

Перв. примен.															
Справ. №															
Подпись и дата															
Инв. № дубл.															
Взам. инв. №															
Подпись и дата															
Инв. № подл.															
<div> <div>1 Техническое задание</div> <div> <div>1.1 Назначение разработки и область применения</div> <div> <p>Разрабатываемая система предназначена для рекомендации музыки пользователям на основе эмоционального состава аудиозаписей.</p> <p>Это система является прототипом системы рекомендации для социальных сетей.</p> </div> </div> <div> <div>1.2 Технические требования</div> <div> <p>Рассмотрим требования, предъявляемые разрабатываемой системой к ЭВМ:</p> <ul style="list-style-type: none"> – операционная система Microsoft Windows не ниже версии XP; – требования к аппаратному обеспечению определяются операционной системой; – клавиатура и мышь; <p>Разрабатываемая система рекомендаций, должна обладать следующим функционалом :</p> <ul style="list-style-type: none"> - должна позволять пользователю использовать его файлы для анализа; - формировать на основании звукового файла его эмоциональный состав; - производить поиск похожих по эмоциональному составу аудиозаписей в базе данных; - отображать в графическом интерфейсе список рекомендаций. </div> </div> </div>															
<table border="1"> <tr> <td>Изм.</td> <td>Лист</td> <td>№ докум.</td> <td>Подпись</td> <td>Дата</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </table>					Изм.	Лист	№ докум.	Подпись	Дата						<div> <div>ВКР-НГТУ-09.03.01-12В-2-009-2016ПЗ</div> <div>Лист</div> <div>5</div> </div>
Изм.	Лист	№ докум.	Подпись	Дата											

Перв. примен.																	
Справ. №																	
<p>2 Анализ технического задания.</p> <p>2.1 Выбор операционной системы</p> <p>Одна из важнейших задач, которую предстоит решить перед началом разработки – это выбор операционной системы. Использование каждой из них имеет свои особенности</p> <p>Рассмотрим наиболее распространенные операционные системы:</p> <ul style="list-style-type: none"> – Windows – операционная система корпорации Microsoft. Наиболее распространённая пользовательская система. – Mac OS – это операционная система, разработанная компанией Apple. В настоящее время является второй наиболее по популярности операционной системой. Mac OS является частью семейства операционных систем Unix. – Linux – общее название Unix-подобных операционных систем, основанных на одноимённом ядре. Является открытой операционной системой. <p>После анализа возможностей операционных систем было принято решение использовать операционную систему Windows, так как она содержит все возможности для разработки данного ПО.</p>																	
Подпись и дата																	
Инд. № дубл.																	
Взам. инв. №																	
Подпись и дата																	
Инд. № подл.																	
<table border="1"> <tr> <td>Изм.</td> <td>Лист</td> <td>№ докум.</td> <td>Подпись</td> <td>Дата</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </table>					Изм.	Лист	№ докум.	Подпись	Дата						<table border="1"> <tr> <td>Лист</td> </tr> <tr> <td>6</td> </tr> </table>	Лист	6
Изм.	Лист	№ докум.	Подпись	Дата													
Лист																	
6																	

Перв. примен.					
Справ. №					
Подпись и дата					
Инв. № дубл.					
Взам. инв. №					
Подпись и дата					
Инв. № подл.					

2.2 Выбор языка программирования

Выбор языка программирования важны этап разработки программного обеспечения. Учитывая то, что разработка ведётся под операционную систему Windows, можно выделить такие распространенные языки программирования как: Python, R, и Java.

Рассмотрим их более подробно:

- Python – высокоуровневый язык, поддерживающий множество парадигм программирования: ООП, функциональное, структурное программирование и другие. Python – скриптовый язык, т.е. исходный код программы не компилируется, а выполняется с помощью интерпретатора.
- R – интерпретируемый язык, созданный для обработки информации. Поддерживает множество статических и численных методов. Почти все стандартные функции R написаны на самом R – это делает его легким в изучении, так как каждый пользователь может посмотреть и понять, как реализован тот или иной интересующий его метод. Этот язык так же является легко расширяемым за счет скачиваемых пакетов, что позволяет «заточить» его под задачи, для которых он изначально не был предназначен. Так же R позволяет внутри себя вызвать функции языков C, Java и C#, что позволяет увеличить скорость работы и функционал языка.
- Java – объектно-ориентированный язык. Данный язык компилируется в промежуточный байт код, который затем выполняется на виртуальной машине Java – это позволяет уйти от особенностей архитектуры на котором будет выполняться код и достичь платформенезависимости. Данный язык программирования широко распространен и имеет множество бесплатных библиотек для научных вычислений.

Решающим факторам выбора языка стало наличие библиотеки для работы со звуковыми файлами – seewave и библиотеки для классификации с помощью метода опорных векторов - e1071 в языке R.

Seewave предоставляет:

- функции для анализа сигнала;
- функции для открытия звуковых файлов;
- функции для генерации звукового сигнала.

Изм.	Лист	№ докум.	Подпись	Дата	ВКР-НГТУ-09.03.01-12В-2-009-2016ПЗ	Лист
						7

Перв. примен.						
Справ. №						
Подпись и дата						
Инв. № дубл.						
Взам. инв. №						
Подпись и дата						
Инв. № подл.						
Изм.	Лист	№ докум.	Подпись	Дата	ВКР-НГТУ-09.03.01-12В-2-009-2016ПЗ	Лист
						8

е1071 – библиотека с различными методами, относящимися к области статистики и теории вероятности. Содержит в себе функции для анализа латентных классов, быстрого преобразования Фурье, нечеткой кластеризации и многие другие, но главное эта библиотека содержит возможности для создания системы классификации на основе метода опорных векторов

Перв. примен.					
Справ. №					
Подпись и дата					
Инв. № дубл.					
Взам. инв. №					
Подпись и дата					
Инв. № подл.					
<div> <div> <div>Изм.</div> <div>Лист</div> <div>№ докум.</div> <div>Подпись</div> <div>Дата</div> </div> <div> <div>ВКР-НГТУ-09.03.01-12В-2-009-2016ПЗ</div> <div>Лист</div> <div>9</div> </div> </div>					

2.3 Выбор среды разработки

Для каждого языка программирования существует множество сред разработки. Так и для языка R: Weka, RStudio, R Commander. Была выбрана интегрированная среда разработки RStudio. Это открытая кроссплатформенная среда для разработки на языке R. Ее особенностями являются:

- Интерактивный отладчик, чтобы диагностировать и быстро исправить ошибки;
- Интегрированная помощь и документация;
- Простое управление несколькими рабочими каталогами;
- Поддержка выполнения кода прямо из редактора;
- Подсветка синтаксиса;
- Простое управление пакетами.

