|  |
| --- |
| **Содержание**[**ВВЕДЕНИЕ 8**](#_Toc42140639)[**4.1 МАТЕМАТИЧЕСКОЕ ОПИСАНИЕ И АЛГОРИТМЫ ВЫЯИСЛЕНИЯ СПЕКТОГРАММ АУДИОФАЙЛОВ 10**](#_Toc42140640)[**4.2 КЛАССИФИКАЦИЯ СИСТЕМ РАСПОЗНАНИЯ РЕЧИ 14**](#_Toc42140641)[**4.3 АРХИТЕКТУРА СИСТЕМ РАСПОЗНАНИЯ РЕЧИ 16**](#_Toc42140642)[**4.4 ПРИЗНАКИ ЭМОЦИОНАЛЬНО ОКРАШЕННОЙ РЕЧИ В СИСТЕМАХ РАСПОЗНОВАНИЯ 19**](#_Toc42140643)[**4.5 РАЗНОВИДНОСТИ ЦИФРОВЫХ АУДИОФОРМАТОВ 23**](#_Toc42140644)[**4.6 ОБЪЕКТНО-ОРИЕНТИРОВАННЫЕ ТЕХНОЛОГИИ 26**](#_Toc42140645)[**ПРОГРАММИРОВАНИЯ 26**](#_Toc42140646)[**4.7 ПРОЕКТИРОВАНИЕ ГРАФИЧЕСКОГО ПОЛЬЗОВА- 29**](#_Toc42140647)[**ТЕЛЬСКОГО ИНТЕРФЕЙСА СРЕДСТВАМИ COM- 29**](#_Toc42140648)[**ОБЪЕКТОВ 29**](#_Toc42140649)**5.1 ГРАФИЧЕСКИЙ ДИЗАЙН………………………………………………………………… ПРОГРАММЫ……………………………………………………………………………………**[**ЗАКЛЮЧЕНИЕ 32**](#_Toc42140650)[**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 33**](#_Toc42140651)[**Приложение А 34**](#_Toc42140652)[**(обязательное) Листинг программного кода 34**](#_Toc42140653) |
|  |

# ВВЕДЕНИЕ

 Тема данной курсовой работы: «Программа анализа спектрограмм речевых команд».

 Цель работы - проектирование пользовательского интерфейса для реализации работы с аудио форматом.

 Данная задача – довольно нетипичная задача для современного мира веб разработки, в виду обилия на рынке других продуктов, связанных со сферой услуг. Но, постепенно, данное направление начинает набирать обороты. Системы распознавания речи получают, в настоящее время, все более широкое распространение в тех приложениях, где естественный речевой диалог позволяет повысить удобство управления и обмена информацией с различными информационными системами.

На всех этапах развития техники делаются попытки возложения на машины функций выполняемых человеком. Сначала это были только наиболее трудоемкие функции, связанные с большими затратами энергии. Затем, с дальнейшим развитием техники, появилась возможность выполнения машинами различных интеллектуальных функций, т.е. в некоторых областях интеллектуальной деятельности, например обработки больших объемов информации, человек заменяется машиной. Какие бы не были простые или сложные эти технические средства всегда существует необходимость управления ими. Человек может передавать управляющее воздействие с помощью органов движения и речи. Наиболее простым оказалось решение задачи передачи управляющего воздействия с использованием рук человека. Однако более удобной и естественной для человека является речевая коммуникация. Поэтому уже много десятилетий развивается область науки связанная с проблемой организации речевого управления техническими системами. Попытка решения этой проблемы привела к формулировке и решению огромного числа задач связанных с моделированием и обработкой, передачей и распознаванием человеческой речи.

Такие системы распознавания могут потребоваться, например, для речевого запроса информации в какой-либо информационной системе; для запроса оперативной информации о состоянии объекта управления; для подачи речевых команд системе управления, в тех случаях, когда руки оператора заняты контролем других функций управляемого процесса; или в случае, когда оператор находится в скафандре и ему неудобно руками воздействовать на органы управления.

Для таких систем требуется узко специализированный словарь командного языка и контекстно-зависимая грамматика, а от блока распознавания требуется высокая достоверность идентификации команды.

Системы распознавания речи получают, в настоящее время, все более широкое распространение в тех приложениях, где естественный речевой диалог позволяет повысить удобство управления и обмена информацией с различными информационными системами. Повышение достоверности распознавания ведет, как правило, к усложнению систем распознавания и стоимости их изготовления. Вместе с тем в некоторых приложениях требуется реализовать компактную и простую систему распознавания речевых команд с небольшим словарем, которую достаточно просто реализовать аппаратно и встроить в уже имеющуюся или разрабатываемую систему.

Такие системы распознавания могут потребоваться, например:

* Для речевого запроса информации в какой-либо информационной системе;
* Для запроса оперативной информации о состоянии объекта управления;
* Для подачи речевых команд системе управления, в тех случаях, когда руки оператора заняты контролем других функций управляемого процесса; или в случае, когда оператор находится в скафандре и ему неудобно руками воздействовать на органы управления.

Для таких систем требуется узко специализированный словарь командного языка и контекстно-зависимая грамматика, а от блока распознавания требуется высокая достоверность идентификации команды.

В связи с вышеизложенным, представляется актуальнойзадача разработки алгоритмов распознавания речевого сигнала просто реализуемых аппаратно и позволяющих достигнуть высокой достоверности распознавания.

# 4.1 МАТЕМАТИЧЕСКОЕ ОПИСАНИЕ И АЛГОРИТМЫ ВЫЯИСЛЕНИЯ СПЕКТОГРАММ АУДИОФАЙЛОВ

Звук - волнообразные колебания твердых, жидких и газообразных тел воспринимаются органом слуха, ухом, в форме особого ощущения, звука. Звуковые волны распространяются от получившего удар и приведенного в дрожательное колебание тела во все стороны окружающей тела среды. Все тела по отношению к звуку делятся на проводящие и на неупругие (например - воск) и потому не проводящие звук. Обычным проводником звука является воздух. Звук распространяется со скоростью, различной для разных тел, в воздухе при 0° в 1 сек.332,5 м. Скорость распространения звука наблюдается при громе и молнии. Гром, несмотря на то, что возникает одновременно с разрядом электричества, молнией, слышен через некоторый промежуток времени, в зависимости от отдаленности места электрического разряда. Звук характеризуется высотой, силой и оттенком. Высота звука зависит от числа колебаний звучащего тела; сила звука - в данном месте есть количество звуковой энергии, проходящей в единицу времени через единицу площади, перпендикулярной к направлению распространения звука. Оттенок звука или тембр зависит от высших или гармонических тонов, сопровождающих во всяком звуке основной тон.

