

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»

Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

ДОПУСТИТЬ К ЗАЩИТЕ
Заведующий кафедрой ЭВМ
_____ Д.В. Топольский
«___» _____ 2022 г.

РАЗРАБОТКА ЭКРАННОЙ КЛАВИАТУРЫ БРАЙЛЯ ДЛЯ МОБИЛЬНЫХ
УСТРОЙСТВ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
ЮУРГУ-090301.2022.220 ПЗ ВКП

Руководитель работы,
к.пед.н., доцент каф. ЭВМ
_____ М.А. Алтухова
«___» _____ 2022 г.

Автор работы,
студент группы КЭ-405
_____ А.В. Мелехин
«___» _____ 2022 г.

Нормоконтролёр,
к.пед.н., доцент каф. ЭВМ
_____ М.А. Алтухова
«___» _____ 2022 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

УТВЕРЖДАЮ

Заведующий кафедрой ЭВМ

_____ Д.В. Топольский

«___» _____ 2022 г.

ЗАДАНИЕ

на выпускную квалификационную работу бакалавра

студенту группы КЭ-405

Мелехину Артему Викторовичу

обучающемуся по направлению

09.03.01 «Информатика и вычислительная техника»

1) Тема работы: «Разработка экранной клавиатуры Брайля для мобильных устройств» утверждена приказом по университету от «___» _____ 202___ г. № _____

2) Срок сдачи студентом законченной работы: «___» _____ 2022 г.

3) Исходные данные к работе.

Приложение должно работать на ОС Android и обеспечивать выполнение функций:

- ввод текста с помощью азбуки Брайля из 6 символов;
- автоматический запуск клавиатуры при возникновении ситуации ввода текста;
- возможность быстрой смены раскладки с кириллицы на латиницу без перезапуска клавиатуры;
- возможность аудиоконтроля вводимых символов и слов;

– минимальный набор приемов быстрого редактирования с помощью жестов.

4) Перечень подлежащих разработке вопросов:

- рассмотрение существующих программных решений, реализующих экранные клавиатуры Брайля;
- рассмотрение оптимальных сред разработки;
- проектирование и разработка собственного программного приложения экранной клавиатуры Брайля;
- тестирование разработанного программного продукта.

5) Дата выдачи задания: «___» _____ 2022 г.

Руководитель работы _____ / М.А. Алтухова /

Студент _____ / А.В. Мелехин /

КАЛЕНДАРНЫЙ ПЛАН

Этап	Срок сдачи	Подпись руководителя
Введение и обзор литературы (рассмотрение программных решений, рассмотрение оптимальных сред разработки)	07.02.2022	
Разработка модели, проектирование	07.03.2022	
Реализация системы	04.04.2022	
Тестирование, отладка, эксперименты	10.05.2022	
Компоновка текста работы и сдача на нормоконтроль	16.05.2022	
Подготовка презентации и доклада	24.05.2022	

Руководитель работы _____ / М.А. Алтухова /

Студент _____ / А.В. Мелехин /

АННОТАЦИЯ

А.В. Мелехин. Разработка экранной клавиатуры Брайля для мобильных устройств. – Челябинск: ФГАОУ ВО «ЮУрГУ (НИУ)», ВШЭКН; 2022, 44 с., 16 ил., библиогр. список –8 наим.

В рамках выпускной квалификационной работы были рассмотрены существующие программные решения, реализующие экранные клавиатуры Брайля. Проанализированы среды разработки. Спроектировано и разработано собственное программное приложение. По завершению этапа разработки выполнено альфа-тестирование.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	7
1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	9
1.1 Обзор существующих аналогичных решений	9
1.2 Анализ основных технологических решений	12
1.2.1 Выбор среды разработки	12
1.2.2 Описание языка программирования	13
2 ОСНОВНЫЕ ТРЕБОВАНИЯ.....	15
2.1 Основные требования к функционалу системы.....	15
2.2 Основные нефункциональные требования.....	15
3 ПРОЕКТИРОВАНИЕ	16
3.1 Основные сведения	16
3.2 Проектирование концепции системы	17
3.3 Проектирование структуры интерфейса.....	18
4 РЕАЛИЗАЦИЯ	20
5 ТЕСТИРОВАНИЕ	22
5.1 Тестирование появления клавиатуры	22
5.2 Тестирование корректности ввода	24
5.3 Тестирование корректности ввода жестов	25
ЗАКЛЮЧЕНИЕ	27
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	28
ПРИЛОЖЕНИЕ А Исходный код	29

ВВЕДЕНИЕ

Многие люди имеют разного вида физиологические особенности, из-за которых им трудно коммуницировать с окружающим миром. Для них создается безбарьерная среда, что позволяет свободно передвигаться и взаимодействовать наравне со всеми [1]. Разработчикам программного обеспечения необходимо учитывать запросы таких пользователей, включая специальные возможности в функционал приложений.

В 1824 году французом Луи Брайлем был разработан рельефно-точечный тактильный шрифт, предназначенный для письма и чтения незрячими и плохо видящим людям. Идея заключается в кодировании букв и других символов на гладкой поверхности с помощью выпуклых точек, расположенных на определённых позициях. Для каждой буквы выделяется шесть позиций — две колонки по три позиции в каждой. Наличие выпуклых точек в той или иной позиции и задаёт код буквы. Незрячие люди трогают поверхность пальцами и «считывают» буквы. Для людей с проблемами зрения алфавит Брайля, состоящий из рельефно-точечного шрифта, является единственной возможностью получить образование и полноценно участвовать в жизни общества. Незрячих людей обучают «слепому» чтению тифлопедагоги. Сегодня шрифт Брайля применяется не только в специальных книгах, но и для помощи печати на мобильном устройстве [2].

Идея о создании виртуальной клавиатуры Брайля не нова, и это вполне закономерно, так как данный метод печати является самым быстрым и удобным. В сравнении с голосовым набором можно отметить меньшее количество ошибок и неточностей, так как исключается влияние шумов, мешающих алгоритмам распознавания речи. Стоит отметить то, что находящиеся вокруг пользователя люди не будут слышать информацию, которую он набирает на своем мобильном устройстве. Именно поэтому уже сегодня существуют клавиатуры такого вида.

По данным статистического сервиса statcounter большинство пользователей сегодня использует мобильные устройства под операционными

системами iOS (Apple) или Android (Google) [3]. Разработчики этих операционных систем уже внедрили клавиатуру Брайля в заводскую конфигурацию. Но, так как это произошло относительно недавно, данная возможность имеет ряд замечаний. Основная проблема для русскоязычных пользователей состоит в том, что в стандартном приложении нет возможности печати на кириллице, либо нет быстрого переключения раскладки (раскладку клавиатуры возможно поменять только при перезапуске виртуальной клавиатуры).

Актуальной задачей является разработка виртуальной клавиатуры Брайля, которая должна иметь:

- русскую и английскую раскладку;
- возможность быстро переключать раскладки;
- быструю настройку.

Цель работы.

Выпускная квалификационная работа посвящена разработке виртуальной клавиатуры Брайля, позволяющей русскоязычным пользователям с ограниченными возможностями печатать текст на мобильном устройстве.

Задачи работы:

- изучить принцип печати текста по алфавиту Брайля;
- найти существующие на данный момент аналоги системы;
- провести анализ с целью выявления преимуществ и недостатков этих систем, сделать выводы;
- составить техническое задание и эскизный проект системы;
- выполнить программную реализацию проекта;
- определить методы тестирования и провести тестирование разработанного приложения на корректность выполнения задач, удобство интерфейса.

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Обзор существующих аналогичных решений

На сегодняшний день существуют несколько реализаций виртуальной клавиатуры Брайля. Они имеют разные проблемы, например, отсутствие русской раскладки или затруднительное переключение раскладки. Подробно разберем примеры, выявим достоинства и недостатки. Хорошие решения аналогов реализуем в проекте, а недочеты исключим.

Клавиатура Брайля talkback присутствует в заводской конфигурации на всех смартфонах под операционной системой Android. Чтобы использовать клавиатуру Брайля, необходимо отвести экран смартфона от себя и держать смартфон в горизонтальном положении впереди. Таким образом, задняя часть смартфона будет ориентирована лицом вверх, и руки будут держать устройство за оба коротких конца, порт USB-C должен быть справа от нас. Для букв есть шесть областей на экране, показанные на рисунке 1.1 кружками, которые позволяют вводить слова, используя различные комбинации. Например, коснувшись цифр один и пять, можно написать символ «Е» [4].

