# PROJET AVL

I.Introduction

II.Structure du projet

III.Code coverage & Mutation testing

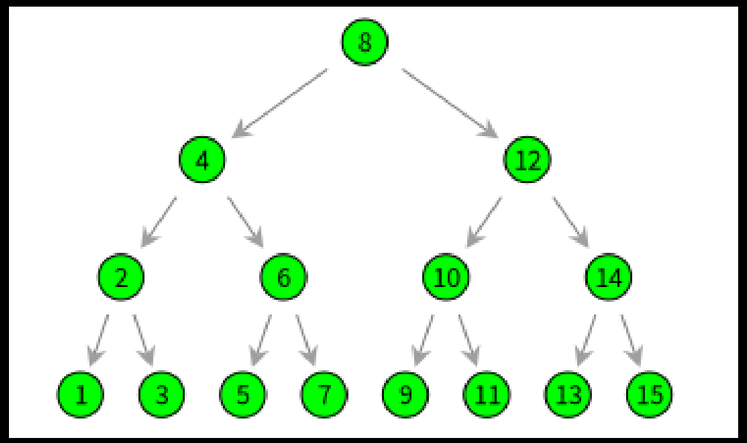
IV.Conclusion

Besbas Mélik
Leulmi Mohamed

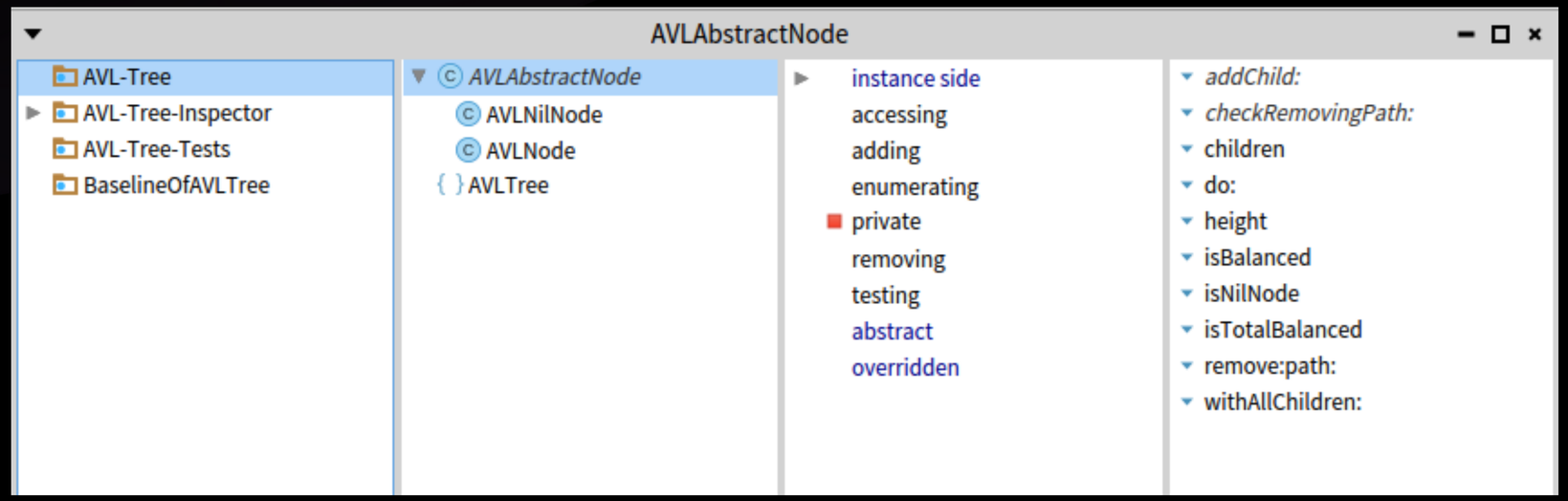
# I. INTRODUCTION

## AVL C’est quoi ?

- Arbre binaire de recherche auto-équilibré possédant une racine et de sorte que chaque nœud possède au maximum 2 éléments fils.
- Les hauteurs des deux sous-arbres d'un même nœud diffèrent au plus de un



# II.STRUCTURE DU PROJET




- AVL-Tree : Le package principal pour l'implémentation de l'arbre AVL

- AVL-Tree-Inspector : contient des outils d'inspection spécifiques pour l'arbre AVL.

- AVL-Tree-Tests :  contient les tests unitaires pour l'implémentation de l'arbre AVL.

- AVL-Tree : un package de configuration ou de gestion des dépendances.