Перв. примен.	<h2>2.4 Обзор существующих систем</h2> <p>Рассмотрим принцип работы наиболее популярных рекомендательных систем:</p> <p>1) Яндекс музыка</p> <p>Анализирует профиль пользователя, какие песни и исполнители ему понравились. Каждый раз, когда пользователь совершает то или иное действие система заново формирует рекомендации, это позволяет быстро менять рекомендации в случае изменения настроения пользователя.</p> <p>Делая прогноз, алгоритм также учитывает информацию о том, как связаны друг с другом объекты из каталога Яндекс.Музыки: треки, альбомы, исполнители, жанры. Кроме того, система сравнивает профили всех пользователей Яндекс.Музыки. Это делается для того, чтобы выявить людей со схожими музыкальными предпочтениями: то, что нравится одному, может понравиться и другому. [5]</p> <p>2) Вконтакте</p> <p>Подбор рекомендаций основан на предпочтении всех пользователей у которых есть такие же аудиозаписи – система анализирует их список и выбирает те песни, которые встречаются чаще всего у таких людей, но которых нет у текущего пользователя. Результат таких рекомендаций не всегда идеален, так как у людей слишком разные вкусы. [6]</p> <p>3) Apple music</p> <p>Эта система учитывает следующие действия пользователя: лайки, проигрывания, содержимое медиатеки и на основе их формирует рекомендации</p> <p>Чем дольше пользователь слушает музыку при помощи сервиса, тем точнее сервис должен понимать его интересы. [7]</p> <p>Все рассмотренные выше системы построены на методы коллаборативной фильтрации.</p> <p>В таких системах оценивается поведении пользователей в прошлом: их оценки той или иной музыки, покупки и частота прослушиваний тех или иных треков.</p> <p>Такая система не лишена недостатков основным, из которых является – холодный старт. Т.е. когда рекомендательная система только начинает свою работы и пользователи еще не успели совершить много действий, результат работы системы будет посредственным, т.к. ей не на чем основывать свои рекомендации.</p>				
	Справ. №				
Подпись и дата					
	Инв. № дубл.				
Взам. инв. №					
Подпись и дата					
Инв. № подл.					
Изм.	Лист	№ докум.	Подпись	Дата	<div>ВКР-НГТУ-09.03.01-12В-2-009-2016ПЗ</div> <div>Лист 10</div>

Перв. примен.					
Справ. №					
Подпись и дата					
Инв. № дубл.					
Взам. инв. №					
Подпись и дата					
Инв. № подл.					
<div> <div>Изм.</div> <div>Лист</div> <div>№ докум.</div> <div>Подпись</div> <div>Дата</div> </div>					<div>ВКР-НГТУ-09.03.01-12В-2-009-2016ПЗ</div> <div>Лист</div> <div>11</div>

Разрабатываемая в данной работе система построена на принципе фильтрации содержимого – т.е. признаки для рекомендации выстраиваются не на основе действий пользователей, а на непосредственно самом контенте.

Такая система позволяет избежать проблемы «холодного старта», т.к. ей нет необходимо ожидать действий пользователей для генерации рекомендации.

Перв. примен.	<h2>2.5 Выбор подхода к решению задачи распознавания и классификации</h2> <p>Рассмотрим задачу распознавания эмоций с точки зрения системного подхода. Задача состоит из 3 этапов:</p> <ol style="list-style-type: none"> 1) предварительная обработка сигнала – фильтрация сигнала от помех 2) вычисление признаков – создание системы признаков на основе обработанного сигнала 3) классификация – разделение сигналов на группы на основе признаков <p>Однако для большинства методов обработки данных характерны следующие проблемы [8]:</p> <ol style="list-style-type: none"> 1) проблема формирования исходного описания: данная проблема связана с тем, что существующие модели и методы распознавания адаптированы к конкретному классу прикладных задач и требуют априорного знания свойств анализируемых сигналов; 2) проблема формирования системы признаков, связанная с выбором конечного множества признаков, обеспечивающих однозначность решения задачи классификации на этапе распознавания и отвечающая требованиям необходимости и достаточности. Этап выбора системы признаков необходим для сокращения размерности входного описания. Учитывая, что задача сокращения размерности — оптимизационная задача, то для её решения необходимо использование критерия информативности. Отсутствие модели априорной неопределённости и модели её раскрытия породило большое количество методов в выборе критерия информативности, что, в свою очередь, порождает большое число возможных вариантов признаков; 3) проблема принятия решений в условиях априорной неопределённости. Этап принятия решения заключается в сравнении с имеющимся эталоном признакового описания анализируемого сигнала. Предполагается, что эталону соответствует компактное множество точек в системе признаков. Однако помехи, структурные изменения одного и того же представителя класса приводят к перекрытию классов. Поэтому проблема принятия решения замыкается на проблемы формирования системы признаков, позволяющей сформировать эталон, имеющий компактное представление. <p>Чтобы избежать этих проблем, для этапов предварительной обработки сигнала и формирования системы признаков было решено использовать теорию активного восприятия, в которой данные недостатки отсутствуют. Для этапа классификации было решено использовать метод опорных векторов.</p>																	
	Справ. №																	
Подпись и дата		Инв. № дубл.	Взам. инв. №	Подпись и дата	Инв. № подл.													
<table border="1"> <tr> <td>Изм.</td> <td>Лист</td> <td>№ докум.</td> <td>Подпись</td> <td>Дата</td> <td rowspan="2"> ВКР-НГТУ-09.03.01-12В-2-009-2016ПЗ </td> <td>Лист</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td>12</td> </tr> </table>						Изм.	Лист	№ докум.	Подпись	Дата	ВКР-НГТУ-09.03.01-12В-2-009-2016ПЗ	Лист						12
Изм.	Лист	№ докум.	Подпись	Дата	ВКР-НГТУ-09.03.01-12В-2-009-2016ПЗ	Лист												
						12												