Приёмники звука - служат для восприятия звуковой энергии и преобразования её в другие формы. К приёмникам звука относится, в частности, слуховой аппарат человека и животных. В технике для приёма звука применяются главным образом электроакустические преобразователи: в воздухе - микрофоны, в воде - гидрофоны и в земной коре - геофоны.

Чистый звуковой тон представляет собой звуковую волну, подчиняющуюся синусоидальному закону:

у =am\* sin (wt) = аm\*sin (2пft),

где am - максимальная амплитуда синусоиды;

w - частота (w=2пf);

f - количество колебаний упругой среды в секунду (f=1\T);

Т-период;

t - время (параметрическая переменная).

Звук характеризуется частотой (f), обычно измеряемой в герцах, т.е. количеством колебаний в секунду, и амплитудой (у). Амплитуда звуковых колебаний определяет громкость звука.

Цифровой звук - аналоговый звуковой сигнал, представленный посредством дискретных численных значений его амплитуды.

Оцифровка звука - технология деления временным шагом и последующей записи полученных значений в численном виде.

Другое название оцифровки звука - аналогово-цифровое преобразование звука.

Звуковые волны при помощи микрофона превращаются в аналоговый переменный электрический сигнал. Он проходит через звуковой тракт и попадает в аналого-цифровой преобразователь (АЦП) - устройство, которое переводит сигнал в цифровую форму.

Оцифровка звука включает в себя два процесса:

* процесс дискретизации (осуществление выборки) сигнала по времени
* процесс квантования по амплитуде

При воспроизведении звука на компьютере происходит обратное преобразование цифрового сигнала в аналоговую непрерывную форму. Сигнальный процессор рассчитывает параметры звукового сигнала, а цифро-аналоговый преобразователь преобразует электрический сигнал в аналоговый, который слышен человеческому уху.

 PCM это метод, используемый для преобразования аналогового сигнала в цифровой. Импульсная кодовая модуляция имеет хорошее соотношение сигнал / шум.

PCD (плотностно-импульсная модуляция) — формат, разработанный компаниями Sony и Philips в конце прошлого века и предназначенный для архивации в цифровой форме аналоговых фонограмм.

PCM, даже в силу возраста, намного более изучен, имеет гораздо лучшую совместимость с огромным количеством самых разных устройств, подразумевает возможность редактирования (эквализации, деления на частотные полосы, преобразования). PCD — формат фактически закрытый, в нем можно записывать, его можно воспроизводить, на этом всё. Однако же, он по своей природе гораздо ближе к первоначальному аналоговому сигналу.

Аудиопоток – смешанные или не смешанные звуковые данные которое непрерывно получает пользователь от провайдера потокового вещания.

Спектр цифрового сигнала - сигнал из повторяющихся прямоугольных импульсов имеет очень широкий спектр, причем, несмотря на общую тенденцию к убыванию амплитуд, наблюдаются то увеличения, то спады гармоник на различных частотах

Математической основой спектрального анализа сигналов является преобразование Фурье.

Преобразование Фурье позволяет представить непрерывную функцию f(x) (сигнал), определенную на отрезке {0, T} в виде суммы бесконечного числа (бесконечного ряда) тригонометрических функций (синусоид и\или косинусоид) с определёнными амплитудами и фазами, также рассматриваемых на отрезке {0, T}. Такой ряд называется рядом Фурье.

Спектрогра́мма (соногра́мма) — изображение, показывающее зависимость спектральной плотности мощности сигнала от времени. Спектрограммы применяются для идентификации речи, анализа звуков животных, в различных областях музыки, радио- и гидролокации, обработке речи, сейсмологии и в других областях.

Один из важных вопросов, который является общим для всех классических методов спектрального оценивания, связан с применением функции окна. Обработка с помощью окна (windowing) используется для управления эффектами, обусловленными наличием боковых лепестков в спектральных оценках. Заметим, что термин "взвешивание" (weighting) используется как синоним термина "обработка с помощью окна". Это контрастирует с практикой тех специалистов, которые употребляют термин "окно" только применительно к преобразованиям, связанным с применением весовой функции во временной области. Введение функции окна позволяет управлять формой частотных характеристик фильтров анализатора как в полосе пропускания, так и, что особенно важно, в полосе подавления. Функция W(f) является частотным окном в том случае, когда она получена в результате дискретно-временного преобразования Фурье (ДВПФ) окна данных w(n). Окна данных называются также обуживающими функциями (tapering functions), т. е. плавно спадающими к краям. Функция W(f) обозначает спектральное окно в том случае, когда она является ДВПФ корреляционного окна w(m), применяемого к дискретно-временной автокорреляционной последовательности. Основное назначение окна данных - уменьшить величину смещения в периодограммных спектральных оценках. Основное назначение корреляционного окна - уменьшить дисперсию коррелограммной оценки СПМ.

Прямоугольное окно и БПФ, можно предложить ряд других функций окна, применение которых позволяет снизить уровень боковых лепестков по сравнению с тем их уровнем, который они имеют в случае прямоугольного окна. Снижение уровня боковых лепестков будет уменьшать смещение. Однако это дается ценой расширения главного лепестка спектра окна, что, естественно, приводит к ухудшению разрешения. Следовательно, должен выбираться какой-то компромисс между шириной главного лепестка и уровнем подавления боковых лепестков. Для классификации функций окна используется несколько показателей оценки их качества. Ширина полосы частот главного лепестка позволяет судить о частотном разрешении. Для количественной оценки ширины полосы главного лепестка используются два показателя. Традиционным показателем является ширина полосы на уровне половинной мощности, т. е. на уровне, который на 3 дБ ниже максимума главного лепестка.

**Интегральные спектральные характеристики речи.** Как следует из названия, интегральные спектральные характеристики речи характеризуют свойства речевого процесса в целом. Сюда относят:

· спектр речи  ;

· спектр формант  ;

· относительную встречаемость формант по спектру  .

Спектром речи называют оценку спектральной плотности мощности речевого сигнала  , вычисленную по отрезку речевого сигнала значительной протяженности (более минуты). Спектр речи характеризует распределение мощности речевого сигнала по частоте.

Спектром формант  называют зависимость наиболее вероятного уровня формант от частоты. Чтобы оценить спектр формант, необходимо также располагать отрезком речи большой протяженности. Спектр формант на всех частотах меньше спектра речи.

# 4.2 КЛАССИФИКАЦИЯ СИСТЕМ РАСПОЗНАНИЯ РЕЧИ

 Распознавание речи — автоматический процесс преобразования речевого сигнала в цифровую информацию (например, текстовые данные). Обратной задачей является синтез речи.

