МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное образовательное учреждение высшего образования «Крымский федеральный университет имени В.И. Вернадского»

**ТАВРИЧЕСКИЙ КОЛЛЕДЖ**

(структурное подразделение)

**КУРСОВАЯ РАБОТА**

по **МДК.1.2 Прикладное программирование**

на тему:

**Разработка сетевого приложения «Пианино»**

Выполнил: **Халевин Кирилл Александрович** \_\_\_\_\_\_\_\_\_\_\_

Специальность: 09.02.03 Программирование в компьютерных системах

Группа **3ПКС10**

Научный руководитель: **Железняк Александр Владимирович** \_\_\_\_\_\_\_\_\_\_

Г. Симферополь, 2018

Оглавление

[Введение 3](#_Toc512199276)

[Глава 1 – Подготовка к созданию приложения 4](#_Toc512199277)

[1.1 Выбор среды разработки NetBeans с языком Java. Плюсы и минусы. 4](#_Toc512199278)

[Глава 2 – Разработка сетевого приложение «Пианино» 5](#_Toc512199279)

[2.1 Создание нового проекта 5](#_Toc512199280)

[2.2 Проектирование формы 7](#_Toc512199281)

[2.3 Создание пианино для одного пользователя 10](#_Toc512199282)

[2.4 Создание сервера и клиента, для проигрывания звука на другом компьютере 15](#_Toc512199283)

[Глава 3 – Тестирование приложения 18](#_Toc512199284)

[3.1 Подготовка приложения к сети 18](#_Toc512199285)

[3.2 Тестирование приложения 20](#_Toc512199286)

[Выводы 22](#_Toc512199287)

[Список литературы 23](#_Toc512199288)

# Введение

В наше время, всё больше предметов оцифровывается. Теперь можно оплатить онлайн покупку цифровыми деньгами не выходя из дома. Можно общаться в цифровом чате и при этом не говорить ни слова. Можно играть в цифровом мире и совсем забыть про реальный.

Музыка тоже обрела цифровой вид. И если раньше её можно было слышать только когда кто-то играл рядом на музыкальном инструменте, то теперь музыка есть в цифровом виде почти у каждого человека в его цифровом устройстве.

Большая часть музыкальных инструментов уже воспроизводят цифровой звук, а большое множество музыкальных инструментов уже есть в цифровом виде. Актуальность курсовой работы в том, что при помощи сетевого приложения, двое человек, сидев каждый за своим компьютером, смогут играть музыку вдвоём на расстоянии.

Цель – разработать сетевое приложение “Пианино” на языке Java в котором можно будет играть при помощи клавиатуры и играть вдвоём по сети. А так же разработать переключатель инструментов.

Задачи:

-Рассмотреть возможности Java при работе со звуками.

-Разработать пианино

-Сделать переключатель октавы и инструмента

-Разработать серверную и клиентскую часть

Объект исследования – процесс создания приложения.

Предмет – создание сетевого приложения “Пианино” в среде разработки NetBeans на языке Java.

# Глава 1 – Подготовка к созданию приложения

## Выбор среды разработки NetBeans с языком Java. Плюсы и минусы.

Один из плюсов среды разработки NetBeans, это дизайн кнопок. В NetBeans кнопки имеют градиентную заливку, а не монотонный цвет как в некоторых других средах разработки. А при наведении курсора на кнопку, она подсвечивается и при нажатии темнеет. В других же средах разработки чтобы прийти к такому же дизайну, нужно создать свою прорисовку кнопки в разных состояниях и запрограммировать несколько событий на смену картинки.

Следующий плюс, это то что уже присутствует библиотека в языке Java для работы с MIDI интерфейсом. То есть, для проигрывания нот, вместо использования аудио файлов с записанными нотами, можно создать объект Synthesizer и командами проигрывать ноты. Взаимодействие с MIDI интерфейсом позволяет избежать поиска и записи нот некоторых музыкальных инструментов.

Один из основных недостатков NetBeans, это динамическое размещение объектов. Конечно при изменении размера формы в режиме разработки, динамическое размещение позволяет не перемещать объекты, но иногда это создаёт лишние пустые поля. А при переносе объекта на объект, может сдвинуться половина объектов, которые сдвигать не нужно. Динамическое размещение можно выключить, но если это не сделать в самом начале, то все панели изменят свою ширину и высоту на ноль, а все объекты поменяют свою позицию. Что приведёт к разрушению готового дизайна и введёт в дискомфорт разработчика.

Ещё недостатком является требование – должна быть установлена на компьютере Java Virtual Machine, чтобы можно было запустить приложение.

