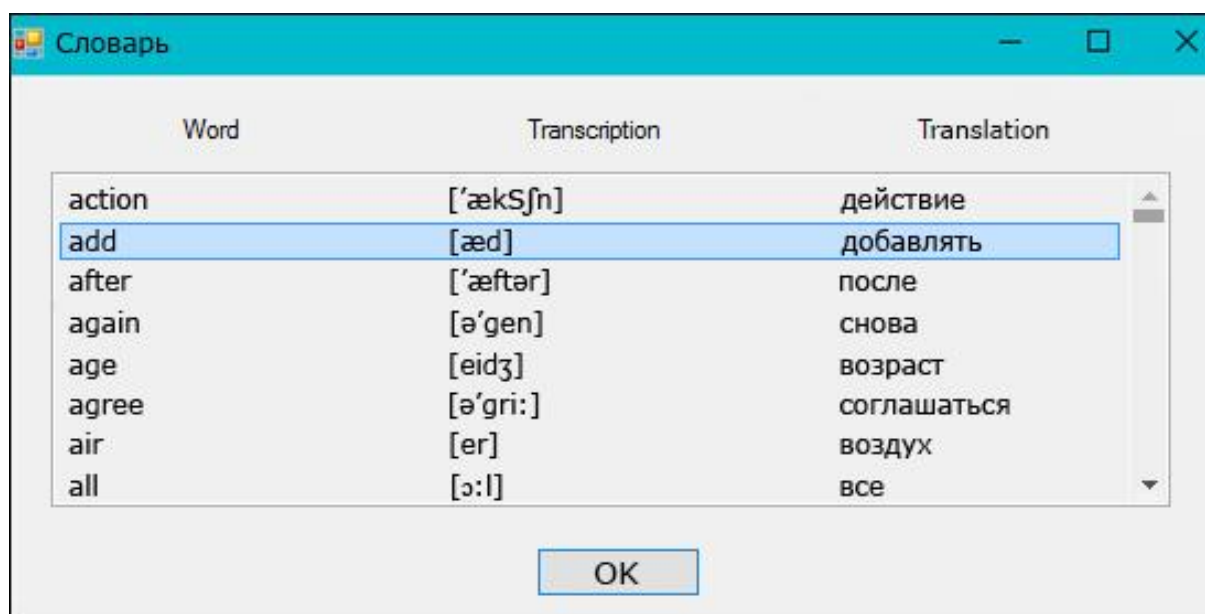


Рис. 2.3.6 Выбор слова



Все, что остается сделать – это нажать на старт. Обучаемый увидит слово и перевод в верхнем и нижнем блоках соответственно и услышит звучание слова в правом ухе и закодированное это же слово морзянкой в левом. Также будет происходить световая стимуляция с помощью изменения цвета элементов в разных частях экрана:

