Citezen

Louis Deloffre Sabrina Kernouf Romain Morel

Groupe **SLR**

Sommaire

I Le Projet Citezen

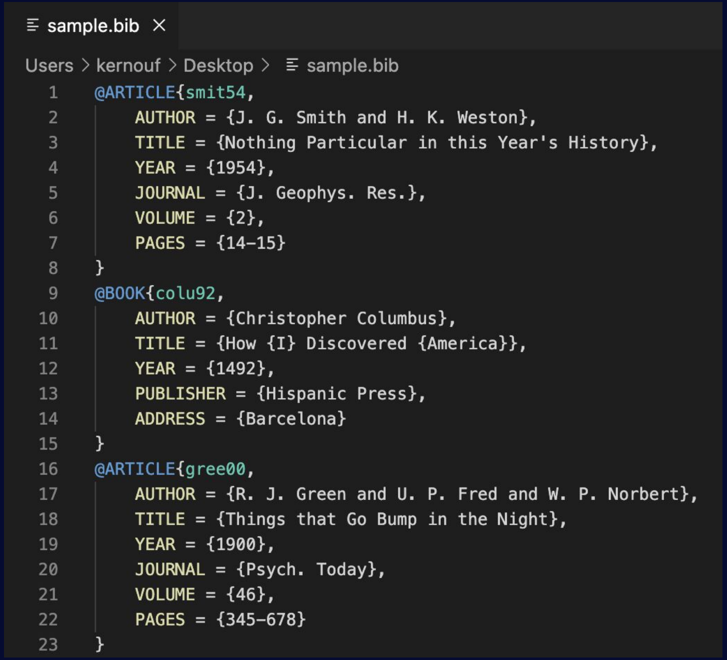
Vue d’ensemble du projet

Qu’est ce que Citezen ?

- Bib

- Outils

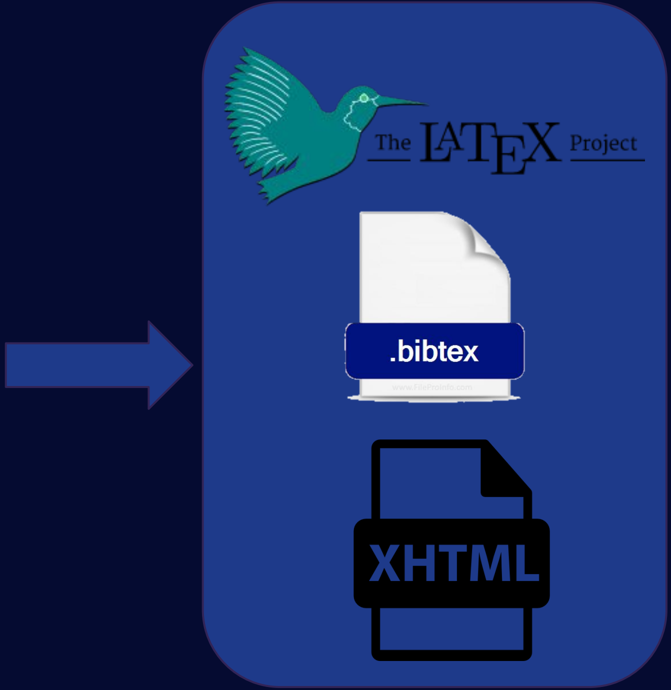
Utilité et fonctionnement du projet



sample.bib

Users > kernouf > Desktop > sample.bib

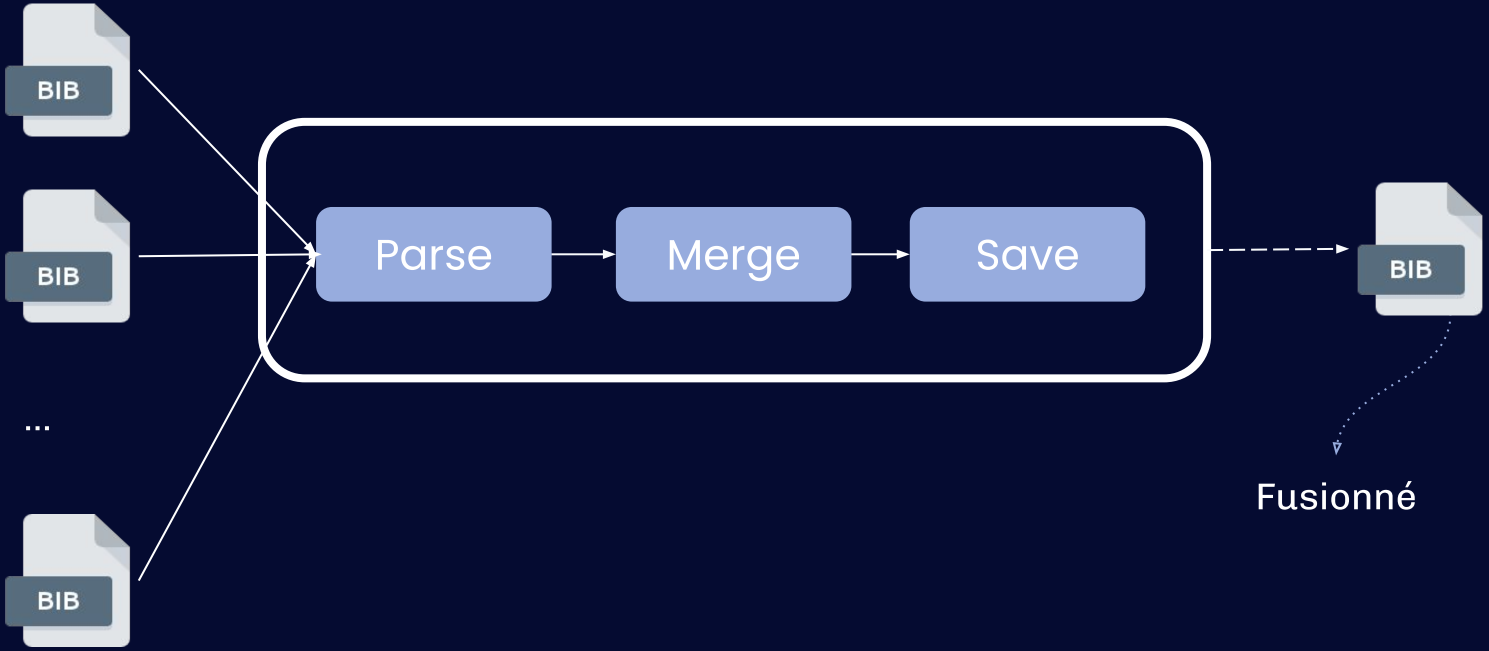
```
@ARTICLE{smit54,
    AUTHOR = {J. G. Smith and H. K. Weston},
    TITLE = {Nothing Particular in this Year's History},
    YEAR = {1954},
    JOURNAL = {J. Geophys. Res.},
    VOLUME = {2},
    PAGES = {14–15}
}
@BOOK{colu92,
    AUTHOR = {Christopher Columbus},
    TITLE = {How {I} Discovered {America}},
    YEAR = {1492},
    PUBLISHER = {Hispanic Press},
    ADDRESS = {Barcelona}
}
@ARTICLE{gree00,
    AUTHOR = {R. J. Green and U. P. Fred and W. P. Norbert},
    TITLE = {Things that Go Bump in the Night},
    YEAR = {1900},
    JOURNAL = {Psych. Today},
    VOLUME = {46},
    PAGES = {345–678}
}
```



