Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Глазовский государственный педагогический институт имени В.Г. Короленко»

**КУРСОВАЯ РАБОТА**

**Использование технологии речевого синтеза в мобильном приложении**

**Специальность 09.02.03 Программирование в компьютерных системах**

Выполнил:

Студент ОП СПО группа 33

Шаталов Кирилл Романович

Руководитель курсовой работы: ст. преподаватель кафедры

Математики и информатики Касаткин К.А

г. Глазов 2019

**Оглавление**

[Введение 3](#_Toc7525189)

[§ 1. Мобильные технологии 4](#_Toc7525190)

[§ 2. Инструменты для создания приложений 11](#_Toc7525191)

[§ 3. Создание мобильного приложения 17](#_Toc7525203)

[Заключение 19](#_Toc7525204)

[Список литературы 20](#_Toc7525205)

[Приложение 22](#_Toc7525206)

# Введение

Актуальность: Вопросы синтеза и распознавания речи человека компьютером становятся все более актуальными. Речевые технологии уже внедряются в нашу жизнь. Успехи в развитии естественно-языковых технологий обещают широкий доступ к on-line информации и электронным сервисам. Так как почти каждый говорит и понимает речь, развитие естественно-языковых систем позволит человеку без специальных навыков общаться с компьютером в любое время и в любом месте без дополнительного обучения, используя такие устройства, как мобильный телефон, получать доступ к информации или к управлению устройствами. Cотовый телефон так же можно использовать как устройство перевода речи в речь, которое распознает сказанную вами фразу на одном из трех языков – японском, английском и немецком, и переводит на любой их этих языков. В настоящее время технология распознавания речи является очень востребованной технологией в связи с экономией времени, экономических средств и доступностью для обычных пользователей. Также данная технология позволяет повысить производительность набора поисковых запросов путем ввода их при помощи речевого синтеза. Для этого требуется всего лишь правильно разработанная программа с возможностью голосового поиска [1-6, 8, 9, 12, 13 15].

Цель: разработка мобильного приложения с поддержкой речевого синтеза.

Задача: научиться разрабатывать мобильные приложения с использованием технологии TTS (Text To Speech).

# § 1. Мобильные технологии

Мобильные устройства — это такие устройства как смартфоны, интернет планшеты, электронные книги, телефоны, КПК, нетбуки. Главная их особенность – это, конечно же размер, а так же количество выполняемых ими функций. В эпоху тотальной миниатюризации мобильные устройства вытесняют так называемые «стационарные»: на смену компьютерам приходят планшеты и смартфоны, умные часы, очки виртуальной реальности и прочая «мобильная», переносная, легкая техника [18].

Электронные книги по характеру напоминают планшеты, однако они узко специализированы. Основной их задачей считается чтение книг и электронных файлов. Эти мобильные устройства основаны на матрице e-ink, которая по своим свойствам имитирует обычную бумагу, т.е. экран не имеет подсветки и на вид глазом воспринимается как обычный лист бумаги.

Время автономной работы электронных книг составляет от трех до десяти дней. Далее следуют смартфоны, которые работают автономно от одного до трех дней, а планшеты работают не более 10 часов в зависимости от интенсивности нагрузки.

Рассмотрим структуру смартфонов, т.е. их составляющие, которые позволяют обеспечить широкие функциональные возможности [19].

**Аппаратное обеспечение**. Наиболее значимым элементом смартфонов является процессор, тактовая частота которых в настоящее время в некоторых моделях достигает 1 ГГц. Еще 10 лет назад подобные процессоры были обычными явлением для ПК, а их применение в мобильных устройствах было исключено из-за высокого потребления энергии. Современные процессоры обладают на порядки более низким энергопотреблением и, соответственно, тепловым выделением. Высокопроизводительные процессоры позволяют использовать сложное ПО реального времени с красочными визуальными эффектами.

В состав аппаратного обеспечения смартфонов обычно входит яркий дисплей с поддержкой сотен тысяч цветов. Это позволяет воспроизводить качественные видеоролики и даже просматривать полнометражные фильмы, а также запускать игры по сложности соизмеримые с теми, которые запускают на игровых приставках. Многие современные смартфоны оснащаются сенсорными дисплеями, которые значительно расширяют их мультимедийные возможности. По некоторым оценкам в 2011 году уже каждый третий телефон будет обладать сенсорным экраном.

Неотъемлемым атрибутом любого смартфона является емкая небольшая батарея, т.к. она определяет возможность автономной работы «умного» устройства, что является очень важной характеристикой. Современные батареи обычно Li-Ion (литий-ионные) с емкостью от 0,7 до 1,5 А⋅ч. Причем для увеличения времени автономной работы развитие ведется не только в сторону увеличения емкости, но и снижения энергопотребления компонентами телефона.

Кроме этого, большинство смартфонов оснащается камерами, позволяющими не только фотографировать, но и снимать видеоролики. Специальные мультимедийные смартфоны могут обладать камерой с матрицей 12 Mpx и более, а также записывать видеоролики с разрешением 1024х768.