# II.STRUCTURE DU PROJET

## Implémentation du code

Classe abstraite

```
Object subclass: #AVLAbstractNode
    instanceVariableNames: ''
    classVariableNames: ''
    package: 'AVL-Tree'
```

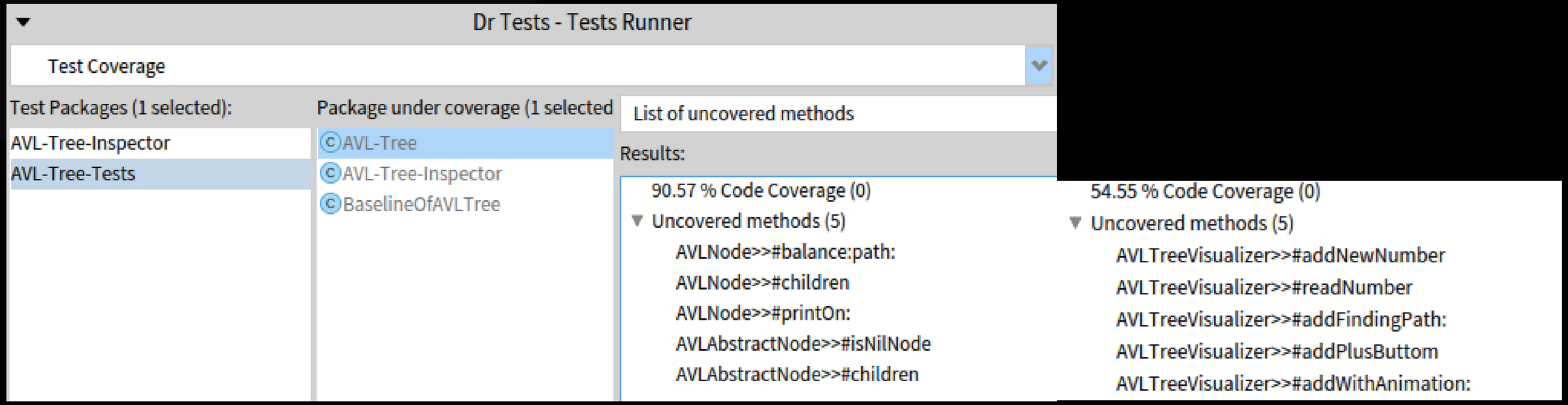
Classe représentant un noeud vide

```
AVLAbstractNode subclass: #AVLNilNode
    instanceVariableNames: ''
    classVariableNames: ''
    package: 'AVL-Tree'
```

Classe représentant un noeud non vide

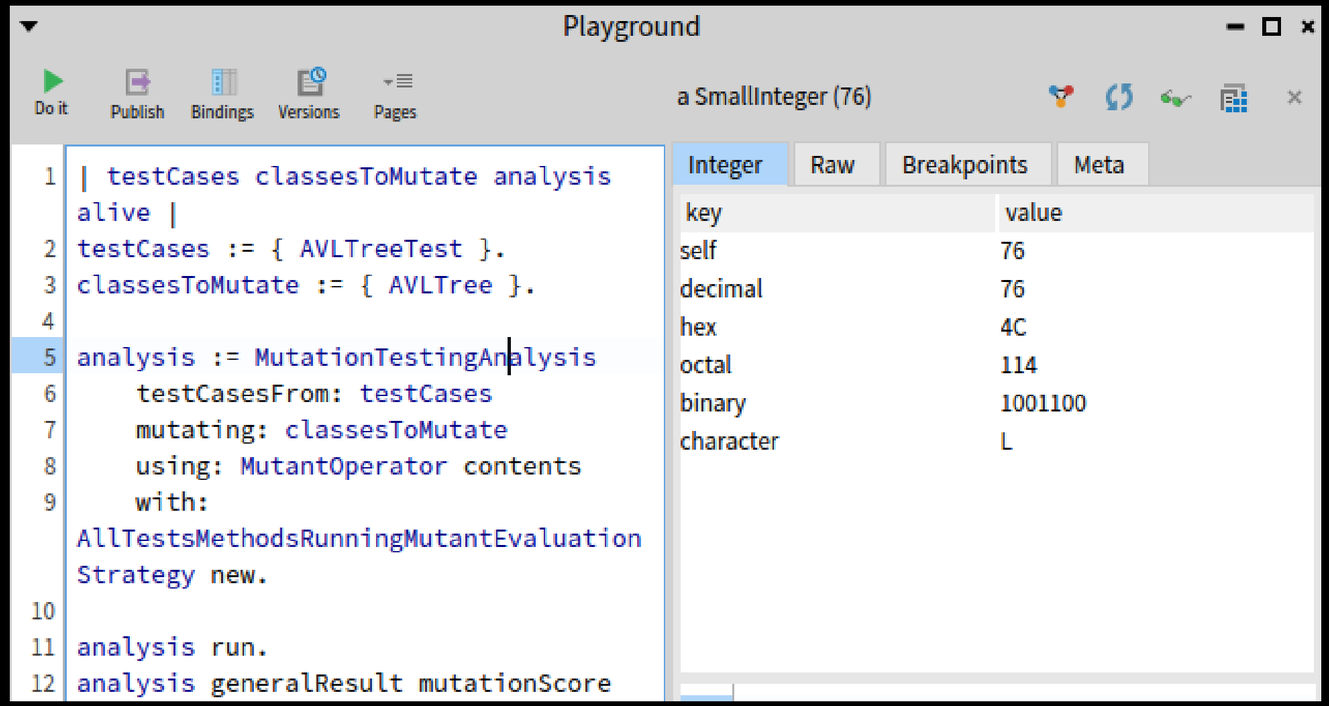
```
AVLAbstractNode subclass: #AVLNode
    instanceVariableNames: 'left contents right'
    classVariableNames: ''
    package: 'AVL-Tree'
```