II Nos actions

Implémentation de nos outils

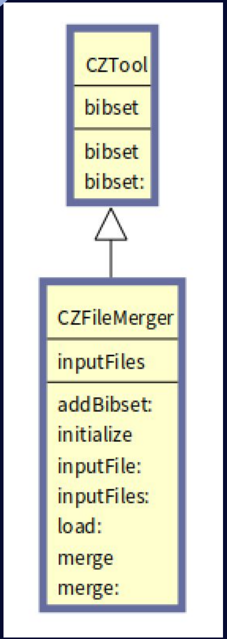
Merger

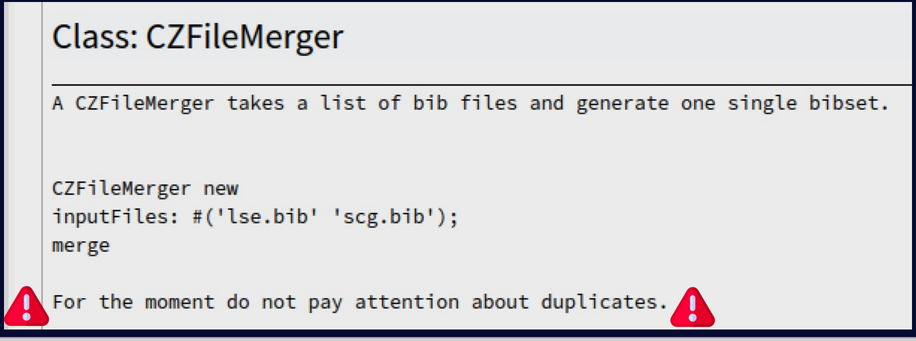


Nos objectifs

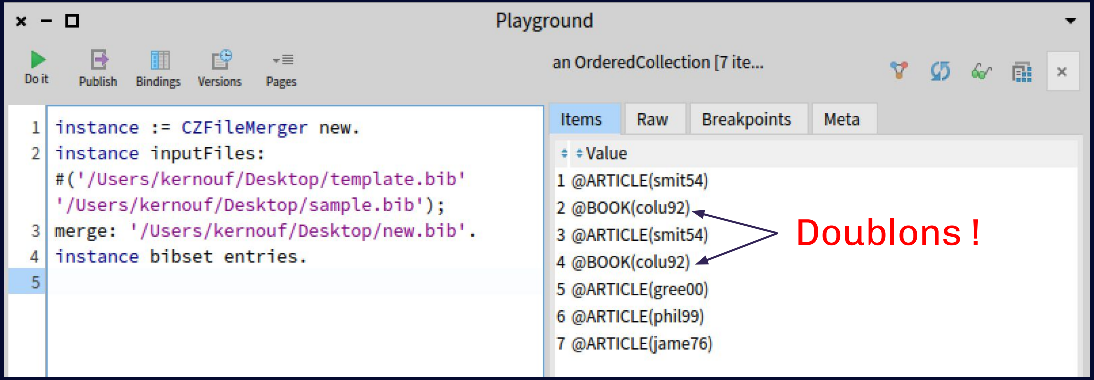
- Fusionner un ou plusieurs fichiers .
- Une fois fusionnés, enregistrer le fichier généré.
- Tester et commenter le code écrit.
- Phase de refactoring.

Reverse Engineering




Class: CZFileMerger

```
A CZFileMerger takes a list of bib files and generate one single bibset.


CZFileMerger new
inputFiles: #('lse.bib' 'scg.bib');
merge

For the moment do not pay attention about duplicates.
```

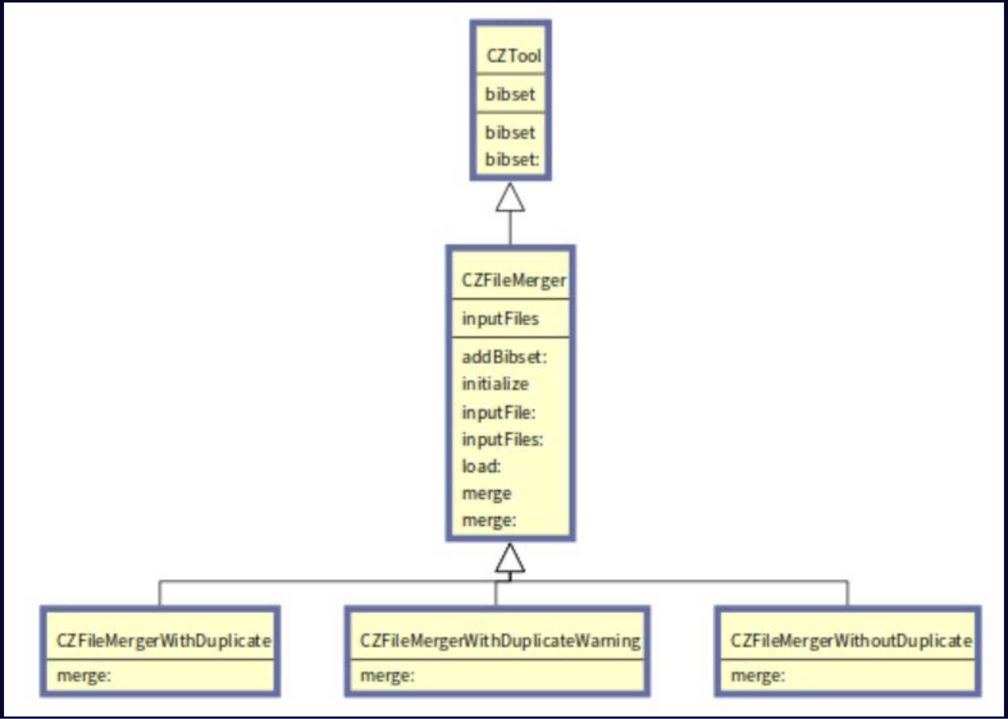


```
instance := CZFileMerger new.
instance inputFiles:
#('/Users/kernouf/Desktop/template.bib'
'/Users/kernouf/Desktop/sample.bib');
merge: '/Users/kernouf/Desktop/new.bib'.
instance bibset entries.
```