Смартфон также может быть оснащен различными датчиками, например, освещенности (для изменения яркости дисплея), акселерометром (для изменения ориентации дисплея) и т.п. Часть смартфонов оснащается приемником GPS для работы различных приложений и сервисов, учитывающих местоположение абонента. Обычно смартфоны оборудуются набором различных модулей для связи с другими устройствами и телекоммуникационными сетями: Инфракрасный порт, Bluetooth, WiFi и др [16].

**Программное обеспечение.** Для того, чтобы все описанное выше аппаратное обеспечение могло взаимодействовать и предоставляло интерфейс для пользователя, прикладных приложений и других устройств необходимо надежное программное обеспечение. В отличие от Software, предназначенного для ПК к ПО для смартфонов предъявляются некоторые специфические требования, связанные с их телекоммуникационной и мобильной составляющими. Рассмотрим программное обеспечение в виде стека уровней:

1. Kernel – система управления процессами и драйвера для работы с аппаратным обеспечением

2. Middleware – библиотеки программного обеспечения, которые позволяют работать приложениям

3. Application execution environment (AEE) – программные интерфейсы, которые предоставляют доступ для работы пользовательских приложений.

4. User interface framework – набор дополнительных библиотек и ПО, которое обеспечивает доступ к пользовательским интерфейсам, например, к экрану.

5. Application suite – прикладные приложения пользователя.

Главной составляющей ПО смартфона является операционная система (ОС), которая объединяет уровни с 2 по 5 из вышеперечисленного стека. На сегодняшний день количество операционных систем не меньше чем для ПК. Некоторые производители смартфонов разрабатывают собственные ОС, а другие полагаются на уже существующие успешные разработки. Перечислим наиболее распространенные ОС для смартфонов [10]:

* Android
* Java
* Linux
* OS X
* Palm OS
* Symbian
* Windows Mobile

**Android («Андро́ид»)** — операционная система для смартфонов, интернет-планшетов, электронных книг, цифровых проигрывателей, наручных часов, игровых приставок, нетбуков, смартбуков, очков Google, телевизоров и других устройств. В будущем планируется поддержка автомобилей и бытовых роботов. Основана на ядре Linux и собственной реализации виртуальной машины Java от Google. Изначально разрабатывалась компанией Android, Inc., которую затем купила Google. Впоследствии Google инициировала создание альянса Open Handset Alliance (OHA), который сейчас занимается поддержкой и дальнейшим развитием платформы. Android позволяет создавать Java-приложения, управляющие устройством через разработанные Google библиотеки. Android Native Development Kit позволяет портировать библиотеки и компоненты приложений, написанные на C, C++, Java.

Приложения под операционную систему Android являются программами в нестандартном байт-коде для виртуальной машины Dalvik, для них был разработан формат установочных пакетов .APK. Для работы над приложениями доступно множество библиотек: Bionic (библиотека стандартных функций, несовместимая с glibc); мультимедийные библиотеки на базе PacketVideo OpenCORE (поддерживают такие форматы, как MPEG-4, H.264, MP3, AAC, AMR, JPEG и PNG); SGL (движок двухмерной графики); OpenGL ES 1.0 ES 2.0 (движок трёхмерной графики); Surface Manager (обеспечивает для приложений доступ к 2D/3D); WebKit (готовый движок для веб-браузера; обрабатывает HTML, JavaScript); FreeType (движок обработки шрифтов); SQLite (легковесная СУБД, доступная для всех приложений); SSL (протокол, обеспечивающий безопасную передачу данных по сети). По сравнению с обычными приложениями Linux приложения Android подчиняются дополнительным правилам:

* Content Providers — обмен данными между приложениями;
* Resource Manager — доступ к таким ресурсам, как файлы XML, PNG, JPEG;
* Notification Manager — доступ к строке состояния;
* Activity Manager — управление активными приложениями.

Google предлагает для свободного скачивания инструментарий для разработки (Software Development Kit), который предназначен для x86-машин под операционными системами Linux, Mac OS X (10.4.8 или выше), Windows XP, Windows Vista и Windows 7. Для разработки требуется JDK 5 или более новый.

Разработку приложений для Android можно вести на языке Java (не ниже Java 1.5). Существует плагин для Eclipse — Android Development Tools (ADT), предназначенный для Eclipse версий 3.3—3.7. Также существует плагин для IntelliJ IDEA, облегчающий разработку Android-приложений, и для среды разработки NetBeans IDE, который, начиная с версии NetBeans 7.0, перестал быть экспериментальным, хоть пока и не является официальным. Кроме того, существует Motodev Studio for Android — комплексная среда разработки на базе Eclipse, позволяющая работать непосредственно с Google SDK.

В 2009 году в дополнение к ADT был опубликован Android Native Development Kit (NDK) — пакет инструментариев и библиотек, позволяющий реализовать часть приложения на языке С/C++. NDK рекомендуется использовать для разработки участков кода, критичных к скорости.

В 2013 году Google представила новую среду разработки Android Studio, основанную на IntelliJ IDEA от JetBrains.

