ACO Project Report
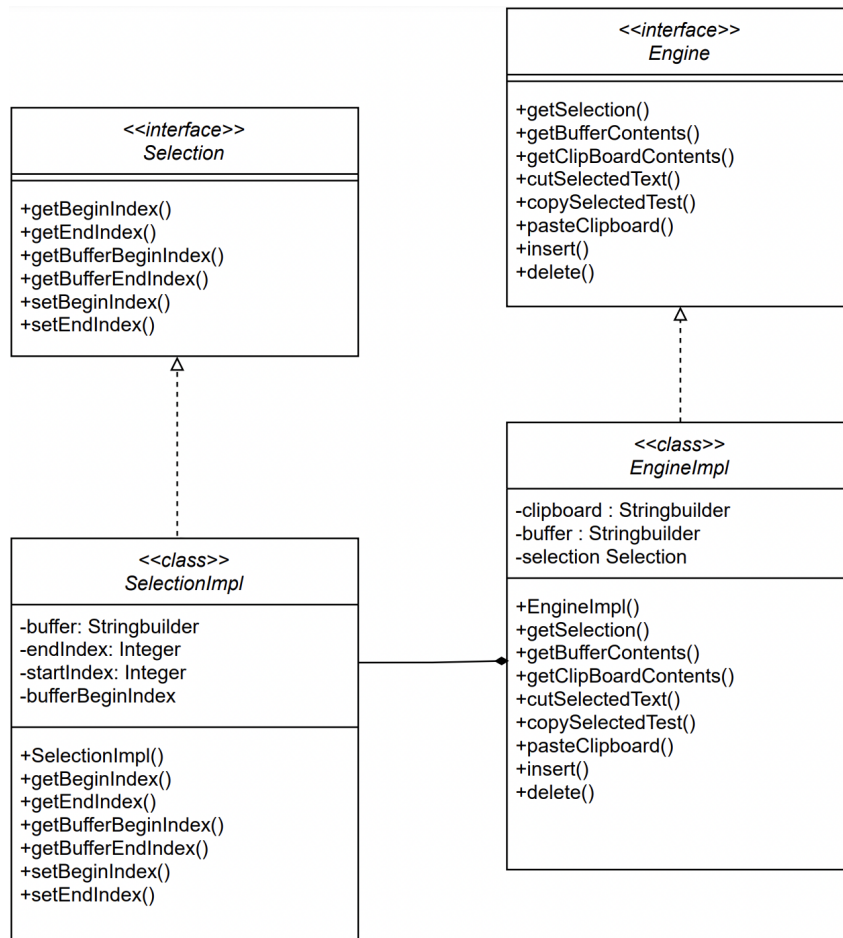
- **Authors**: Zofia Walczewska and Marta Moreno
- **TP group**: EIT
- **TP supervisor**: Adrien Le Roche
- Link to gitlab:
  https://gitlab.istic.univ-rennes1.fr/zwalczewska/texteditor
- Launching of all versions is done by running tests.
- Javadoc available for all versions in the gitlab repository (file: index.html).

VERSION 1

1. Branch for version 1: origin/v1
   https://gitlab.istic.univ-rennes1.fr/zwalczewska/texteditor/-/tree/origin/v1/src/TextEditor
2. Tests for version 1: file EngineTest.java and SelectionTest.java
   https://gitlab.istic.univ-rennes1.fr/zwalczewska/texteditor/-/blob/origin/v1/src/TextEditor/EngineTest.java
   https://gitlab.istic.univ-rennes1.fr/zwalczewska/texteditor/-/blob/origin/v1/src/TextEditor/SelectionTest.java
3. Test results:
   a. All tests are successful for version 1 and 99% test coverage achieved.
4. UML architecture design:

Selection (interface)

<<interface>>
Selection

+getBeginIndex()
+getEndIndex()
+getBufferBeginIndex()
+getBufferEndIndex()
+setBeginIndex()
+setEndIndex()

Engine (interface)

