m_memBuffer: std::vector<char> m_putP: size_t m_subject: std::shared_ptr<AudioFile> addChunk(Chunk_iterator&): void begin(): Chunk_const_iterator {query} begin(): Chunk_iterator end(): Chunk_const_iterator {query} end(): Chunk_iterator Model::AbsAudioFile getSubject(): AbsAudioFile& defaultChunkSize: size_t = 10 {readOnly} getSubject(): AbsAudioFile& {query} headerSize: size_t = sizeof(size_t) {readOnly} MemAudioFile(std::string&) m chunkSize: size t ~MemAudioFile() m_fname: std::string readChunk(char*): void m_numberChunks: size_t seekChunkg(size_t): void seekChunkp(size_t): void + AbsAudioFile(std::string&, size_t) tellChunkg(): size_t ~AbsAudioFile() tellChunkp(): size_t addChunk(Chunk_iterator&): void writeChunk(char*): void begin(): Chunk_const_iterator {query} begin(): Chunk_iterator end(): Chunk_const_iterator {query} end(): Chunk_iterator getChunkSize(): size_t {query} getName(): std::string& {query} getNumberChunks(): size_t {query} Model::AudioFile # readChunk(char*): void seekChunkg(size_t): void m_stream: std::fstream # seekChunkp(size_t): void addChunk(Chunk_iterator&): void # setChunkSize(size_t): void setNumberChunks(size_t): void AudioFile(std::string&) tellChunkg(): size_t ~AudioFile() begin(): Chunk_const_iterator {query} tellChunkp(): size_t writeChunk(char*): void begin(): Chunk_iterator end(): Chunk_const_iterator {query} end(): Chunk_iterator # readChunk(char*): void # seekChunkg(size_t): void # seekChunkp(size_t): void # tellChunkg(): size_t tellChunkp(): size_t writeChunk(char*): void

Model::MemAudioFile