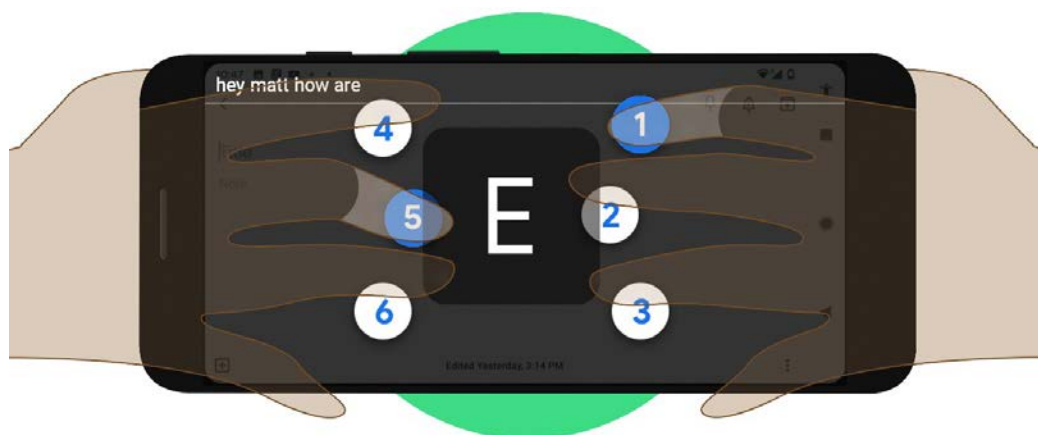


Рисунок 1.1 – Интерфейс клавиатуры от Google

К достоинствам этой клавиатуры можно отнести то, что в ней есть система жестов, которые позволяют редактировать текст.

Недостатком является отсутствие поддержки кириллицы.

Клавиатура Брайля от Apple, также как и в мобильных устройствах под операционной системой Android, добавлена в заводскую конфигурацию. В отличие от аналога в ней имеется русская раскладка. Кроме того, она дает возможность набора текста, когда устройство расположено на горизонтальной поверхности. Настольный режим включается, когда устройство находится перед пользователем и активные зоны на экране размещаются как в печатной машинке Брайля (рисунок 1.2). Данный режим удобен только для планшетов.

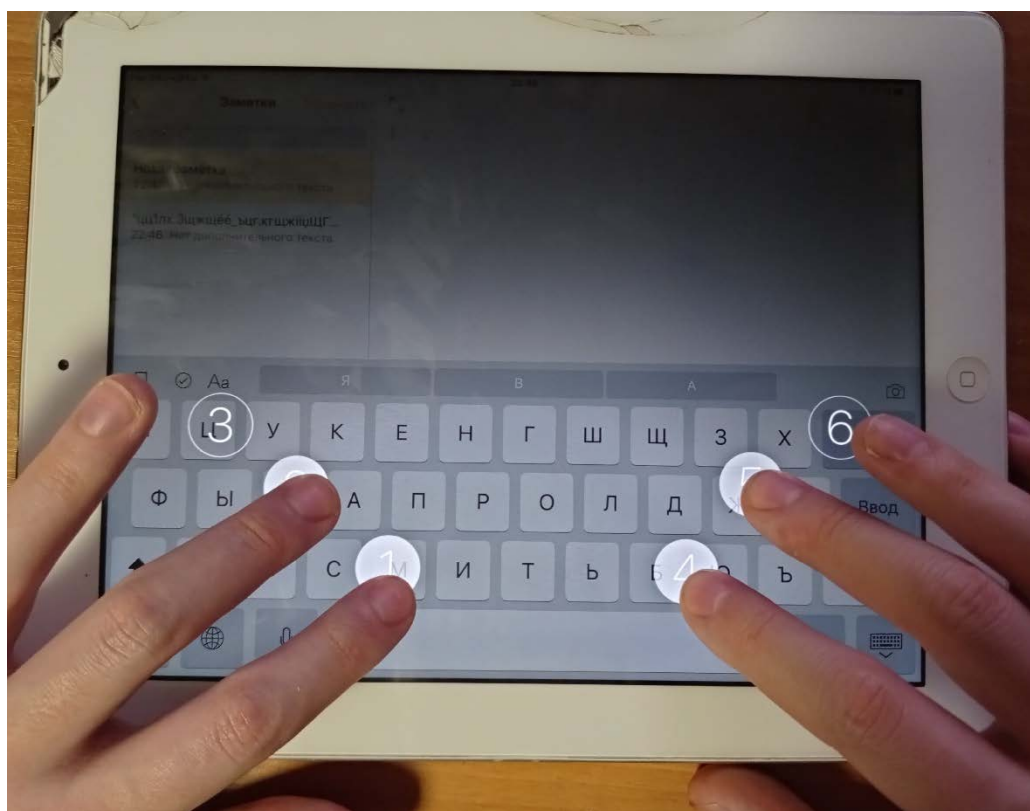


Рисунок 1.2 – Настольный режим шрифта Брайля

К достоинствам данной клавиатуры можно отнести то, что в ней есть несколько режимов набора текста, русская раскладка и система жестов для редактирования набранного текста.

Недостаток заключается в том, что нет быстрого способа переключения раскладки клавиатуры, для этого требуется закрывать клавиатуру и при новом запуске выбирать раскладку.

Также на площадке Google Play есть несколько приложений:

- 1) Braillic: Braille Keyboard;

- 2) Braille Keyboard;
- 3) ASB Braille Keyboard;
- 4) Advanced Braille Keyboard.

Сравнительный анализ приведен в таблице 1.1.

Условные обозначения: «+» - критерий есть у сайта, «-» - критерий отсутствует.

Таблица 1.1 – Сравнительный анализ существующих аналогов

Название приложения	Наличие русской раскладки	Наличие настольного режима	Наличие системы жестов	Возможность быстро менять раскладку	Предоставляется бесплатно
Клавиатура talkback (Android)	-	+	+	-	+
Клавиатура от Apple	+	+	+	-	+
Braillic: Braille Keyboard	-	-	+	-	+
Braille Keyboard	-	-	+	-	+
ASB Braille Keyboard	-	-	-	-	+
Advanced Braille Keyboard	+	-	+	-	-

Проанализировав аналоги, можно сделать вывод о том, что у большинства приложений отсутствует возможность ввода кириллицы, а быстро поменять раскладку невозможно ни в одном приложении. Для удобства пользования русскоязычным пользователям, разрабатываемая клавиатура должна иметь:

- возможность ввода русских и латинских символов;
- возможность быстрого переключения между раскладками;
- наличие системы жестов для редактирования текста;
- должно быть полностью бесплатным для пользователя.

1.2 Анализ основных технологических решений

1.2.1 Выбор среды разработки

Самыми популярными средами разработки под операционную систему Android являются:

- Microsoft Xamarin;
- Eclipse;
- Android Studio.

Microsoft Xamarin — это платформа с открытым исходным кодом, предназначенная для разработки производительных приложений для iOS, Android и Windows с .NET. Платформа Xamarin представляет собой уровень абстракции, который обеспечивает управление взаимодействием между общим кодом и кодом базовой платформы. Xamarin выполняется в управляемой среде, которая реализует такие возможности, как выделение памяти и сборка мусора. Приложения, написанные с помощью Xamarin и C#, имеют полный доступ к интерфейсам API базовой платформы и возможность создавать нативные пользовательские интерфейсы, а также компилировать код в машинный, поэтому влияние на производительность во время выполнения является незначительным [5].

Однако из-за невысокой популярности данной среды разработки и предполагаемых проблем с поиском информации от нее пришлось отказаться.

Eclipse является бесплатной программной платформой с открытым исходным кодом, контролируется организацией Eclipse Foundation [6].

Написана на языке программирования Java и основной целью её создания является повышение продуктивности процесса разработки программного обеспечения. Eclipse сама по себе не является средой разработки приложений для мобильных устройств, но к ней можно подключить отдельный плагин ADT (Android Development Tools). Все же, данная среда разработки не ориентирована

конкретно на разработку мобильных приложений, поэтому возможны различные проблемы на этапе разработки, которых можно избежать, выбрав другую среду разработки.

Android Studio — это интегрированная среда разработки для работы с платформой Android [7]. Она популярна среди разработчиков под ОС Android и имеет ряд положительных особенностей. Вот некоторые из них:

- способность работать с графическим интерфейсом и компонентами при помощи Drag-and-Drop;
- 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;
- шаблоны основных макетов и компонентов Android;
- Android Studio поддерживает Android N Preview SDK, что позволяет начать работу по созданию приложения для новой программной платформы.

Также, данная среда разработки, как следует из названия, ориентирована на разработку приложений именно под ОС Android, что является существенным преимуществом перед другими средами разработки.

В итоге, после анализа существующих сред разработки, была выбрана среда разработки Android Studio, так как она подходит под нужды разрабатываемого приложения, а также в интернете имеется много информации по ее настройке и работе с ней.

1.2.2 Описание языка программирования

Для создания мобильных приложений для платформы Android используется язык программирования Java. Однако при компиляции программы основной Java код преобразуется в нестандартный байт-код и используется дополнительный инструмент Android SDK (Software Development Kit) [8].

Возможности языка:

- расширенные обработки исключений;
- большой набор средств ввода-вывода;
- существование классов, которые позволяют выполнять запросы HTTP и обрабатывать ответы;
- встроенные инструменты для функционального программирования;
- возможность использования лямбд-выражений;
- возможность управлять памятью автоматически.

Особенности языка:

- наличие аппаратных средств, которые обеспечивают быструю обработку байт-кода;
- широкое использование машинного кода (платформенно-ориентированного языка) в стандартных библиотеках;
- реализация технологий трансляции байтового кода в машинный во время работы программы с возможностью сохранения версий класса в нативном коде.

Еще одним преимуществом является относительно высокая безопасность. По той причине, что все программы выполняются и управляются виртуальной машиной Java, и программа просто прервет операцию, если кто-то попытается выполнить попытку несанкционированного доступа или соединения.