Перв. примен.					
	<p>Рисунок 1 – Этапы обработки информации</p>				
Справ. №					
Подпись и дата					
Инв. № дубл.					
Взам. инв. №					
Подпись и дата					
Инв. № подл.					

Рассмотрим более подробно эти этапы:

В теории активного восприятия предварительная обработка сигнала заключается в выполнении интегрирования. Сигнал разбивается на сегменты и для каждого сегмента выполняется U -преобразование. [1]

Таким образом используется следующий алгоритм для формирования признаков:

- 1) сигнал разбивается на множество сегментов
- 2) к каждому сегменту применяется U -преобразование и в результате формируется спектральное представление каждого сегмента.
- 3) на основе спектрального представления формируется описание с помощью закрытых групп.

На этом этап вычисления признаков заканчивается. Затем полученные признаки предаются в классификатор, который определяет, к какому классу они принадлежат.

В данной работе в качестве механизмов классификации был выбран метод опорных векторов– алгоритм для классификации основанный на обучении с учителем. Идея метода состоит в переводе исходных векторов в пространство более высокой размерности и поиск разделяющей гиперплоскости с максимальным зазором в этом пространстве. Две параллельных гиперплоскости строятся по обеим сторонам гиперплоскости, разделяющей наши классы. Разделяющей гиперплоскостью будет гиперплоскость, максимизирующая расстояние до двух параллельных гиперплоскостей[3].

В качестве классов в данной системы было решено использовать следующий список эмоций: веселье, воодушевление, покой, раздражение, разрушение.

					<p>ВКР-НГТУ-09.03.01-12В-2-009-2016ПЗ</p>	Лист
						13
Изм.	Лист	№ докум.	Подпись	Дата		

Перв. примен.					
Справ. №					
Подпись и дата					
Инв. № дубл.					
Взам. инв. №					
Подпись и дата					
Инв. № подл.					

3.Разработка структуры программной части

3.1 Разработка общей структуры системы

Программная часть моей системы будет состоять из 2х частей:

1) программа для создания классификатора

Так как классификатор SVM необходимо обучить перед использованием было принято решение сделать его обучение в виде отдельной программы и сохранять созданный им классификатор. Это позволяет не тратить время на то что бы заново создать классификатор при каждом запуске клиента.

Это часть использует заранее подготовленную подборку звуковых файлов, с уже определенными классами для обучения классификатора, который будет использоваться для рекомендаций в дальнейшем.

2)клиентская программа рекомендательной системы

Программа с помощью, которой пользователь может получить музыкальные рекомендации.

Позволяет открыть пользователю музыкальный файл и получить его эмоциональный состав. Выводит список рекомендованных аудиозаписей основываясь на поиске схожих по эмоциональному составу файлов в базе данных.

Программа для создания классификатора

```
graph TD; A[Модуль обработки аудио файлов] --> B[Модуль создания обучающей выборки]; B --> C[Модуль обучения SVM];
```

Клиентское приложение

```
graph TD; G[Графический интерфейс] --> D[Модуль подбора рекомендаций]; D --> C[Классификатор]; C --> B[База данных]; B --> D;
```

Рисунок 2 – Общая структура программы

					ВКР-НГТУ-09.03.01-12В-2-009-2016ПЗ	Лист 14
Изм.	Лист	№ докум.	Подпись	Дата		

Перв. примен.						
Справ. №						
Подпись и дата						
Инв. № дубл.						
Взам. инв. №						
Подпись и дата						
Инв. № подл.						
Изм.	Лист	№ докум.	Подпись	Дата	ВКР-НГТУ-09.03.01-12В-2-009-2016ПЗ	Лист
						15

Исходя из того что, язык R поддерживает сохранение переменных в виде файлов, то было принято решение связать две эти части с помощью файловой системы. Таким образом переменная содержащая модель классификатора в первой части программы сохранится в виде файла, а затем с помощью стандартных средств языка восстанавливается в клиенте.

Рассмотрим каждую часть более подробно.

Перв. примен.		<h3>3.2 Разработка структуры программы для создания классификатора</h3> <p>Программа для создания классификатора необходима для того, чтобы с помощью заранее заготовленного набора аудиозаписей обучить классификатор и сохранить его для дальнейшего использования в клиенте.</p> <p>Ниже приведена его обобщенная структура:</p> <div><p>Программа для создания классификатора</p><div><div>Модуль обработки аудио файлов</div><div>Модуль создания обучающей выборки</div><div>Модуль обучения SVM</div></div></div> <p>Рисунок 3 – Структура программы для создания классификатора</p> <p>Рассмотрим каждый модуль более подробно:</p> <p>1) модуль обработки аудиофайлов</p> <p>На вход этого модуля подается заранее подготовленный набор файлов, разбитых по 5 разным эмоциям, которые они вызывают. Далее модуль выполняет для каждого из файлов одну и ту же последовательность действий:</p> <p>1.1) открытие файла</p> <p>1.2) разные отрезки музыки могут вызывать различные эмоции, поэтому аудиозапись делится на 10 частей по 5 секунд и для каждой части выполняются следующие действия:</p> <p>1.2.1) выполняется U преобразование</p> <p>1.2.2) вычисление закрытых групп на основе U-преобразования</p> <p>1.2.3) полученные признаки добавляются в общий массив</p>				
Справ. №						
Подпись и дата						
Инв. № дубл.						
Взам. инв. №						
Подпись и дата						
Инв. № подл.						
Изм.	Лист	№ докум.	Подпись	Дата	ВКР-НГТУ-09.03.01-12В-2-009-2016ПЗ	
					Лист	
					16	