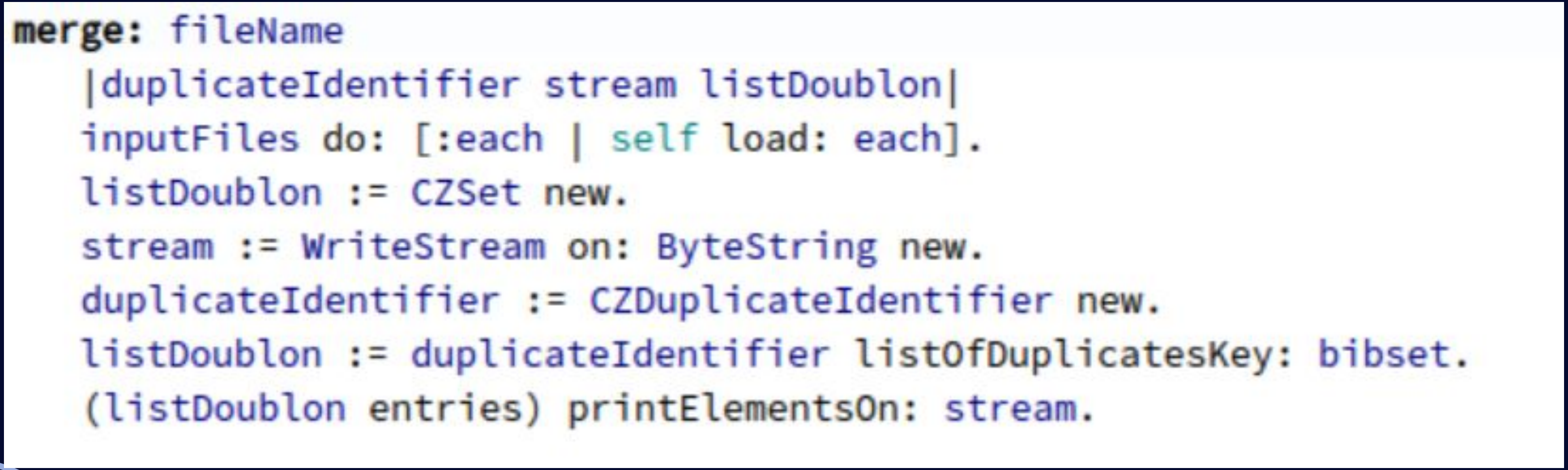
```
1 @ARTICLE(smit54)
2 @BOOK(colu92)
3 @ARTICLE(smit54)
4 @BOOK(colu92)
5 @ARTICLE(gree00)
6 @ARTICLE(phil99)
7 @ARTICLE(jame76)
```

Doublons !

Nos changement



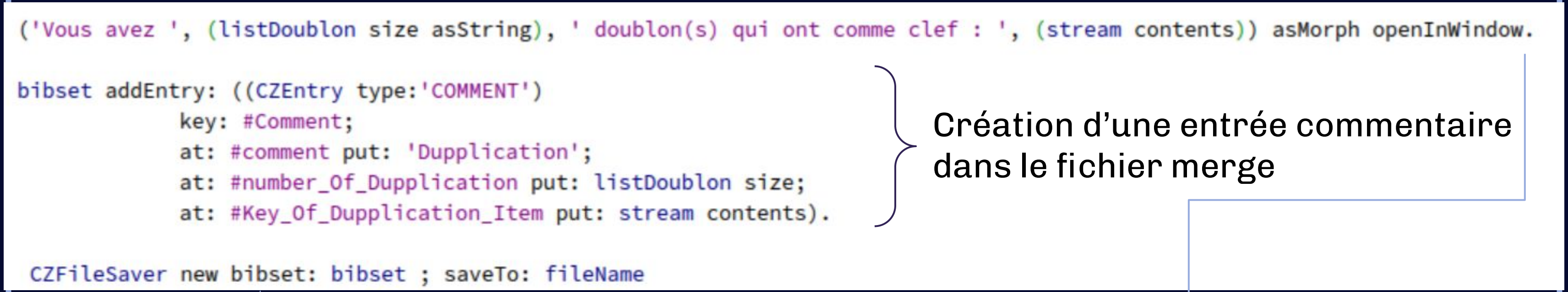
Exemple : MergerWithDuplicateWarning



```
merge: fileName
   |duplicateIdentifier stream listDoublon|
   inputFiles do: [:each | self load: each].
   listDoublon := CZSet new.
   stream := WriteStream on: ByteString new.
   duplicateIdentifier := CZDuplicateIdentifier new.
   listDoublon := duplicateIdentifier listOfDuplicatesKey: bibset.
   (listDoublon entries) printElementsOn: stream.
```

Première partie de la méthode, permet de faire le trie entre liste des doublon et sans doublon

Exemple : MergerWithDuplicateWarning



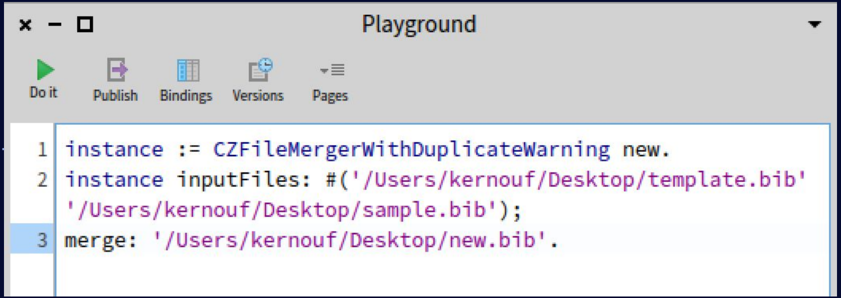
```
('Vous avez ', (listDoublon size asString), ' doublon(s) qui ont comme clef : ', (stream contents)) asMorph openInWindow.

bibset addEntry: ((CZEntry type:'COMMENT')
            key: #Comment;
            at: #comment put: 'Dupplication';
            at: #number_Of_Dupplication put: listDoublon size;
            at: #Key_Of_Dupplication_Item put: stream contents).

 CZFileSaver new bibset: bibset ; saveTo: fileName
```

Création d'une entrée commentaire dans le fichier merge

Création du nouveau fichier mergé

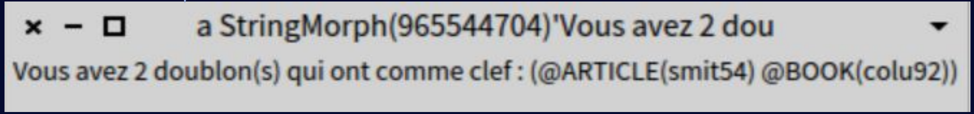
Message de doublon dans la console pharo

Test Fonctionnel



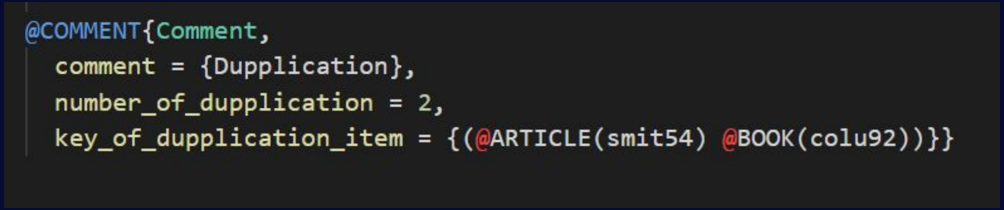
```
instance := CZFileMergerWithDuplicateWarning new.
instance inputFiles: #('/Users/kernouf/Desktop/template.bib'
'/Users/kernouf/Desktop/sample.bib');
merge: '/Users/kernouf/Desktop/new.bib'.
```