Для реализации клавиатуры Брайля будет использоваться язык программирования Java и интегрированная среда разработки Android Studio.

2 ОСНОВНЫЕ ТРЕБОВАНИЯ

2.1 Основные требования к функционалу системы

На этапе проектирования системы были выявлены следующие функциональные требования:

- система должна выполнять функцию считывания нажатых кнопок на экране, и жестов;
- система должна выполнять функцию печати нужного символа в строке ввода;
- система должна выполнять функцию оповещения пользователя о каком-либо действии с клавиатурой (ввод, удаление, переключение регистра).

2.2 Основные нефункциональные требования

Разрабатываемое приложение должно быть полностью бесплатным.

Взаимодействие с приложением должно быть простым и понятным, без лишних элементов, которые могут помешать пользователю вводить текст.

Приложение не должно требовать и собирать личные данные пользователя.

3 ПРОЕКТИРОВАНИЕ

3.1 Основные сведения

Идея шрифта Брайля заключается в кодировании букв и других символов на гладкой поверхности с помощью выпуклых точек, расположенных на определённых местах. Для представления каждой буквы выделяется шесть позиций, организованных в виде матрицы 3x2. Наличие выпуклых точек в том или ином положении и задаёт код буквы. Незрячие люди ощупывают поверхность пальцами и «считывают» буквы. Кодировка символов представлено на рисунке 3.1.

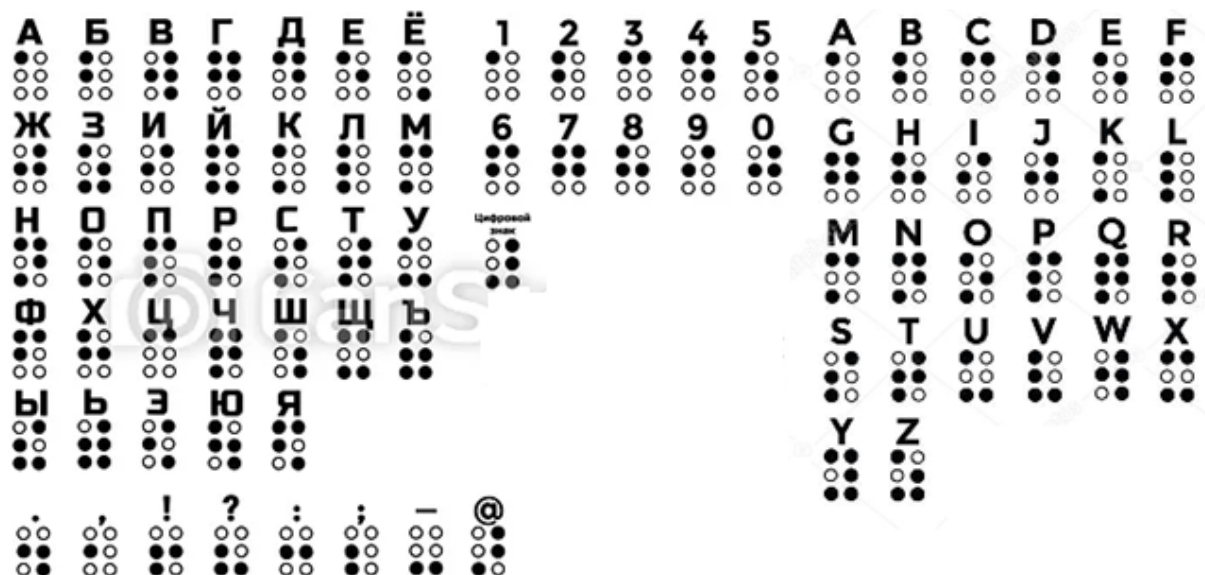


Рисунок 3.1 – Кодировка символов

В таблице кодировки можно заметить, что некоторые буквы и цифры кодируются одинаковой комбинацией точек. Поэтому для написания цифр применяется специальный символ – «цифровой знак». Он печатается перед числом и после него. Перед прописной буквой ставится символ – «прописная буква», а перед латинской буквой символ – «латинская буква».

3.2 Проектирование концепции системы

Программа будет разделяться на следующие подсистемы:

- подсистема обработки нажатий – выполняет функцию считывания нажатых кнопок на экране и жестов;
- подсистема вывода символа на экран – выполняет функцию печати нужного символа в строке ввода;
- подсистема аудио контроля – выполняет функцию оповещения пользователя о каком-либо действии с клавиатурой (ввод, удаление, переключение регистра).

На рисунках 3.2 и 3.3 показана функциональная модель программы.

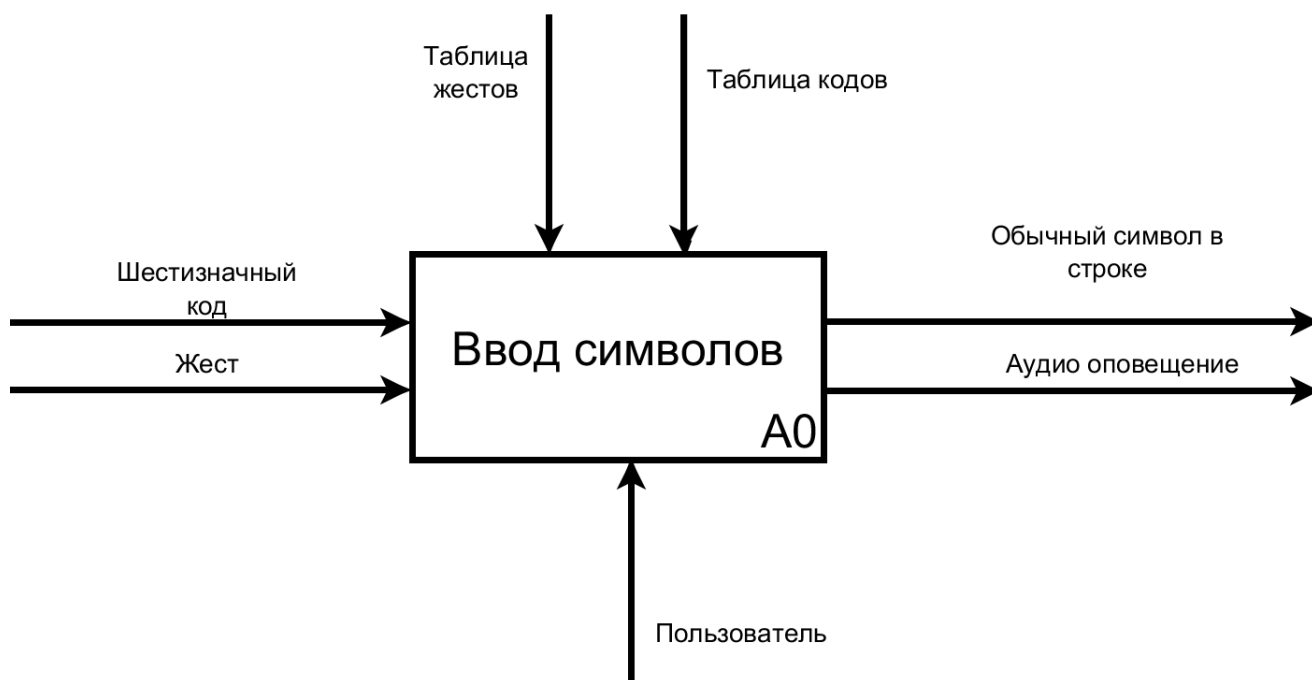


Рисунок 3.2 – Контекстная диаграмма

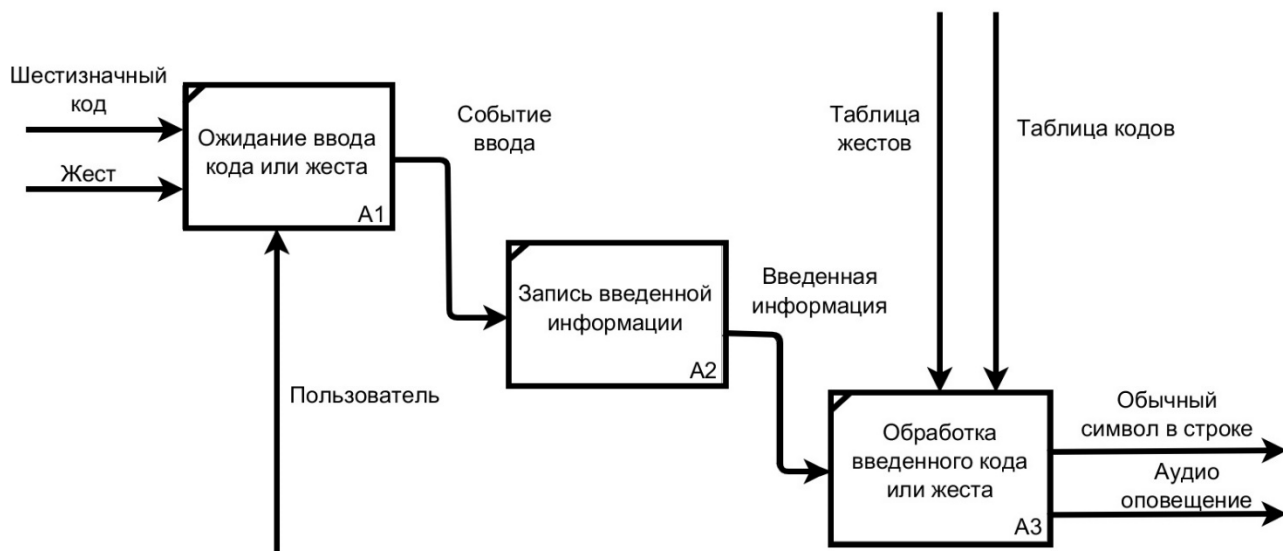


Рисунок 3.3 – Декомпозиция контекстной диаграммы

3.3 Проектирование структуры интерфейса

Интерфейс экранной клавиатуры состоит из одного основного окна. В этом окне пользователю предоставляется шесть кнопок для нажатия, расположенных возле левого и правого края экрана. Центральная часть отводится для ввода жестов (свайпов). Такая конфигурация окна исключит неточности ввода символов. Также стоит предусмотреть возможность зеркального ввода, для печати текста, когда устройство расположено экраном перед пользователем и когда оно расположено экраном от пользователя. Схема интерфейса представлена на рисунке 3.4.