Перв. примен.					
Справ. №					
Подпись и дата					
Инв. № дубл.					
Взам. инв. №					
Подпись и дата					
Инв. № подл.					
<div> <div> <div></div> <div></div> <div></div> <div></div> <div></div> </div> <div> <div></div> <div></div> <div></div> <div></div> <div></div> </div> </div>					<div> <div>Лист</div> <div>17</div> </div>
<div> <div>Изм.</div> <div>Лист</div> <div>№ докум.</div> <div>Подпись</div> <div>Дата</div> </div>					<div>ВКР-НГТУ-09.03.01-12В-2-009-2016ПЗ</div>

Таким образом в конце этого этапа мы получаем массив, состоящий из признаков для 10 отрезков из каждого аудиофайла.

2) модуль создания обучающей выборки

На этом этапе из массива признаков создается обучающая выборка. Для этого необходимо указать названия классов для каждого из элементов массива. Затем измененный массив конвертируется в понятный для библиотеки e1071 вид, и этап подходит к концу.

3) модуль обучения

В этом модуле на основе обучающей выборке создается классификатор. Классификатор создается с использованием средств библиотеке e1071, таким образом, нам необходимо передать в метод библиотеки созданную обучающую выборку, а также ряд параметров, которые были подобраны в результате эксперимента (эти параметры влияют на то как будет происходить классификация).

Перв. примен.	3.3 Разработка структуры клиента				
	Данная часть системы представляет собой программу, в которую пользователь загружает свои аудиозаписи и получает рекомендации основываясь на них.				
Справ. №	<div>Клиентское приложение</div> <div><div>База данных</div><div>Классификатор</div><div>Модуль подбора рекомендаций</div><div>Графический интерфейс</div></div>				
	Рисунок 4 – Структура клиентской части программы				
Подпись и дата	Рассмотрим каждый модуль более подробно:				
	1) модуль рекомендаций – центральный модуль клиента. Запускается каждый раз, когда из графического интерфейса приходит запрос на поиск рекомендаций и выполняет следующий действия:				
Инв. № дубл.	1.1) получение признаков из файла, полученного из графического интерфейса – файл подвергается такой же обработке, как и при создании классификатора (файл делится на 10 частей, выполняется U преобразование, высчитываются закрытые группы)				
	1.2) подключение классификатора – программа загружает заранее подготовленный в первом приложении классификатор для использования				
Взам. инв. №	1.3) признаки, полученные на первом этапе, передаются в классификатор. В итоге получаем эмоциональный состав этой аудиозаписи.				
Подпись и дата					
Инв. № подл.					

					ВКР-НГТУ-09.03.01-12В-2-009-2016ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		18

Перв. примен.	<p>1.4) в базе данных происходит поиск записей наиболее похожих по эмоциональному составу с той которую передал пользователь. Получаем список записей, рекомендованных пользователю, которые передаем в графический интерфейс.</p> <p>1.5) состав текущей записи добавляется в базу данных, для того что бы она могла быть рекомендована в дальнейшем.</p>																							
	Справ. №	<p>2) БД – база данных хранящая в себе информацию о эмоциональном составе аудиозаписей, которые уже были обработаны программой. Именно отсюда берутся треки для рекомендаций.</p> <p>Таблица базы данных имеет следующую структуру:</p> <p>Таблица 1 – Структура базы данных</p> <table border="1"> <tr> <th>Id</th> <th>Имя файла</th> <th>Эмоция1,Эмоция2,Эмоция3,Эмоция4,Эмоция5</th> </tr> <tr> <td>Уникальный идентификатор аудиозаписи</td> <td>Имя аудиозаписи которое будет выводится пользователю</td> <td>Хранит эмоциональный состав каждой аудиозаписи</td> </tr> </table> <p>Для взаимодействия с базой данных необходимы следующие функции:</p> <p>2.1) функция поиска</p> <p>Эта функция необходима для того что бы искать в базе данных аудиозаписи с эмоциональным составом схожим с заданным.</p> <p>2.2) функция добавления</p> <p>Эта функция добавляет в базу данных информацию о новых аудиозаписях.</p> <p>3) Графический интерфейс – интерфейс для взаимодействия пользователем. Позволяет выбрать пользователю файл для классификации, а также показывает рекомендованные музыкальные файлы.</p> <p>При разработке прототипа графического интерфейса учитывались следующие необходимые возможности:</p> <ul style="list-style-type: none"> - Наличие интерфейса для выбора файла - Отображение текущего состояния программы - Отображение эмоционального состава текущей записи - Отображение списка рекомендаций 				Id	Имя файла	Эмоция1,Эмоция2,Эмоция3,Эмоция4,Эмоция5	Уникальный идентификатор аудиозаписи	Имя аудиозаписи которое будет выводится пользователю	Хранит эмоциональный состав каждой аудиозаписи													
Id		Имя файла	Эмоция1,Эмоция2,Эмоция3,Эмоция4,Эмоция5																					
Уникальный идентификатор аудиозаписи	Имя аудиозаписи которое будет выводится пользователю	Хранит эмоциональный состав каждой аудиозаписи																						
Подпись и дата	Инв. № дубл.	Взам. инв. №	Подпись и дата	Инв. № подл.	<table border="1"> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td rowspan="3"> ВКР-НГТУ-09.03.01-12В-2-009-2016ПЗ </td> <td>Лист</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td>19</td> </tr> <tr> <td>Изм.</td> <td>Лист</td> <td>№ докум.</td> <td>Подпись</td> <td>Дата</td> <td></td> </tr> </table>						ВКР-НГТУ-09.03.01-12В-2-009-2016ПЗ	Лист						19	Изм.	Лист	№ докум.	Подпись	Дата	
					ВКР-НГТУ-09.03.01-12В-2-009-2016ПЗ	Лист																		
						19																		
Изм.	Лист	№ докум.	Подпись	Дата																				