Popup Warning dans la console pharo



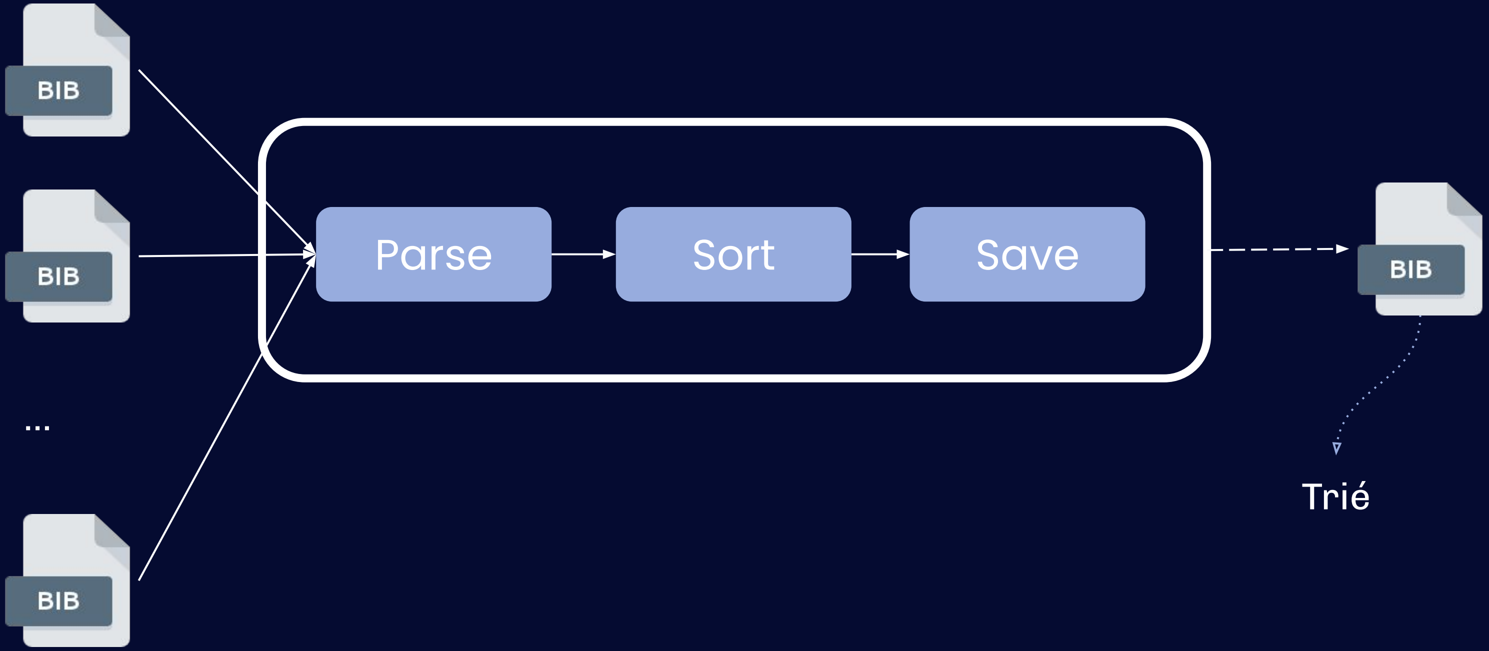
```
a StringMorph(965544704)'Vous avez 2 dou
Vous avez 2 doublon(s) qui ont comme clef : (@ARTICLE(smit54) @BOOK(colu92))
```

Nouveau fichier généré



```
@COMMENT{Comment,
  comment = {Dupplication},
  number_of_dupplication = 2,
  key_of_dupplication_item = {(@ARTICLE(smit54) @BOOK(colu92))}}
```

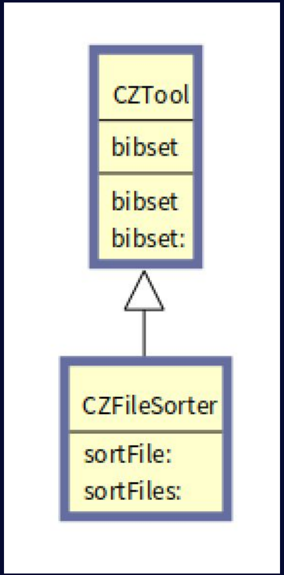
Sorter

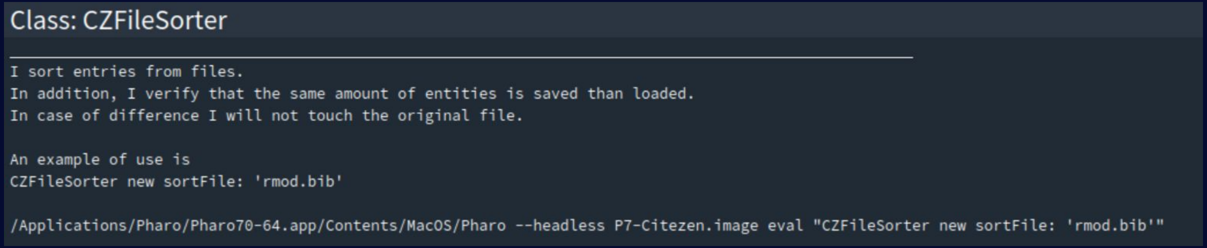


Nos objectifs

- Trier un ou plusieurs fichiers par rapport à différents arguments (Titre, clé, Date , ...).
- Une fois trié, enregistrer le fichier généré à la place de celui passé en argument.
- Tester et commenter le code écrit.
- Phase de refactoring.

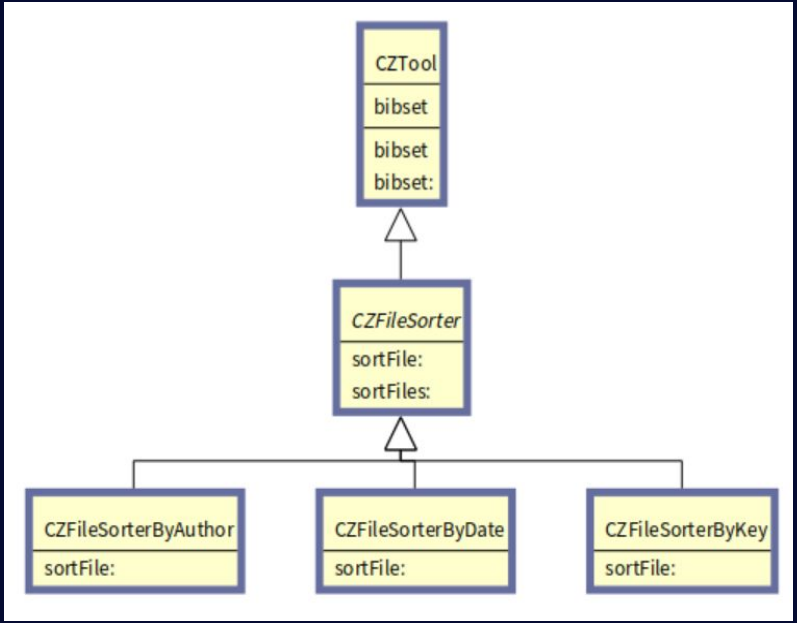
Reverse Engineering



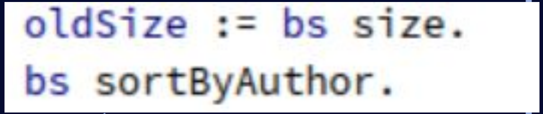

Class: CZFileSorter

```
I sort entries from files.
In addition, I verify that the same amount of entities is saved than loaded.
In case of difference I will not touch the original file.

An example of use is
CZFileSorter new sortFile: 'rmod.bib'

/Applications/Pharo/Pharo70-64.app/Contents/MacOS/Pharo --headless P7-Citezen.image eval "CZFileSorter new sortFile: 'rmod.bib'"
```