Рисунок 3.4 – Схема интерфейса

Для ввода обычных символов будет применяться стандартная кодировка, представленная на рисунке 3.1. Кодировка специальных символов описана в таблице 3.1.

Таблица 3.1 – Кодировка специальных символов

Символ	Кодировка
Цифровой регистр	3 – 4 – 5 – 6
Регистр заглавных букв	1 – 2 – 3 – 4 – 5 – 6
Переключение раскладки	Свайп одним пальцем вверх
Отзеркаливание кнопок	Свайп одним пальцем вниз
Пробел	Свайп одним пальцем вправо
Удаление предыдущего символа	Свайп одним пальцем влево

После каждого действия устройство воспроизводит определенную аудиодорожку, а при включенном вводе прописных символов после ввода любой буквы пользователю сообщается, что введена прописная буква.

4 РЕАЛИЗАЦИЯ

Реализованное приложение воспринимается устройством как обычная клавиатура, которую можно назначить в настройках. Для этого пользователю нужно перейти во вкладку «расширенные настройки», «язык и ввод», затем в пункте «управление клавиатурами», необходимо включить новую клавиатуру «EDMTKeyboard», далее требуется в пункте «Текущая клавиатура» выбрать «EDMTKeyboard». На рисунке 4.1 представлена настройка включения клавиатуры.

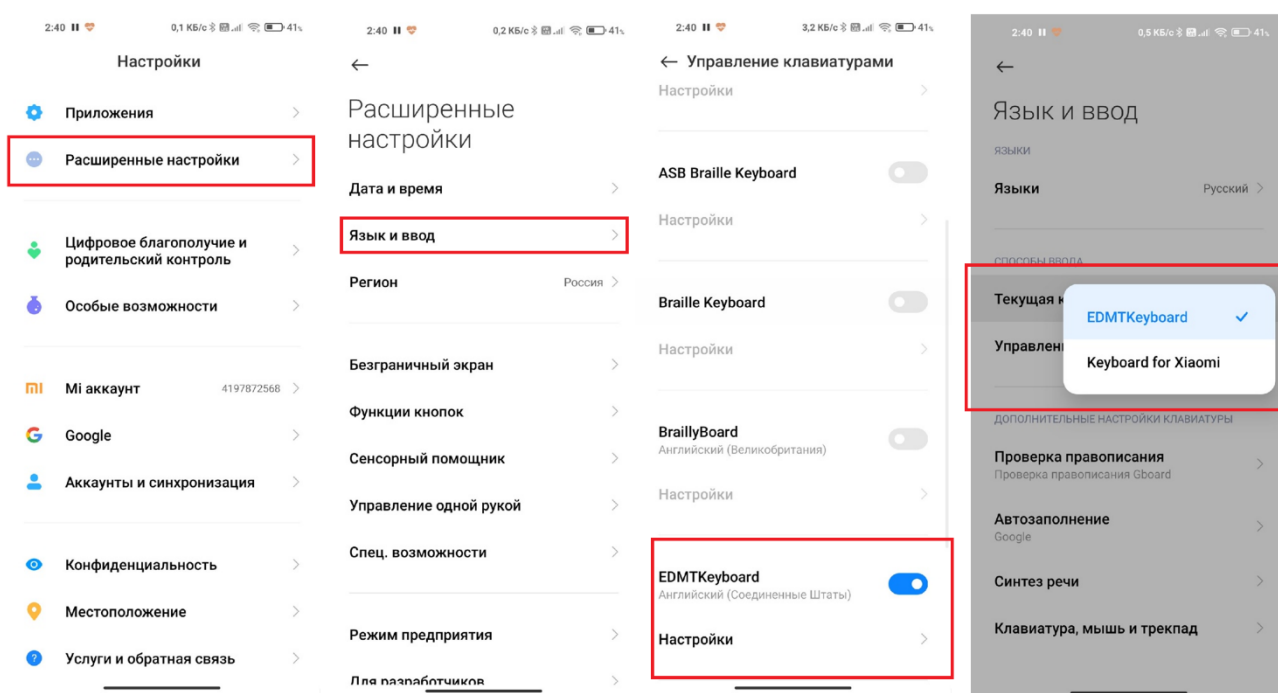


Рисунок 4.1 – Включение клавиатуры на устройстве

Разработанная клавиатура будет работать также, как и стандартная, то есть при возникновении ситуации ввода, она будет автоматически открываться, отодвигая интерфейс приложения, в котором производится ввод текста.

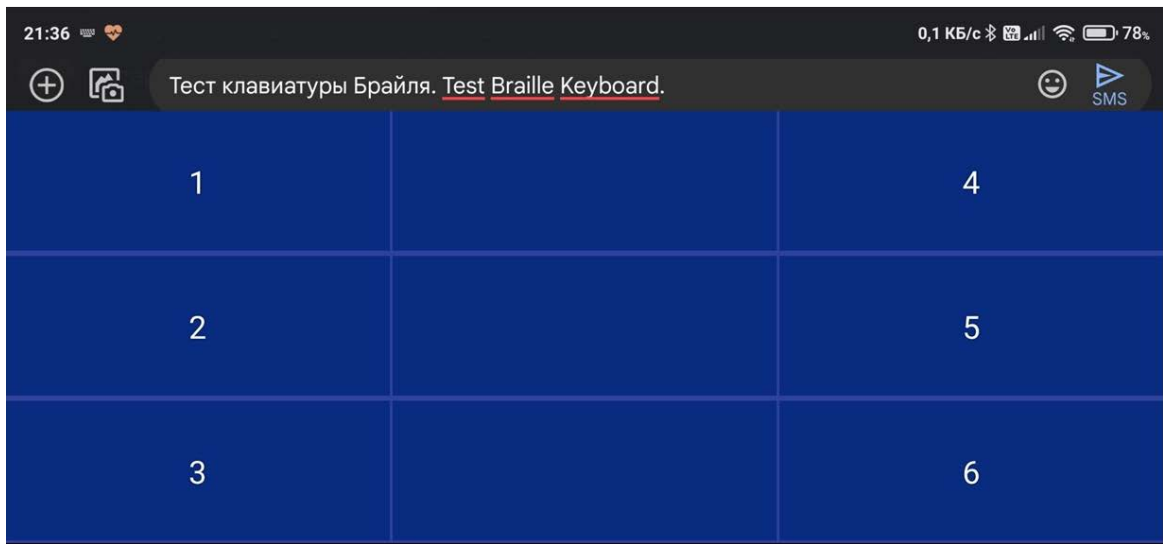


Рисунок 4.2 – Интерфейс разработанной клавиатуры
Исходный код программы представлен в приложении А.

5 ТЕСТИРОВАНИЕ

Поскольку приложение разрабатывалось в качестве выпускной квалификационной работы, было проведено только альфа-тестирование разработчиком. Для предотвращения ошибок в коде были проведены модульные тесты.

5.1 Тестирование появления клавиатуры

Для тестирования появления клавиатуры создадим различные ситуации ввода текста в популярных приложениях. На рисунках 5.1-5.5 представлены тесты появления клавиатуры.



