Package1

Package in package 'Analyse'

<div align="right">

Package1
Version 1.0 Phase 1.0 Proposed
rozho created on 2019-03-19. Last modified 2019-03-19

</div>

Package1 diagram

Class diagram in package 'Package1'

<div align="right">

Package1
Version 1.0
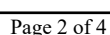rozho created on 2019-03-19. Last modified 2019-03-19

</div>

Package in package 'Analyse'

Chunk_iterator_base *iterator*

- m_audioFile: AbsAudioFile&
- m_chunkIdx: size_t
- m_currentChunk: Chunk

+ Chunk_iterator_base(class AbsAudioFile&, size_t)
+ Chunk_iterator_base(Chunk_iterator_base&)
# dec(): void
# dec(size_t): void
# equal(Chunk_iterator_base&): bool {query}
# get(): Chunk& {query}
# get(size_t): Chunk& {query}
# getp(): Chunk* {query}
# inc(): void
# inc(size_t): void
+ operator!=(Chunk_iterator_base&): bool {query}
+ operator==(Chunk_iterator_base&): bool {query}
«friend»
+ operator-(Chunk_iterator_base&, size_t): Chunk_iterator_base
+ operator-(Chunk_iterator_base, Chunk_iterator_base): difference_type
+ operator!=(Chunk_iterator_base&, Chunk_iterator_base&): bool
+ operator+(Chunk_iterator_base&, size_t): Chunk_iterator_base
+ operator+(size_t, iterator&): iterator
+ operator<(Chunk_iterator_base&, Chunk_iterator_base&): bool
+ operator<=(Chunk_iterator_base&, Chunk_iterator_base&): bool
+ operator==(Chunk_iterator_base&, Chunk_iterator_base&): bool
+ operator>(Chunk_iterator_base&, Chunk_iterator_base&): bool
+ operator>=(Chunk_iterator_base&, Chunk_iterator_base&): bool
+ swap(Chunk_iterator_base&, Chunk_iterator_base&): void

Chunk_const_iterator

+ Chunk_const_iterator(class AbsAudioFile&, size_t)
+ Chunk_const_iterator(Chunk_const_iterator&)
+ operator--(): Chunk_const_iterator&
+ operator--(int): Chunk_const_iterator
+ operator-=(size_t): Chunk_const_iterator&
+ operator->(): Chunk* {query}
+ operator!=(Chunk_const_iterator&): bool {query}
+ operator*(): Chunk& {query}
+ operator[](size_t): Chunk& {query}
+ operator++(): Chunk_const_iterator&
+ operator++(int): Chunk_const_iterator
+ operator+=(size_t): Chunk_const_iterator&
+ operator==(Chunk_const_iterator&): bool {query}

Chunk

- m_buffer: std::vector<char>

+ Chunk(size_t)
+ Chunk(size_t, char*)
+ get(): char {query}
+ get(): char*
+ operator!=(Chunk&): bool {query}
+ operator==(Chunk&): bool {query}
+ size(): size_t

-m_currentChunk

Chunk_iterator

+ Chunk_iterator(class AbsAudioFile&, size_t)
+ Chunk_iterator(Chunk_iterator&)
+ operator--(): Chunk_iterator&
+ operator--(int): Chunk_iterator
+ operator-=(size_t): Chunk_iterator&
+ operator->(): Chunk* {query}
+ operator!=(Chunk_iterator&): bool {query}
+ operator*(): Chunk& {query}
+ operator[](size_t): Chunk& {query}
+ operator++(): Chunk_iterator&
+ operator++(int): Chunk_iterator
+ operator+=(size_t): Chunk_iterator&
+ operator==(Chunk_iterator&): bool {query}

AbsTransform

- m_empty_transforms: TransformContainer

+ AbsTransform()
+ ~AbsTransform()
+ addChild(AbsTransform&): void
+ begin(): TransformIterator_const {query}
+ begin(): TransformIterator
+ clone(): AbsTransform* {query}
+ end(): TransformIterator_const {query}
+ end(): TransformIterator
+ removeChild(TransformIterator_const): void
+ transform(class Chunk_iterator&, class AbsAudioFile&): void {query}

-m_empty_transforms

std::vector<TransformPtr>
«alias»
TransformContainer

-m_transforms

CompositeTransform

- m_transforms: TransformContainer

+ addChild(AbsTransform&): void
+ begin(): TransformIterator_const {query}
+ begin(): TransformIterator
+ clone(): CompositeTransform* {query}
+ CompositeTransform()
+ CompositeTransform(CompositeTransform&)
+ ~CompositeTransform()
+ end(): TransformIterator_const {query}
+ end(): TransformIterator
+ removeChild(TransformIterator_const): void
+ transform(class Chunk_iterator&, class AbsAudioFile&): void {query}