<<interface>>
Engine

+getSelection()
+getBufferContents()
+getClipBoardContents()
+cutSelectedText()
+copySelectedTest()
+pasteClipboard()
+insert()
+delete()

SelectionImpl (class)

<<class>>
SelectionImpl

-buffer: Stringbuilder
-endIndex: Integer
-startIndex: Integer
-bufferBeginIndex

+SelectionImpl()
+getBeginIndex()
+getEndIndex()
+getBufferBeginIndex()
+getBufferEndIndex()
+setBeginIndex()
+setEndIndex()

EngineImpl (class)

<<class>>
EngineImpl

-clipboard : Stringbuilder
-buffer : Stringbuilder
-selection Selection

+EngineImpl()
+getSelection()
+getBufferContents()
+getClipBoardContents()
+cutSelectedText()
+copySelectedTest()
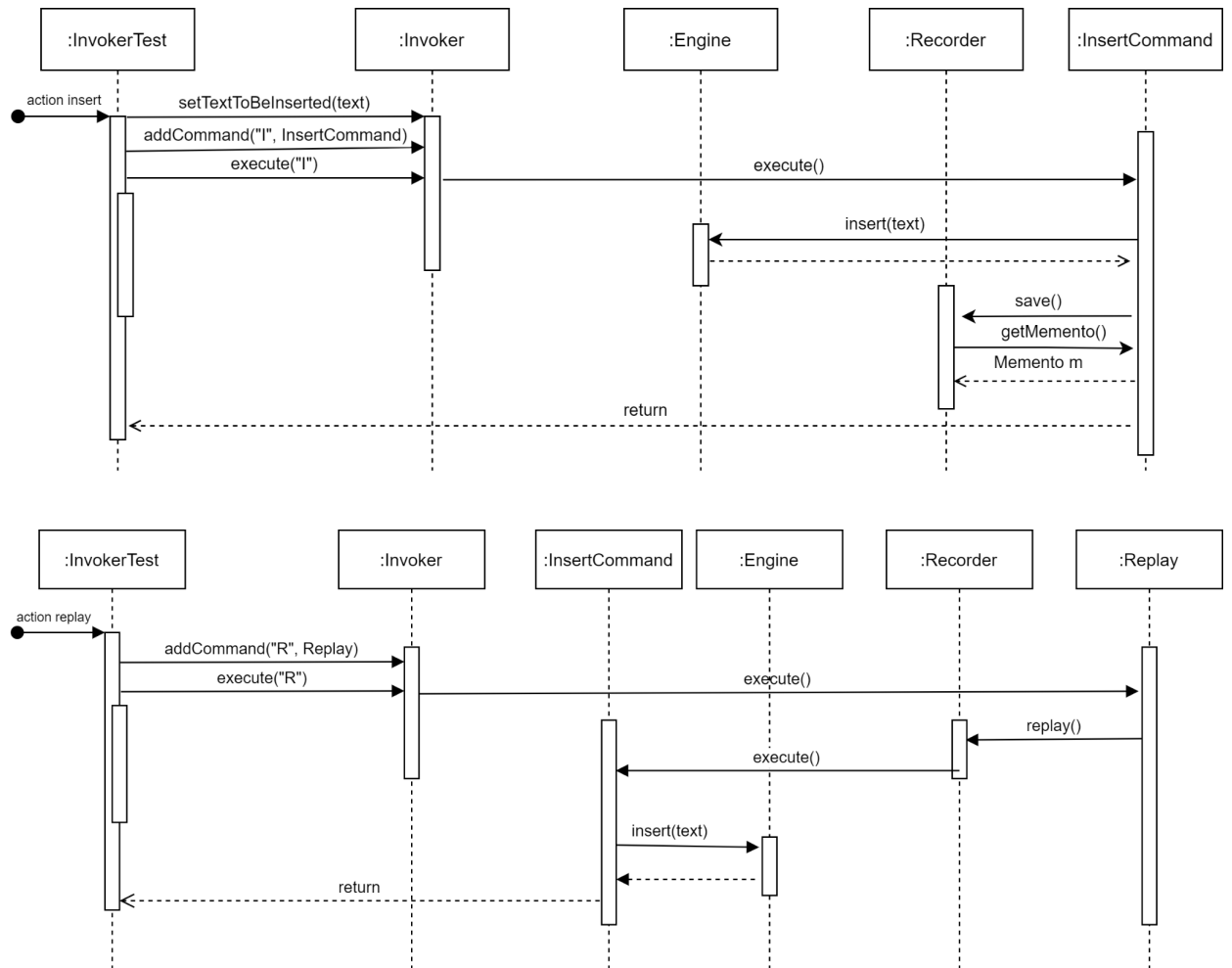+pasteClipboard()
+insert()
+delete()