# III.A. CODE COVERAGE




- Plus de 90% de couverture de test pour la classe principale
- 5 Méthodes non couvertes
- Ne mesure pas la pertinence et la précision des Tests
- Pas un indicateur de performance des tests

# III.B. MUTATION TESTING




- 76% de Couverture de Mutation
- 24% de Mutations non Détectées

# III.B. MUTATION TESTING

| Variable | Value |
| --- | --- |
| self | 17 mutants, 13 killed, 4 alive, 0 terminated. Mutation Score: 76%. |
| particularResults | an OrderedCollection [17 items] (Remove ^ in AVLTree>>#add: Remove ^ in AVLTree>>#height Remove [..] |
| elapsedTime | 0:00:00:00.243 |

| | Value |
| --- | --- |
| 1 | Remove ^ in AVLTree>>#add: |
| 2 | Remove ^ in AVLTree>>#includes: |
| 3 | Remove ^ in AVLTree>>#remove:ifAbsent: |
| 4 | Remove ^ in AVLTree>>#inspectorCanvas |

- 4 mutants toujours en vie
- Amélioration du score de mutation possible

# III.B. MUTATION TESTING

## Suppression des mutants

### Mutant 1 (add)

```
add: newObject

    root := root addChild: newObject.
    ^ newObject
```

### Mutant 2 (includes)

```
includes: anObject

    anObject ifNil: [ ^ nil ].
    ^ (self search: anObject) notNil
```

### Mutant 3 (remove)

```
    | toRemove path |
    path := OrderedCollection new.
    toRemove := root remove: oldObject path: path.
    toRemove ifNil: [ ^ anExceptionBlock value ].

    toRemove == root ifTrue: [
       root := root successor: path.
       root ifNil: [ root := AVLNilNode new ] ].
    root checkRemovingPath: path.


    ^ toRemove contents
```

### Mutant 4 (inspectorCanvas)

```
inspectorCanvas

    <inspectorPresentationOrder: 90 title: 'AVL'>
    ^ AVLTreeVisualizer new
        tree: self;
        asPresenter
```

# III.B. MUTATION TESTING

## Modifications apportées

| Variable | Value |
| --- | --- |
| self | 13 mutants, 13 killed, 0 alive, 0 terminated. Mutation Score: 100%. |