В 2013-м году состоялся релиз Embarcadero RAD Studio — XE5. Возможность разработки нативных приложений для платформы Android. Процесс создания Android приложения не требует дополнительных устройств, кроме, собственно, Android устройства (в принципе, можно обойтись и эмулятором).

В Android 4.4 появилась возможность сменить виртуальную машину Dalvik на ART (Android Runtime). ART отличается повышенной скоростью загрузки приложения. Проработан механизм оптимизации памяти.

В Android 5 перерисован дизайн, базирующийся на концепции Material Design, добавлен режим энергосбережения Project Volta, выбор машины пропал, вместо Dalvik стала использоваться ART.

В Android 6 появился интеллектуальный режим расхода энергии Doze и запрет выхода в интернет и работы в фоне давно не используемых приложений App Standby.

Достоинства Android:

* Открытость, благодаря чему на ней можно реализовать больше полезных функций.
* Установка программ без подключения к интернету.
* Доступность для разных аппаратных платформ.
* Наличие альтернативных магазинов приложений.
* Поддержка многопользовательского режима (с Android3).
* Наличие альтернативных прошивок.

Критика Android:

* В некоторых Android-устройствах есть сервисы Google, обеспечивающие возможность передачи идентификационной информации на серверы компании, например, информации о перемещении пользователя в реальном времени.
* Возможность установки программ из «непроверенных источников» способствует пиратству на платформе Android.
* В версии Android 1.6 разработчики добавили Native Development Kit, который позволяет писать собственные низкоуровневые модули для системы на языках C/C++, опираясь на стандартные Linux-библиотеки. Хотя, например, стандартная библиотека языка C на платформе Android, известная как Bionic, как раз не является стандартной и полностью совместимой с libc.
* Для доступа к Google Play и другим сервисам от Google необходимо использовать проприетарные приложения, которые производитель телефона имеет право устанавливать на телефон только после заключения контракта с Google.

Конкуренты Android выступили с критикой платформы, обвиняя её в чрезмерной фрагментации, создающей препятствия разработчикам. Google опровергла все обвинения, заявив, что никаких подобных проблем нет.

* Подверглось критике решение Google не размещать в широком доступе код Android 3.0 Honeycomb, доступный только для участников Open Handset Alliance или по индивидуальному запросу после подписания соглашения. Google мотивирует это неготовностью платформы и мерой предупреждения небрежной её реализации. Ричард Столлман заявлял, что «всё просто и ясно: за исключением ядра Linux, Android 3 представляет собой несвободный софт» и «несмотря на то, что телефоны с Android на сегодняшний день не так плохи, как смартфоны Apple или Windows, нельзя сказать, что они уважают вашу свободу». В настоящий момент Google открыл исходный код для всех версий Android вплоть до 6.0, а также отправил в Linux все изменения, в соответствии с GPL.
* По данным Lookout Security Mobile, за 2011 год у пользователей Android-смартфонов было украдено около миллиона долларов США (напр., отправка СМС без ведома владельца телефона) [20].

# § 2. Инструменты для создания приложений

Идеальное мобильное приложение — это многофункциональное, простое и удобное решение. Для его создания требуется меньше времени чем для полноценного сайта, хотя сегодняшние достижения в разработке красноречиво подтверждают — мобильные приложения ничем не хуже ни по удобству, ни по выполняемым функциям. Если вы готовы приступить к их созданию, то вам только нужно продумать такие вещи как: среда разработки приложений, набор инструментов, программы-помощники и откуда черпать информацию.

Если вы не хотите погрязнуть в тонкостях разработки, а предпочитаете готовые решения, то ваш выход — конструктор. Сегодня их появляется все больше и больше, поэтому сделать что-то стандартное и незамысловатое для мобильного трафика не составит труда.

### 1 Xamarin

Начнем с тяжеловесов. Этот конструктор не для начинающих, поскольку требует минимального знания C#. Из плюсов — подходит для приложений как Android или iOS, а также содержит полезные внутренние метрики для отслеживания поведения пользователей.

### 2 jQuery Mobile

Если вам нужно интерактивное, кросс-платформенное и адаптивное приложение, то jQuery — ваше решение. Огромное количество модулей позволят реализовать любую по сложности задумку, а разнообразные новинки от сообщества (преимущество открытого кода) и вовсе идеальное поле для творчества.

### 3 Android Studio

Комплексная кроссплатформенная среда для разработки приложений на Android. По функционалу разве что не летает. Помимо инструментов для самого приложения, вы можете рассчитывать на полный набор полезных штук от Google (Аналитика, интеграция с облачными хранилищами и т.д.).

### 4 Como

Неплохая, средняя по функциональным возможностям платформа для разработки мобильных приложений, которая подходит для решений среднего и малого бизнеса в сфере общепита, развлечений и др.

### 5 Appery.io

Для тех, кто нуждается в кросс-платформенном и гибком в настройке решении. Своим главным преимуществом компания называет скорость создания приложения и, с учетом приемлемого качества на выходе, так оно и есть. Очень хорошо подходит для проработки бэк-энда.