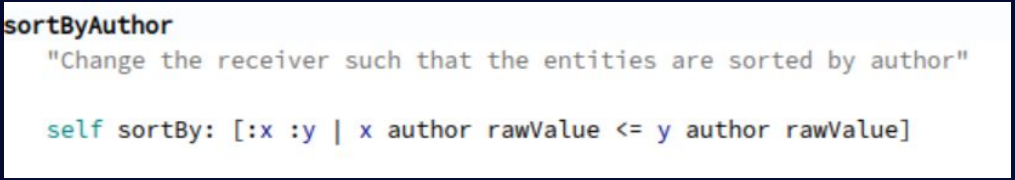
Nos changement



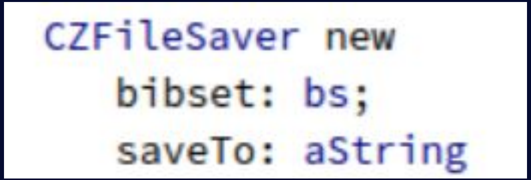
Exemple : SorterByAuthor



```
oldSize := bs size.
bs sortByAuthor.
```



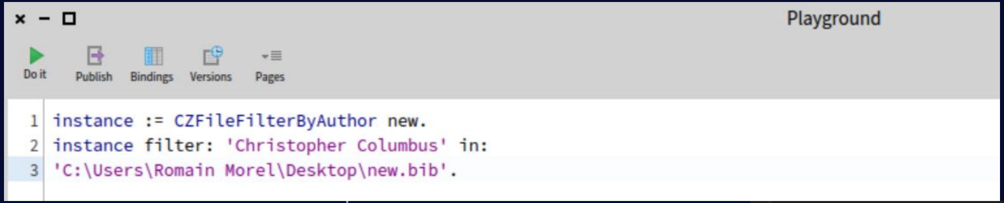
```
sortByAuthor
   "Change the receiver such that the entities are sorted by author"

   self sortBy: [:x :y | x author rawValue <= y author rawValue]
```



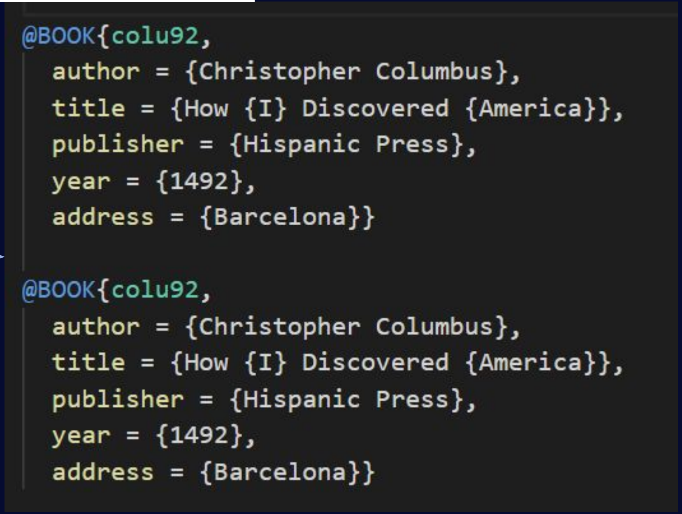
```
CZFileSaver new
   bibset: bs;
   saveTo: aString
```

Permet de save le nouveau bib

Test Fonctionnel

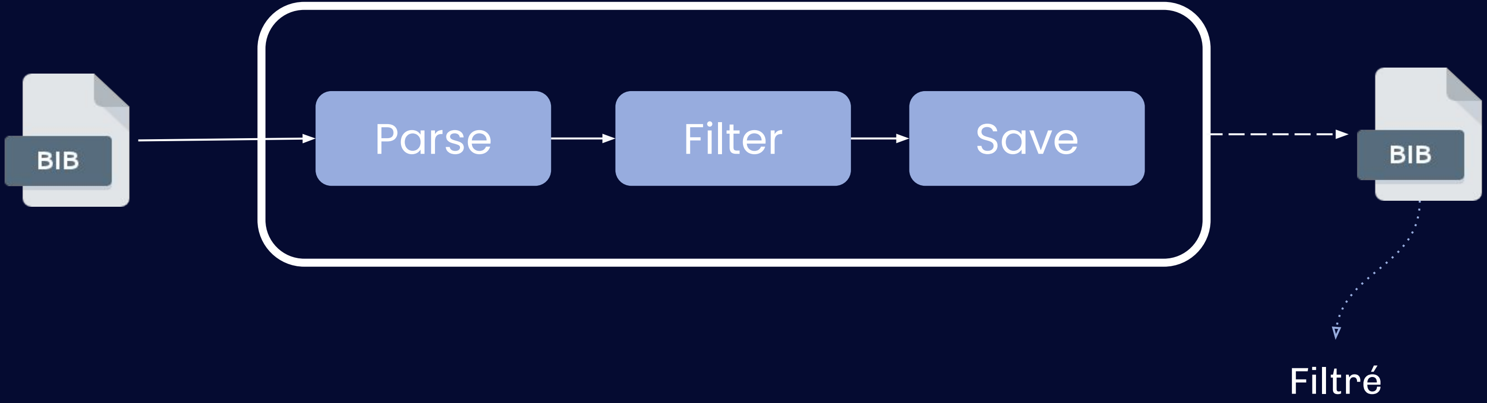


```
instance := CZFileFilterByAuthor new.
instance filter: 'Christopher Columbus' in:
'C:\Users\Romain Morel\Desktop\new.bib'.
```



```
@BOOK{colu92,
  author = {Christopher Columbus},
  title = {How {I} Discovered {America}},
  publisher = {Hispanic Press},
  year = {1492},
  address = {Barcelona}}

@BOOK{colu92,
  author = {Christopher Columbus},
  title = {How {I} Discovered {America}},
  publisher = {Hispanic Press},
  year = {1492},
  address = {Barcelona}}
```

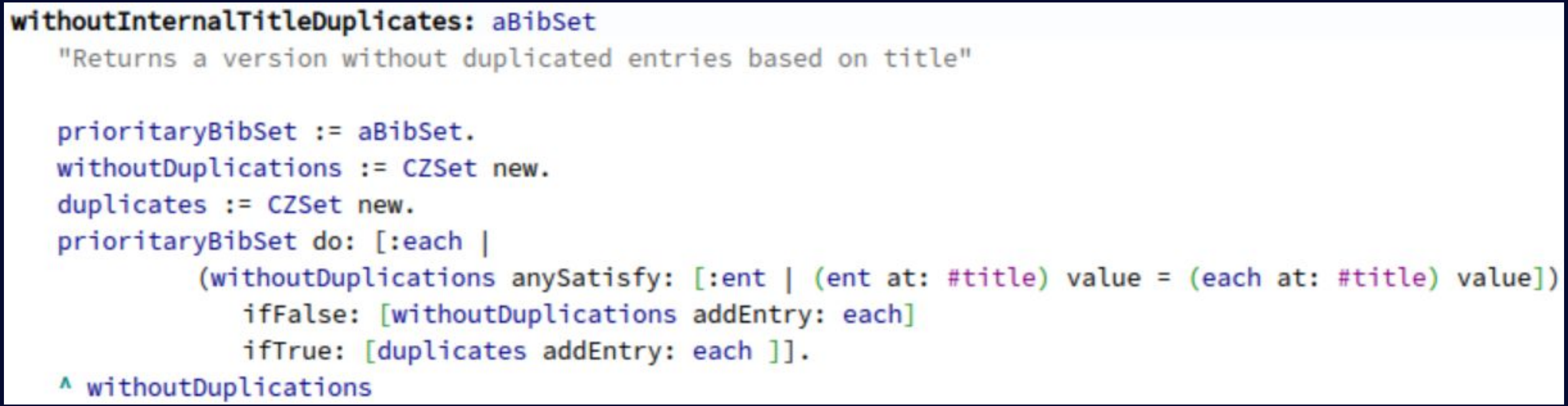
Filter



Nos objectifs

- Filtrer un ou plusieurs fichiers par rapport à différents arguments (Titre, clé, Date , ...).
- Une fois filtré, enregistrer le fichier généré à la place de celui passé en argument.
- Tester et commenter le code écrit.
- Phase de refactoring.