VERSION 2

1. Branch for version 2: origin/v2
   https://gitlab.istic.univ-rennes1.fr/zwalczewska/texteditor/-/tree/origin/v2/src/TextEditor
2. Tests for version 2: file InvokerTest.java
   https://gitlab.istic.univ-rennes1.fr/zwalczewska/texteditor/-/blob/origin/v2/src/TextEditor/InvokerTest.java
3. Test results:
   a. All tests are successful for version 2 and 99% test coverage achieved.
4. UML architecture diagram at the end of the document.
5. UML sequence diagrams- executing a command and replaying it:

Sequence Diagram 1 (action insert)

Participants: `:InvokerTest`, `:Invoker`, `:Engine`, `:Recorder`, `:InsertCommand`

- action insert → `:InvokerTest`
- `:InvokerTest` → `:Invoker` : setTextToBeInserted(text)
- `:InvokerTest` → `:Invoker` : addCommand("I", InsertCommand)
- `:InvokerTest` → `:Invoker` : execute("I")
- `:Invoker` → `:InsertCommand` : execute()
- `:InsertCommand` → `:Engine` : insert(text)
- `:Engine` ⤑ `:InsertCommand` (return)
- `:InsertCommand` → `:Recorder` : save()
- `:Recorder` → `:InsertCommand` : getMemento()
- `:InsertCommand` ⤑ `:Recorder` : Memento m
- `:InsertCommand` ⤑ `:InvokerTest` : return

Sequence Diagram 2 (action replay)

Participants: `:InvokerTest`, `:Invoker`, `:InsertCommand`, `:Engine`, `:Recorder`, `:Replay`

- action replay → `:InvokerTest`
- `:InvokerTest` → `:Invoker` : addCommand("R", Replay)
- `:InvokerTest` → `:Invoker` : execute("R")
- `:Invoker` → `:Replay` : execute()
- `:Replay` → `:Recorder` : replay()
- `:Recorder` → `:InsertCommand` : execute()
- `:InsertCommand` → `:Engine` : insert(text)
- `:Engine` ⤑ `:InsertCommand`
- `:InsertCommand` ⤑ `:InvokerTest` : return