AudioFile

- m_stream: std::fstream

+ addChunk(Chunk_iterator&): void
+ AudioFile(std::string&)
+ ~AudioFile()
+ begin(): Chunk_const_iterator {query}
+ 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

AbsAudioFile

defaultChunkSize: size_t = 10 {readOnly}
headerSize: size_t = sizeof(size_t) {readOnly}
- m_chunkSize: size_t
- m_fname: std::string
- m_numberChunks: size_t

+ AbsAudioFile(std::string&, size_t)
+ ~AbsAudioFile()
+ addChunk(Chunk_iterator&): 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}
# readChunk(char*): void
# seekChunkg(size_t): void
# seekChunkp(size_t): void
# setChunkSize(size_t): void
# setNumberChunks(size_t): void
# tellChunkg(): size_t
# tellChunkp(): size_t
# writeChunk(char*): void

-m_audioFile

InvertTransform

+ clone(): InvertTransform* {query}
+ InvertTransform()
+ transform(Chunk_iterator&, AbsAudioFile&): void {query}

RepeatTransform

- m_nRepetitions: size_t

+ clone(): RepeatTransform* {query}
+ getNRepetitions(): size_t
+ RepeatTransform(size_t)
+ transform(class Chunk_iterator&, class AbsAudioFile&): void {query}

MemAudioFile

- m_getP: size_t
- 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
+ getSubject(): AbsAudioFile&
+ getSubject(): AbsAudioFile& {query}
+ MemAudioFile(std::string&)
+ ~MemAudioFile()
# readChunk(char*): void
# seekChunkg(size_t): void
# seekChunkp(size_t): void
# tellChunkg(): size_t
# tellChunkp(): size_t
# writeChunk(char*): void

TransformIterator

+ operator->(): class AbsTransform*
+ operator*(): class AbsTransform&
+ TransformIterator(TransformContainer::iterator&)

TransformContainer::iterator
«alias»
TransformBaseIterator

TransformIterator_const

+ operator->(): class AbsTransform*
+ operator*(): class AbsTransform&
+ TransformIterator_const(TransformContainer::const_iterator&)

TransformContainer::const_iterator
«alias»
TransformBaseIterator_const

«Enumeration» ResultatTest

echec
succes

Test_TP4

- m_nameAudiof_composite: std::string = "audiof_composi... {readOnly}
- m_nameAudiof_composite_ref: std::string = "audiof_composi... {readOnly}
- m_nameAudiof_inv: std::string = "audiof_inv.bin" {readOnly}
- m_nameAudiof_inv_ref: std::string = "audiof_inv_ref.bin" {readOnly}
- m_nameAudiof_proxy: std::string = "audiof_proxy.bin" {readOnly}
- m_nameAudiof_ref1: st...
- m_nameAudiof_rt: std::...
- m_nameAudiof_rt_ref: s...
- m_nameAudiof1: std::st...
- m_racineComposite: std...

CompareFilesEqual(AbsAudioFile&, AbsAudioFile&): bool
+ executeCompositeTest(): void
+ executeProxyTest(): void
+ Test_TP4()
+ ~Test_TP4()

std::unique_ptr<class AbsTransform>
«alias»
TransformPtr

Figure 1: Package1

Proxy diagram

Class diagram in package 'Package1'