Перв. примен.	В результате прототипирования был получен следующий UI:				
	<div><div>Открыть файл</div><div>Текущий статус</div><div>Список рекомендованных аудиозаписей:<ul style="list-style-type: none">- Аудиозапись 1- Аудиозапись 2- Аудиозапись 3</div><div>Эмоциональный состав текущей аудиозаписи</div></div>				
Справ. №	Рисунок 5 – прототип графического интерфейса				
В дальнейшем этот прототип был реализован с помощью библиотеки RGtk2.					
Подпись и дата	Инв. № дубл.	Взам. инв. №	Подпись и дата	Инв. № подл.	
ВКР-НГТУ-09.03.01-12В-2-009-2016ПЗ					Лист
Изм.	Лист	№ докум.	Подпись	Дата	20

Перв. примен.	<h3 style="margin: 0;">4. Разработка программных средств</h3> <p style="margin: 5px 0;">Рассмотрим основные методы, реализованные в системе:</p> <p style="margin: 5px 0;">getClass - метод для получения эмоционального состава аудиозаписи</p> <p style="margin: 5px 0;">Таблица 2 – метод getClass</p> <table border="1" style="width: 100%; border-collapse: collapse; margin: 10px 0;"> <tr> <th style="width: 5%;">№</th> <th style="width: 25%;">Имя параметра</th> <th style="width: 70%;">Комментарий</th> </tr> <tr> <td style="text-align: center;">1</td> <td>fp</td> <td>Путь к треку для анализа</td> </tr> <tr> <td style="text-align: center;">2</td> <td>svm</td> <td>Классификатор</td> </tr> </table> <pre style="font-family: monospace; font-size: 0.9em; margin: 10px 0;"> getClass = function(fp) { song <- NULL summyArray <- NULL for (k in 1:10) { #файл делится на 10 частей w<-NULL w <- readWave(fp,units= "seconds", from = (1+(k-1)*3), to = (5+(k-1)*3))@left #для каждой части выполняется U преобразование u<-getUTrans(w,128,64,oper,flt) #получение закрытых групп a <- getfeatures('cls1d',u,fullGrp,clsGrp,oper) #признаки добавляются в общий массив summyArray <- c(summyArray,a) } #массив конвертируется дата фрейм df <- matrix(summyArray,10,840,byrow=TRUE) df <- as.data.frame(df) #получение эмоциональных составляющих файла с помощью SVM svm.pred <- predict(svm.model, df) t<-table(svm.pred) print(t) return(t) } searchSong – метод для поиска похожих аудиозаписей в базе данных Таблица 3 – метод searchSong</pre> <table border="1" style="width: 100%; border-collapse: collapse; margin: 10px 0;"> <tr> <th style="width: 5%;">№</th> <th style="width: 25%;">Имя параметра</th> <th style="width: 70%;">Комментарий</th> </tr> <tr> <td style="text-align: center;">1</td> <td>classes</td> <td>Массив из эмоций исходной аудиозаписи</td> </tr> </table>					№	Имя параметра	Комментарий	1	fp	Путь к треку для анализа	2	svm	Классификатор	№	Имя параметра	Комментарий	1	classes	Массив из эмоций исходной аудиозаписи
	№	Имя параметра	Комментарий																	
1	fp	Путь к треку для анализа																		
2	svm	Классификатор																		
№	Имя параметра	Комментарий																		
1	classes	Массив из эмоций исходной аудиозаписи																		
Справ. №																				
Подпись и дата																				
Инв. № дубл.																				
Взам. инв. №																				
Подпись и дата																				
Инв. № подл.																				

					ВКР-НГТУ-09.03.01-12В-2-009-2016ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		21

Перв. примен.	<pre> searchSong = function(classes) { library(RSQLite) con <- dbConnect(SQLite(), "db1.sqlite") rs <- dbSendQuery(con, paste0("SELECT * FROM songs WHERE fun = ", classes[1], " AND inspiration = ", classes[2], " AND calm = ", classes[3], " AND irritation = ", classes[4], " AND destruction = ", classes[5], ";")) gettedSongs= fetch(rs, n = -1) dbDisconnect(con) return(gettedSongs["name"]) } addSong – метод для добавления треков в базу данных Таблица 4 – метод addSong </pre>				
	Справ. №				
Подпись и дата					
	Инв. № дубл.				
Взам. инв. №					
	Подпись и дата				
Инв. № подл.					

```

function(name, classes)
{
  library(RSQLite)
  con <- dbConnect(SQLite(), "db1.sqlite")
  query      <-      paste0("INSERT      INTO      songs (name,
fun, inspiration, calm, irritation, destruction) VALUES

('", name, "'", ", ", classes[1], ", ", classes[2], ", ", classes[3], ", ", c
lasses[4], ", ", classes[5], ");")
  rs <- dbSendQuery(con, query)
  rs <- dbSendQuery(con, "SELECT * FROM songs;")
  print(fetch(rs, n = -1))
  dbDisconnect(con)
}

getSVM – метод для обучения SVM, а так же выводит статистику по угаданным эмоциям
для тестового набора.

```

					ВКР-НГТУ-09.03.01-12В-2-009-2016ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		22

Перв. примен.					
Справ. №					
Подпись и дата					
Инв. № дубл.					
Взам. инв. №					
Подпись и дата					
Инв. № подл.					