### 6 Bizness Apps

Есть разделение по специфике бизнеса, но только для маленьких и уже проработанных сегментов. Если вы “в списке”, то можете рассчитывать на шаблоны со всеми необходимыми инструментами для дальнейшей оптимизации. Подходит для Android и iOS.

### 7 Dojo Mobile

Для тех, кто в курсе HTML, JavaScript и CSS. Порадуйтесь количество виджетов, готовых тем и оцените кросс-платформенность.

### 8 GameSalad

Поговорим и об игровых приложениях, поскольку инструменты разработки мобильных приложений и здесь не отстают в своих функциональных возможностях.

### 9 AppMachine

Достаточно известный и проверенный временем конструктор, который предлагает нативное, кросс-платформенное приложение с хорошей внутренней аналитикой. Можно добавить пуш-уведомления, продвигаться и тестировать — словом, заменить полноценную команду девелоперов (конечно, нет).

### 10 AppMakr

Качественный конструктор, который может похвастаться реализацией любой задумки на достойном уровне. Кросс-платформенность, адаптивность, быстрая обработка контента, множество модулей и самое главное — очень развитое сообщество адептов.

**Android Studio —**это интегрированная среда разработки (IDE) для работы с платформой Andoid, анонсированная 16 мая 2013 года на конференции Google I/O.

IDE находилась в свободном доступе начиная с версии 0.1, опубликованной в мае 2013, а затем перешла в стадию бета-тестирования, начиная с версии 0.8, которая была выпущена в июне 2014 года. Первая стабильная версия 1.0 была выпущена в декабре 2014 года, тогда же прекратилась поддержка плагина Android Development Tools (ADT) для Eclipse.

Android Studio, основанная на программном обеспечении IntelliJ IDEA от компании JetBrains, – официальное средство разработки Android приложений. Данная среда разработки доступна для Windows, OS X и Linux. 17 мая 2017, на ежегодной конференции Google I/O, Google анонсировал поддержку языка Kotlin, используемого в Android Studio, как официального языка программирования для платформы Android в добавление к Java и С++.

Новые функции появляются с каждой новой версией Android Studio. На данный момент доступны следующие функции:

* Расширенный редактор макетов: WYSIWYG, способность работать с UI компонентами при помощи Drag-and-Drop, функция предпросмотра макета на нескольких конфигурациях экрана.
* Сборка приложений, основанная на Gradle.
* Различные виды сборок и генерация нескольких .apk файлов
* Рефакторинг кода
* Статический анализатор кода (Lint), позволяющий находить проблемы производительности, несовместимости версий и другое.
* Встроенный ProGuard и утилита для подписывания приложений.
* Шаблоны основных макетов и компонентов Android.
* Поддержка разработки приложений для Android Wear и Android TV.
* Встроенная поддержка Google Cloud Platform, которая включает в себя интеграцию с сервисами Google Cloud MEssaging и App Engine.
* Android Studio 2.1 поддерживает Android N Preview SDK, а это значит, что разработчики смогут начать работу по созданию приложения для новой программной платформы.
* Новая версия Android Studio 2.1 способна работать с обновленным компилятором Jack, а также получила улучшенную поддержку Java 8 и усовершенствованную функцию Instant Run.
* Начиная с Platform-tools 23.1.0 для Linux исключительно 64-разрядная.
* В Android Studio 3.0 будут по стандарту включены инструменты языка Kotlin основанные на JetBrains IDE [14].

**Язык программи́рования** — формальный язык, предназначенный для записи компьютерных программ. Язык программирования определяет набор лексических, синтаксических и семантических правил, определяющих внешний вид программы и действия, которые выполнит исполнитель (обычно — ЭВМ) под её управлением.

Со времени создания первых программируемых машин человечество придумало более восьми тысяч языков программирования (включая эзотерические, визуальные и игрушечные). Каждый год их число увеличивается. Некоторыми языками умеет пользоваться только небольшое число их собственных разработчиков, другие становятся известны миллионам людей. Профессиональные программисты могут владеть десятком и более разных языков программирования.

Язык программирования предназначен для написания компьютерных программ, которые представляют собой набор правил, позволяющих компьютеру выполнить тот или иной вычислительный процесс, организовать управление различными объектами, и т. п. Язык программирования отличается от естественных языков тем, что предназначен для управления ЭВМ, в то время как естественные языки используются, прежде всего, для общения людей между собой. Большинство языков программирования использует специальные конструкции для определения и манипулирования структурами данных и управления процессом вычислений.

Как правило, язык программирования определяется не только через спецификации *стандарта языка*, формально определяющие его синтаксис и семантику, но и через *воплощения (реализации) стандарта* — программные средства, обеспечивающих трансляцию или интерпретацию программ на этом языке; такие программные средства различаются по производителю, марке и варианту (версии), времени выпуска, полноте воплощения стандарта, дополнительным возможностям; могут иметь определённые ошибки или особенности воплощения, влияющие на практику использования языка или даже на его стандарт.