Рисунок 5.1 – Тест появления клавиатуры в мессенджере Vк



Рисунок 5.2 – Тест появления клавиатуры в мессенджере Telegram



Рисунок 5.3 – Тест появления клавиатуры в браузере Chrome

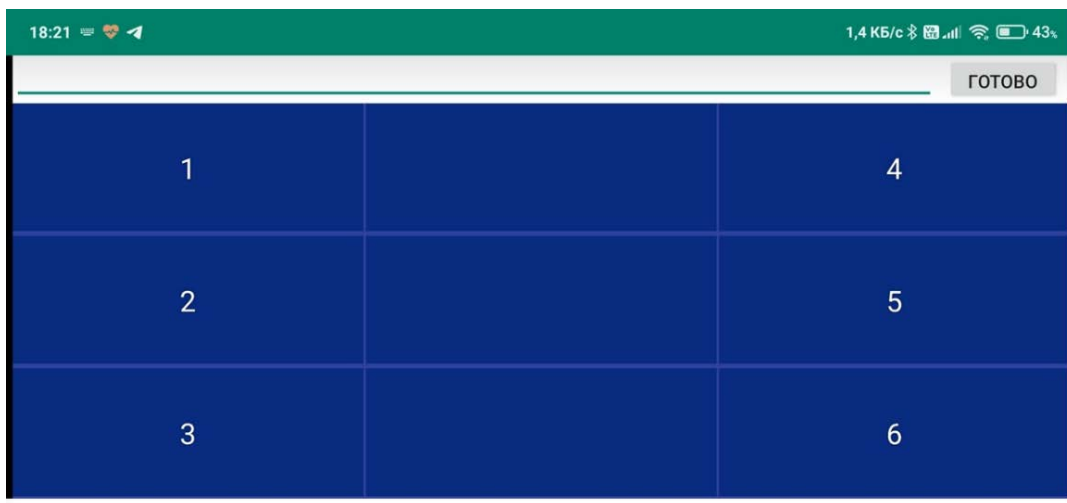


Рисунок 5.4 – Тест появления клавиатуры в мессенджере WhatsApp

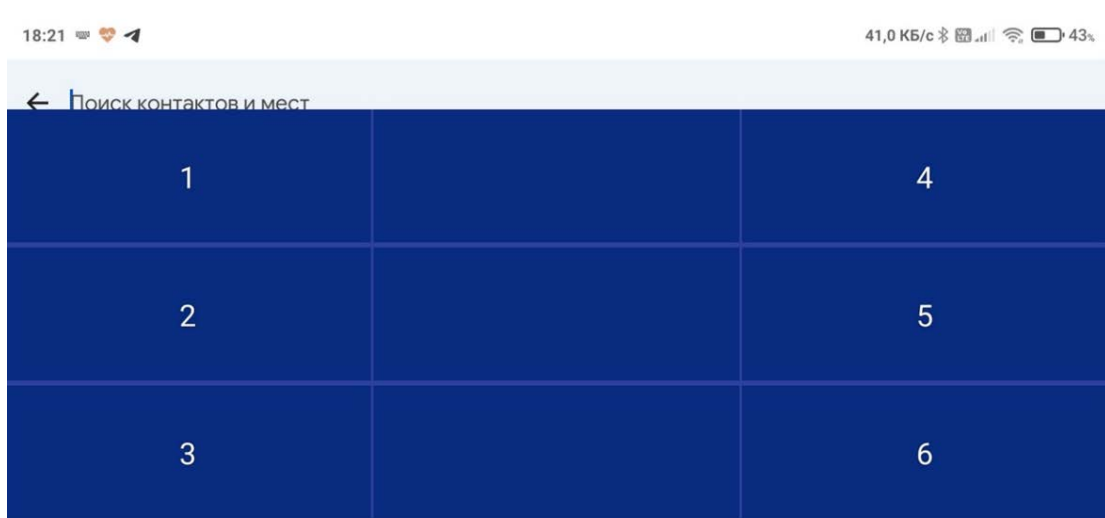


Рисунок 5.5 – Тест появления клавиатуры в приложении Телефон

Во всех приложениях появилась клавиатура, что означает корректное срабатывание в ситуации ввода.

5.2 Тестирование корректности ввода

Для тестирования ввода символов, создадим ситуацию ввода. В данном случае откроем приложение «Сообщения» и нажмем на строку ввода.

Введем все возможные символы. На рисунках 5.6 и 5.7 представлено тестирование ввода символов.

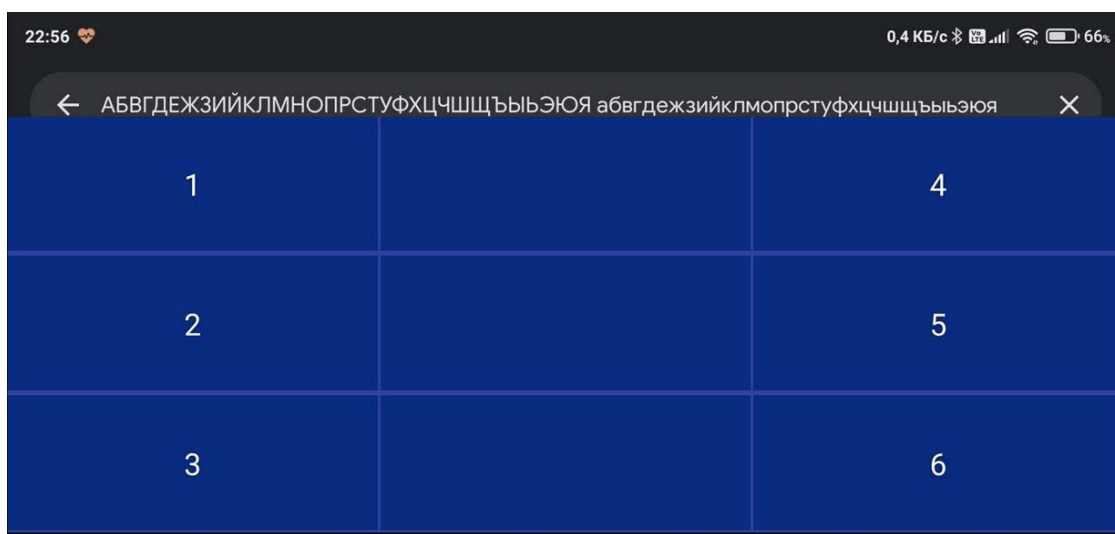


Рисунок 5.6 – Ввод всех символов русского алфавита

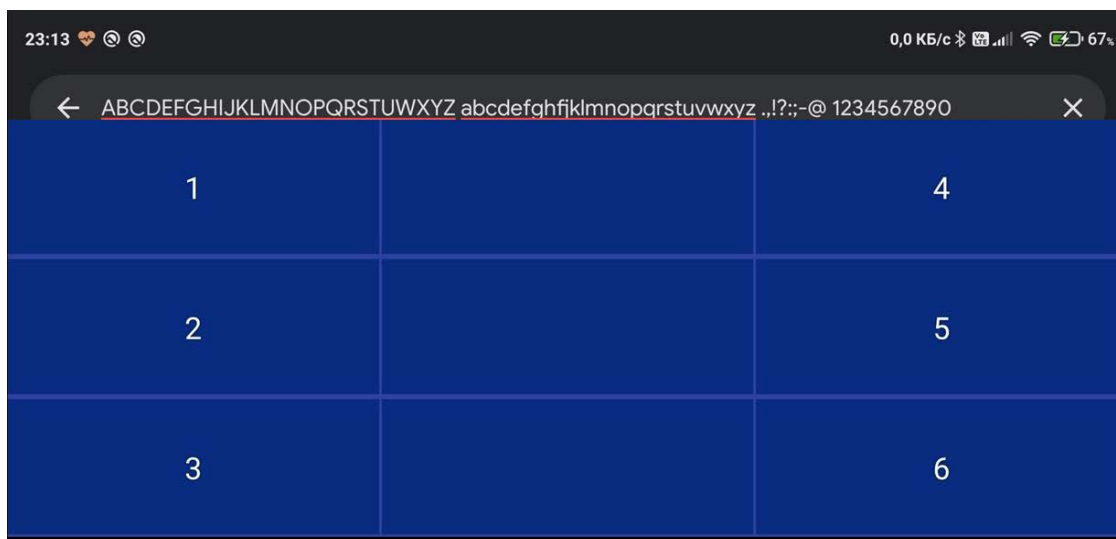


Рисунок 5.7 – Ввод всех латинских символов

Для печати всех символов нам потребовалось переключать регистры и сменять раскладку, данные действия мы совершили, не закрывая клавиатуру, что означает корректную работу ввода.

5.3 Тестирование корректности ввода жестов

Для тестирования жестов напечатаем текст, удалим символ, а после введем пробел, далее произведем смену раскладки и отзеркалим кнопки ввода. На рисунке 5.8 представлен тест жестов.

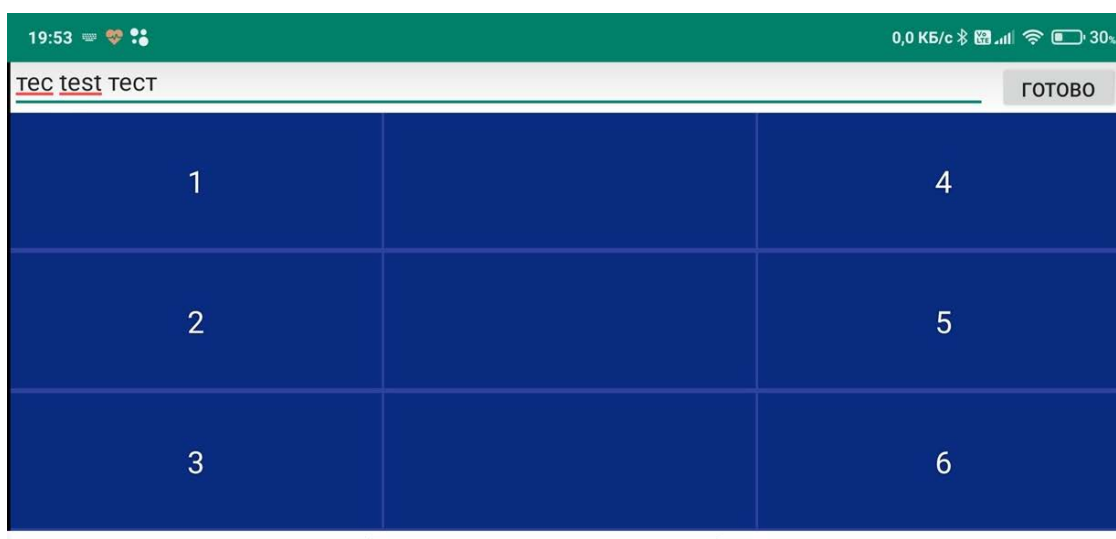


Рисунок 5.8 – Проверка жестов

Клавиатура обработала все жесты, что говорит нам о корректной работе системы жестов.