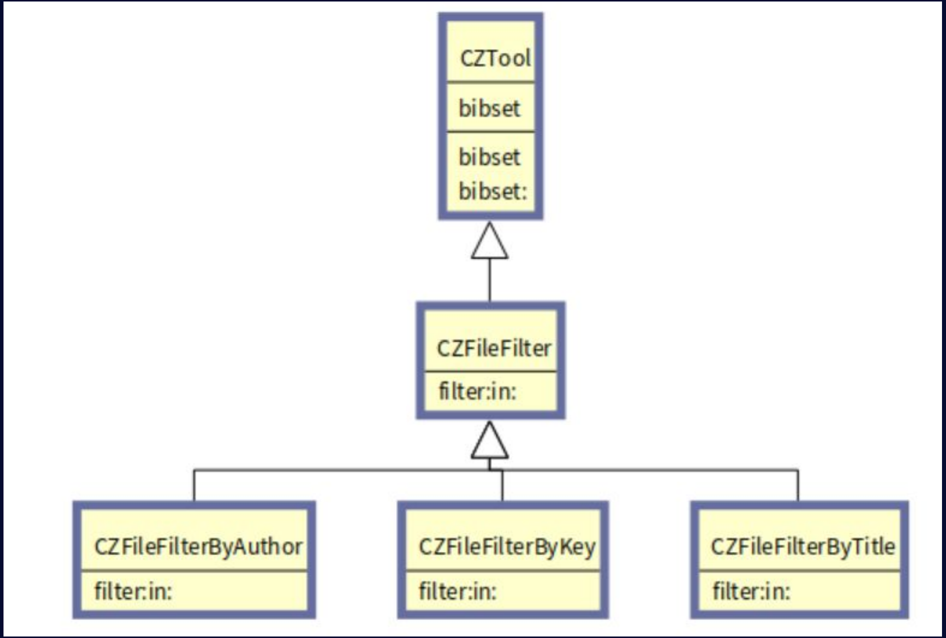
Reverse Engineering



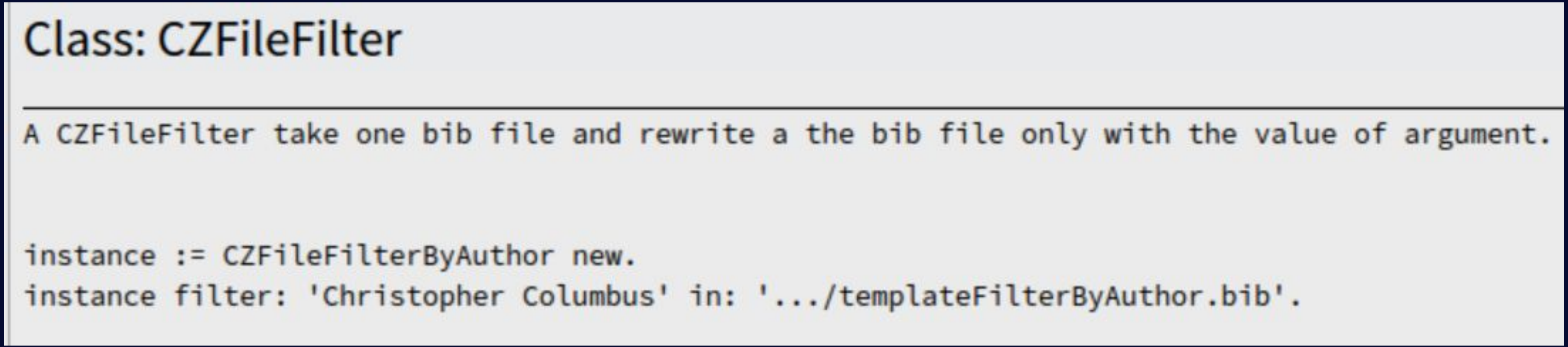
```
withoutInternalTitleDuplicates: aBibSet
    "Returns a version without duplicated entries based on title"

    prioritaryBibSet := aBibSet.
    withoutDuplications := CZSet new.
    duplicates := CZSet new.
    prioritaryBibSet do: [:each |
             (withoutDuplications anySatisfy: [:ent | (ent at: #title) value = (each at: #title) value])
                ifFalse: [withoutDuplications addEntry: each]
                ifTrue: [duplicates addEntry: each ]].
    ^ withoutDuplications
```

Pour faire Filter nous nous sommes inspirés du fonctionnement de la fonction withoutInternalTitleDuplicate

Nos changement



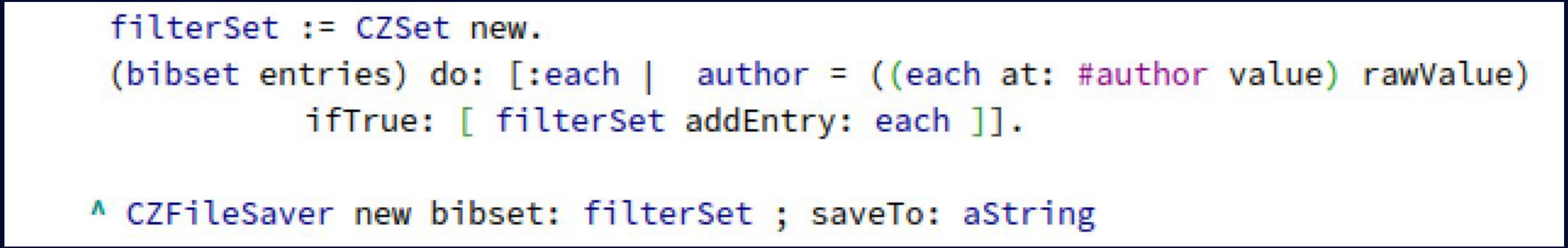
Ajout de Commentaire



Class: CZFileFilter

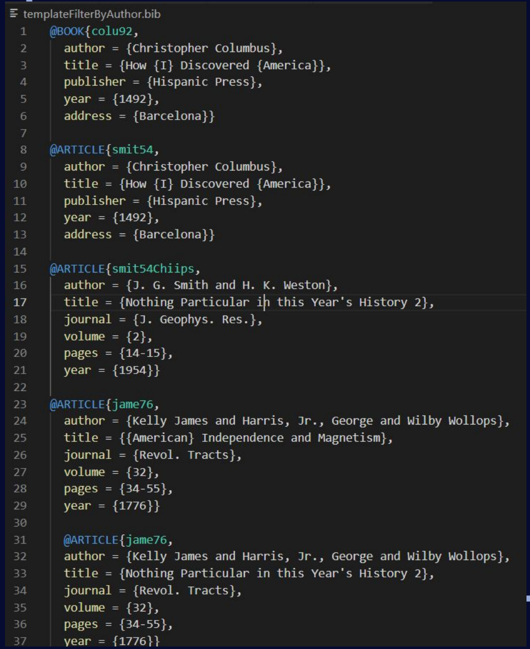
```
A CZFileFilter take one bib file and rewrite a the bib file only with the value of argument.



instance := CZFileFilterByAuthor new.
instance filter: 'Christopher Columbus' in: '.../templateFilterByAuthor.bib'.
```