Таблица 5 – метод getSVM

№	Имя параметра	Комментарий
1	TrainPath	Путь к тренировочным данным
2	TestPath	Путь к тестовым данным
3	ct	Параметр цены для svm
4	gama	Параметр гамма для svm

```

function(TrainPath,TestPath, ct,gama)
{
load(paste(TrainPath, ".rdata", sep = ""))
svm.model <- svm(V841 ~ ., data = frame, type = "C-
classification", cost = ct, gamma = gama, cross = 4)
load(paste(TestPath, ".rdata", sep = ""))
svm.pred <- predict(svm.model, frame)
tb<-table(pred = svm.pred, true = frame[, (840)+1])
tb[,1]
}

```

					ВКР-ИГТУ-09.03.01-12В-2-009-2016ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		23

Перв. примен.	5 Тестирование системы				
	5.1 Тестирование клиентской части системы				
Справ. №	Пользовательский интерфейс разработанной клиентской части системы:				
	<div><div>Дипломная работа</div><div><div>Открыть файл</div><div>Анализ завершен...<div>Список рекомендуемых записей Nina Simone - Felling Good Go Down, Mosses Fly Me to The Moon</div></div><div>Веселье: 0 Раздражение: 0 Покой: 2 Разрушение: 0 Воодушевление: 8</div></div></div>				
Рисунок 6 – Интерфейс программы					
Подпись и дата	Рассмотрим порядок работы с клиентской частью:				
	<div>1) открыть приложение</div> <div>2) нажать кнопку «Открыть файл»</div> <div>3) появится окно выбора файла</div>				
Инв. № дубл.	<div><div>Open File</div><div><div><div>Места</div><div>Поиск</div><div>Недавние докум...</div><div>Documents</div><div>Рабочий стол</div><div>Локальный диск ...</div><div>Grand Theft Auto...</div><div>CD-дисковод (F:)</div></div><div><div>Имя</div><div>src</div><div>.DS_Store</div><div>I.ZIP</div><div>Список эмоций</div></div><div><div>Размер</div><div>6.0 КБ</div><div>63.5 КБ</div><div>239 байт</div></div><div><div>Изменён</div><div>21:03</div><div>06.05.2016</div><div>05.05.2016</div><div>11.05.2016</div></div></div><div><div>Добавить</div><div>Удалить</div><div>Отменить</div><div>Открыть</div></div></div>				
	Рисунок 7 – Окно выбора фалов				
Инв. № подл.	ВКР-ИГТУ-09.03.01-12В-2-009-2016ПЗ				
	Изм.	Лист	№ докум.	Подпись	Дата

Перв. примен.	<p>4) выбрать файл для подбора рекомендаций</p> <p>5) после окончания анализа появится список рекомендованных аудиозаписей</p> <p>Тестирование системы выполнялось следующим образом:</p> <p>Была создана подборка аудиозаписей (5 файлов), которые вызывают у людей схожие эмоции</p> <p>Для первых четырех аудиозаписей был получен следующий состав:</p> <p>Таблица 6 – результаты анализа аудиозаписей</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 20px;"> <tr> <th style="width: 20%;"></th> <th style="width: 15%;">Веселье</th> <th style="width: 15%;">Раздражение</th> <th style="width: 15%;">Покой</th> <th style="width: 15%;">Разрушение</th> <th style="width: 15%;">Воодушевление</th> </tr> <tr> <td>Vector-Hardcore</td> <td>1</td> <td>2</td> <td>0</td> <td>7</td> <td>0</td> </tr> <tr> <td>Dead Go - I wanna be your</td> <td>0</td> <td>2</td> <td>0</td> <td>8</td> <td>0</td> </tr> <tr> <td>I See You – Rules</td> <td>2</td> <td>3</td> <td>0</td> <td>5</td> <td>0</td> </tr> <tr> <td>Pulse – Dead</td> <td>1</td> <td>1</td> <td>0</td> <td>8</td> <td>0</td> </tr> </table> <p style="margin-top: 20px;">Как видно их эмоциональный состав схож, т.е. система определения эмоций работает.</p>						Веселье	Раздражение	Покой	Разрушение	Воодушевление	Vector-Hardcore	1	2	0	7	0	Dead Go - I wanna be your	0	2	0	8	0	I See You – Rules	2	3	0	5	0	Pulse – Dead	1	1	0	8	0	Справ. №
		Веселье	Раздражение	Покой	Разрушение	Воодушевление																														
Vector-Hardcore	1	2	0	7	0																															
Dead Go - I wanna be your	0	2	0	8	0																															
I See You – Rules	2	3	0	5	0																															
Pulse – Dead	1	1	0	8	0																															
Подпись и дата						Инв. № дубл.																														
Взам. инв. №						Подпись и дата																														
Инв. № подл.						Лист																														

Перв. примен.	Теперь загрузим в программу оставшийся файл и посмотрим список рекомендованных аудиозаписей:																					
	<div><div>Дипломная работа</div><div><div>Открыть файл</div><div>Анализ завершен...<div>Список рекомендуемых записей Vector- Hardcore Dead Go - I wanna be your I See You -Rules Pulse – Dead</div></div><div>Веселье: 1 Раздражение: 0 Покой: 0 Разрушение: 9 Воодушевление: 0</div></div></div>																					
Справ. №	Рисунок 8 – Результат работы программы																					
В результате список рекомендаций содержит аудиозаписи, которые были загружены ранее и схожи по вызываемым эмоциям с последним. Т.е. система работает.																						
Инов. № подл.	Подпись и дата																					
	Инов. № дубл.																					
	Взам. инв. №																					
	Подпись и дата																					
<table><tr><td></td><td></td><td></td><td></td><td></td><td rowspan="3">ВКР-НГТУ-09.03.01-12В-2-009-2016ПЗ</td><td rowspan="3">Лист 26</td></tr><tr><td>Изм.</td><td>Лист</td><td>№ докум.</td><td>Подпись</td><td>Дата</td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr></table>											ВКР-НГТУ-09.03.01-12В-2-009-2016ПЗ	Лист 26	Изм.	Лист	№ докум.	Подпись	Дата					
					ВКР-НГТУ-09.03.01-12В-2-009-2016ПЗ	Лист 26																
Изм.	Лист	№ докум.	Подпись	Дата																		