В ходе тестирования мы проверили все возможности разработанной клавиатуры. Ошибок в работе не выявлено.

ЗАКЛЮЧЕНИЕ

В рамках выпускной квалификационной работы была разработана и реализована экранная клавиатура Брайля, при этом были решены следующие задачи:

- изучены принципы печати текста по алфавиту Брайля, и определены специальные символы для обозначения цифр и регистров;
- найдены существующие аналогичные системы, в которых были выявлены достоинства и недостатки;
- составлено техническое задание и эскизный проект системы;
- выполнена программная реализация проекта;
- проведено альфа-тестирование разработанного приложения на корректность ввода всех символов и удобство интерфейса.

Перспективой разработки является внедрение клавиатуры в существующую систему для слабовидящих talkback.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Создание доступной среды для маломобильных групп населения: проблемы и пути решения. – Текст : электронный // ГлавГосЭкспертиза России : [сайт]. – URL: <https://gge.ru/press-center/news/sozдание-dostupnoy-sredy-dlya-malomobilnykh-grupp-naseleniya-problemy-i-puti-resheniya/> (дата обращения: 03.05.2022).
2. Ввод символов шрифта Брайля на экране iPhone, iPad и iPod touch. – Текст : электронный // Apple Support : [сайт]. – URL: <https://support.apple.com/ru-ru/HT210066> (дата обращения: 01.05.2022).
3. Operating System Market Share Russian Federation. – Текст : электронный // Statcounter GlobalStats : [сайт]. – URL: <https://gs.statcounter.com/os-market-share/all/russian-federation> (дата обращения: 03.05.2022).
4. TalkBack и клавиатура Брайля. – Текст : электронный // Справка - Google : [сайт]. – URL: https://support.google.com/accessibility/android/answer/9728765?hl=ru&ref_topic=10601975 (дата обращения: 02.05.2022).
5. Visual Studio и Xamarin. – Текст : электронный // microsoft.com : [сайт]. – URL: <https://msdn.microsoft.com/ru-ru/library/mt299001.aspx> (дата обращения: 05.05.2022).
6. Обзор платформы Eclipse. – Текст : электронный // hightech.in.ua : [сайт]. – URL: <http://hightech.in.ua/content/art-eclipse-platform> (дата обращения: 04.05.2022).
7. Android studio. – Текст : электронный // developer.android.com : [сайт]. – URL: <https://developer.android.com/studio> (дата обращения: 06.05.2022).
8. Java. – Текст : электронный // developer.android.com : [сайт]. – URL: <https://developer.android.com/studio/intro> (дата обращения: 06.05.2022).

ПРИЛОЖЕНИЕ А

Исходный код

Листинг А1 – содержимое файла AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="edmt.dev.androidcustomkeyboard">

    <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"
    >
        <service
            android:name=".EDMTKeyboard"
            android:label="EDMTKeyboard"
            android:permission="android.permission.BIND_INPUT_METHOD">
            <meta-data android:name="android.view.im"
android:resource="@xml/method" />
            <intent-filter>
                <action android:name="android.view.InputMethod" />
            </intent-filter>

        </service>

    </application>
</manifest>
```

Листинг А2 – содержимое файла qwerty.xml

```
<?xml version="1.0" encoding="utf-8"?>
<Keyboard xmlns:android="http://schemas.android.com/apk/res/android"
    android:keyWidth="100%p"
    android:horizontalGap="5px"
    android:verticalGap="5px"
    android:keyHeight="100dp"
>

    <Row>
        <Key android:keyLabel="1"
            android:keyEdgeFlags="left"
            android:codes="1"
        />
        <Key android:keyLabel=""
            android:codes="0" />
        <Key android:keyLabel="4"
            android:keyEdgeFlags="right"
            android:codes="4"
        />

    </Row>
    <Row>
        <Key android:keyLabel="2"
            android:keyEdgeFlags="left"

```

```

        android:codes="2" />

        <Key android:keyLabel=""
            android:codes="0" />
        <Key android:keyLabel="5"
            android:keyEdgeFlags="right"
            android:codes="5" />

    </Row>
    <Row>
        <Key android:keyLabel="3"
            android:keyEdgeFlags="left"
            android:codes="3" />
        <Key android:keyLabel=""
            android:codes="0" />
        <Key android:keyLabel="6"
            android:keyEdgeFlags="right"
            android:codes="6" />

    </Row>

</Keyboard>

```

Листинг А3 – содержимое файла keyboard.xml

```

<?xml version="1.0" encoding="utf-8"?>
<android.inputmethodservice.KeyboardView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/keyboard"
    android:layout_alignParentBottom="true"

    android:keyTextColor="@android:color/white"
    android:keyBackground="@drawable/key_background"
    android:keyPreviewLayout="@layout/key_preview"
    android:background="@color/colorPrimaryDark"

    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    >

</android.inputmethodservice.KeyboardView>

```

Листинг А4 – содержимое файла EDMTKeyboard.java

```

package edmt.dev.androidcustomkeyboard;

import android.annotation.SuppressLint;
import android.app.DownloadManager;
import android.content.pm.ActivityInfo;
import android.inputmethodservice.InputMethodService;
import android.inputmethodservice.Keyboard;
import android.inputmethodservice.KeyboardView;
import android.media.AudioManager;
import android.media.MediaPlayer;
import android.os.SystemClock;
import android.text.TextUtils;
import android.view.KeyEvent;
import android.view.View;

```

```

import android.view.inputmethod.InputConnection;
import android.view.MotionEvent;

import android.view.View.OnTouchListener;
import android.widget.TextView;

@SuppressLint("ClickableViewAccessibility")
public class EDMTKKeyboard extends InputMethodService implements
KeyboardView.OnKeyboardActionListener {

    private MediaPlayer sound[]= new MediaPlayer[90];
    private KeyboardView kv;
    private Keyboard keyboard;

    private boolean isCaps = false;
    TextView tv;
    char symbol ;
    int code1 =0;
    //Press Ctrl+O
    int key;

    int upPI = 0;

    boolean inTouch = false;

    boolean[] passed = new boolean[7];
    boolean tmp =true;
    boolean digital_number = false;
    public String register = "rus";
    float fromPositionX;
    float fromPositionY;
    boolean swipe = false;
    boolean mirror = false;
    boolean caps = false;

    @Override
    public View onCreateInputView() {
        sound[0] = MediaPlayer.create(this,R.raw.gen6);
        sound[1] = MediaPlayer.create(this,R.raw.gen7);
        sound[2] = MediaPlayer.create(this,R.raw.gen8);
        sound[3] = MediaPlayer.create(this,R.raw.gen9);
        sound[4] = MediaPlayer.create(this,R.raw.gen10);
        sound[5] = MediaPlayer.create(this,R.raw.gen11);
        sound[6] = MediaPlayer.create(this,R.raw.gen12);
        sound[7] = MediaPlayer.create(this,R.raw.gen13);
        sound[8] = MediaPlayer.create(this,R.raw.gen14);
        sound[9] = MediaPlayer.create(this,R.raw.gen15);
        sound[10] = MediaPlayer.create(this,R.raw.gen16);
        sound[11] = MediaPlayer.create(this,R.raw.gen17);
        sound[12] = MediaPlayer.create(this,R.raw.gen18);
        sound[13] = MediaPlayer.create(this,R.raw.gen19);
        sound[14] = MediaPlayer.create(this,R.raw.gen20);
        sound[15] = MediaPlayer.create(this,R.raw.gen21);
        sound[16] = MediaPlayer.create(this,R.raw.gen22);
        sound[17] = MediaPlayer.create(this,R.raw.gen23);
        sound[18] = MediaPlayer.create(this,R.raw.gen24);
    }

```