 Первое устройство для распознавания речи появилось в 1952 году, оно могло распознавать произнесённые человеком цифры. В 1962 году на ярмарке компьютерных технологий в Нью-Йорке было представлено устройство IBM Shoebox.

В 1963 году в США были презентованы разработанные инженерами корпорации «Сперри» миниатюрные распознающие устройства с волоконно-оптическим запоминающим устройством под названием «Септрон», выполняющие ту или иную последовательность действий на произнесённые человеком-оператором определённые фразы. «Септроны» годились для применения в сфере фиксированной (проводной) связи для автоматизации набора номеров голосом и автоматической записи надиктовываемого текста телетайпом, могли применяться в военной сфере (для голосового управления сложными образцами военной техники), авиации (для создания «умной авионики», реагирующей на команды пилота и членов экипажа), автоматизированных системах управления и др.

В 1983 году был презентован интерактивный комплекс «умной авионики» для ударных вертолётов «Апач», распознающий команды и запросы пилота, преобразующий их в сигналы управления на бортовое оборудование и односложно отвечающий ему голосом относительно возможности реализации поставленной им задачи.

Коммерческие программы по распознаванию речи появились в начале девяностых годов. Обычно их используют люди, которые из-за травмы руки не в состоянии набирать большое количество текста. Эти программы переводят голос пользователя в текст, таким образом, разгружая его руки. Надёжность перевода у таких программ не очень высока, но с годами она постепенно улучшается.

Увеличение вычислительных мощностей мобильных устройств позволило и для них создать программы с функцией распознавания речи. Среди таких программ стоит отметить приложение Microsoft Voice Command, которое позволяет работать со многими приложениями при помощи голоса. Например, можно включить воспроизведение музыки в плеере или создать новый документ.

Все большую популярность применение распознавания речи находит в различных сферах бизнеса, например, врач в поликлинике может проговаривать диагнозы, которые тут же будут внесены в электронную карточку. Или другой пример. Наверняка каждый хоть раз в жизни мечтал с помощью голоса выключить свет или открыть окно. В последнее время в телефонных интерактивных приложениях все чаще стали использоваться системы автоматического распознавания и синтеза речи. В этом случае общение с голосовым порталом становится более естественным, так как выбор в нём может быть осуществлен не только с помощью тонового набора, но и с помощью голосовых команд. При этом системы распознавания являются независимыми от дикторов, то есть распознают голос любого человека.

Следующим шагом технологий распознавания речи можно считать развитие так называемых интерфейсов безмолвного доступа (silent speech interfaces, SSI). Эти системы обработки речи базируются на получении и обработке речевых сигналов на ранней стадии артикулирования. Данный этап развития распознавания речи вызван двумя существенными недостатками современных систем распознавания: чрезмерная чувствительность к шумам, а также необходимость четкой и ясной речи при обращении к системе распознавания. Подход, основанный на SSI, заключается в том, чтобы использовать новые сенсоры, не подверженные влиянию шумов в качестве дополнения к обработанным акустическим сигналам.

Системы распознавания речи классифицируются:

– по размеру словаря (ограниченный набор слов, словарь большого размера);

– по зависимости от диктора (дикторозависимые и дикторонезависимые системы);

– по типу речи (слитная или раздельная речь);

– по назначению (системы диктовки, командные системы);

– по используемому алгоритму (нейронные сети, скрытые Марковские модели, динамическое программирование);

– по типу структурной единицы (фразы, слова, фонемы, дифоны, аллофоны);

– по принципу выделения структурных единиц (распознавание по шаблону, выделение лексических элементов).

Для систем автоматического распознавания речи, помехозащищённость обеспечивается, прежде всего, использованием двух механизмов:

– использование нескольких, параллельно работающих, способов выделения одних и тех же элементов речевого сигнала на базе анализа акустического сигнала;

– параллельное независимое использование сегментного (фонемного) и целостного восприятия слов в потоке речи.

# 4.3 АРХИТЕКТУРА СИСТЕМ РАСПОЗНАНИЯ РЕЧИ

Современные системы распознавания речи дают возможность пользователям диктовать слова (фразы) в обычной разговорной манере. Однако процесс непрерывного распознавания речи, дающий до 95 % качества распознавания при оптимальных условиях, все-таки дает на 100 знаков 4–5 ошибок. Около 200 ошибок на странице формата A4 — слишком много для профессиональной работы.

Как правило, система распознавания речи состоит из двух моделей: акустической и лингвистической. Компьютер записывает звук речи в виде цифрового сигнала и делит его на аудиофрагменты длительностью несколько миллисекунд. Акустическая модель отвечает за преобразование речевого сигнала в набор признаков, в которых отображена информация о содержании речевого сообщения. Программа выполняет сложный анализ речи, сравнивая аудиофрагменты с записанными в память речевыми образцами. Лингвистическая модель анализирует информацию, получаемую от акустической модели, и формирует окончательный результат распознавания. На основе вероятностного расчета компьютер определяет, что именно мог произнести пользователь. В основе модели лежит понятие фонемы — наименьшей акустической единицы языка. В процессе обучения, компьютер распознает наиболее важные признаки произношения пользователем фонем и записывает полученные данные в виде профиля пользователя. Для таких систем важно, чтобы в дальнейшем во время диктовки пользователь, по возможности, Возможности современных технологий Увеличение вычислительных мощностей мобильных устройств позволило создать для них программы с функцией распознавания речи.

Среди таких программ стоит отметить приложение Microsoft Voice Command, Siri, голосовой переводчик Google Translate, Алиса и т. д. Эти приложения способны распознавать фразы, произнесенные пользователем, и выполнять команду, либо переводить их на другие языки. Интеллектуальные речевые системы, позволяющие автоматически синтезировать и распознавать речевой сигнал, являются следующей ступенью развития интерактивных голосовых систем (IVR). Использование интерактивных телефонных приложений в настоящее время не веяние моды, а необходимость. Снижение нагрузки на операторов контакт-центров, и как следствие: сокращение расходов на оплату труда и повышение производительности систем обслуживания — вот только некоторые преимущества, доказывающие целесообразность использования подобных продуктов. Таким образом, в приложениях все чаще стали использоваться системы автоматического распознавания и синтеза речи. При этом системы распознавания являются независимыми от дикторов, то есть распознают голос любого человека.

**Проблемы развития систем распознавания речи.**

Рассмотрим некоторые аспекты, препятствующие глобальному решению проблемы качественного распознавания речи.