- Tout les mutants ont été éliminés
- Amélioration du Mutation score

# IV. CONCLUSION

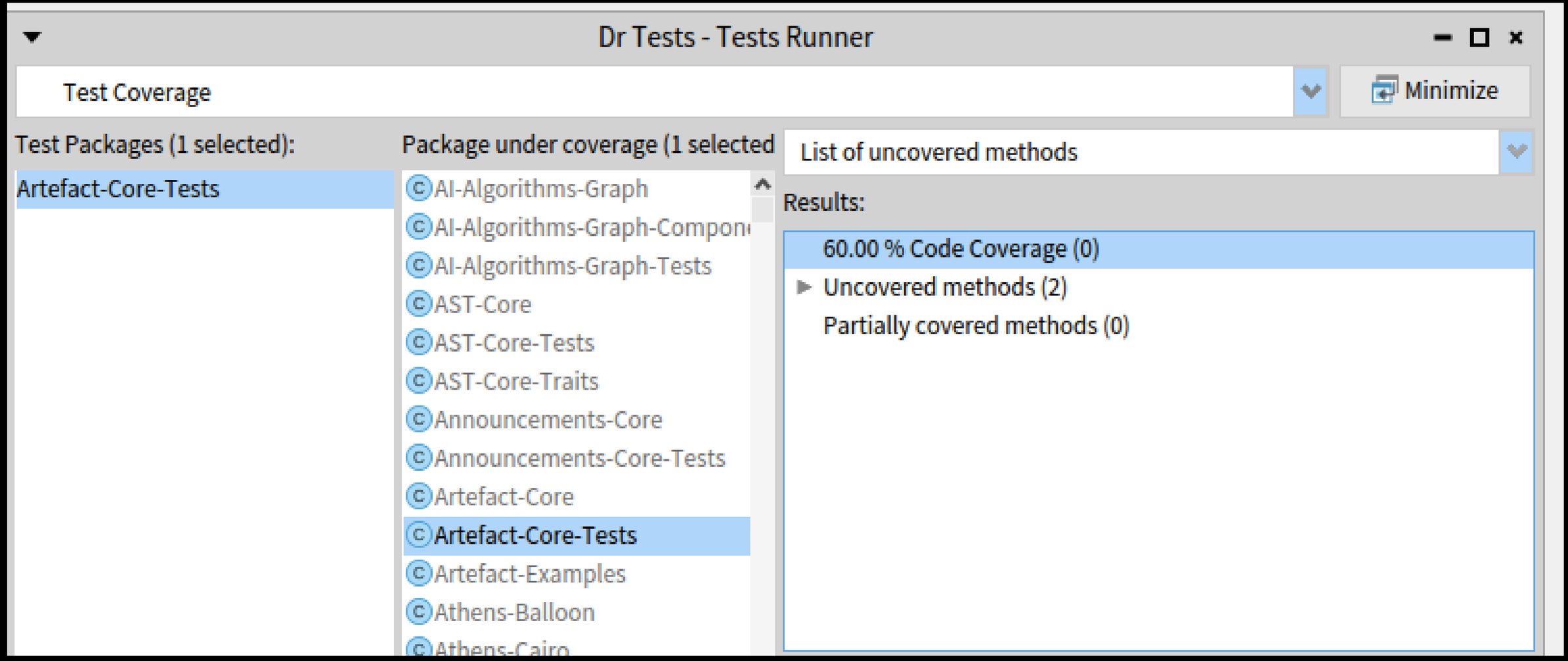
- Petit projet avec très peu de classe
- Pas de Design Pattern évident
- Méthodes assez courtes
- Coverage de test améliorable
- Très peu de documentation