VERSION 3
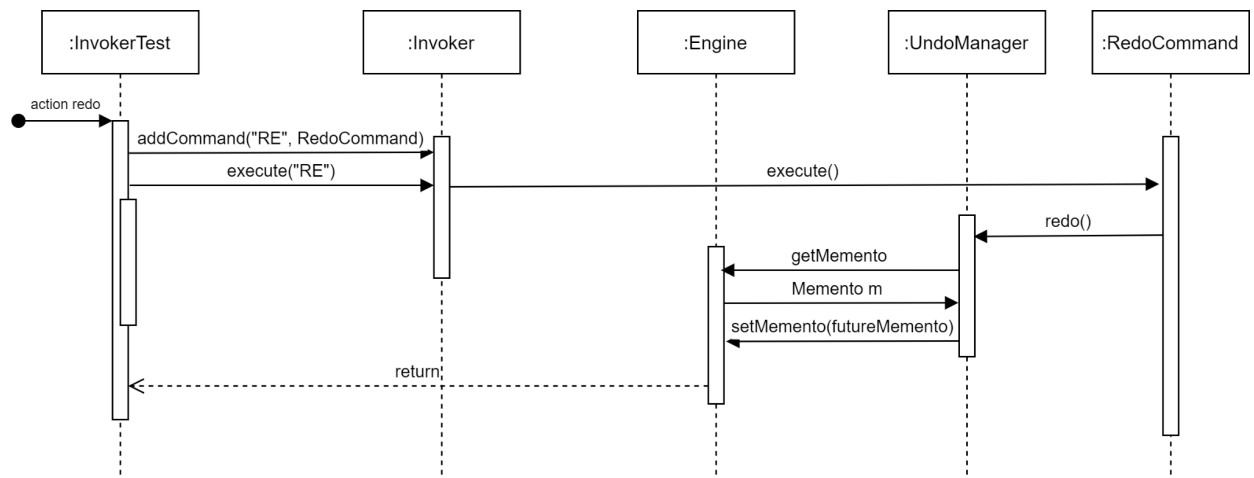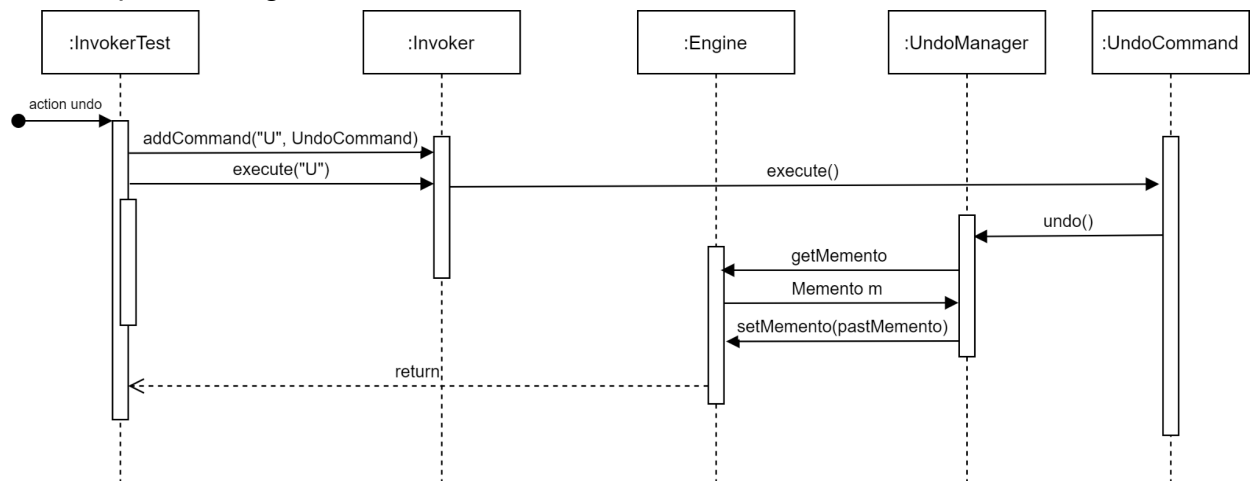
1. Branch for version 3: origin/v3
   https://gitlab.istic.univ-rennes1.fr/zwalczewska/texteditor/-/tree/origin/v3/src/TextEditor
2. Tests for version 3: files UndoManagerTest.java and InvokerTest.java
   https://gitlab.istic.univ-rennes1.fr/zwalczewska/texteditor/-/blob/origin/v3/src/TextEditor/UndoManagerTest.java
   https://gitlab.istic.univ-rennes1.fr/zwalczewska/texteditor/-/blob/origin/v3/src/TextEditor/InvokerTest.java
3. Personal decisions for version 3:

a. Both Redo and Undo are commands. The UndoManager class is the caretaker that keeps all the states of the Engine in cases where the buffer gets modified. EngineImpl acts as the originator and EditorMemento is the concrete memento class that keeps record of the Editor's state.
b. We decided to delete set methods from all concrete mementos to make them unmodifiable. The properties of concrete mementos are set only once during creation.
4. Test results:
   a. All tests are successful for version 3 and 99% test coverage achieved.
5. UML architecture diagram at the end of the document.
6. UML sequence diagram- undo and redo:

7. UML architecture design (versions 2 and 3):

UML Class Diagram — Memento and Command patterns

CARETAKER — Recorder
- -memento: Memento
- -command : Command
- +save():
- +execute():

<<interface>> Command
- +execute():

<<interface>> Originator
- +getMemento():memento
- +setMemento(memento):

<<interface>> Recordable

<<interface>> Memento

CONCRETE MEMENTO — SelectionMemento
- -beginIndex: Integer
- -endIndex: Integer
- +getBeginIndex(): Integer
- +getEndIndex(): Integer
- +setBeginIndex(index):
- +setEndIndex(index):

INVOKER
- -engine: Engine
- -textToBeInserted: String
- -mapOfCommands: Map<String,Command>
- -beginIndex: Integer
- -endIndex: Integer
- +getBeginIndex(): Integer
- +getEndIndex(): Integer
- +setBeginIndex(index):
- +setEndIndex(index):
- +getTextToBeInserted(): String
- +setTextToBeInserted(String):
- +execute(String):

COMMANDS -ORIGINATORS

UndoCommand
- -engine: Engine
- +execute()
- +setMemento(memento):
- +getMemento():memento

RedoCommand
- -engine: Engine
- +execute()
- +setMemento(memento):
- +getMemento():memento

Replay
- -engine: Engine
- +execute()
- +setMemento(memento):
- +getMemento():memento

CopyCommand
- -engine: Engine
- +execute()
- +setMemento(memento):
- +getMemento():memento

PasteCommand
- -engine: Engine
- +execute()
- +setMemento(memento):
- +getMemento():memento

InsertCommand
- -engine: Engine
- -text: String
- +execute()
- +setMemento(memento):
- +getMemento():memento

SelectionCommand
- -engine: Engine
- -beginIndex: Integer
- -endIndex: Integer
- +execute()
- +setMemento(memento):
- +getMemento():memento

CutCommand
- -engine: Engine
- +execute()
- +setMemento(memento):
- +getMemento():memento

CONCRETE MEMENTO — InsertMemento
- -text: String
- +setTextToBeInserted():
- +getTextTobeInserted(): String

ORIGINATOR — <<class>> EngineImpl
- -clipboard : Stringbuilder
- -buffer : Stringbuilder
- -selection Selection
- +EngineImpl()
- +getSelection()
- +getBufferContents()
- +getClipBoardContents()
- +cutSelectedText()
- +copySelectedTest()
- +pasteClipboard()
- +insert()
- +delete()
- +getMemento() : Memento
- +setMemento(Memento m):

<<class>> SelectionImpl
- -buffer: Stringbuilder
- -endIndex: Integer
- -startIndex: Integer
- -bufferBeginIndex
- +SelectionImpl()
- +getBeginIndex()
- +getEndIndex()
- +getBufferBeginIndex()
- +getBufferEndIndex()
- +setBeginIndex()
- +setEndIndex()

CONCRETE MEMENTO — <<class>> EditorMemento
- -bufferContent : String
- -endIndex : Integer
- -beginIndex : integer
- -Memento(buffer)
- -getState(): buffer, selection

CLIENT

<<interface>> Engine
- +getSelection()
- +getBufferContents()
- +getClipBoardContents()
- +cutSelectedText()
- +copySelectedTest()
- +pasteClipboard()
- +insert()
- +delete()
- +getMemento() : Memento
- +setMemento(Memento m):

<<interface>> Selection
- +getBeginIndex()
- +getEndIndex()
- +getBufferBeginIndex()
- +getBufferEndIndex()
- +setBeginIndex()
- +setEndIndex()

CARETAKER — <<class>> UndoManager
- -List<EditorMemento> : pastStates
- -List<EditorMemento>: futureStates
- -Engine: engine
- +void: store()
- +void: undo()
- +void: redo()