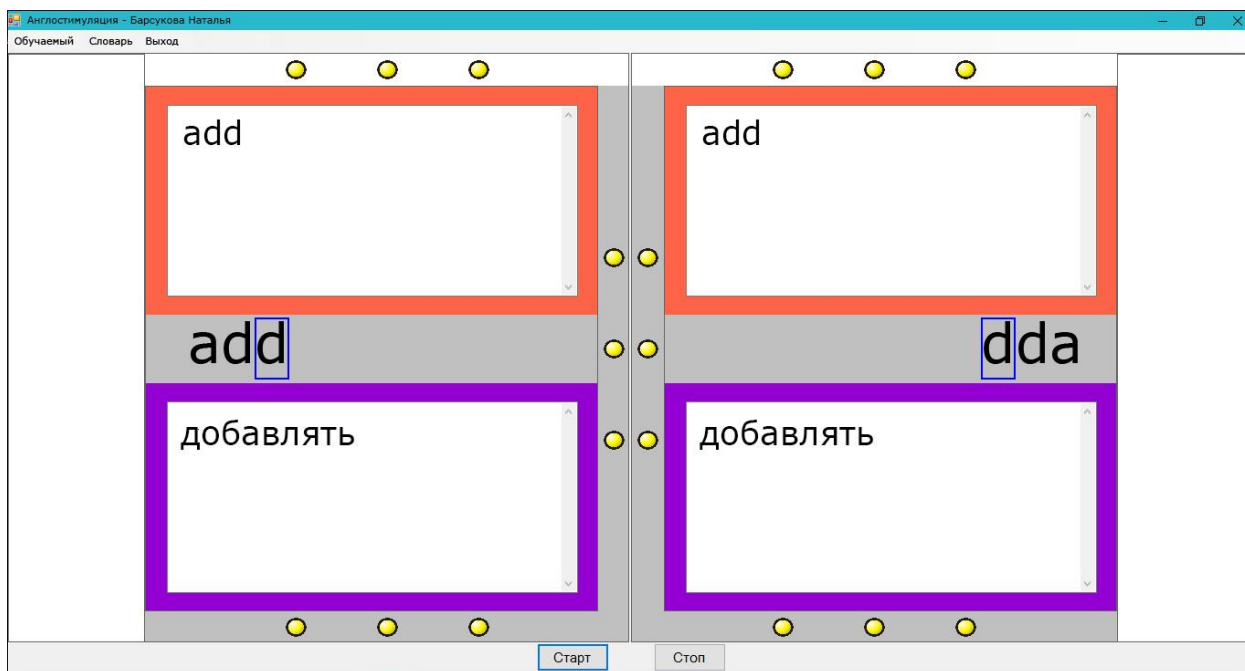


Рис. 2.3.7 Вывод результата

## 2.4. Тестирование работы приложения

На базе Санкт-петербургского научно-исследовательского института физической культуры (возможность проведения исследования была предоставлена заведующим сектором спортивной медицины и физиологии Голубом Я. В.) было выполнено тестирование работоспособности и эффективности системы. Группа обучаемых формировалась из студентов, желающих пройти небольшой курс английского языка (10-15 слов) по новой экспериментальной методике. Количество обучаемых составило 12 человек.

Для каждого из испытуемых, в зависимости от их уровня владения языком, педагогом были подобраны слова для изучения. После проведения занятия через небольшой промежуток времени производилась контрольная

проверка результатов психостимуляции. Также контроль знаний проводился повторно через несколько дней, чтобы удостовериться в степени усваивания информации.

Результатом исследования стало 100% положительных отзывов о работе системы. Кроме того, все обучаемые подтвердили гипотезу о значительном повышении эффективности запоминания слов с применением разработанной методики.

Проведенное тестирование является лишь первым практическим исследованием выполненной разработки. Предполагаются также и дальнейшие, более масштабные проверки эффективности работы. Однако, их проведению должна предшествовать доработка и расширение функциональности программы.

## **Выводы**

В данной работе была создана программная система, предназначенная для изучения слов английского языка, новизна которой заключается в применении аудио-визуальных воздействий, повышающих возможности запоминания изучаемых слов.

Были выполнены поставленные задачи:

- проанализирована предметная область, в частности работа головного мозга человека и методика светозвуковых стимуляций;
- спроектировано программное решение, создана наглядная блок-схема работы алгоритма, выполнено построение объектной модели программы;
- выбраны средства разработки;
- разработан программный продукт, реализующий простую систему обучения словам английского языка;
- было выполнено тестирование разработанной системы, продемонстрировавшее ее эффективность.

Полученная программа может быть использована людьми, вне зависимости от возраста и уровня подготовки, для накопления словарного запаса и приобретения навыков разговорной речи на английском языке. Завершение процесса изучения английского языка, связанного с освоением грамматических правил, должно осуществляться отдельно с использованием общепринятых методик.

## **Заключение**

Практическое использование разработанной компьютерной технологии накопления словарного запаса и приобретения навыков разговорной речи на иностранном языке подтвердило принципиальную возможность ее использования как в качестве методического инструмента для первичного изучения английского языка, так и для совершенствования знаний.

Повышению эффективности накопления словарного запаса способствовала предварительная светозвуковая стимуляция перед началом сеанса и тематическое музыкальное сопровождение обучения. Пробные занятия по этой программе психологически воспринимались положительно.

Реализованный алгоритм позволяет применить данную технологию для изучения различных иностранных языков, используя для этого разнообразный демонстрационный материал и необходимую базу транскрипционных звуков выбранного языка, что придаёт ей свойство универсальности.

### ***Перспективы развития:***

- реализация возможности составления из слов, записанных в словаре, предложений для изучения;
- создание базы данных пользователей для хранения уже изученных слов и возможности записи слов для последующего изучения в виде нескольких уроков;
- подключение мышечной работы с помощью передвижения курсора по словам в предложении;
- модифицирование алгоритма склейки слов не по звукам, а по слогам для достижения более гладкого звучания;
- расширение программы для возможности изучения других иностранных языков.

## Список литературы

1. Miller, G. A. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 1956.
2. Squire, L. R., & Knowlton, B. J. The medial temporal lobe, the hippocampus, and the memory systems of the brain. // In M. Gazzaniga (Ed.), *The new cognitive neurosciences*. Cambridge, MA: MIT Press., 2000.
3. Швейцер А. Д. Перевод и лингвистика: учебник для институтов и факультетов иностранных языков. М.: Воениздат, 1973.
4. Шилдт Г. С# 3.0 Полное руководство, 2010.
5. BASS.NET. <http://BASS.radio42.com/>
6. NAudio Overview. <https://naudio.codeplex.com/>
7. WAV on Wikipedia. <https://ru.wikipedia.org/wiki/WAV/>
8. WAVE PCM soundfile format. <http://soundfile.sapp.org/doc/WAVEFormat/>
9. Английский язык Wikipedia.  
[https://ru.wikipedia.org/wiki/Английский\\_язык/](https://ru.wikipedia.org/wiki/Английский_язык/)
10. Голуб Я.В. Светозвуковая стимуляция мозга.  
[http://magicspeedreading.com/books/a\\_mysl/mind\\_machine/maind\\_machine\\_000.html/](http://magicspeedreading.com/books/a_mysl/mind_machine/maind_machine_000.html/)
11. Журнал «Знание-сила № 2». <http://pcworl.narod.ru/pamit.index.html>
12. Килстром Джон. Лекция 10. Память.  
[http://univertv.ru/video/psihologiya/psihofiziologiya/lekcija\\_10\\_pamyat](http://univertv.ru/video/psihologiya/psihofiziologiya/lekcija_10_pamyat)
13. Козаренко В.А. Все о памяти и способах запоминания.  
[http://mnemonikon.ru/kozarenko\\_1p.pdf](http://mnemonikon.ru/kozarenko_1p.pdf)
14. Нейроны и нейросети. <http://neutrino.mk.ua/pamyat/neyroni-i-neyroseti>
15. Новости науки.  
[http://elementy.ru/novosti\\_nauki/430482/Vo\\_vremya\\_fazy\\_medlennogo\\_sna\\_a\\_ktivno\\_zakrepyayutsya\\_novye\\_znaniya](http://elementy.ru/novosti_nauki/430482/Vo_vremya_fazy_medlennogo_sna_a_ktivno_zakrepyayutsya_novye_znaniya)
16. Память Wikipedia. <https://ru.wikipedia.org/wiki/Память>
17. Физиология человека / под редакцией В.М.Покровского, Г.Ф.Коротько

<http://www.bibliotekar.ru/447/55.htm>

18. Шевцов А. А., Фундовная Е. В. Многофункциональные особенности гиппокампа. <http://www.scienceforum.ru/2014/pdf/2154.pdf>

# Приложение 1

## Класс *SoundManager*

### Метод *LeftStream*:

```
private void LeftStream(string Word)
{
    Morse morse = new Morse(Word);
    morse.ConvertToMorse();
    string MorseCode = morse.MorseText;
    string[] MorseCodeSplited = MorseCode.Split();

    foreach (string i in MorseCodeSplited) SignsInMorse += 1;
    string[] MorseCodeInFiles = new string[SignsInMorseCode];

    for (int i = 0; i < SignsInMorseCode; i++)
    {
        switch (MorseCodeSplited[i])
        {
            case ".":
            {
                MorseCodeInFiles[i] = "Dot.wav";
                LengthOfMorseAudioMs += 125;
                break;
            }
            case "-":
            {
                MorseCodeInFiles[i] = "Line.wav";
                LengthOfMorseAudioMs += 325;
                break;
            }
        }
    }

    Concatenator ConcatenateLeft = new Concatenator();
    ConcatenateLeft.Concatenate(MorseCodeInFiles, "ResultLeft.wav");
}
```

## Метод *RightStream*:

```
private void RightStream(string Transcription)
{
    string[] TranscriptionInFiles = new string[Transcription.Length];
    int index;
    string[] Vowels = new string[] { "I", "E", "U", "Y", "Э", "A",
                                     "И", "Е", "Ё", "О", "А", "О",
                                     "Ц", "Г", "Щ", "С", "Я", "Ы", "Х"};

    string[] VowelSounds = new string[]{ "i.wav", "e.wav", "u.wav", "u.wav",
                                           "ae.wav", "a.wav", "i.wav", "ee.wav",
                                           "yo.wav", "o.wav", "a.wav", "o.wav",
                                           "ie.wav", "ea.wav", "ei.wav", "oi.wav",
                                           "ai.wav", "au.wav", "ou.wav"};

    string[] Consonants = new string[] { "P", "T", "K", "Ч", "F", "Ф", "S",
                                           "Ш", "B", "D", "G", "Д", "V", "З", "Ж",
                                           "M", "N", "H", "H", "L", "R", "Й", "W"};

    string[] ConsonantSounds = new string[] { "p.wav", "t.wav", "k.wav",
                                                "ch.wav", "th.wav", "s.wav", "sh.wav",
                                                "b.wav", "d.wav", "g.wav", "dg.wav",
                                                "v.wav", "zth.wav", "z.wav", "zs.wav",
                                                "m.wav", "n.wav", "nn.wav",
                                                "h.wav", "l.wav", "r.wav",
                                                "j.wav", "W.wav"};

    int NumberOfVowels = 0;
    int LengthOfEachVowelMs = LengthOfMorseAudioMs;
    foreach (char letter in Transcription)
    {
        index = -1;
        index = Array.IndexOf<string>(Vowels,
                                     letter.ToString().ToUpper());
        if (index != -1)
        {
            NumberOfVowels += 1;
            continue;
        }
        else index = Array.IndexOf<string>(Consonants,
                                           letter.ToString().ToUpper());
    }
}
```



```

        if (index != -1)
        {
            LengthOfEachVowelMs -= 300;
            continue;
        }
        if (index == -1) continue;
    }
    if (NumberOfVowels != 0) LengthOfEachVowelMs =
        LengthOfEachVowelMs / NumberOfVowels;
    int Counter = 0;
    foreach (char letter in trans)
    {
        index = -1;
        index = Array.IndexOf<string>(Vowels,
            letter.ToString().ToUpper());
        if (index != -1)
        {
            TranscriptionInFiles[Counter] = VowelSounds[index];
            Counter += 1;
            continue;
        }
        else
        {
            index = Array.IndexOf<string>(Consonants,
                letter.ToString().ToUpper());
            if (index != -1)
            {
                TranscriptionInFiles[Counter] =
                    ConsonantSounds[index];
                Counter += 1;
                continue;
            }
        }
    }
    Concatenator ConcatenateRight = new Concatenator();
    ConcatenateLeft.Concatenate(TranscriptionInFiles, "ResultRight.wav");
}