```
sound[19] = MediaPlayer.create(this,R.raw.gen25);
sound[20] = MediaPlayer.create(this,R.raw.gen26);
sound[21] = MediaPlayer.create(this,R.raw.gen27);
sound[22] = MediaPlayer.create(this,R.raw.gen28);
sound[23] = MediaPlayer.create(this,R.raw.gen29);
sound[24] = MediaPlayer.create(this,R.raw.gen30);
sound[25] = MediaPlayer.create(this,R.raw.gen31);
sound[26] = MediaPlayer.create(this,R.raw.gen32);
sound[27] = MediaPlayer.create(this,R.raw.gen33);
sound[28] = MediaPlayer.create(this,R.raw.gen34);
sound[29] = MediaPlayer.create(this,R.raw.gen35);
sound[30] = MediaPlayer.create(this,R.raw.gen36);
sound[31] = MediaPlayer.create(this,R.raw.gen37);
sound[32] = MediaPlayer.create(this,R.raw.gen38);
sound[33] = MediaPlayer.create(this,R.raw.enternumber);
sound[34] = MediaPlayer.create(this,R.raw.offnumber);
sound[35] = MediaPlayer.create(this,R.raw.num0);
sound[36] = MediaPlayer.create(this,R.raw.num1);
sound[37] = MediaPlayer.create(this,R.raw.num2);
sound[38] = MediaPlayer.create(this,R.raw.num3);
sound[39] = MediaPlayer.create(this,R.raw.num4);
sound[40] = MediaPlayer.create(this,R.raw.num5);
sound[41] = MediaPlayer.create(this,R.raw.num6);
sound[42] = MediaPlayer.create(this,R.raw.num7);
sound[43] = MediaPlayer.create(this,R.raw.num8);
sound[44] = MediaPlayer.create(this,R.raw.num9);
sound[45] = MediaPlayer.create(this,R.raw.delete);
sound[46] = MediaPlayer.create(this,R.raw.space);
sound[47] = MediaPlayer.create(this,R.raw.en);
sound[48] = MediaPlayer.create(this,R.raw.rus);
sound[49] = MediaPlayer.create(this,R.raw.en1);
sound[50] = MediaPlayer.create(this,R.raw.en2);
sound[51] = MediaPlayer.create(this,R.raw.en3);
sound[52] = MediaPlayer.create(this,R.raw.en4);
sound[53] = MediaPlayer.create(this,R.raw.en5);
sound[54] = MediaPlayer.create(this,R.raw.en6);
sound[55] = MediaPlayer.create(this,R.raw.en7);
sound[56] = MediaPlayer.create(this,R.raw.en8);
sound[57] = MediaPlayer.create(this,R.raw.en9);
sound[58] = MediaPlayer.create(this,R.raw.en10);
sound[59] = MediaPlayer.create(this,R.raw.en11);
sound[60] = MediaPlayer.create(this,R.raw.en12);
sound[61] = MediaPlayer.create(this,R.raw.en13);
sound[62] = MediaPlayer.create(this,R.raw.en14);
sound[63] = MediaPlayer.create(this,R.raw.en15);
sound[64] = MediaPlayer.create(this,R.raw.en16);
sound[65] = MediaPlayer.create(this,R.raw.en17);
sound[66] = MediaPlayer.create(this,R.raw.en18);
sound[67] = MediaPlayer.create(this,R.raw.en19);
sound[68] = MediaPlayer.create(this,R.raw.en20);
sound[69] = MediaPlayer.create(this,R.raw.en21);
sound[70] = MediaPlayer.create(this,R.raw.en22);
sound[71] = MediaPlayer.create(this,R.raw.en23);
sound[72] = MediaPlayer.create(this,R.raw.en24);
sound[73] = MediaPlayer.create(this,R.raw.en25);
sound[74] = MediaPlayer.create(this,R.raw.en26);
sound[75] = MediaPlayer.create(this,R.raw.mirroron);
sound[76] = MediaPlayer.create(this,R.raw.mirroroff);
sound[77] = MediaPlayer.create(this,R.raw.dot);
sound[78] = MediaPlayer.create(this,R.raw.comma);
sound[79] = MediaPlayer.create(this,R.raw.exclamationpoint);
```



```

sound[80] = MediaPlayer.create(this,R.raw.questionmark);
sound[81] = MediaPlayer.create(this,R.raw.colon);
sound[82] = MediaPlayer.create(this,R.raw.semicolon);
sound[83] = MediaPlayer.create(this,R.raw.dash);
sound[84] = MediaPlayer.create(this,R.raw.at);
sound[85] = MediaPlayer.create(this,R.raw.capson);
sound[86] = MediaPlayer.create(this,R.raw.capsoff);
sound[87] = MediaPlayer.create(this,R.raw.uppercase);

kv = (KeyboardView)getLayoutInflater().inflate(R.layout.keyboard,null);
keyboard = new Keyboard(this,R.xml.qwerty);
kv.setKeyboard(keyboard);
kv.setOnKeyboardActionListener(this);

kv.setOnTouchListener(new OnTouchListener() {
    @SuppressWarnings("ClickableViewAccessibility")
    @Override
    public boolean onTouch(View view, MotionEvent event) {

        // событие
        int actionMask = event.getActionMasked();
        // индекс касания
        int pointerIndex = event.getActionIndex();
        // число касаний
        int pointerCount = event.getPointerCount();

        switch (actionMask) {
            case MotionEvent.ACTION_DOWN: // первое касание

                if((event.getX()>760)&&(event.getX()<1500)) {
                    fromPositionX = event.getX();
                    fromPositionY = event.getY();
                    swipe = true;
                }else swipe=false;

                inTouch = false;
                code1 =0;

            case MotionEvent.ACTION_POINTER_DOWN: // последующие касания

                inTouch = false;
                float X = event.getX(pointerIndex) ;
                float Y = event.getY(pointerIndex) ;
                if(!mirror) {
                    if ((X > 0) && (X < 754) && (Y > 0) && (Y < 280))
passed[1] = true;

                    if ((X > 0) && (X < 754) && (Y > 281) && (Y < 560))
passed[2] = true;

                    if ((X > 0) && (X < 754) && (Y > 561) && (Y < 841))
passed[3] = true;

```

```

280)) passed[4] = true;
                    if ((X > 1507) && (X < 2400) && (Y > 0) && (Y <
560))
                    if ((X > 1507) && (X < 2400) && (Y > 281) && (Y <
                        passed[5] = true;
841))
                    if ((X > 1507) && (X < 2400) && (Y > 561) && (Y <
                        passed[6] = true;
                    }
                    if(mirror) {
passed[4] = true;
                        if ((X > 0) && (X < 754) && (Y > 0) && (Y < 280))
passed[5] = true;
                        if ((X > 0) && (X < 754) && (Y > 281) && (Y < 560))
passed[6] = true;
                        if ((X > 0) && (X < 754) && (Y > 561) && (Y < 841))
280)) passed[1] = true;
                        if ((X > 1507) && (X < 2400) && (Y > 0) && (Y <
560)) passed[2] = true;
                        if ((X > 1507) && (X < 2400) && (Y > 281) && (Y <
841)) passed[3] = true;
                        if ((X > 1507) && (X < 2400) && (Y > 561) && (Y <
                    }
                    break;

                    case MotionEvent.ACTION_UP: // прерывание последнего касания
                        float toPositionX = event.getX();
                        float toPositionY = event.getY();

                        InputConnection ic = getCurrentInputConnection();
                        if ((swipe)&&(fromPositionX >
toPositionX)&&(Math.abs(toPositionY-fromPositionY)<200)) {
                            CharSequence selectedText =
ic.getSelectedText(0);
                                if (TextUtils.isEmpty(selectedText)) {
                                    // no selection, so delete previous
character
                                    ic.deleteSurroundingText(1, 0);
                                    soundPlay(sound[45]);

                                } else {
                                    // delete the selection
                                    ic.commitText("", 1);

                                }
                                }
                                break;
                            }
                            if ((swipe)&&(fromPositionX <
toPositionX)&&(Math.abs(toPositionY-fromPositionY)<200)) {
                                ic.commitText(String.valueOf(" "), 1);
                                soundPlay(sound[46]);
                            }
                        }
                    if ((swipe)&&(fromPositionY >

```

```

toPositionY)&&(Math.abs(toPositionX-fromPositionX)<200)) {

    if(register=="rus") {
        soundPlay(sound[47]);
        register="en";
        break;
    }
    if(register=="en") {
        soundPlay(sound[48]);
        register = "rus";
        break;
    }

    break;
}
if ((swipe)&&(fromPositionY <
toPositionY)&&(Math.abs(toPositionX-fromPositionX)<200)) {

    if(mirror) {
        soundPlay(sound[76]);
        mirror = false;
    } else {
        soundPlay(sound[75]);
        mirror = true;
    }
    break;

}

fromPositionX =800;
fromPositionY =0;

swipe = false;
inTouch = false;

case MotionEvent.ACTION_POINTER_UP: // прерывания касаний
    upPI = pointerIndex;
    code1=0;

    inTouch = true;

    break;

case MotionEvent.ACTION_MOVE: // движение
    inTouch = false;

    break;
}

if (inTouch) {
if(passed[1]) code1+=1;
if(passed[2]) code1+=10;
if(passed[3]) code1+=100;
if(passed[4]) code1+=1000;

```

