> AudioTrack是MediaStreamTrack的一个子类，负责音频的调节。 VideoTrack和Audio几乎完全一样，只是多了一个free的方法，然后添加的Renderer的类型不一样。

```
//一个list里面存的是音频的渲染器
private final LinkedList<AudioRenderer> renderers = new LinkedList<AudioRenderer>();
```

```
//构造方法
public AudioTrack(long nativeTrack) {
   super(nativeTrack);
 }
```

```
//添加音频渲染器，与去除音频渲染器
public void addRenderer(AudioRenderer renderer){
    renderers.add(renderer);
    nativeAddRenderer(nativeTrack, renderer.nativeAudioRenderer);
}

public void removeRenderer(AudioRenderer renderer){
    if(!renderers.remove(renderer)){
        return;
    }
    nativeRemoveRenderer(nativeTrack,renderer.nativeAudioRenderer);
    renderer.dispose();
}

private static native void nativeAddRenderer(long nativeTrack, long nativeRenderer);
private static native void nativeRemoveRenderer(long nativeTrack, long nativeRenderer);
```

```
//释放掉AudioTrack
public void dispose(){
    while (!renderers.isEmpty()) {
        removeRenderer(renderers.getFirst());
    }
    super.dispose();
}
```

```
//方法含义同AudioTrack
private final LinkedList<VideoRenderer> renderers = new LinkedList<VideoRenderer>();

public VideoTrack(long nativeTrack) {
  super(nativeTrack);
}

public void addRenderer(VideoRenderer renderer) {
  renderers.add(renderer);
  nativeAddRenderer(nativeTrack, renderer.nativeVideoRenderer);
}

public void removeRenderer(VideoRenderer renderer) {
  if (!renderers.remove(renderer)) {
    return;
  }
  nativeRemoveRenderer(nativeTrack, renderer.nativeVideoRenderer);
  renderer.dispose();
}

public void dispose() {
  while (!renderers.isEmpty()) {
    removeRenderer(renderers.getFirst());
  }
  super.dispose();
}

private static native void free(long nativeTrack);

private static native void nativeAddRenderer(long nativeTrack, long nativeRenderer);

private static native void nativeRemoveRenderer(long nativeTrack, long nativeRenderer);
```