1. Темп речи пользователей варьируется в широких пределах, часто в несколько раз. При этом различные звуки речи растягиваются или сжимаются не пропорционально. Например, гласные изменяются значительно сильнее, чем полугласные и особенно смычные согласные. Для так называемых щелевых звуков есть свои закономерности. (Полугласные — это звуки, при генерации которых необходимо участие голосовых связок, как и для гласных звуков, но сами они в обиходе считаются согласными). Образование щелевых звуков связано с шипением и прочими эффектами турбулентности в органах артикуляции. Это свойство называется временной нестационарностью образцов речевого сигнала.

2. Произнося одно и то же слово или фразу в разное время, под влиянием различных факторов (настроения, состояния здоровья и др.), мы генерируем заметно не совпадающие спектрально-временные распределения энергии. Это справедливо даже для дважды подряд произнесенного слова. Намного сильнее этот эффект проявляется при сравнении спектрограмм одной и той же фразы, произнесенной разными людьми. Обычно этот эффект называют спектральной нестационарностью образцов речевого сигнала.

3. Изменение темпа речи и четкости произношения является причиной коартикуляционной нестационарности, означающей изменение взаимовлияния соседних звуков от образца к образцу.

4. Проблема кластеризации слитной речи: в непрерывном речевом потоке трудно распознать речевые единицы из-за неточного определения границ. Это лишь часть причин, препятствующих полной реализации систем распознавания речи.

Одна из архитектур систем автоматической обработки речи, основанной на статистических данных, может быть следующей:

1. Модуль шумоочистки и отделение полезного сигнала.
2. Акустическая модель – позволяет оценить распознавание речевого сегмента с точки зрения схожести на звуковом уровне. Для каждого звука изначально строится сложная статистическая модель, которая описывает произнесение этого звука в речи.
3. Языковая модель – позволяют определить наиболее вероятные последовательности слов. Сложность построения языковой модели во многом зависит от конкретного языка. Так, для английского языка, достаточно использовать статистические модели (так называемые N-граммы). Для высокофлективных языков (языков, в которых существует много форм одного и того же слова), к которым относится и русский, языковые модели, построенные только с использованием статистики, уже не дают такого эффекта – слишком много нужно данных, чтобы достоверно оценить статистические связи между словами. Поэтому применяют гибридные языковые модели, использующие правила русского языка, информацию о части речи и форме слова и классическую статистическую модель.
4. Декодер – программный компонент системы распознавания, который совмещает данные, получаемые в ходе распознавания от акустических и языковых моделей, и на основании их объединения, определяет наиболее вероятную последовательность слов, которая и является конечным результатом распознавания слитной речи.

# 4.4 ПРИЗНАКИ ЭМОЦИОНАЛЬНО ОКРАШЕННОЙ РЕЧИ В СИСТЕМАХ РАСПОЗНОВАНИЯ

Основные понятия, которые характеризуют параметры речи человека, связанные с формой, размерами, динамикой изменения речеобразующего тракта и описывающие эмоциональное состояния человека, можно разделить на четыре группы объективных признаков, позволяющих различать речевые образцы: спектрально-временные, кепстральные, амплитудно-частотные и признаки нелинейной динамики.

**Спектральные признаки:**

* среднее значение спектра анализируемого речевого сигнала;
* нормализованные средние значения спектра;
* относительное время пребывания сигнала в полосах спектра;
* нормализованное время пребывания сигнала в полосах спектра;
* медианное значение спектра речи в полосах;
* относительная мощность спектра речи в полосах;
* вариация огибающих спектра речи;
* нормализованные величины вариации огибающих спектра речи;
* коэффициенты кросскорреляции спектральных огибающих между полосами спектра.

**Временные признаки:**

* длительность сегмента, фонемы;
* высота сегмента;
* коэффициент формы сегмента.

**Спектрально-временные признаки характеризуют речевой сигнал в его физико-математической сущности исходя из наличия компонентов трех видов:**

* периодических (тональных) участков звуковой волны;
* непериодических участков звуковой волны (шумовых, взрывных);
* участков, не содержащих речевых пауз.

Спектрально-временные признаки позволяют отражать своеобразие формы временного ряда и спектра голосовых импульсов у разных лиц и особенности фильтрующих функций их речевых трактов. Характеризуют особенности речевого потока, связанные с динамикой перестройки артикуляционных органов речи говорящего, и являются интегральными характеристиками речевого потока, отражающими своеобразие взаимосвязи или синхронности движения артикуляционных органов говорящего.

**Кепстральные признаки:**

* мел-частотные кепстральные коэффициенты;
* коэффициенты линейного предсказания с коррекцией на неравномерность чувствительности человеческого уха;
* коэффициенты мощности частоты регистрации;
* коэффициенты спектра линейного предсказания;
* коэффициенты кепстра линейного предсказания.

Большинство современных автоматических систем распознавания речи сосредотачивают усилия на извлечении частотной характеристики речевого тракта человека, отбрасывая при этом характеристики сигнала возбуждения. Это объяснено тем, что коэффициенты первой модели обеспечивают лучшую разделимость звуков. Для отделения сигнала возбуждения от сигнала речевого тракта прибегают к кепстральному анализу.

**Амплитудно-частотные признаки:**

* интенсивность, амплитуда
* энергия
* частота основного тона (ЧОТ)
* формантные частоты
* джиттер (jitter) – дрожание частотная модуляция основного тона (шумовой параметр);
* иммер (shimmer) – амплитудная модуляция на основном тоне (шумовой параметр);
* радиальная базисная ядерная функция
* нелинейный оператор Тигер

Амплитудно-частотные признаки позволяют получать оценки, значения которых могут меняться в зависимости от параметров дискретного преобразования Фурье (вида и ширины окна), а также при незначительных сдвигах окна по выборке. Речевой сигнал акустически представляют собой распространяемые в воздушной среде сложные по своей структуре звуковые колебания, которые характеризуются в отношении их частоты (числа колебаний в секунду), интенсивности (амплитуды колебаний) и длительности. Амплитудно-частотные признаки несут необходимую и достаточную информацию для человека по речевому сигналу при минимальном времени восприятия. Но применение этих признаков не позволяет в полной мере использовать их в качестве инструмента идентификации эмоционально окрашенной речи.

**Признаки нелинейной динамики:**

* Отображение пуанкаре;
* Рекуррентный график;
* Максимальный характеристический показатель ляпунова – эмоциональное состояние человека, которому соответствует определенная геометрия аттрактора (фазовый портрет);
* Фазовый портрет (аттрактор);
* Размерность каплана-йорка – количественная мера эмоционального состояния человека, от «спокойствия» до «гнева» (деформация и последующее смещение спектра речевого сигнала).