### Языки низкого и высокого уровня

Обычно под «уровнем языка» понимается:

* степень отличия семантики языка от машинного кода целевой архитектуры процессора — другими словами, наименьший масштаб преобразований, которые должен претерпеть код программы перед тем, как он сможет исполняться (зачастую с существенной потерей эффективности)
* степень, в которой семантика языка учитывает особенности мышления человека, нежели машины — то есть уровень языка тем «ниже», чем он «ближе к машине», и тем «выше», чем он «ближе к человеку».

К языкам низкого уровня относят, в первую очередь, машинные языки (или, на общеупотребимом жаргоне — машинные коды), то есть языки, реализованные непосредственно на аппаратном уровне. Их относят к **первому поколению языков программирования**. Вскоре после них появились **языки второго поколения** — так называемые «языки ассемблера». В простейшем случае они реализуют мнемонику над машинным языком для записи команд и их параметров (в частности, адресов в памяти). Кроме того, многие языки ассемблера включают и весьма развитый макроязык. Языки *первого* и *второго* поколения позволяют точно контролировать, *как* требуемая функциональность будет исполняться на данном процессоре с учётом особенностей его архитектуры. С одной стороны, это обеспечивает высокое быстродействие и компактность программ, но с другой, для переноса программы на другую аппаратную платформу её нужно перекодировать (а часто из-за различий архитектуры процессоров — и перепроектировать) с нуля. Большинство языков ассемблера являются бестиповыми, но существуют и типизированные языки ассемблера, нацеленные на обеспечение минимальной безопасности низкоуровневых программ [22].

# § 3. Создание мобильного приложения

Главной целью курсовой работы является создание мобильного приложения с использованием технологии TTS.

MainActivity.java:

packageru.krll.texttospeech;  
  
importandroid.speech.tts.TextToSpeech;  
importandroid.support.v7.app.AppCompatActivity;  
importandroid.os.Bundle;  
importandroid.widget.Button;  
importandroid.widget.EditText;  
importandroid.widget.Toast;  
  
importjava.util.Locale;  
  
public classMainActivity extendsAppCompatActivity {  
  
 EditText textEdit;  
 Button playButton;  
 TextToSpeech tts;  
  
 @Override  
 protected voidonCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_main);  
  
 textEdit = findViewById(R.id.textEdit);  
 playButton = findViewById(R.id.playButton);  
  
 playButton.setOnClickListener(v -> {  
 String text = textEdit.getText().toString();  
  
 tts = newTextToSpeech(this, status -> {  
 if(status == TextToSpeech.SUCCESS) {  
 if(tts != null) {  
 intresult;  
if(tts.isLanguageAvailable(newLocale(Locale.getDefault().getLanguage()))  
 == TextToSpeech.LANG\_AVAILABLE) {  
 result = tts.setLanguage(newLocale(Locale.getDefault().getLanguage()));  
 } else{  
 result = tts.setLanguage(Locale.US);  
 }  
  
 if (result == TextToSpeech.LANG\_MISSING\_DATA || result ==  
 TextToSpeech.LANG\_NOT\_SUPPORTED) {  
 Toast.makeText(this, "TTS language is not supported",  
 Toast.LENGTH\_LONG).show();  
 } else {  
 tts.speak(text, TextToSpeech.QUEUE\_ADD, null);  
 }  
 }  
 } else {  
 Toast.makeText(this, "TTS initialization failed",  
 Toast.LENGTH\_LONG).show();  
 }  
 });  
 });  
 }  
}

AndroidManifest.xml:

*<?*xml version="1.0" encoding="utf-8"*?>*<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="ru.krll.texttospeech">  
  
<application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/AppTheme">  
<activity android:name=".MainActivity">  
<intent-filter>  
<action android:name="android.intent.action.MAIN" />  
  
<category android:name="android.intent.category.LAUNCHER" />  
</intent-filter>  
</activity>  
</application>  
  
</manifest>

# Заключение

Была поставлена задача изучить мобильные технологии, инструменты и архитектуру мобильной операционной системы, инструменты разработки и особенности применения библиотеки синтеза речи.

Была изучена информация о мобильных технологиях и их разновидности.

Более подробно изучена операционная система Android, какие языки программирования поддерживаются и на каких языках была написана.

Изучены наиболее популярные инструменты для создания приложений, более углубленно изучались инструменты для создания приложений на операционной системе Android.

Было выбрана среда разработки Android Studio, на которой и начинался разрабатываться проект.