FilterByAuthor



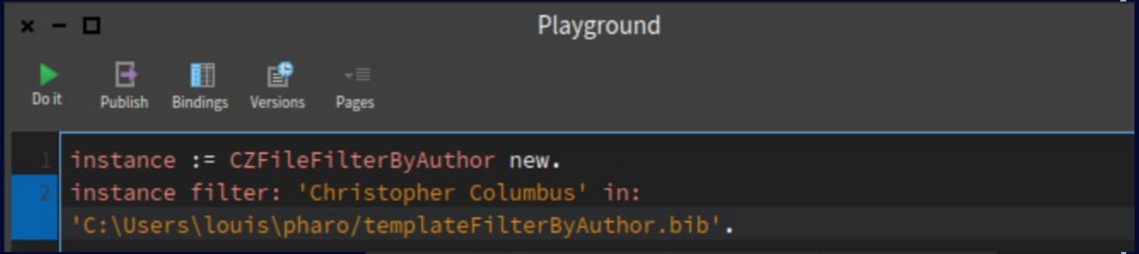
```
    filterSet := CZSet new.
    (bibset entries) do: [:each |  author = ((each at: #author value) rawValue)
              ifTrue: [ filterSet addEntry: each ]].

 ^ CZFileSaver new bibset: filterSet ; saveTo: aString
```

Test Fonctionnel



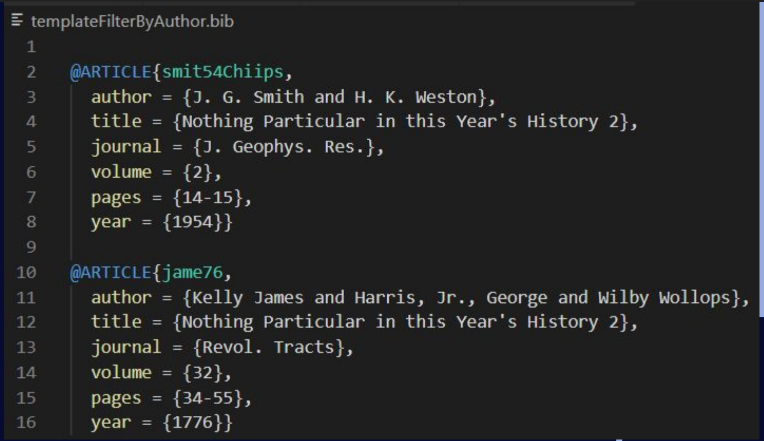
templateFilterByAuthor.bib

```
@BOOK{colu92,
  author = {Christopher Columbus},
  title = {How {I} Discovered {America}},
  publisher = {Hispanic Press},
  year = {1492},
  address = {Barcelona}}

@ARTICLE{smit54,
  author = {Christopher Columbus},
  title = {How {I} Discovered {America}},
  publisher = {Hispanic Press},
  year = {1492},
  address = {Barcelona}}

@ARTICLE{smit54Chiips,
  author = {J. G. Smith and H. K. Weston},
  title = {Nothing Particular in this Year's History 2},
  journal = {J. Geophys. Res.},
  volume = {2},
  pages = {14-15},
  year = {1954}}

@ARTICLE{jame76,
  author = {Kelly James and Harris, Jr., George and Wilby Wollops},
  title = {{American} Independence and Magnetism},
  journal = {Revol. Tracts},
  volume = {32},
  pages = {34-55},
  year = {1776}}

  @ARTICLE{jame76,
  author = {Kelly James and Harris, Jr., George and Wilby Wollops},
  title = {Nothing Particular in this Year's History 2},
  journal = {Revol. Tracts},
  volume = {32},
  pages = {34-55},
  year = {1776}}
```



```
instance := CZFileFilterByAuthor new.
instance filter: 'Christopher Columbus' in:
'C:\Users\louis\pharo/templateFilterByAuthor.bib'.
```

Filter : ‘Nothing Par…’
in :
templateFilterByAuthor.bib

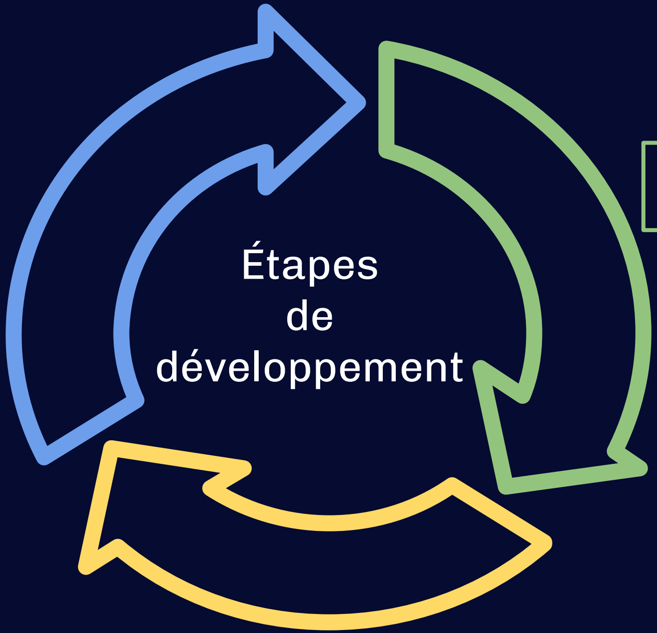


templateFilterByAuthor.bib

```

@ARTICLE{smit54Chiips,
  author = {J. G. Smith and H. K. Weston},
  title = {Nothing Particular in this Year's History 2},
  journal = {J. Geophys. Res.},
  volume = {2},
  pages = {14-15},
  year = {1954}}

@ARTICLE{jame76,
  author = {Kelly James and Harris, Jr., George and Wilby Wollops},
  title = {Nothing Particular in this Year's History 2},
  journal = {Revol. Tracts},
  volume = {32},
  pages = {34-55},
  year = {1776}}
```