# Глава 2 – Разработка сетевого приложение «Пианино»

## 2.1 Создание нового проекта

Разрабатывать проект будем на языке программирование Java в среде разработки NetBeans. Запустим среду разработки и создадим новый проект. Для этого нажмём Ctrl+Shift+N и выберем Java>Приложение Java(Рисунок 1)

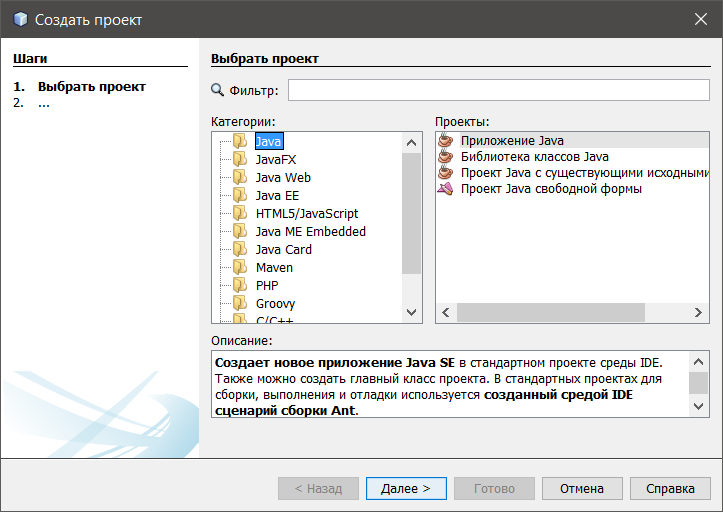


Рисунок 1 – Выбор языка программирования для проекта

Нажимаем далее, задаём имя проекта, указываем путь расположения проекта, убираем галочку с чекбокса “Создать главный класс” и нажимаем готово(Рисунок 2)

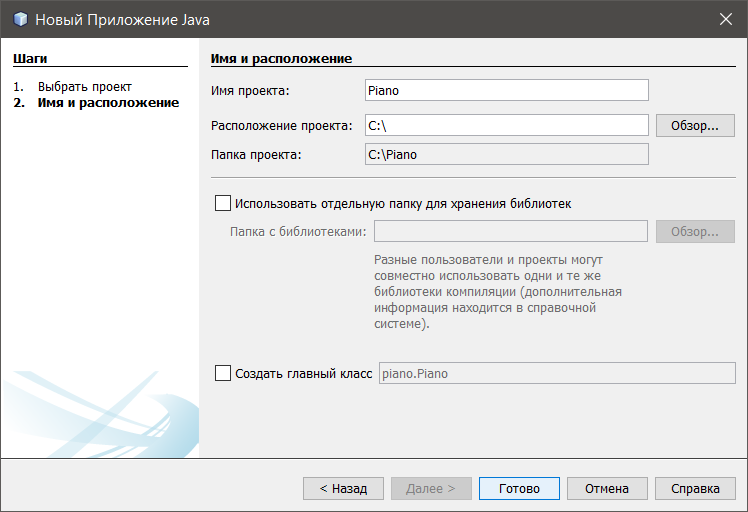


Рисунок 2 – Создание проекта

Проект создан, но пока ещё нет одной главной детали. Нужно добавить графический объект – форму, с которой будет взаимодействовать пользователь. Для этого нажимаем Ctrl+N и выбираем форму JFrame во вкладке – формы Swing GUI.(Рисунок 3)

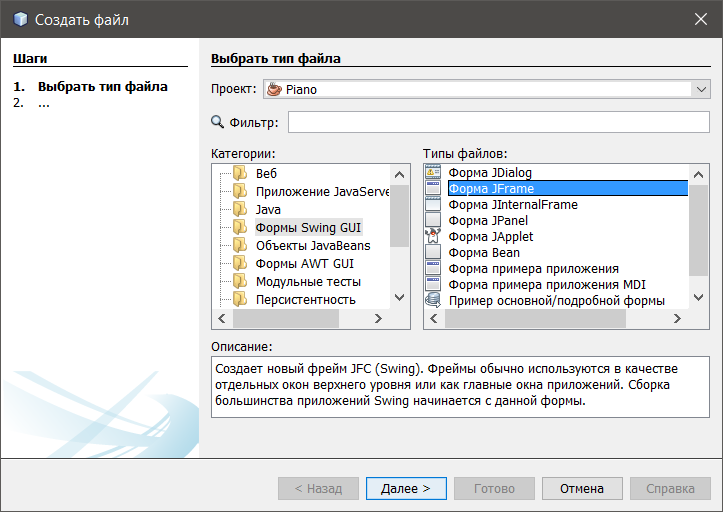


Рисунок 3 – создание формы JFrame

Нажимаем далее и нажимаем готово. Запускаем проект клавишей F6 на клавиатуре и при первом запуске появится окно с выбором главного класса. Выбираем форму которую добавили и нажимаем ОК(Рисунок 4)

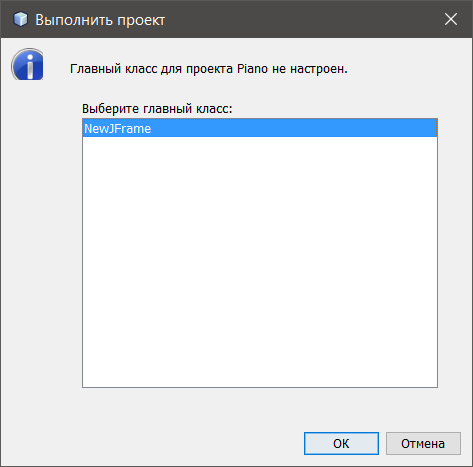


Рисунок 4 – Выбор главного класса

## 2.2 Проектирование формы

Добавляем на форму панель и в свойствах панели, меняем background(цвет панели) на тёмный цвет. (Рисунок 5)

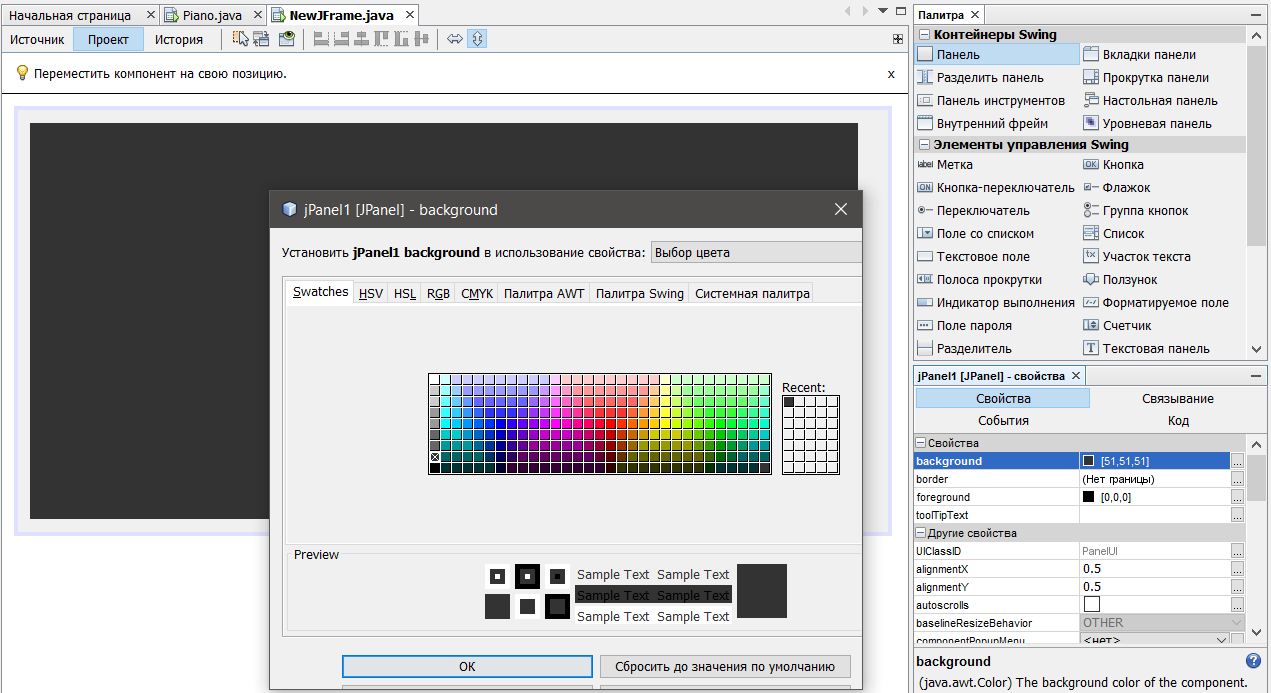


Рисунок 5 – выбор цвета для панели

Теперь нажимаем правой кнопкой мыши по панели и в контекстном меню выбираем “Установить расположение”>”Нет размещения”, чтобы можно было поставить один объект, поверх другого.

Меняем размер панели и добавляем на панель две кнопки. У одной кнопки меняем Background в свойствах на чёрный цвет, а у другой на белый. И меняем их размер. (Рисунок 6)

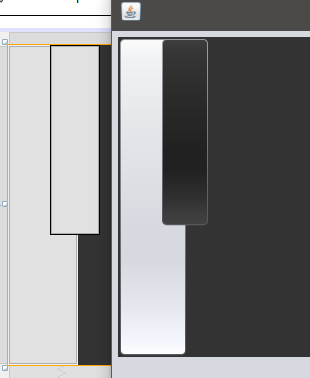


Рисунок 6 – белая и чёрная кнопка

Делаем точно такие же кнопки в количестве двух октав = 14 белых кнопок + 10 чёрных кнопок. Для это можно просто выделить кнопку и нажать Ctrl+C чтобы скопировать и Ctrl+V чтобы добавить ещё одну кнопку. В окне дерева объектов, переносим чёрные кнопки наверх, чтобы они были поверх белых кнопок и их было видно.(Рисунок 7)

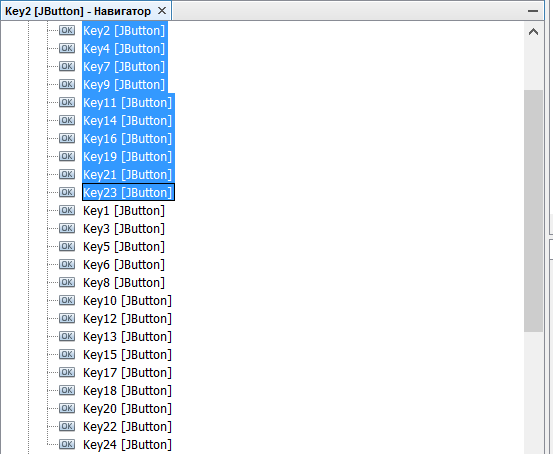


Рисунок 7 – Перенос чёрных кнопок в дереве объектов

Расставляем кнопки и делаем из них две октавы. (Рисунок 8)