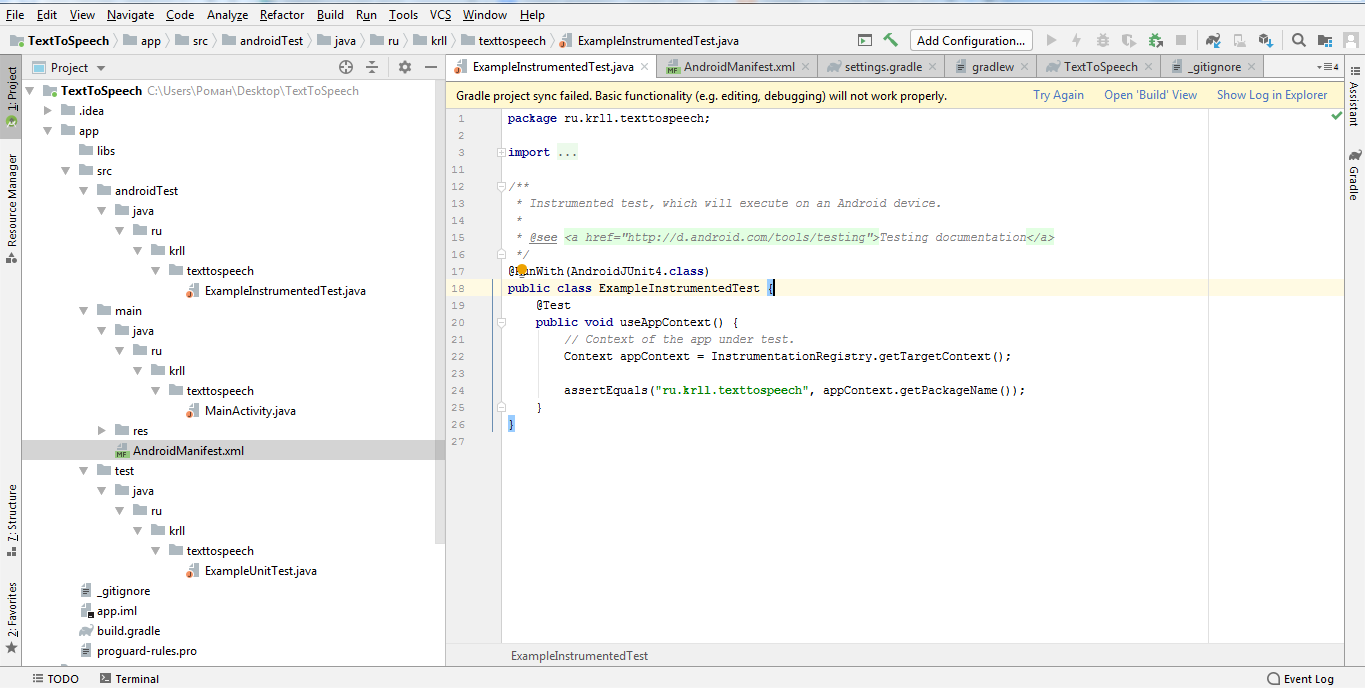
Были разобраны языки программирования, их виды и характеристики, подразделения на низкие и высокие языки программирования.

В ходе всей этой проделанной работы было создано мобильное приложение на языке программирования с Java, в среде разработки Android Studio, с применением библиотеки TTS (Text To Speech), которая предназначена для синтеза речи из текста.