```

## Приложение 2

### Класс *Concatenator* (некоторые важные фрагменты)

```
/* Как нам известно из структуры WAVE файлов, первые 8 байт – это заголовок RIFF-файла. Нам необходимо определить ChunkSize и получить информацию из NumChannels, SampleRate и BitsPerSample. */
```

```
private void WaveHeaderIN(string spath)
{
    FileStream fs = new FileStream(spath, FileMode.Open,
        FileAccess.Read);
    BinaryReader br = new BinaryReader(fs);
    length = (int)fs.Length - 8;
    fs.Position = 22;
    channels = br.ReadInt16();
    fs.Position = 24;
    samplerate = br.ReadInt32();
    fs.Position = 34;
    BitsPerSample = br.ReadInt16();
    DataLength = (int)fs.Length - 44;
    br.Close ();
    fs.Close();
}
```

```
/*Информация о каналах содержится в заголовке WAV-файла в 22м байте, а ее размер составляет 2 байта, поэтому с помощью функции br.ReadInt16() мы считываем только эти два байта, и так далее.*/
```

```
private void WaveHeaderOUT(string sPath)
{
    FileStream fs = new FileStream(sPath, FileMode.Create,
        FileAccess.Write );
    BinaryWriter bw = new BinaryWriter(fs);
    bw.Write(new char[4] { 'R', 'I', 'F', 'F' });
    bw.Write(length);
    bw.Write(new char[8] { 'W', 'A', 'V', 'E', 'f', 'm', 't', ' ' });
    bw.Write((int)16);
    bw.Write((short)1);
}
```

```

        bw.Write(channels);
        bw.Write(samplerate );
        bw.Write((int)(samplerate * ((BitsPerSample * channels) / 8)));
        bw.Write((short )((BitsPerSample * channels) / 8));
        bw.Write(BitsPerSample);
        bw.Write(new char[4] {'d','a','t','a'});
        bw.Write(DataLength);
        bw.Close();
        fs.Close();
    }

```

```

public void Concatenate(string[] files, string outfile)

```

```

{
    WaveIO wa_IN = new WaveIO();
    WaveIO wa_out = new WaveIO();
    wa_out.DataLength = 0;
    wa_out.length = 0;

    foreach (string path in files)
    {
        wa_IN.WaveHeaderIN(@path);
        wa_out.DataLength += wa_IN.DataLength;
        wa_out.length += wa_IN.length;
    }

    wa_out.BitsPerSample = wa_IN.BitsPerSample;
    wa_out.channels = wa_IN.channels;
    wa_out.samplerate = wa_IN.samplerate;
    wa_out.WaveHeaderOUT(@outfile);

    foreach (string path in files)
    {
        FileStream fs = new FileStream(@path, FileMode.Open, FileAccess.Read);
        byte[] arrfile = new byte[fs.Length - 44];
        fs.Position = 44;
        fs.Read(arrfile, 0, arrfile.Length);
        fs.Close();
        FileStream fo = new FileStream(@outfile, FileMode.Append,
            FileAccess.Write);
        BinaryWriter bw = new BinaryWriter(fo);
        bw.Write(arrfile);
    }
}

```

```
        bw.Close();
        fo.Close();
    }
}
/* После вычисления общего размера файлов и объема данных в них и
определения всех необходимых параметров для заголовка, алгоритм
переходит к чтению информации начиная с 44го байта и записывает ее в
итоговый файл. */
```