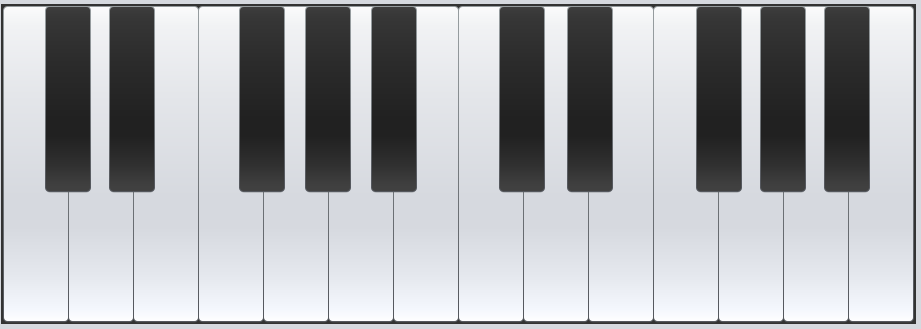


Рисунок 8 – расставленные кнопки по своим местам

Добавим ещё сверху 8 кнопок, 6 меток и одно текстовое поле. Текстовое поле нужно будет для подключения. Добавим кнопку Connect и ещё добавим метку с портом 676 к которому будет подключение. Правее поставим метку с надписью Server и две кнопки On и Off, для включения и выключения сервера. Ещё две метки будут называться Octave и Instrument и под ними поставим по одной метки и две кнопки, чтобы можно было переключать октаву и инструмент. Добавим кнопку stop, для остановки звука. (Рисунок 9)

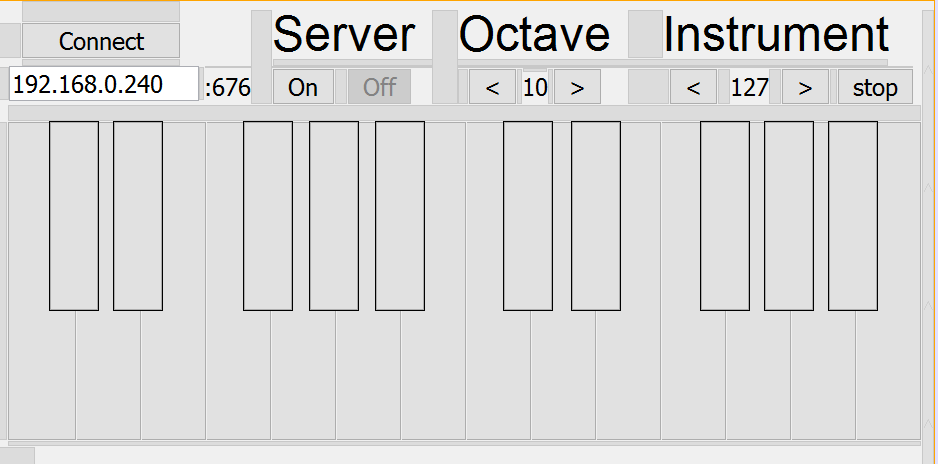


Рисунок 9 – Новые кнопки, метки и текстовое поле, поверх клавиш октав

## 2.3 Создание пианино для одного пользователя

Создадим новые классы. Введём название первого класса MainSounds и создадим второй класс с названием Ready. Объект класса MainSounds будет проигрывать звуки, а объект класса Ready будет передавать информацию объекту класса MainSounds, какую ноту пользователь нажал и какой инструмент он выбрал.(Рисунок 10)



Рисунок 10 – Передача информации в Ready>MainSounds

В классе Ready создадим пустой конструктор. Для этого внутри класса пропишем public Ready(){} Добавим классу свойства

Private int note, instrumentN;

Public Boolean Ready = false;

И добавим методы для получения и обновления информации в объекте. (Рисунок 11)

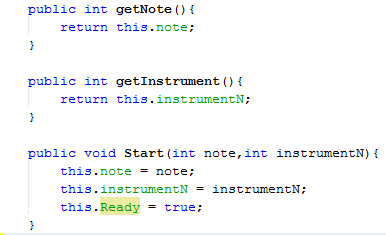


Рисунок 11 – методы класса Ready

Создадим новый объект класса Ready, в главной форме JFrame

public Ready ready = new Ready();

Теперь перейдём к написанию класса MainSounds. Добавим к классу implements Runnable, и добавим функцию run(), чтобы функция выполнялась в другом потоке, параллельно. Добавим пустой конструктор (Рисунок 12)

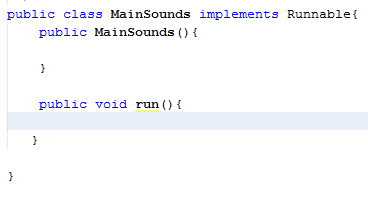


Рисунок 12 – создание класса потока

Инициализируем объект в главной форме.

private Thread MainThread = new Thread(new MainSounds());

И в событии главной формы windowOpened запускаем поток

MainThread.start();

Теперь запрограммируем функцию run в новом потоке. Добавим объект класса Synthesizer, который позволит нам работать с MIDI интерфейсом. Включим бесконечный цикл с задержкой в 0.01 секунду, чтобы сильно не нагружать процессор. И добавим оператор условия, который будет проверять объект ready на готовность к воспроизведению звуков. (Рисунок 13)

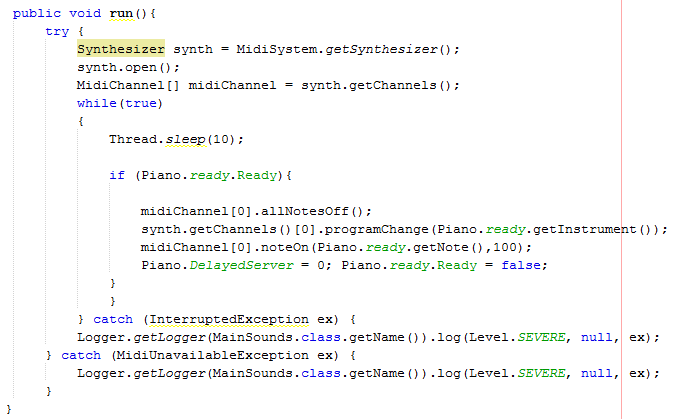


Рисунок 13 – функция run

Для обозначения октавы и id инструмента, добавим две переменные в главной форме.

private int Octave = 4;

private int ins = 1;

Теперь для каждой кнопки для события actionPerformed добавляем

ready.Start((N-1)+Octave\*12, ins);

Где N – это порядковый номер кнопки. После добавления этой функции к каждой кнопки, по нажатию будет проигрываться звук. Для переключения октавы, добавим на кнопку ‘<’ следующий код:

Octave--;

if (Octave < 0){ Octave = 9;}

OctaveField.setText(String.valueOf(Octave+1));

А для кнопки ‘>’:

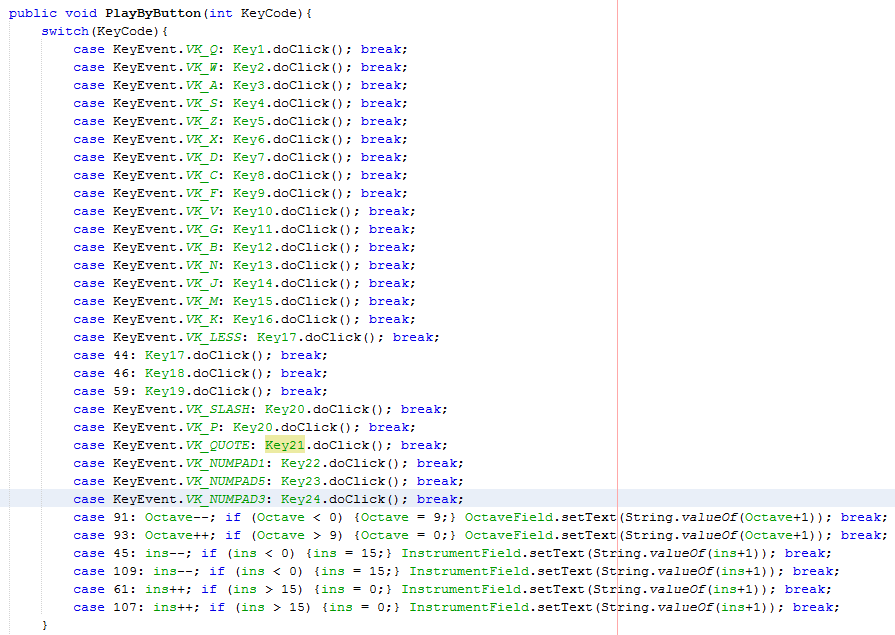
Octave++;

if (Octave > 9){ Octave = 0;}

OctaveField.setText(String.valueOf(Octave+1));

То же самое прописываем в кнопках для смены инструментов, только вместо Octave берём переменную ins и в условии 9 меняем на 15.

Чтобы пользователь мог нажимать клавиши не только мышкой, но ещё и при помощи клавиатуры, для этого нужно считывать код клавиш и нажимать клавиши на форме программно. Сделаем функцию PlayByButton(int KeyCode) в главной форме (Рисунок 14)

Рисунок 14 – функция PlayByButton

Чтобы эта функция запускалась, нам нужно добавить её в событие KeyPressed формы. Но событие будет работать только когда фокус настроен на форму. Для этого со всех кнопок формы, нужно переключить свойство focusable с true на false, а для поля ввода нужно добавить событие KeyTyped и разрешить вводить только цифры, точку и BackSpace. При нажатии на любую другую клавишу, делаем фокусировку на форму.(Рисунок 15)

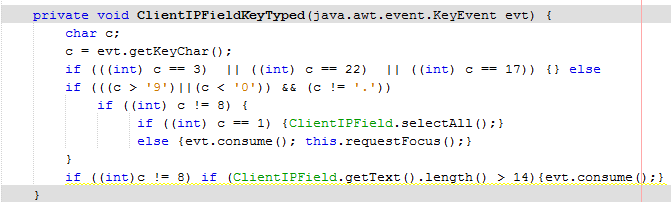


Рисунок 15 – событие поля ввода KeyTyped

## 2.4 Создание сервера и клиента, для проигрывания звука на другом компьютере

Для создания сервера, создадим новый класс Server и сделаем из него поток и добавим функцию run(). Добавим свойство port и сделаем конструктор с изменением свойства port. (Рисунок 16)

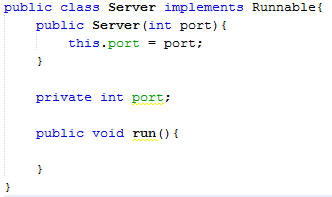


Рисунок 16 – класс Server

В функции run() класса Server создадим серверный сокет и новый объект класса Synthesizer. Будем ожидать сообщений от клиента в виде двух чисел. Где первое число нота, а второе число номер инструмента в MIDI интерфейсе. Переведём сообщение из строки в числа и проиграем полученную ноту на сервере. (Рисунок 17)



Рисунок 17 – функция run которая принимает сообщения

Добавим две функции. startServer – которая будет запускать сервер. И stopServer – которая будет останавливать сервер. В startServer сделаем инициализацию объекта server и запустим новый поток. А так же заблокируем кнопку On и разблокируем кнопку Off, чтобы не было ошибки по повторному нажатию кнопки включения сервера. В функции stopServer остановим поток и сделаем снова инверсию кнопок. К кнопке On добавим запуск функции startServer, а к кнопке Off добавим функцию stopServer. (Рисунок 18)

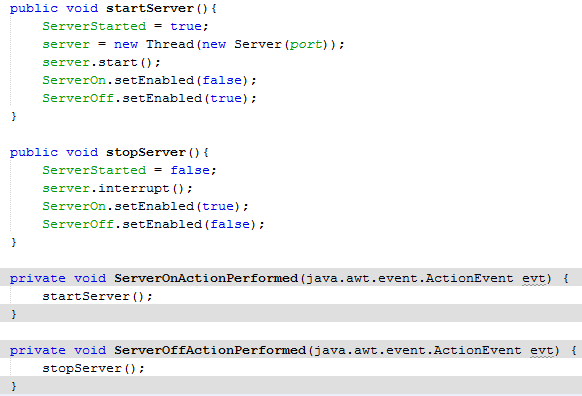


Рисунок 18 – Функции включения и выключения сервера

Теперь в приложении можно включить сервер и при получении текстового сообщения из двух чисел через пробел, поток Server запустит на компьютере звук.

Перед созданием клиента добавим строковую переменную IP и булевую переменную Connect в главную форму. И в класс MainSounds добавим условие: если Connect будет равно истине, то создадим новый клиентский сокет который подключится к серверному сокету по IP указанному пользователем. После чего, передадим серверу текстовое сообщение с двумя числами(номер ноты и номер инструмента). (Рисунок 19)

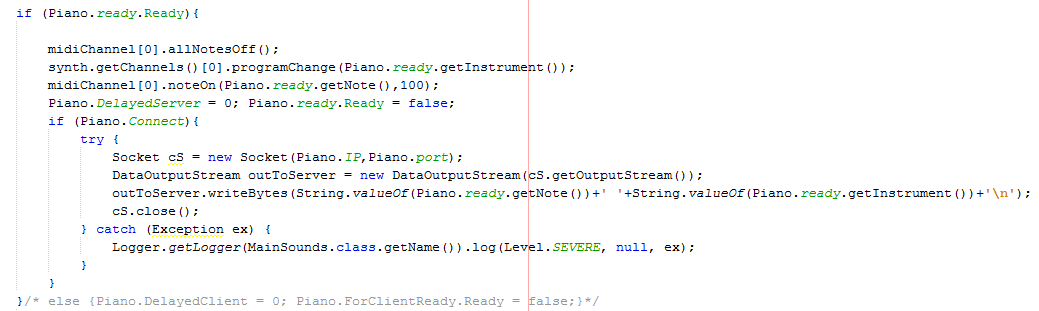


Рисунок 19 – код клиента

Запрограммируем кнопку Connect для подключения к другому пользователю. При нажатии на кнопку, IP адрес из поля ввода занесётся в переменную IP. Булевой переменной Connect изменим значение на истину и поменяем название кнопки с Connect на Disconnect. А если приложение подключено, то кнопка поменяет название обратно, а Connect примет значение ложь. (Рисунок 20)

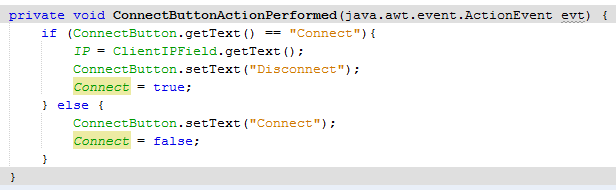


Рисунок 20 – код кнопки Connect

Теперь можно подключаться к другому серверу, но обратного подключения не будет. Чтобы было обратное подключение, при нажатии на Connect добавляем включение сервера. В классе Server после получения сообщения, меняем значение переменной Connect на true и из сокета берём IP адрес клиента и заносим его в переменную IP. После чего оба устройства являются и клиентом и сервером.

# Глава 3 – Тестирование приложения

## 3.1 Подготовка приложения к сети

Для подключения по локальной сети к другому человеку, у него должен быть включён сервер и он должен передать свой локальный IP адрес. Чтобы узнать локальный IP адрес компьютера в Windows, можно нажать Win+R и открыть командную строку прописав там cmd. А в командной строке прописать ipconfig и найти IPv4.(Рисунок 21)

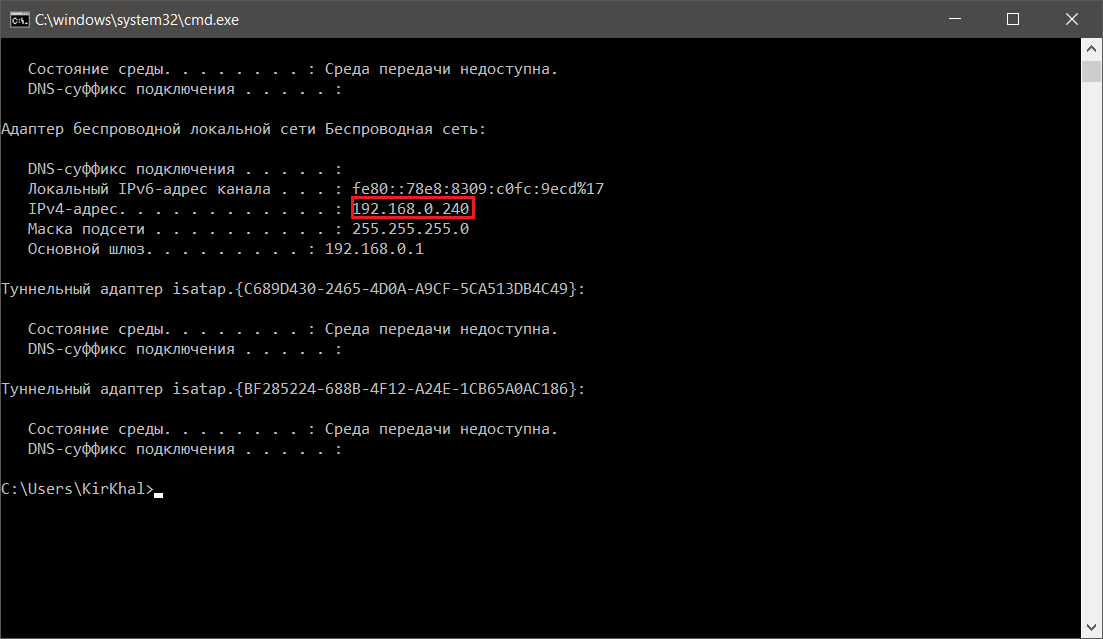


Рисунок 21 – ipconfig и IP локальной сети

Теперь первому человеку нужно ввести в поле ввода полученный IP адрес и кликнуть на кнопку Connect. После этого, нажатие на любую клавишу пианино, будет проигрывать звук на обоих устройствах в локальной сети.

При подключении по глобальной сети, нужно узнать глобальный IP адрес. Для этого пишем в онлайн поисковике “Мой IP адрес”, потом переходим на любой сайт и там берём глобальный IP адрес. Пример сайта где можно взять глобальный IP – 2ip.ru (Рисунок 22)

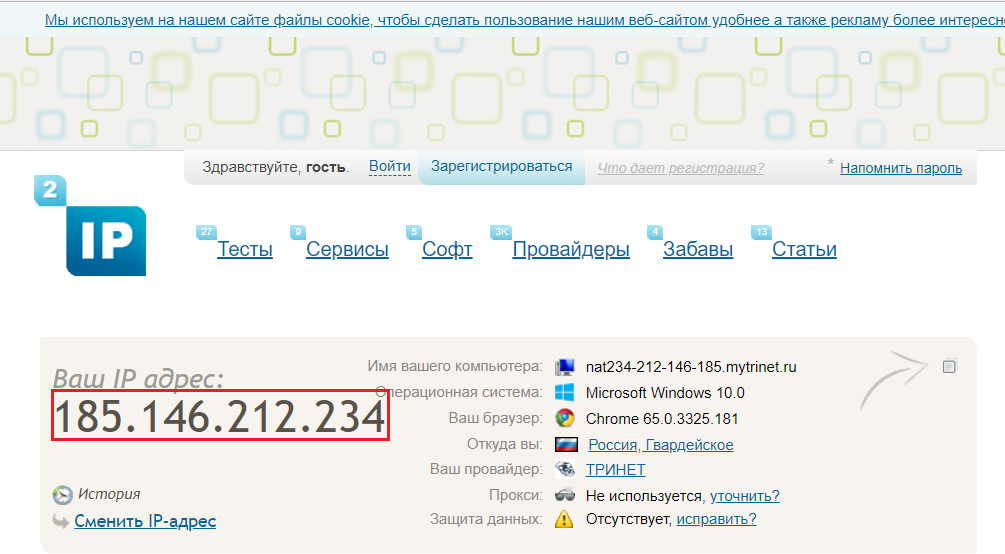


Рисунок 22 – 2ip.ru сайт для проверки глобального IP

Для подключения по глобальному IP адресу, нужно чтобы все устройства маршрутизации в сети, перенаправляли запросы компьютеру с приложением. В основном в домах за маршрутизацию отвечает только одно устройство – роутер. Для перенаправления запросов в роутере, нужно добавить в личном кабинете триггер. Чтобы это сделать нужно зайти в командную строку и прописать ipconfig. Ниже IPv4 находим основной шлюз и копируем его IP. Открываем веб-браузер и вводим скопированный IP адрес в адресную строку. Вводим логин и пароль для входа в личный кабинет. По умолчанию, логином и паролем обычно является слово – admin, в противном случае логин и пароль можно найти в инструкции или в Интернете. Добавляем в настройках триггер или виртуальный сервер и вносим порт 676 с перенаправлением на локальный IP адрес. Теперь сохраняем настройки и перезагружаем роутер, если это требуется.