Для группы признаков нелинейной динамики речевой сигнал рассматривается как скалярная величина, наблюдаемая в системе голосового тракта человека. Процесс речеобразования можно считать нелинейным и анализировать его методами нелинейной динамики. Задача нелинейной динамики состоит в нахождении и подробном исследовании базовых математических моделей и реальных систем, которые исходят из наиболее типичных предложений о свойствах отдельных элементов, составляющих систему, и законах взаимодействия между ними. В настоящее время методы нелинейной динамики базируются на фундаментальной математической теории, в основе которой лежит теорема Такенса (англ.)русск., которая подводит строгую математическую основу под идеи нелинейной авторегрессии и доказывает возможность восстановления фазового портрета аттрактора по временному ряду или по одной его координате. (Под аттрактором понимают множество точек или подпространство в фазовом пространстве, к которому приближается фазовая траектория после затухания переходных процессов.) Оценки характеристик сигнала из восстановленных речевых траекторий используются в построении нелинейных детерминированных фазово-пространственных моделей наблюдаемого временного ряда. Выявленные отличия в форме аттракторов можно использовать для диагностических правил и признаков, позволяющих распознать и правильно идентифицировать различные эмоции в эмоционально окрашенном речевом сигнале.

**Параметры качества речи по цифровым каналам:**

* слоговая разборчивость речи;
* фразовая разборчивость речи;
* качество речи по сравнению с качеством речи эталонного тракта;
* качество речи в реальных условиях работы.

**Основные понятия:**

* Разборчивость речи – относительное количество правильно принятых элементов речи (звуков, слогов, слов, фраз), выраженное в процентах от общего числа переданных элементов.
* Качество речи – параметр, характеризующий субъективную оценку звучания речи в испытуемой системе передачи речи.
* Нормальный темп речи – произнесение речи со скоростью, при которой средняя длительность контрольной фразы равна 2,4 с.
* Ускоренный темп речи – произнесение речи со скоростью, при которой средняя длительность контрольной фразы равна 1,5-1,6 с.
* Узнаваемость голоса говорящего – возможность слушателей отождествлять звучание голоса, с конкретным лицом, известным слушателю ранее.
* Смысловая разборчивость – показатель степени правильного воспроизведения информационного содержания речи.
* Интегральное качество – показатель, характеризующий общее впечатление слушателя от принимаемой речи.

# 4.5 РАЗНОВИДНОСТИ ЦИФРОВЫХ АУДИОФОРМАТОВ

Цифровой аудиоформат – формат представления звуковых данных, используемый при цифровой звукозаписи, а также для дальнейшего хранения записанного материала на компьютере и других электронных носителях информации, так называемых звуковых носителях.

Аудиофайл (файл, содержащий звукозапись) – компьютерный файл, состоящий из информации об амплитуде и частоте звука, сохранённой для дальнейшего воспроизведения на компьютере или проигрывателе.

Существуют различные понятия звукового формата. Формат представления звуковых данных в цифровом виде зависит от способа квантования аналогово-цифровым преобразователем (АЦП). В звукотехнике в настоящее время наиболее распространены два вида квантования:

* импульсно-кодовая модуляция;
* сигма-дельта-модуляция.

Зачастую разрядность квантования и частоту дискретизации указывают для различных звуковых устройств записи и воспроизведения как формат представления цифрового звука (24 бита/192 кГц; 16 бит/48 кГц).

Формат файла определяет структуру и особенности представления звуковых данных при хранении на запоминающем устройстве ПК. Для устранения избыточности аудиоданных используются аудиокодеки, при помощи которых производится сжатие аудиоданных. Выделяют три группы звуковых форматов файлов:

* Аудиоформаты без сжатия, такие как WAV, AIFF;
* Аудиоформаты со сжатием без потерь (APE, FLAC);
* Аудиоформаты со сжатием с потерями (MP3, Ogg).

Особняком стоят модульные музыкальные форматы файлов, созданные синтетически или из семплов заранее записанных живых инструментов. Они в основном служат для создания современной электронной музыки (MOD). Также сюда можно отнести формат MIDI, который не является звукозаписью, но с помощью секвенсора позволяет записывать и воспроизводить музыку, используя определённый набор команд в текстовом виде.

Форматы носителей цифрового звука применяют как для массового распространения звуковых записей (CD, SACD), так и в профессиональной звукозаписи (DAT, минидиск).

Для систем пространственного звучания также можно выделить форматы звука, в основном являющиеся звуковым многоканальным сопровождением к кинофильмам. Такие системы имеют целые семейства форматов, созданные двумя крупными конкурирующими компаниями.

Также форматом называют количество каналов в системах многоканального звука. Изначально такая система была разработана для кинотеатров, но впоследствии была расширена для систем домашнего кинотеатра.

Сжатие аудиофайлов происходит для того, чтобы уменьшать их размер. Например, существует некий аудиофайл длительностью 3:39. В WAV 24 бит данный файл будет занимать объём в 76 Мб, в FLAC 24 бит - 52 Мб, а в MP3 320кбит/с - всего 8 Мб. Разница значительно ощутима.

Сжатием называется процесс уменьшения цифрового потока за счёт показателей сигнала. Большинство методов сжатия основаны на несовершенстве человеческого слуха. При сжатии из аудиофайлов убираются те компоненты, которые особо неразличимы.

Конечно, существуют общества аудиофилов, которые прослушивают музыку исключительно в самом высоком качестве воспроизведения, но для некоторых это перебор. (Извиняемся, господа аудиофилы)

Существуют определённые показатели, отвечающие за качество звука:

* битность и частота дискретизации (звук без потерь)
* битрейт (звук с потерями)

Отсюда следует, чем выше битность и частота дискретизации, тем более качественным будет звук. Например, звук 32 бит и 192 кГц будет очень качественным, только разницу мало кто заметит. Услышать её можно лишь на очень качественной аппаратуре, и то не всегда.

Нынешним стандартом является 16 бит и 44,1 кГц. В DVD применяется 48 кГц, а в DVD-Audio - 96 кГц и 192 кГц.

Для систем пространственного звучания также можно выделить форматы звука, в основном являющиеся звуковым многоканальным сопровождением к кинофильмам. Такие системы имеют целые семейства форматов от двух крупных конкурирующих компаний Digital Theater Systems Inc. — DTS и Dolby Laboratories Inc. — Dolby Digital.

Также форматом называют количество каналов в системах многоканального звука (5.1; 7.1). Изначально такая система была разработана для кинотеатров, но впоследствии была расширена для систем домашнего кинотеатра.