```

        if(passed[5]) code1+=10000;
        if(passed[6]) code1+=100000;

        character_selection();

        if((tmp)&&(symbol!=0)) {

if((caps&&(symbol>='a')&&(symbol<='z'))||(caps&&(symbol>='А')&&(symbol<='Я'))){

            sound[87].start();

            symbol-=32;
        }

            InputConnection ic = getCurrentInputConnection();

            ic.commitText(String.valueOf(symbol), 1);
        }
        tmp = true;

        symbol = 0;
        code1 =0;
        passed[1]= false;
        passed[2]= false;
        passed[3]= false;
        passed[4]= false;
        passed[5]= false;
        passed[6]= false;
    }

        return false;
    }
});
return kv;
}

@Override
public void onPress(int i) {
    key = i;
}

@Override
public void onRelease(int i) {

}

@Override
public void onKey(int i, int[] ints) {

}

private void playClick(int i) {

```

```

AudioManager am = (AudioManager) getSystemService(AUDIO_SERVICE);
switch(i)
{
    case 32:
        am.playSoundEffect(AudioManager.FX_KEYPRESS_SPACEBAR);
        break;
    case Keyboard.KEYCODE_DONE:
    case 10:
        am.playSoundEffect(AudioManager.FX_KEYPRESS_RETURN);
        break;
    case Keyboard.KEYCODE_DELETE:
        am.playSoundEffect(AudioManager.FX_KEYPRESS_DELETE);
        break;
    default: am.playSoundEffect(AudioManager.FX_KEYPRESS_STANDARD);
}
}

@Override
public void onText(CharSequence charSequence) {

}

@Override
public void swipeLeft() {

}

@Override
public void swipeRight() {

}

@Override
public void swipeDown() {

}

@Override
public void swipeUp() {

}

public void soundPlay (MediaPlayer sound) {
    sound.start();
}

public void character_selection () {
    if(code1==111111){
        swipe=false;
        if(!caps) {
            soundPlay(sound[85]);
            caps=true;
        } else {
            soundPlay(sound[86]);
            caps = false;
        }
    }
}

```

```
}
if (code1==111100){
    swipe=false;
    if((register=="rus")||(register=="en")) {
        soundPlay(sound[33]);
        register="num";
    } else if(register=="num"){
        soundPlay(sound[34]);
        register = "rus";
    }
}
if(register=="rus") {

    switch (code1) {
        case 1:
            symbol = 'a'; //a

            soundPlay(sound[0]);
            break;

        case 11:
            symbol = 'б'; //б
            soundPlay(sound[1]);
            break;

        case 111010:
            symbol = 'в'; //в
            soundPlay(sound[2]);
            break;
        case 11011:
            symbol = 'г'; //г
            soundPlay(sound[3]);
            break;

        case 11001:
            symbol = 'д'; //д
            soundPlay(sound[4]);
            break;

        case 10001:
            symbol = 'e'; //e
            soundPlay(sound[5]);
            break;

        case 100001:
            symbol = 'ë'; //ë
            soundPlay(sound[6]);
            break;

        case 11010:
            symbol = 'ж'; //ж
            soundPlay(sound[7]);
            break;

        case 110101:
            symbol = 'з'; //з
            soundPlay(sound[8]);
            break;

        case 1010:
```

```
        symbol = 'и'; //и
        soundPlay(sound[9]);
        break;

    case 101111:
        symbol = 'й'; //й
        soundPlay(sound[10]);
        break;

    case 101:
        symbol = 'к'; //к
        soundPlay(sound[11]);
        break;

    case 111:
        symbol = 'л'; //л
        soundPlay(sound[12]);
        break;

    case 1101:
        symbol = 'м'; //м
        soundPlay(sound[13]);
        break;

    case 11101:
        symbol = 'н'; //н
        soundPlay(sound[14]);
        break;

    case 10101:
        symbol = 'о'; //о
        soundPlay(sound[15]);
        break;

    case 1111:
        symbol = 'п'; //п
        soundPlay(sound[16]);
        break;

    case 10111:
        symbol = 'р'; //р
        soundPlay(sound[17]);
        break;

    case 1110:
        symbol = 'с'; //с
        soundPlay(sound[18]);
        break;

    case 11110:
        symbol = 'т'; //т
        soundPlay(sound[19]);
        break;

    case 100101:
        symbol = 'у'; //у
        soundPlay(sound[20]);
        break;
    case 1011:
        symbol = 'ф'; //ф
```

```
        soundPlay(sound[21]);
        break;

    case 10011:
        symbol = 'x'; //x

        soundPlay(sound[22]);
        break;

    case 1001:
        symbol = 'ц'; //ц
        soundPlay(sound[23]);
        break;

    case 11111:
        symbol = 'ч'; //ч
        soundPlay(sound[24]);
        break;

    case 110001:
        symbol = 'ш'; //ш
        soundPlay(sound[25]);
        break;

    case 101101:
        symbol = 'щ'; //щ
        soundPlay(sound[26]);
        break;
    case 110111:
        symbol = 'ъ'; //ъ
        soundPlay(sound[27]);
        break;
    case 101110:
        symbol = 'ы'; //ы
        soundPlay(sound[28]);
        break;
    case 111110:
        symbol = 'ь'; //ь
        soundPlay(sound[29]);
        break;
    case 101010:
        symbol = 'э'; //э
        soundPlay(sound[30]);
        break;
    case 110011:
        symbol = 'ю'; //ю
        soundPlay(sound[31]);
        break;
    case 101011:
        symbol = 'я'; //я
        soundPlay(sound[32]);
        break;

    default:
        tmp = false;
        break;
}
```



```
}
if(register=="en") {
    switch (code1) {
        case 1:
            symbol = 'a'; //a
            soundPlay(sound[49]);
            break;

        case 11:
            symbol = 'b'; //b
            soundPlay(sound[50]);
            break;
        case 1001:
            symbol = 'c'; //c
            soundPlay(sound[51]);
            break;
        case 11001:
            symbol = 'd'; //d
            soundPlay(sound[52]);
            break;
        case 10001:
            symbol = 'e'; //e
            soundPlay(sound[53]);
            break;
        case 1011:
            symbol = 'f'; //f
            soundPlay(sound[54]);
            break;
        case 11011:
            symbol = 'g'; //g
            soundPlay(sound[55]);
            break;
        case 10011:
            symbol = 'h'; //h
            soundPlay(sound[56]);
            break;
        case 1010:
            symbol = 'i'; //i
            soundPlay(sound[57]);
            break;
        case 11010:
            symbol = 'j'; //j
            soundPlay(sound[58]);
            break;
        case 101:
            symbol = 'k'; //k
            soundPlay(sound[59]);
            break;
        case 111:
            symbol = 'l'; //l
            soundPlay(sound[60]);
            break;
        case 1101:
            symbol = 'm'; //m
            soundPlay(sound[61]);
            break;
        case 11101:
            symbol = 'n'; //n
            soundPlay(sound[62]);
            break;
        case 10101:
```

```

        symbol = 'o'; //o
        soundPlay(sound[66]);
        break;
    case 1111:
        symbol = 'p'; //p
        soundPlay(sound[64]);

        break;
    case 11111:
        symbol = 'q'; //q
        soundPlay(sound[65]);
        break;
    case 10111:
        symbol = 'r'; //r
        soundPlay(sound[66]);
        break;
    case 1110:
        symbol = 's'; //s
        soundPlay(sound[67]);
        break;
    case 11110:
        symbol = 't'; //t
        soundPlay(sound[68]);
        break;
    case 100101:
        symbol = 'u'; //u
        soundPlay(sound[69]);
        break;
    case 100111:
        symbol = 'v'; //v
        soundPlay(sound[70]);
        break;
    case 111010:
        symbol = 'w'; //w
        soundPlay(sound[71]);
        break;
    case 101101:
        symbol = 'x'; //x
        soundPlay(sound[72]);
        break;
    case 111101:
        symbol = 'y'; //y
        soundPlay(sound[73]);
        break;
    case 110101:
        symbol = 'z'; //z
        soundPlay(sound[74]);
        break;

    default:
        tmp = false;
        break;
    }
}
if(register=="num"){
    switch (code1){
    case 1:
        symbol = '1'; //1
        soundPlay(sound[36]);

```

```
        break;

    case 11:
        symbol = '2'; //2
        soundPlay(sound[37]);
        break;

    case 11011:
        symbol = '7'; //7
        soundPlay(sound[42]);
        break;

    case 11001:
        symbol = '4'; //4
        soundPlay(sound[39]);
        break;

    case 10001:
        symbol = '5'; //5
        soundPlay(sound[40]);
        break;

    case 11010:
        symbol = '0'; //0
        soundPlay(sound[35]);
        break;

    case 1010:
        symbol = '9'; //9
        soundPlay(sound[44]);
        break;

    case 1011:
        symbol = '6'; //6
        soundPlay(sound[41]);
        break;

    case 10011:
        symbol = '8'; //8
        soundPlay(sound[43]);
        break;

    case 1001:
        symbol = '3'; //3
        soundPlay(sound[38]);
        break;

    default:
        tmp = false;
        break;
    }
}
tmp = true;
switch (code1){
    case 110010:
        symbol = '.'; //.
        soundPlay(sound[77]);
        break;
    case 10:
```

```
        symbol = ','; //,  
        soundPlay(sound[78]);  
        break;  
    case 10110:  
        symbol = '!'; //!  
        soundPlay(sound[79]);  
        break;  
  
    case 100110:  
        symbol = '?'; //?  
        soundPlay(sound[80]);  
        break;  
    case 10010:  
        symbol = ':'; //:  
        soundPlay(sound[81]);  
        break;  
    case 110:  
        symbol = ';'; //;  
        soundPlay(sound[82]);  
        break;  
    case 100100:  
        symbol = '-'; //-  
        soundPlay(sound[83]);  
        break;  
    case 11100:  
        symbol = '@'; //@  
        soundPlay(sound[84]);  
        break;  
    }  
}  
  
}
```