<div align="right">

Proxy
Version 1.0
rozho created on 2019-03-22. Last modified 2019-03-22
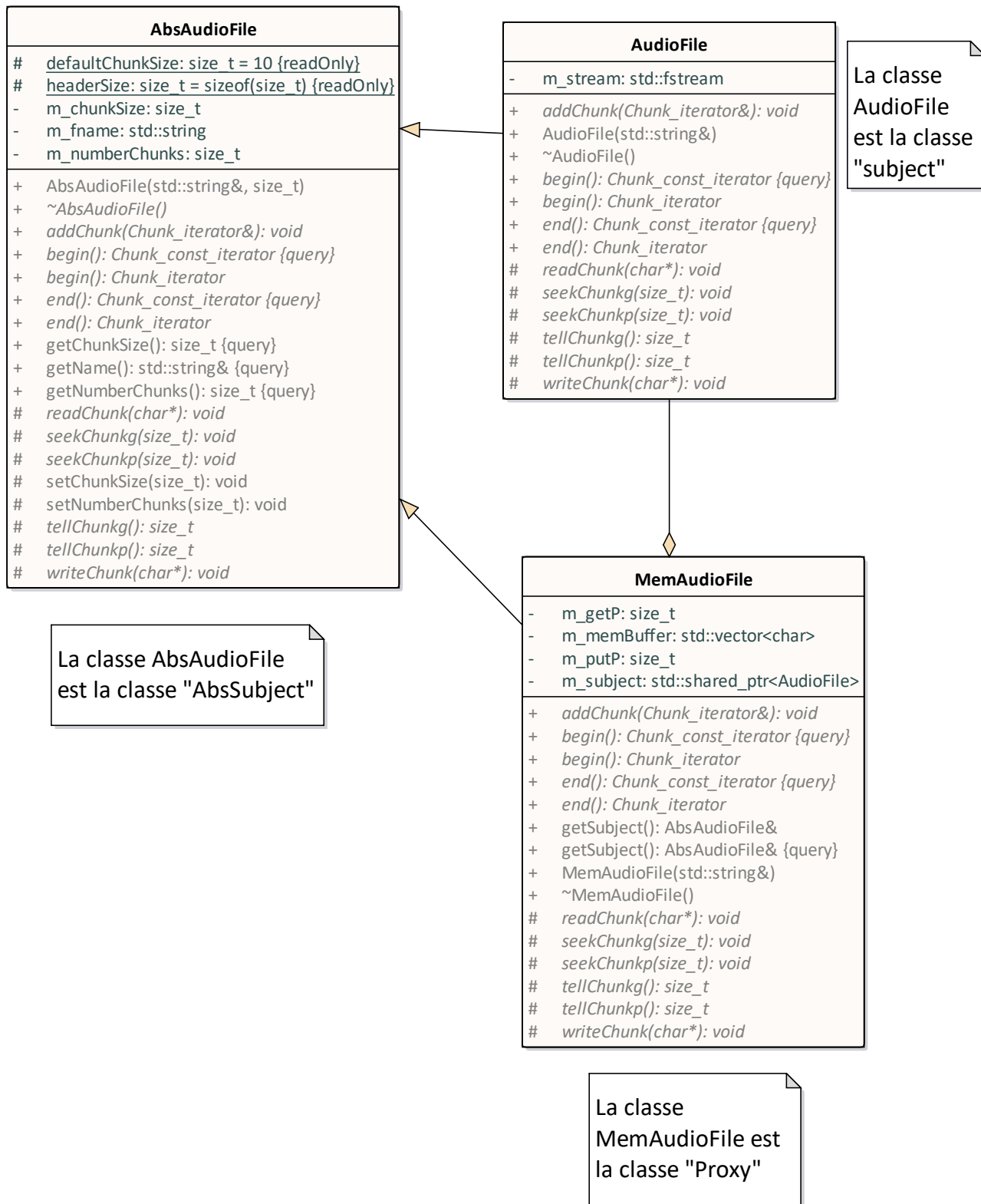
</div>

AbsAudioFile
defaultChunkSize: size_t = 10 {readOnly}
headerSize: size_t = sizeof(size_t) {readOnly}
- m_chunkSize: size_t
- m_fname: std::string
- m_numberChunks: size_t
+ AbsAudioFile(std::string&, size_t)
+ ~AbsAudioFile()
+ addChunk(Chunk_iterator&): 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}
# readChunk(char*): void
# seekChunkg(size_t): void
# seekChunkp(size_t): void
# setChunkSize(size_t): void
# setNumberChunks(size_t): void
# tellChunkg(): size_t
# tellChunkp(): size_t
# writeChunk(char*): void

AudioFile
- m_stream: std::fstream
+ addChunk(Chunk_iterator&): void
+ AudioFile(std::string&)
+ ~AudioFile()
+ begin(): Chunk_const_iterator {query}
+ 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

La classe AudioFile est la classe "subject"

La classe AbsAudioFile est la classe "AbsSubject"

MemAudioFile
- m_getP: size_t
- 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
+ getSubject(): AbsAudioFile&
+ getSubject(): AbsAudioFile& {query}
+ MemAudioFile(std::string&)
+ ~MemAudioFile()
# readChunk(char*): void
# seekChunkg(size_t): void
# seekChunkp(size_t): void
# tellChunkg(): size_t
# tellChunkp(): size_t
# writeChunk(char*): void

La classe MemAudioFile est la classe "Proxy"

Figure 2: Proxy

ResultatTest

Enumeration «Enumeration» in package 'Package1'

ResultatTest
Version 1.0 Phase 1.0 Proposed
rozho created on 2019-03-19. Last modified 2019-03-19