**Список цифровых аудиоформатов:**

**Кодирование без потерь**

**Форматы без сжатия**

* AIFF
* AU
* CDDA — формат, используемый в аудио-CD
* DSD — формат, используемый в SACD
* IFF-8SVX — Interchange File Format
* IFF-16SV
* RAW — сырые замеры без какого-либо заголовка или синхронизации
* WAV — Microsoft Wave (Waveform audio format). Разработан совместно с IBM

**Сжатие звука без потерь**

* FLAC (.flac) — свободный кодек из проекта Ogg
* Lossless Audio (.la)
* Lossless Predictive Audio Compression (LPAC; .pac)
* Apple Lossless (.m4a)
* Monkey's Audio (.ape)
* OptimFROG (.ofr)[1]
* RKAU (.rka)[2]
* Shorten (.shn)
* TTA — True Audio, свободный кодек
* TAK (.tak) — (T)om’s lossless (A)udio (K)ompressor
* WavPack (.wv)
* Windows Media Audio 9 Lossless (.wma)
* ADX — формат звука с максимальным битрейтом в 1 Гбит/с

**Кодирование с потерями**

* MP2 — MPEG Layer 2
* MP3 — MPEG Layer 3
* Vorbis — проект Ogg, свободен и похож по принципам на MP3
* Speex — проект Ogg, свободен, сжатие голоса, низкий битрейт
* Opus — проект Ogg, свободен, передача звука по сети; основан на технологиях кодеков CELT и SILK.
* GSM-FR — GSM Full Rate, изначально для сотовых телефонов
* G.723.1 — один из базовых речевой кодеков IP-телефонии
* G.729 — эффективный узкополосный речевой кодек IP-телефонии
* Windows Media Audio (WMA)
* AAC (.m4a, .mp4, .m4p, .aac) — Advanced Audio Coding, часто в контейнере MPEG-4
* Musepack — Musepack (MPC)
* TwinVQ — Yamaha TwinVQ (VQF)
* RealAudio (RA, RM)
* OTS Audio File — похож на MP3
* SWA — Macromedia Shockwave Audio. Сжатие как в MP3 с дополнительными заголовками для Macromedia Director.[3]

# 4.6 ОБЪЕКТНО-ОРИЕНТИРОВАННЫЕ ТЕХНОЛОГИИ

# ПРОГРАММИРОВАНИЯ

 **Объектно-ориентированное программирование**

Объектно-ориентированное программирование (или ООП) − это парадигма программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является реализацией определенного типа, использующая механизм пересылки сообщений и классы, организованные в иерархию наследования.

Так как само определение понятия ООП содержит многие термины (ключевые для данной парадигмы), логично, что сначала нужно определить значения этих терминов. Они будут рассмотрены на примере языка С++, применённого в данном проекте.

Первое из таких понятий – это класс. **Класс** − это элемент программного обеспечения, описывающий абстрактный тип данных и его частичную или полную реализацию. Класс является составным типом. Он может содержать в себе функции и переменные (в том числе и составных типов данных). Каждый член класса называется его полем. Причем переменные в классе именуются **свойствами** класса, а функции – **методами** класса. Члены классов могут быть трёх типов: private, protected и public.

Если поле класса объявлено как private, то доступ к этому члену класса может быть осуществлён только из этого класса (его полей). По умолчанию, члены объявляются private. Объявление члена класса как public означает, что доступ к данному полю разрешен не только из этого класса, но и из любого другого места, где виден данный класс. Protected член класса имеет такую же видимость, как и private, за тем исключением, что protected член класса виден и классам-наследникам этого класса.

 Объявление класса происходит следующим образом:



 Так как объявленный класс − это тип данных, то для работы с ним нужно объявить переменную этого типа. Переменные этого типа называется объектами. **Объекты** − экземпляр класса. То, что «рождено» по «чертежу», то есть по описанию из класса.

 Объявление объекта класса осуществляется так:

myClass1 objectNumber1;

 Доступ к полям класса через операцию обращения к члену (здесь metNumb1 − это public метод класса myClass1, а если бы он был private или его не существовало – была бы ошибка):

objectNumber1.metNumb1();

 Здесь приведены лишь основные сведения касаемо классов. Помимо этого, существуют такие «фишки» объектно-ориентированного программирования, как дружественные функции и классы, статические поля класса, виртуальные методы, абстрактные классы, наследование классов и так далее.

 **Основные принципы ООП:**

* Абстракция;
* Инкапсуляция;
* Наследование;
* Полиморфизм.

**Абстрагирование** означает выделение значимой информации и исключение из рассмотрения незначимой. В ООП рассматривают лишь абстракцию данных (нередко называя её просто «абстракцией»), подразумевая набор значимых характеристик объекта, доступный остальной программе.

**Инкапсуляция** – свойство системы, позволяющее объединить данные и методы, работающие с ними, в классе. Одни языки (например, С++) отождествляют инкапсуляцию с сокрытием, но другие различают эти понятия.

**Наследование** – свойство системы, позволяющее описать новый класс на основе уже существующего с частично или полностью заимствующейся функциональностью. Класс, от которого производится наследование, называется базовым, родительским или суперклассом. Новый класс − потомком, наследником, дочерним или производным классом.

Наследование осуществляется следующим образом:

class SecondClass : public FirstClass { ...; };

**Полиморфизм** подтипов (в ООП называемый просто «полиморфизмом») – свойство системы, позволяющее использовать объекты с одинаковым интерфейсом без информации о типе и внутренней структуре объекта. Параметрический полиморфизм (свойство семантики системы типов, позволяющее обрабатывать значения разных типов идентичным образом) − в ООП называют обобщённым программированием.

Основным достоинством объектно-ориентированного программирования по сравнению с модульным программированием является "более естественная" декомпозиция программного обеспечения, которая существенно облегчает его разработку. Это приводит к более полной локализации данных и интегрированию их с подпрограммами обработки, что позволяет вести практически независимую разработку отдельных частей (объектов) программы. Кроме этого, объектный подход предлагает новые способы организации программ, основанные на механизмах наследования, полиморфизма, композиции, наполнения. Эти механизмы позволяют конструировать сложные объекты из сравнительно простых. В результате существенно увеличивается показатель повторного использования кодов и появляется возможность создания библиотек классов для различных применений.