## 3.2 Тестирование приложения

Когда на роутерах открыли порт 676 с перенаправлением на локальный компьютер, на одном компьютере нажимаем кнопку On чтобы включить сервер. После нажатия, она должна стать серой, а кнопка Off должна активироваться(Рисунок 23)

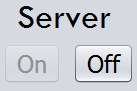


Рисунок 23 – Состояние кнопок, при включенном сервере

А на другом компьютере введём IP адрес и нажмём Connect. Connect должно поменяться на Disconnect (Рисунок 24). А кнопки On и Off должны переключиться, если сервер выключен. (Рисунок 23)

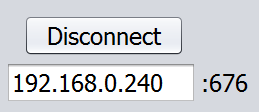


Рисунок 24 – Изменение кнопки Connect после нажатия

Теперь по нажатию на клавиши пианино, звук играет на обоих компьютерах.

При нажатии на кнопку Disconnect, клиент прекращает отправлять сообщения и сервер выключается. Кнопка On снова становиться доступной, а Disconnect обратно переименовывается на Connect (Рисунок 25)

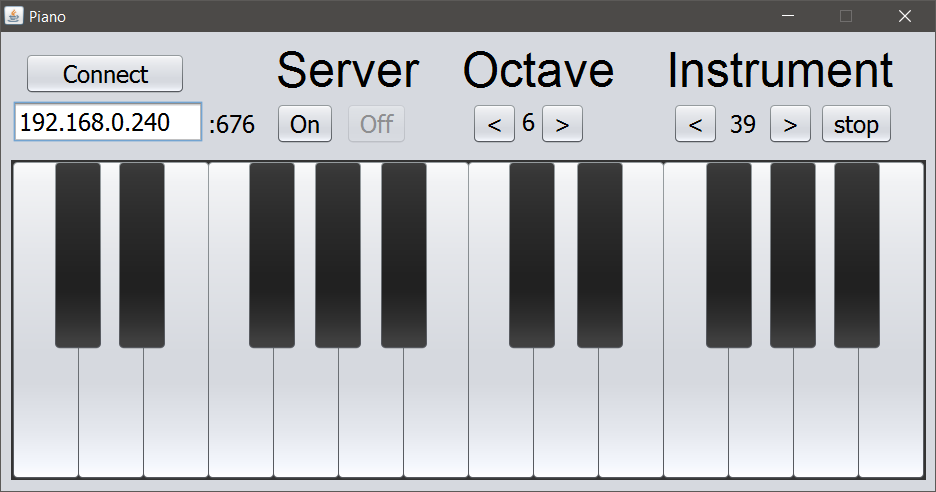


Рисунок 25 – Вид формы после отключения от сервера

Если в подключенном состоянии, на другом компьютере выключится приложение, или человек отключит сервер, или при подключении будет указан IP компьютера которого нет в сети, звуки пианино не будут работать на локальном компьютере секунд десять, пока приложение будет пытаться подключиться.

# Выводы

В течении курсовой работы были выполнены следующие задачи:

Были рассмотрены возможности Java при работе со звуками. Java позволяет работать с готовыми аудио файлами, но так же Java умеет работать с MIDI(Musical Instrument Digital Interface) интерфейсом, что позволяет отказаться от аудио файлов. При помощи MIDI интерфейса можно проигрывать 144 ноты. Так же есть большой выбор из 128 инструментов.

Приложение пианино разработано. Добавлен переключатель октавы и инструмента и так же при помощи клавиатуры можно играть в приложении.

Серверная и клиентская часть разработана. Сетевое приложение “Пианино”, позволяет играть вдвоём с разных компьютеров.

Поставленная Цель достигнута. Сетевое приложение “Пианино”, позволяет играть вдвоём с разных компьютеров. И все задачи выполнены.

# Список литературы

1. Кэти Сиерра и Берт Бейтс - Изучаем Java. 2-е издание, издательство “Эксмо”, город Москва, 2017г. 720 стр.
2. Васильев А. Н. - Программирование на Java для начинающих. Издательство “Эксмо”, город Москва, 2017г. 704 стр.
3. Брюс Эккель - Философия Java. 4-е издание, издательство “Питер” город Санкт-Петербург, 2017г. 1168 стр.
4. Дэвид Хеффельфингер - Разработка приложений Java EE 7 в NetBens 8. издательство “ДМК-Пресс”, город Москва, 2016г. 348 стр.
5. Монахов В. В. - Язык программирования Java и среда NetBeans. 3-е издание, издательство “БХВ-Петербург”, город Санкт-Петербург, 2012г. 704 стр.
6. Арун Гупта - Java EE 7. Основы. Издательство “Вильямс”, город Москва, 2014г. 336 стр.
7. Крэйг Хант - TCP/IP Сетевое администрирование. 3-е издание, издательство “Символ-Плюс”, город Москва, 2008г. 775 стр.
8. Дэвид Гири, Кей С. Хорстманн - JavaServer Faces. Издательство “Вильямс”, город Москва, 2011г. 544 стр.
9. Машнин Т. С. - Современные Java-технологии на практике. Издательство “БХВ-Петербург”, город Санкт-Петербург, 2010г. 560стр.