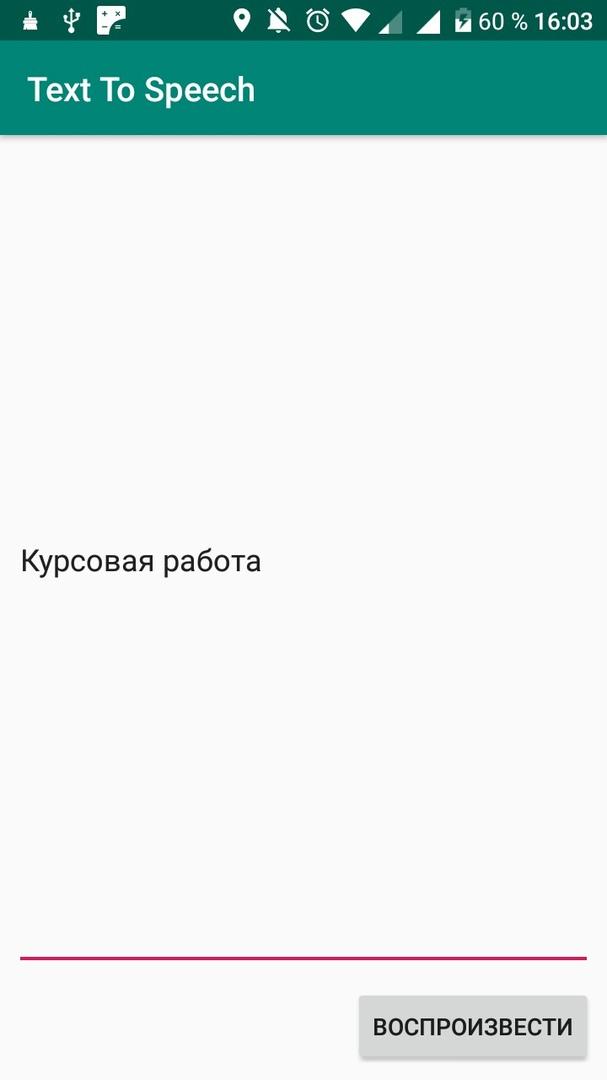
# Список литературы

1. Бондаренко В.П., Квасов А.Н., Конев А.А., Мещеряков Р.В., Чойнзонов Е.Л., Чижевская C.Ю. Программные средства комплекса исследования речевого сигнала при злокачественных заболеваниях гортани // Медицинская техника. – 2009. – № 4. – С. 33–37.
2. Бондаренко В.П., Мещеряков Р.В. Диалог как основа построения речевых систем // Кибернетика и системный анализ. – 2008. – № 2. – С. 30–41.
3. Гренандер У. Лекции по теории образов: Т. 1. Синтез образов / под ред. Ю.И. Журавлева; пер. с англ. – М.: Мир, 1979. – 383 с.
4. Дмитриев Л.Б., Теляева Л.М., Таптапова С.Л., Ермакова И.И. Фониатрия и фонопедия. – М.: Медицина, 1990. – 272 с.
5. Златоустова Л.В., Потапова Р.К., Трунин-Донской В.Н. Общая и прикладная фонетика. – М.: Издво МГУ, 1986. – 304 с.
6. Математическая лингвистика. Сб. переводов / под ред. Ю.А. Шрейдера, И.И. Ревзина, Д.Г. Лахути, В.К. Финна. – М.: Мир, 1964. – 144 с.
7. Мещеряков Р.В., Бондаренко В.П., Организация баз знаний в системе синтеза речи // Теория и практика речевых исследований (АРСО99). Матер. конф. – М.: МГУ им. М.В. Ломоносова, 1999. – С. 37–38.
8. Сорокин В.Н. Синтез речи. – М.: Наука, 1992. – 392 с.
9. Потапова Р.К. Речевое управление роботом. – М.: Радио и связь, 1989. – 246 с.
10. Потапова Р.К. Речь: коммуникация, информация, кибернетика. – М.: Радио и связь, 1997. – 528 с.
11. Трунин-Донской В.Н. Автоматический синтез звучащего текста // Звучащий текст. – М.: Институт научной информации по общественным наукам, 1983. – С. 218–250.
12. Филичева Т.Б., Чевелева Н.А., Чиркина Г.В. Основы логопедии. – М.: Просвещение, 1989. – 223 с.
13. Флейшман Б.С. Элементы теории потенциальной эффективности сложных систем. – М.: Советское радио, 1971. – 223 с.
14. Android Studio [электронный ресурс] — URL:https://ru.wikipedia.org/wiki/Android\_Studio
15. Актуальность речевого синтеза. [электронный ресурс] —http://www.dialog-21.ru/media/2729/rusanova.pdf
16. Аппаратное обеспечение [электронный ресурс] — URL: http://celnet.ru/smartphones.php
17. Инструменты для создания приложений [электронный ресурс] — URL: https://artjoker.ua/ru/blog/luchshie-instrumenty-dlya-sozdaniya-mobilnykh-prilozheniy/
18. Мобильные устройства [электронный ресурс] — URL: https://hi-news.ru/tag/mobilnye-ustrojstva
19. Мобильные устройства [электронный ресурс] — URL: https://www.securitylab.ru/news/tags/%EC%EE%E1%E8%EB%FC%ED%FB%E5+%F3%F1%F2%F0%EE%E9%F1%F2%E2%E0/
20. Операционная система Android [электронный ресурс] — URL: https://ru.wikipedia.org/wiki/Android
21. Синтез и распознавание речи. Современные решения [электронный ресурс] — 2016 г. — URL: http://www.frolov-lib.ru/books/hi/ch07.html
22. Языки программирования [электронный ресурс] — URL: https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA\_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F