Современная технология программирования — компонентный подход, который предполагает построение программного обеспечения из отдельных компонентов — физически отдельно существующих частей программного обеспечения, которые взаимодействуют между собой через стандартизованные двоичные интерфейсы. В отличие от обычных объектов объекты-компоненты можно собрать в динамически вызываемые библиотеки или исполняемые файлы, распространять в двоичном виде (без исходных текстов) и использовать в любом языке программирования, поддерживающем соответствующую технологию. На сегодня рынок объектов стал реальностью. Это позволяет программистам создавать продукты, хотя бы частично состоящие из повторно использованных частей, т.е. использовать технологию, хорошо зарекомендовавшую себя в области проектирования аппаратуры.

Компонентный подход лежит в основе технологий, разработанных на базе *COM (Component Object Model — компонентная модель объектов)*, и технологии создания распределенных приложений *CORBA (Common Object Request Broker Architecture — общая архитектура с посредником обработки запросов объектов)*. Эти технологии используют сходные принципы и различаются лишь особенностями их реализации.

Технология СОМ фирмы Microsoft является развитием технологии *OLE (Object Linking and Embedding — связывание и внедрение объектов)*, которая использовалась в ранних версиях Windows для создания составных документов. Технология СОМ определяет общую парадигму взаимодействия программ любых типов: библиотек, приложений, операционной системы, т. е. позволяет одной части программного обеспечения использовать функции (службы), предоставляемые другой, независимо от того, функционируют ли эти части в пределах одного процесса, в разных процессах на одном компьютере или на разных компьютерах. Модификация СОМ, обеспечивающая передачу вызовов между компьютерами, называется *DCOM (Distributed COM — распределенная СОМ)*.

Таковы, вкратце, основные этапы изменения технологий программирования. Поскольку программное обеспечение на сегодняшний день используется в самых различных устройствах и сферах деятельности человека, то можно прогнозировать дальнейшее совершенствование технологий программирования.

# 4.7 ПРОЕКТИРОВАНИЕ ГРАФИЧЕСКОГО ПОЛЬЗОВА-

# ТЕЛЬСКОГО ИНТЕРФЕЙСА СРЕДСТВАМИ COM-

# ОБЪЕКТОВ

Для проектирования графического пользовательского интерфейса использовалась библиотека Swing языка Java.

Java - сильно типизированный объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems (в последующем приобретённой компанией Oracle). Приложения Java обычно транслируются в специальный байт-код, поэтому они могут работать на любой компьютерной архитектуре с помощью виртуальной Java-машины. Дата официального выпуска - 23 мая 1995 года. На 2018 год Java - один из самых популярных языков программирования.

Программы на Java транслируются в байт-код Java, выполняемый виртуальной машиной Java (JVM) - программой, обрабатывающей байтовый код и передающей инструкции оборудованию как интерпретатор.

Достоинством подобного способа выполнения программ является полная независимость байт-кода от операционной системы и оборудования, что позволяет выполнять Java-приложения на любом устройстве, для которого существует соответствующая виртуальная машина. Другой важной особенностью технологии Java является гибкая система безопасности, в рамках которой исполнение программы полностью контролируется виртуальной машиной. Любые операции, которые превышают установленные полномочия программы (например, попытка несанкционированного доступа к данным или соединения с другим компьютером), вызывают немедленное прерывание.

Часто к недостаткам концепции виртуальной машины относят снижение производительности. Ряд усовершенствований несколько увеличил скорость выполнения программ на Java:

* применение технологии трансляции байт-кода в машинный код непосредственно во время работы программы (JIT-технология) с возможностью сохранения версий класса в машинном коде,
* широкое использование платформенно-ориентированного кода (native-код) в стандартных библиотеках,
* аппаратные средства, обеспечивающие ускоренную обработку байт-кода (например, технология Jazelle, поддерживаемая некоторыми процессорами архитектуры ARM).

По данным сайта shootout.alioth.debian.org, для семи разных задач время выполнения на Java составляет в среднем в полтора-два раза больше, чем для C/C++, в некоторых случаях Java быстрее, а в отдельных случаях в 7 раз медленнее. С другой стороны, для большинства из них потребление памяти Java-машиной было в 10-30 раз больше, чем программой на C/C++. Также примечательно исследование, проведённое компанией Google, согласно которому отмечается существенно более низкая производительность и бо́льшее потребление памяти в тестовых примерах на Java в сравнении с аналогичными программами на C++.

Идеи, заложенные в концепцию и различные реализации среды виртуальной машины Java, вдохновили множество энтузиастов на расширение перечня языков, которые могли бы быть использованы для создания программ, исполняемых на виртуальной машине. Эти идеи нашли также выражение в спецификации общеязыковой инфраструктуры CLI, заложенной в основу платформы .NET компанией Microsoft.

Swing - библиотека для создания графического интерфейса для программ на языке Java. Swing был разработан компанией Sun Microsystems. Он содержит ряд графических компонентов (англ. Swing widgets), таких как кнопки, поля ввода, таблицы и т. д.

Swing относится к библиотеке классов JFC, которая представляет собой набор библиотек для разработки графических оболочек. К этим библиотекам относятся Java 2D, Accessibility-API, Drag & Drop-API и AWT.

Архитектура Swing разработана таким образом, что вы можете изменять «look and feel[en]» (L&F) вашего приложения. «Look» определяет внешний вид компонентов, а «Feel» - их поведение. Sun’s JRE предоставляет следующие L&F[1]:

1. CrossPlatformLookAndFeel - это родной L&F для Java-приложений (так же называется Metal). Он используется по умолчанию, обеспечивая стандартное поведение компонентов и их внешний вид, вне зависимости от платформы, на которой запускается приложение.

2. SystemLookAndFeel - в этом случае приложение использует L&F, который является родным для системы, на которой запущено приложение. Системный L&F определяется во время выполнения. Для Windows используется «Windows» L&F, который имитирует особенности конкретной системы, на которой запущен - классический Windows, XP, или Vista. Для Linux и Solaris используется «GTK+», если установлен GTK+ 2.2 или более поздняя версия, в противном случае используется «Motif».

3. Synth - основа для создания собственных L&F.

4. Multiplexing - предоставляет возможность использования различных L&F одновременно.

Можно сделать вывод, что данная библиотека довольно мощная и имеет обширный функционал для создания современных десктопных GUI приложений для среды Windows (в основном), но не только (среды Linux, Android и т.д.). На этом описание пользовательского интерфейса можно считать завершённым.

# ЗАКЛЮЧЕНИЕ

 В работе был предложен метод разномасштабного описания речевого сигнала.

Эти методы были реализованы в виде алгоритмов, позволяющих получить спектральное представление речевых команд, сформировать эталоны и выполнить сравнение текущей команды с эталоном. Для настройки параметров и исследования эффективности работы алгоритмов был разработан программный комплекс, с помощью которого была проведена серия экспериментов.