Перв. примен.															
Справ. №															
Подпись и дата															
Инв. № дубл.															
Взам. инв. №															
Подпись и дата															
Инв. № подл.															
<div> <div>5.2 Оценка эффективности системы</div> <div> <p>Для оценки эффективности системы был использован следующий метод: была создана тестовая подборка аудиозаписей, для которых были определены классы. В данном случае подборка состояла из 25 музыкальных файлов – по 5 файлов на каждый класс.</p> <p>Эти файлы были переданы в обученный классификатор. Далее в полученном эмоциональном составе для каждой записи выбиралась доминирующая эмоция и она сравнивалась с заранее заданным классом – если они равны, то система верно определила класс.</p> <p>Были получены следующие результаты для тестовой выборки:</p> <p>Веселье - верно определено 4 из 5</p> <p>Воодушевление - верно определено 5 из 5</p> <p>Покой - верно определено 3 из 5</p> <p>Раздражение - верно определено 3 из 5</p> <p>Разрушение - верно определено 4 из 5</p> <p>Итог: Для 19 файлов из 25 классы определены верно, т.е. эффективность системы составляет 76%</p> </div> </div>															
<table border="1"> <tr> <td>Изм.</td> <td>Лист</td> <td>№ докум.</td> <td>Подпись</td> <td>Дата</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </table>					Изм.	Лист	№ докум.	Подпись	Дата						<div> <div>ВКР-НГТУ-09.03.01-12В-2-009-2016ПЗ</div> <div>Лист</div> <div>27</div> </div>
Изм.	Лист	№ докум.	Подпись	Дата											

Перв. примен.					
Справ. №					
Подпись и дата					
Инв. № дубл.					
Взам. инв. №					
Подпись и дата					
Инв. № подл.					
<div> <div> <div>Изм.</div> <div>Лист</div> <div>№ докум.</div> <div>Подпись</div> <div>Дата</div> </div> <div> ВКР-НГТУ-09.03.01-12В-2-009-2016ПЗ </div> <div> <div>Лист</div> <div>28</div> </div> </div>					

Заключение

В результате выполнения выпускной работы спроектирована и программно реализована система рекомендации музыки на основе эмоционального отклика.

Созданная система предназначена для рекомендации музыки людям основываясь на тех эмоциях что музыка вызывает у человека. Тестирование системы подтвердило её работоспособность и возможность использования для решения поставленной задачи.

В дальнейшем на основе созданной системы планируется разработать ее веб-версию.

Перв. примен.		<div>Перечень условных обозначений</div> <div>ООП - Объектно-ориентированное программирование</div> <div>SVM - support vector machine - Метод опорных векторов</div> <div>UI - user interface- пользовательский интерфейс</div> <div>БД – база данных</div>					
Справ. №							
Подпись и дата							
Инв. № дубл.							
Взам. инв. №							
Подпись и дата							
Инв. № подл.							
					ВКР-НГТУ-09.03.01-12В-2-009-2016ПЗ		Лист
Изм.	Лист	№ докум.	Подпись	Дата			29

Перв. примен.	<p align="center">Список литературы</p> <p>1. Информационный подход к описанию звукового сигнала - Гай В.Е. -Труды МФТИ, Том 6, № 2, 2014, С. 167-173.</p> <p>2.Оценка эмоционального состояния человека по голосу с позиций теории активного восприятия - В. Е. Гай, В. А. Утробин, П. А. Родионов, М. О. Дербасов - Системы управления и информационные технологии, №1.1(59), 2015, С. 118-122.</p> <p>3. Машина опорных векторов - http://www.machinelearning.ru/wiki/index.php?title=SVM</p> <p>4. Мастицкий С.Э., Шитиков В.К. "Статистический анализ и визуализация данных с помощью R".</p> <p>5. Как это работает? Рекомендации в Яндекс.Музыке - https://yandex.ru/blog/company/92883</p> <p>6. «ВКонтакте» обновила рекомендации аудиозаписей - https://hi-tech.mail.ru/news/vk-music/</p> <p>7. Как работают рекомендации в Apple Music и как им помочь - http://appleinsider.ru/tips-tricks/kak-rabotayut-rekomendacii-v-apple-music-i-kak-im-pomoch.html</p> <p>8. Утробин, Владимир Александрович “Компьютерная обработка изображений. Основы специальной теории восприятия” - Нижний Новгород : Нижегородский гос. технический ун-т им. Р. Е. Алексеева, 2015</p>								
	Справ. №								
Подпись и дата	Инв. № дубл.	Взам. инв. №	Подпись и дата	Инв. № подл.					
					<div> <div>ВКР-НГТУ-09.03.01-12В-2-009-2016ПЗ</div> <div> <div>Изм.</div> <div>Лист</div> <div>№ докум.</div> <div>Подпись</div> <div>Дата</div> </div> </div>				
					<div> <div>Лист</div> <div>30</div> </div>				

Перв. примен.						
Справ. №						
<pre> loadedData[[i]][j] <- dataList[[1]][2] } } } summaryArray <- NULL for (i in 1:5) { for (j in 1:10) { for (k in 1:10) { summaryArray <- c(summaryArray, waveList[[i]][10*(j-1) + k]) } } } frame <- matrix(summaryArray, 5*5*10, 840, byrow=TRUE) frame <- as.data.frame(frame) classNames <- NULL for (i in 1:5) { for (j in 1:5) { for (k in 1:10) { classNames <- c(classNames, folders[i]) } } } classesMatrix <- matrix(classNames, 5*5*10, 1, byrow=FALSE) frame[, 841] <- classesMatrix frame1 <- frame index <- 1:nrow(frame) testindex <- sample(index, trunc(length(index)/8)) aaa <- -testindex testset <- frame[testindex,] trainset <- frame[-testindex,] frame <- frame1 save(frame, file = paste("C:\\dpr\\dp\\src\\", "frame1PokoiTest", ".rdata", sep = "")) load(paste("C:\\dpr\\dp\\src\\", "frame1PokoiTrain", ".rdata", sep = "")) svm.model <- svm(V841 ~ ., data = frame, type = "C-classification", cost = 5000, gamma = 0.001, cross = 4) load(paste("C:\\dpr\\dp\\src\\", "frame1PokoiTest", ".rdata", sep = "")) svm.pred <- predict(svm.model, frame) tb <- table(pred = svm.pred, true = frame[, (840)+1]) tb[, 1]</pre>						
Подпись и дата						
Инв. № дубл.						
Взам. инв. №						
Подпись и дата						
Инв. № подл.						

					ВКР-НГТУ-09.03.01-12В-2-009-2016ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		32

Перв. примен.	<div>AddSong.R</div> <pre>function(name, classes) { library(RSQLite) con <- dbConnect(SQLite(), "db1.sqlite") query <- paste0("INSERT INTO songs(name, fun, inspiration, calm, irritation, destruction) VALUES ('", name, "'", ", ", classes[1], ", ", classes[2], ", ", classes[3], ", ", class es[4], ", ", classes[5], ");") rs <- dbSendQuery(con, query) rs <- dbSendQuery(con, "SELECT * FROM songs;") print(fetch(rs, n = -1)) dbDisconnect(con) }</pre> <div>SearchSong.R</div> <pre>searchSong = function(classes) { library(RSQLite) con <- dbConnect(SQLite(), "db1.sqlite") rs <- dbSendQuery(con, paste0("SELECT * FROM songs WHERE fun = ", classes[1], " AND inspiration = ", classes[2], " AND calm = ", classes[3], " AND irritation = ", classes[4], " AND destruction = ", classes[5], " ;")) gettedSongs= fetch(rs, n = -1) dbDisconnect(con) return(gettedSongs["name"]) }</pre> <div>GetClass.R</div> <pre>getClass = function(fp) { song <- NULL summyArray <- NULL for (k in 1:10) { #файл делится на 10 частей w<-NULL w <- readWave(fp, units= "seconds", from = (1+(k-1)*3), to = (5+(k-1)*3))@left #для каждой части выполняется U преобразование u<-getUTrans(w, 128, 64, oper, flt) #получение закрытых групп a <- getfeatures('cls1d', u, fullGrp, clsGrp, oper)</pre>					
	Справ. №					
Подпись и дата	Инв. № дубл.					
Взам. инв. №	Подпись и дата					
Инв. № подл.						
		ВКР-НГТУ-09.03.01-12В-2-009-2016ПЗ				
		Лист 33				
		Изм.	Лист	№ докум.	Подпись	Дата

Перв. примен.	<pre> #признаки добавляются в общий массив summyArray <- c(summyArray,a) } #массив конвертируется дата фрейм df <- matrix(summyArray,10,840,byrow=TRUE) df <- as.data.frame(df) #получение эмоциональных составляющих файла с помощью SVM svm.pred <- predict(svm.model, df) t<-table(svm.pred) print(t) return(t) } </pre>								
	Справ. №	<p>Gui.R</p> <pre> library("RGtk2") source("getClass.R") source("addSong.R") source("searchSong.R") window = gtkWindow() window["title"] = «Дипломная работа» frame = gtkFrameNew("") window\$add(frame) vbox = gtkVBoxNew(FALSE, 8) vbox\$setBorderWidth(24) frame\$add(vbox) upView = gtkFrameNew("") vbox\$packStart(upView, FALSE, FALSE, 5) downView = gtkFrameNew("") vbox\$packStart(downView, FALSE, FALSE, 5) hbox = gtkHBoxNew(FALSE, 8) vbox1 = gtkVBoxNew(FALSE, 2) openFl=gtkButton(«Открыть файл») status = gtkLabel("\n\n Идет анализ... \n\n",TRUE) progress = gtkLabel("\n \n\n\n\n\n\n",TRUE) vbox1\$packStart(openFl, FALSE, FALSE, 0) vbox1\$packStart(status, FALSE, FALSE, 0) vbox1\$packStart(progress, FALSE, FALSE, 0) table = gtkLabel(«Список рекомендуемых записей \n\n\n\n\n\n\n\n\n\n\n\n\n",TRUE) hbox\$packStart(vbox1, FALSE, FALSE, 5) hbox\$packStart(table, FALSE, FALSE, 5) upView\$add(hbox) spectr = gtkLabel("",TRUE) downView\$add(spectr) spectr\$setText("") </pre>							
Подпись и дата		Инв. № дубл.	Взам. инв. №	Подпись и дата	Инв. № подл.	<div style="display: flex; justify-content: space-between; align-items: center;"> <div style="width: 40%;"> <p>ВКР-ИГТУ-09.03.01-12В-2-009-2016ПЗ</p> </div> <div style="width: 10%; text-align: center;"> <p>Лист</p> <p>34</p> </div> </div>			
	Изм.					Лист	№ докум.	Подпись	Дата

Перв. примен.						<pre> openFunc<-function (button) { dialog <- gtkFileChooserDialog("Open File", window, "open", "gtk-cancel", GtkResponseType["cancel"], "gtk-open", GtkResponseType["accept"]) if (dialog\$run() == GtkResponseType["accept"]) { filepath <- dialog\$getFilename() filename = basename(filepath) } dialog\$destroy() status\$setText("\n\n Идет анализ... \n\n") qt<-getClass(filepath) formattedStr <- NULL for (z in 1:5) { formattedStr = paste0(formattedStr,paste0(folders[z],": ",qt[z]," ")) } spectr\$setText(formattedStr) formattedStr <- «Список рекомендуемых записей \n" ss = searchSong(qt) for (z in 1:length(ss[,])) { formattedStr = paste0(formattedStr,paste0(ss[z,]," \n")) } table\$setText(formattedStr) addSong(filename,qt) status\$setText("\n\n анализ завершен... \n\n") } gSignalConnect(openFl, "clicked", openFunc) </pre>
Подпись и дата						
Взам. инв. №						
Подпись и дата						
Инов. № подл.						

Изм.	Лист	№ докум.	Подпись	Дата	ВКР-НГТУ-09.03.01-12В-2-009-2016ПЗ	Лист
						35