# Приложение



Разработанный проект в Android Studio



Созданное мобильное приложение Text To Speech

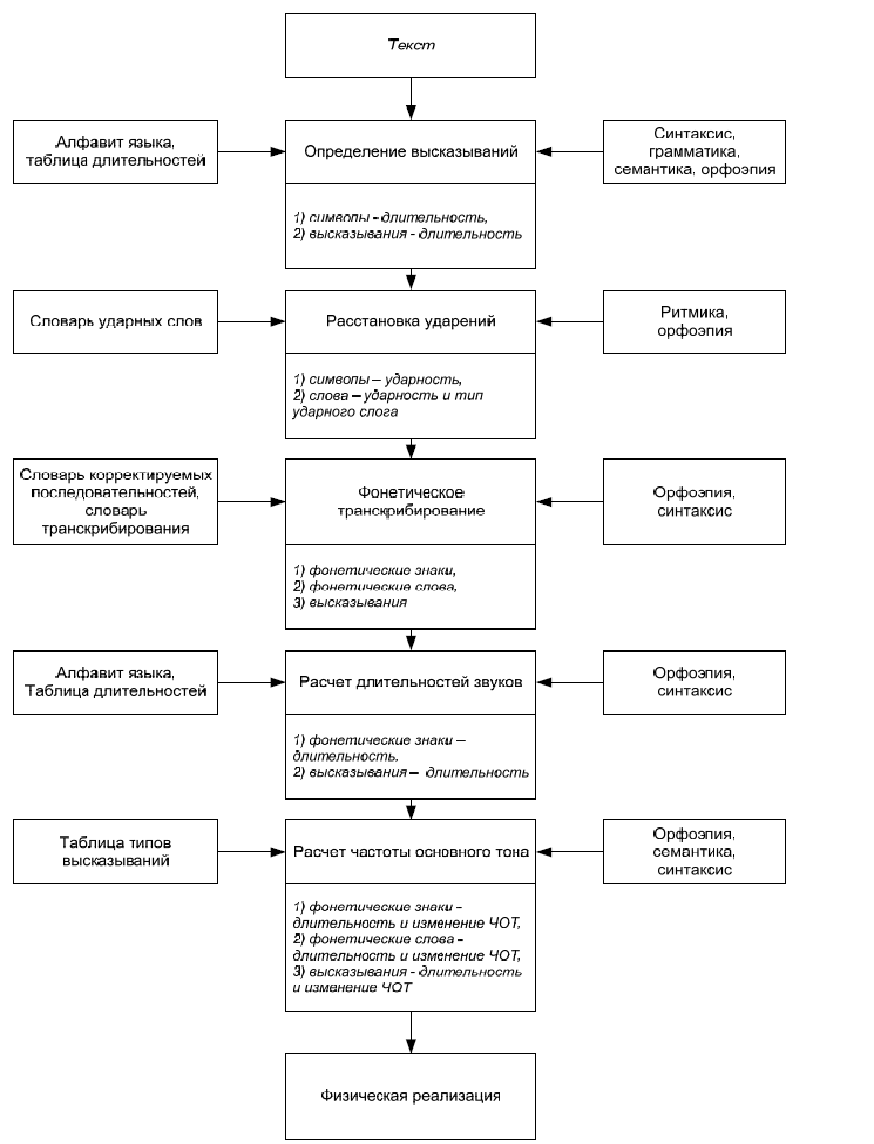


Схема построения модели управления [7]

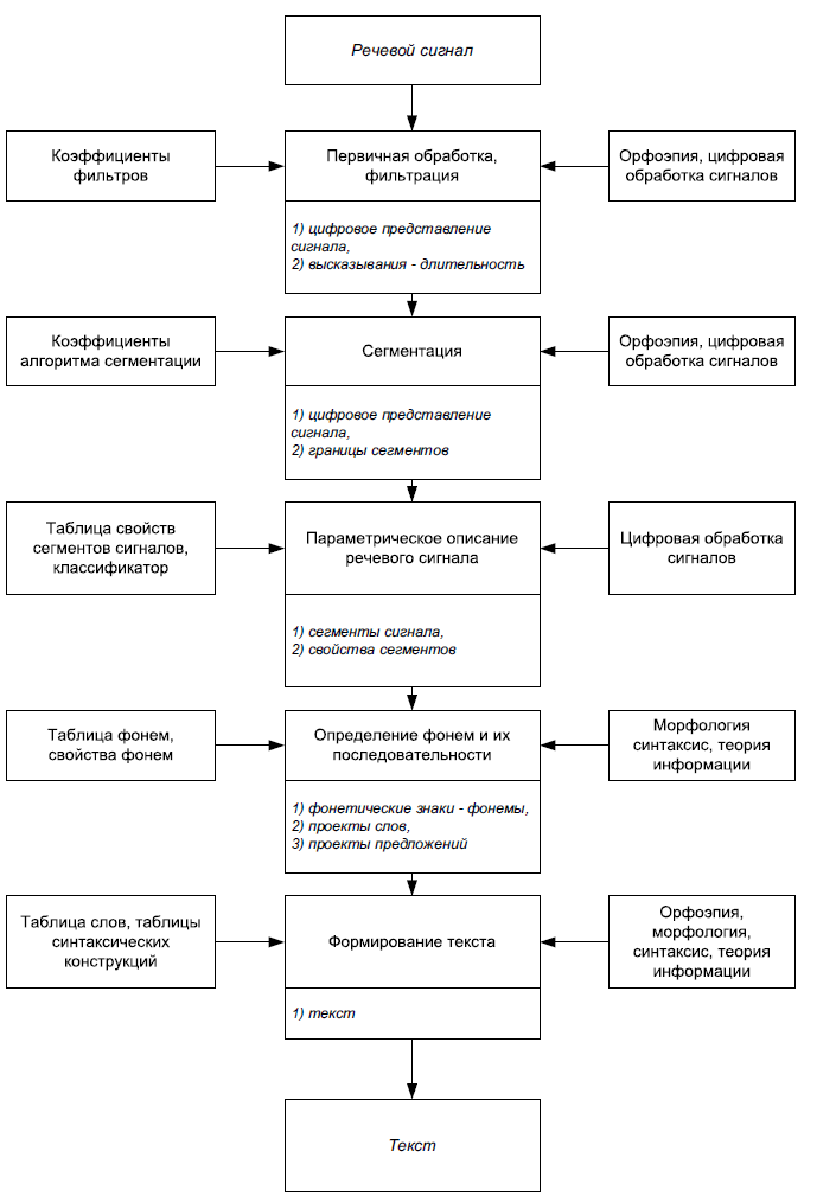


Схема распознавания речевого сигнала для получения печатного текста [11]