В работе показана возможность эффективного применения аппарата пакетного преобразования для анализа речевого сигнала.

Показана также важность использования в системах распознавания информации о сегментации речевого сигнала.

Разработанный в данной работе метод получения информации переходах между соседними звуками речи может применяться при автоматическом сегментировании обучающих речевых баз данных.

Метод получения информации о переходах между звуками может быть использован при реализации системы распознавания речи основанной на теории А.А. Пирогова о фонетической функции речи, в которой говорится о том, что человек воспринимает речевые элементы по их относительным, а не абсолютным спектрам. Выделяя стационарные сегменты и переходы между ними можно затем вычислять усредненные значения относительных спектров стационарных сегментов. Использование такого подхода, возможно позволит распознавать искаженную каналами связи речь, а также речь произнесенную шепотом.

 По данному проекту можно сделать несколько **выводов.**

 Во-первых, были изучены алгоритмы решения данной задачи на практике, с помощью средств языка Java.

 Во-вторых, было изучено построение пользовательского интерфейса в рамках среды InteliJ IDEA на основе библиотеки Swing, изучены практически все встроенные средства и возможности данной системы.

 В-третьих, усовершенствованы навыки программирования на языке программирования Java и навыки применения ООП при реализации.

 Таким образом, в проекте реализованы все поставленные ранее задачи, а сам проект имеет доступный интерфейс. В процессе работы приобретены новые знания и умения.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1] MQTT – Википедия. [Электронный ресурс]. – Режим доступа : <https://ru.wikipedia.org/wiki/MQTT>.

[2] Распознавание речи – Википедия. [Электронный ресурс]. – Режим доступа : https://ru.wikipedia.org/wiki/Ethernet.

[3] Руководство по языку программирования Java. [Электронный ресурс]. – Режим доступа https://metanit.com/java/tutorial/.

[4] java.com: Java и вы. [Электронный ресурс]. – Режим доступа : https://www.java.com/ru.

[5] Java Swing Учебник: примеры для создания графического интерфейса. [Электронный ресурс]. – Режим доступа : <https://guru99.ru/java-swinggui>.

[6] JAVA+Swing. Стоит ли? [Электронный ресурс]. – Режим доступа : https://habr.com/ru/post/179459/.

# Приложение А

# (обязательное)Листинг программного кода

<!DOCTYPE html>

<html>

 <head>

 <meta http-equiv="Content-Type" content="text/html; charset=utf-8">

 <title>AudioSpectrum in JavaScript</title>

 <style>

 /\*СТИЛИ\*/

 body{

 background: url('https://img5.goodfon.ru/original/2560x1440/d/27/minimalizm-naushniki-fon-goluboi-rozovyi.jpg');

 -webkit-background-size: 1300px;

 background-size: 1300px;

 }

 .wrapper{

 padding-top: 120px;

 padding-left: 120px;

 padding-right: 120px;

 }

 .bl, .br{

 display: -webkit-flex;

 display: -moz-flex;

 display: -ms-flex;

 display: -o-flex;

 display: flex;

 justify-content: center;

 margin-top: 10px;

 }

 input[type=button], input[type=submit], input[type=reset] {

 background-color: #9AB9B3;

 color: white;

 padding: 16px 32px;

 text-decoration: none;

 margin: 4px 2px;

 cursor: pointer;

 }

 input[type=text]{

 height: 40px;

 width: 300px

 }

 </style>

 </head>

 <body>

 <div class="wrapper">

 <!-- div, где будут отображаться звуковые волны -->

 <div id="audio-spectrum"></div>

 <!-- кнопки действий -->

 <div class="bl">

 <input type="button" id="btn-stop" value="Next" disabled="disabled" />

 <input type="button" id="btn-play" value="Play" disabled="disabled"/>

 <input type="button" id="btn-pause" value="Pause" disabled="disabled"/>

 <input type="button" id="btn-stop" value="Stop" disabled="disabled" />

 <input type="button" id="btn-stop" value="Next" disabled="disabled" />

 </div>

 <div class="br">

 <input type="text" placeholder="Type something..." id="myInput">

 <button type="button" onclick="getInputValue();">Get Value</button>

 </div>

 </div>

 <!-- Загрузить waveurferscript, в данном случае из CDN -->

 <script src="https://cdnjs.cloudflare.com/ajax/libs/wavesurfer.js/1.3.7/wavesurfer.min.js"></script>

 <script>

 // Сохраняем 3 кнопки в каком-то объекте

 var buttons = {

 play: document.getElementById("btn-play"),

 pause: document.getElementById("btn-pause"),

 stop: document.getElementById("btn-stop")

 };

 // Создать экземпляр Wave Surfer с его конфигурацией

 var Spectrum = WaveSurfer.create({

 container: '#audio-spectrum',

 progressColor: "#03a9f4"

 });

 // Кнопка воспроизведения

 buttons.play.addEventListener("click", function(){

 Spectrum.play();

 // Включить / отключить кнопки

 buttons.stop.disabled = false;

 buttons.pause.disabled = false;

 buttons.play.disabled = true;

 }, false);

 // Кнопка Пауза

 buttons.pause.addEventListener("click", function(){

 Spectrum.pause();

 // Включить / отключить кнопки

 buttons.pause.disabled = true;

 buttons.play.disabled = false;

 }, false);

 // Кнопка «Стоп»

 buttons.stop.addEventListener("click", function(){

 Spectrum.stop();

 // Включить / отключить кнопки

 buttons.pause.disabled = true;

 buttons.play.disabled = false;

 buttons.stop.disabled = true;

 }, false);

 //Добавьте слушателя, чтобы включить кнопку воспроизведения, когда он будет готов

 Spectrum.on('ready', function () {

 buttons.play.disabled = false;

 });

 window.addEventListener("resize", function(){

 // Получить текущий прогресс в соответствии с положением курсора

 var currentProgress = Spectrum.getCurrentTime() / Spectrum.getDuration();

 // Сбросить график

 Spectrum.empty();

 Spectrum.drawBuffer();

 // Установить исходную позицию

 Spectrum.seekTo(currentProgress);

 // Включить / отключить кнопки

 buttons.pause.disabled = true;

 buttons.play.disabled = false;

 buttons.stop.disabled = false;

 }, false);

 function getInputValue(){

// Выбираем элемент ввода и получаем его значение

 var inputVal = document.getElementById("myInput").value;

// Отображение значения

 alert(inputVal);

 Spectrum.load(inputVal);

 }

 </script>

 </body>

</html>