III Conclusion

Notre avancement

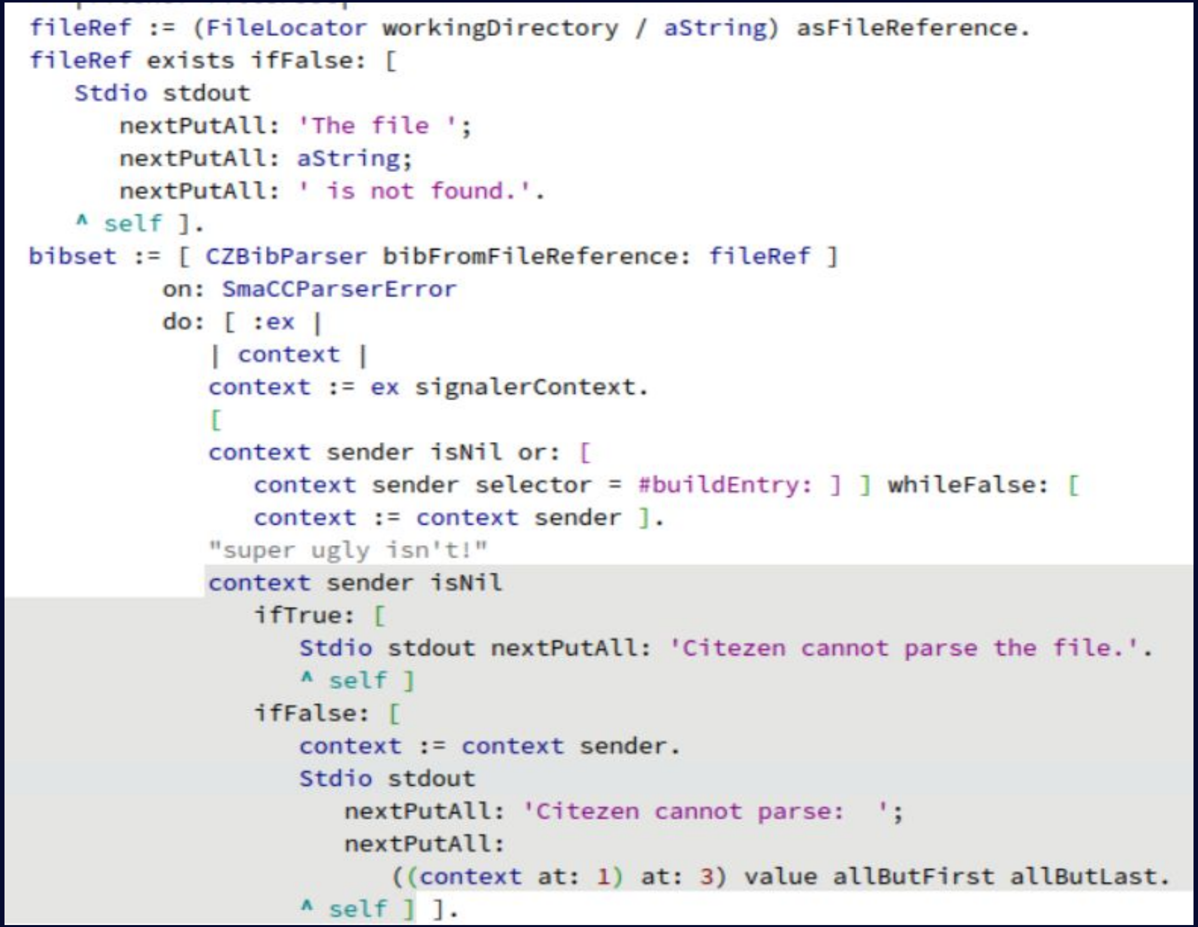


Refactoring

Conception/réalisation

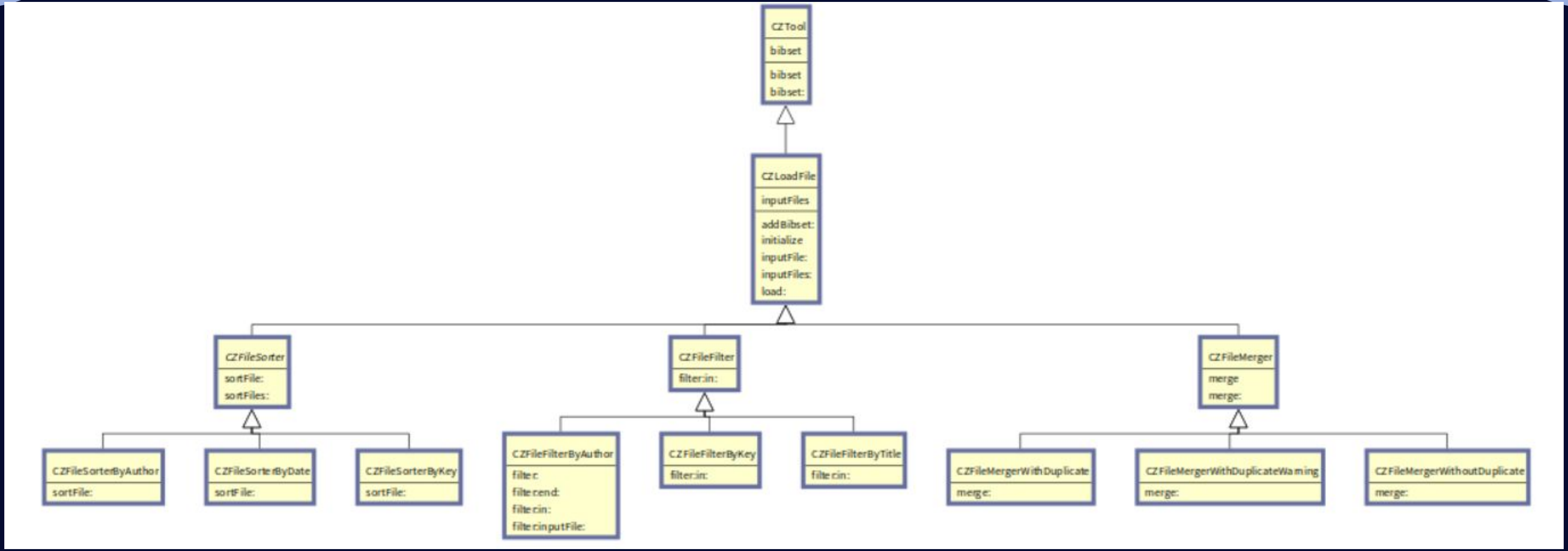
Tests fonctionnels et unitaires

Load un fichier



```
fileRef := (FileLocator workingDirectory / aString) asFileReference.
fileRef exists ifFalse: [
   Stdio stdout
      nextPutAll: 'The file ';
      nextPutAll: aString;
      nextPutAll: ' is not found.'.
   ^ self ].
bibset := [ CZBibParser bibFromFileReference: fileRef ]
         on: SmaCCParserError
         do: [ :ex |
            | context |
            context := ex signalerContext.
            [
            context sender isNil or: [
               context sender selector = #buildEntry: ] ] whileFalse: [
               context := context sender ].
            "super ugly isn't!"
            context sender isNil
               ifTrue: [
                  Stdio stdout nextPutAll: 'Citezen cannot parse the file.'.
                  ^ self ]
               ifFalse: [
                  context := context sender.
                  Stdio stdout
                     nextPutAll: 'Citezen cannot parse:  ';
                     nextPutAll:
                        ((context at: 1) at: 3) value allButFirst allButLast.
                  ^ self ] ].
```

Evolution retenue


CZTool
bibset
bibset
bibset:
CZLoadFile
inputFiles
addBibset:
initialize
inputFile:
inputFiles:
load:
CZFileSorter
sortFile:
sortFiles:
CZFileFilter
filter:in:
CZFileMerger
merge
merge:
CZFileSorterByAuthor
sortFile:
CZFileSorterByDate
sortFile:
CZFileSorterByKey
sortFile:
CZFileFilterByAuthor
filter:
filter:cond:
filter:in:
filter:inputFile:
CZFileFilterByKey
filter:in:
CZFileFilterByTitle
filter:in:
CZFileMergerWithDuplicate
merge:
CZFileMergerWithDuplicateWarning
merge:
CZFileMergerWithoutDuplicate
merge: