VoCe: Voice Conference

Content

1. How to run

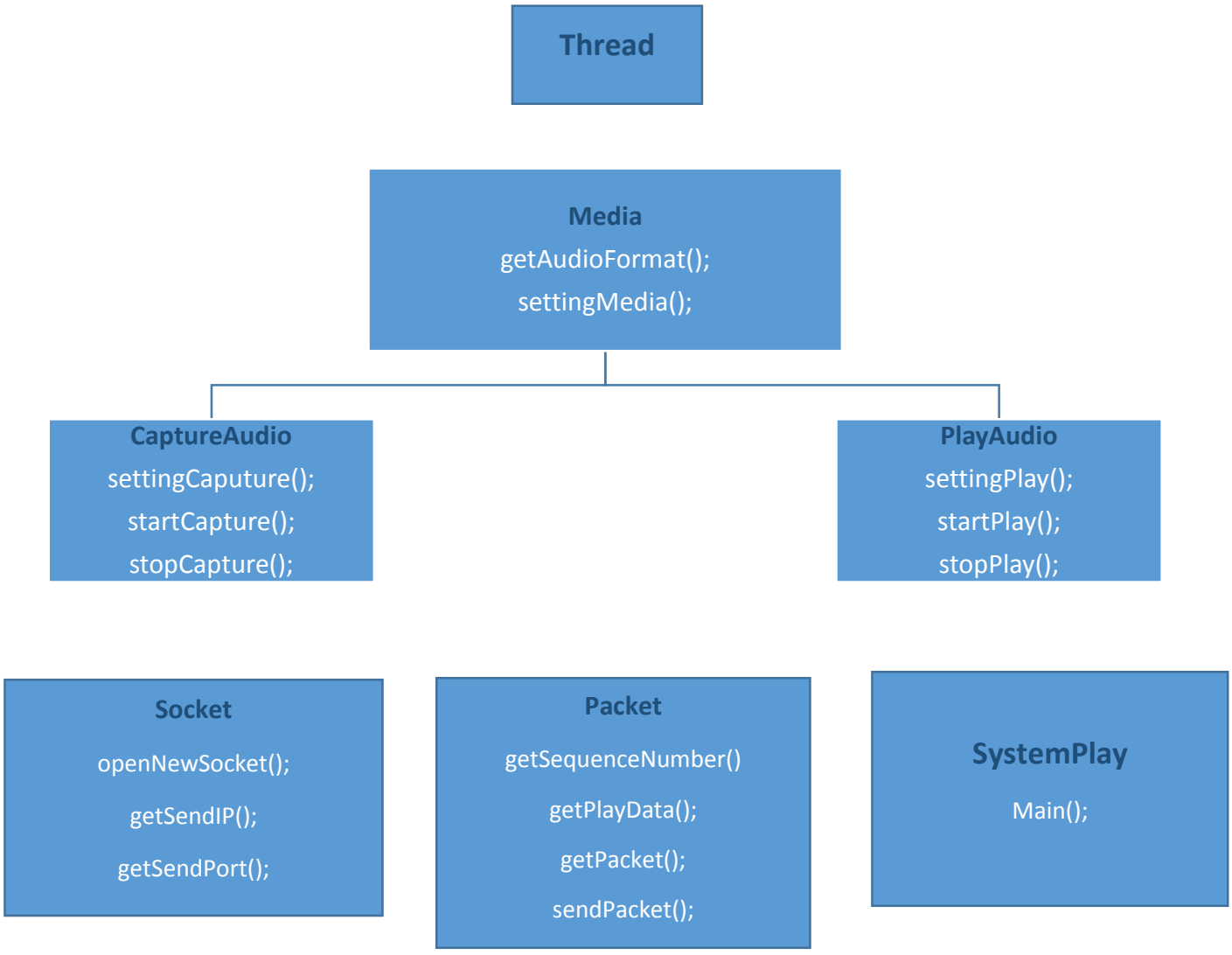
Compile the classes

- Media.java
- CaptureAudio.java
- PlayAudio.java
- Socket.java
- Packet.java
- SystemPlay.java

Then run the System play giving arguments, host receiving port, sending IP and sending port

Eg: - java SystemPlay 2000 192.168.4.23 3000

2. Classes



2.1. Media

This class extends Thread class which need for child classes to make threads for Play audio and send audio data as packets.

2.1.1.Media()

This is the constructer of the class which initiate the mixers for audio capturing and play

2.1.2.getAudioFormat()

method used to setup the audio format for capturing and playing

2.1.3.settingMedia()

method used to set mixers for audio capturing and playing. This method is called by the Media() constructer

2.2. CaptureAudio

2.2.1.CaptureAudio(String ip, String portHost)

This is the constructor of CaptureAudio. This will make Socket Object to send captured audio data by a Packet.

2.2.2.*settingCapture()*

method used to set up capturing settings related to TargetDataLine and Mixer.

2.2.3.*startCapture()*

method used to start Capturing audio

2.2.4.stopCapture()

method used to stop Capturing audio

2.2.5.*run()*

method is override the method in Thread class. This helps to capture audio in a separate thread. Real time audio capturing is done by this method.

2.3. PlayAudio

2.3.1.*PlayAudio(String port)*

Constructer of this class. String port used to create packet accepting socket through given port.

2.3.2.startPlay()

method used to start play received audio

2.3.3.*stopPlay()*

method used to stop play received audio

2.3.4.*settingPlay()*

method used to set up capturing settings related to SourceDataLine and Mixer

2.3.5.*run()*

method is override the method in Thread class. This helps to receive Packet and play audio in a separate thread. Real time audio playing is done by this method.

2.4. Socket

2.4.1.Socket(String hostPort)

Constructor use to make a new DatagramSocket with given port. This Object with constructor used in constructor of PlayAudio class to create a receiving Socket.

2.4.2.Socket(String hostIP,String hostPort)

Constructor use to save sending IP and sending port. It will generate a Socket object with IP and Port. This constructor is used in constructor of CaptureAudio class.

2.4.3.*openNewSocket()*

method used to create a new DatagramSocket to send DatagramPackets. It does not contain port, because it's a sending socket, which can have any port number assigned by system

return DatagramSocket

2.4.4.*getSendIP()*

method used to get the sending IP of the data Packet.

return InetAdress

2.4.5.*getSendPort()*

method used to get the sending port of the data Packet

return int

2.5. Packet

2.5.1. *Packet(Socket sock,byte[] data)*

This get audio byte array , sending IP and sending port from the Socket object to create a DatagramPacket.

Serialization happens here which will add a sequence to each Packet object create by this constructor. Both audio byte array and sequence number in int put into a byte array and make a new DatagramPacket object in here.

This construct use in CaptureAudio class to make Packet objects to send

2.5.2. *Packet(DatagramPacket packet)*

This get receiving DatagramPacket and make a Packet objects by deserializing the byte array on it to sequence number and the audio byte array.

This constructor used in PlayAudio class to separate audio byte array from the DatagramPacket object

2.5.3. *getSequenceNumber()*

method used to get the sequence number of the Packet object

return int

2.5.4. *getPlayData()*

method used to get the audio byte array from the Packet object

return byte[]

2.5.5. *getPacket()*

method used to get DatagramPacket from the Packet object

return DatagramPacket

2.5.6. *sendPacket(DatagramSocket socket)*

method used to send the DatagramPacket in the Packet object through the DatagramSocket passed by the argument

2.6. SystemPlay

2.6.1. *main()*

main method to run the program is implemented in this class

2.7. **HandlePacket**

This class is not implemented yet. But this will handle the packet order in the receiving end. This class have a buffer to store incoming packets and arrange them in order. After implementation of this class, this will take the packet receiving process from the PlayAudio class.

PlayAudio class get this buffer to play the audio.