# ARTEFACT

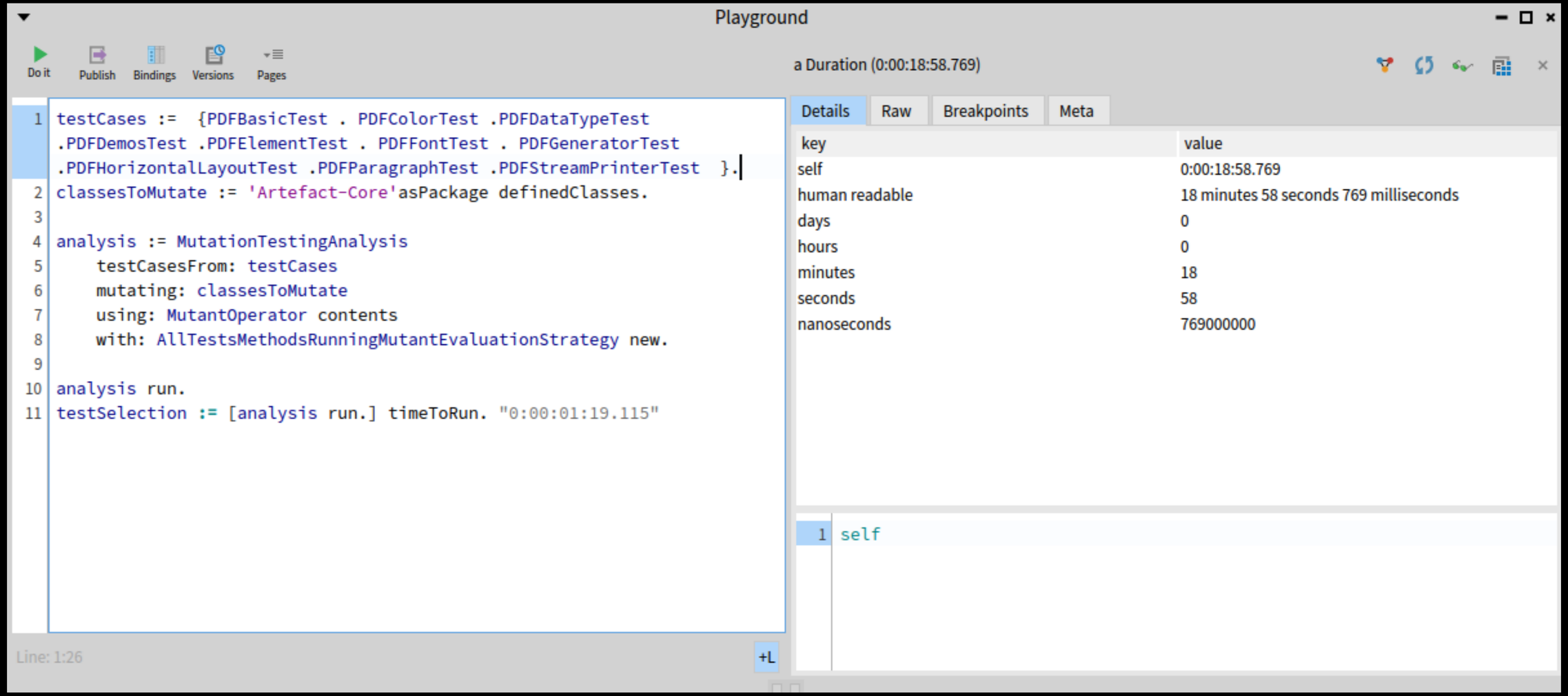
1. **Mutation Testing**
2. **Design pattern**
3. **Est il facilement utilisable ?**
4. **Conclusion**

# I.MUTATION TESTING

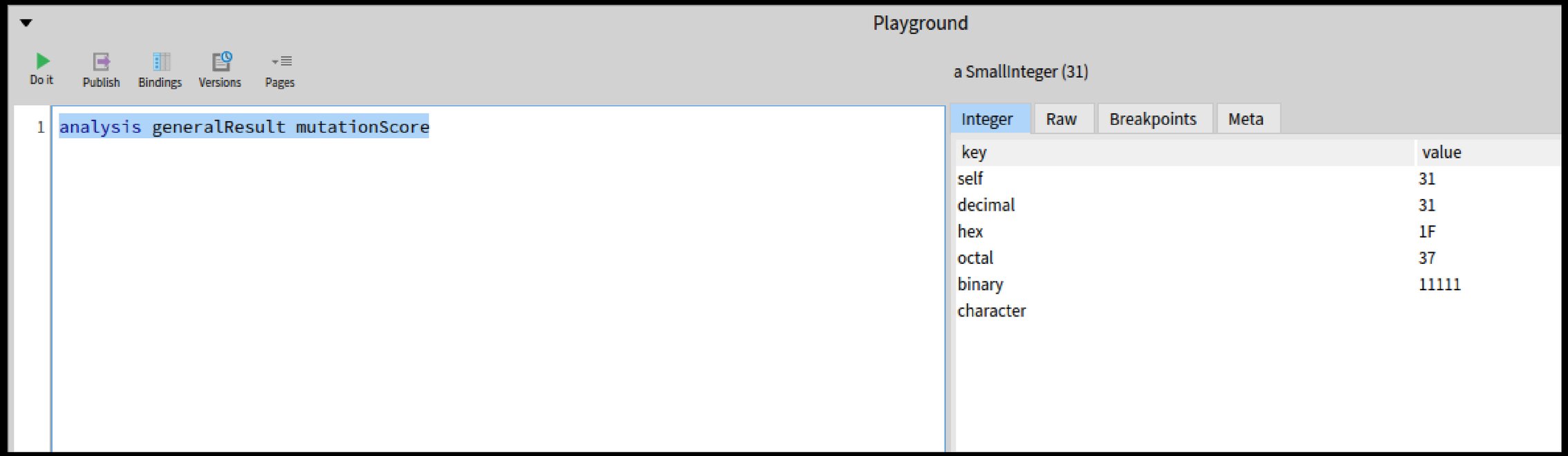
# I.Mutation Testing



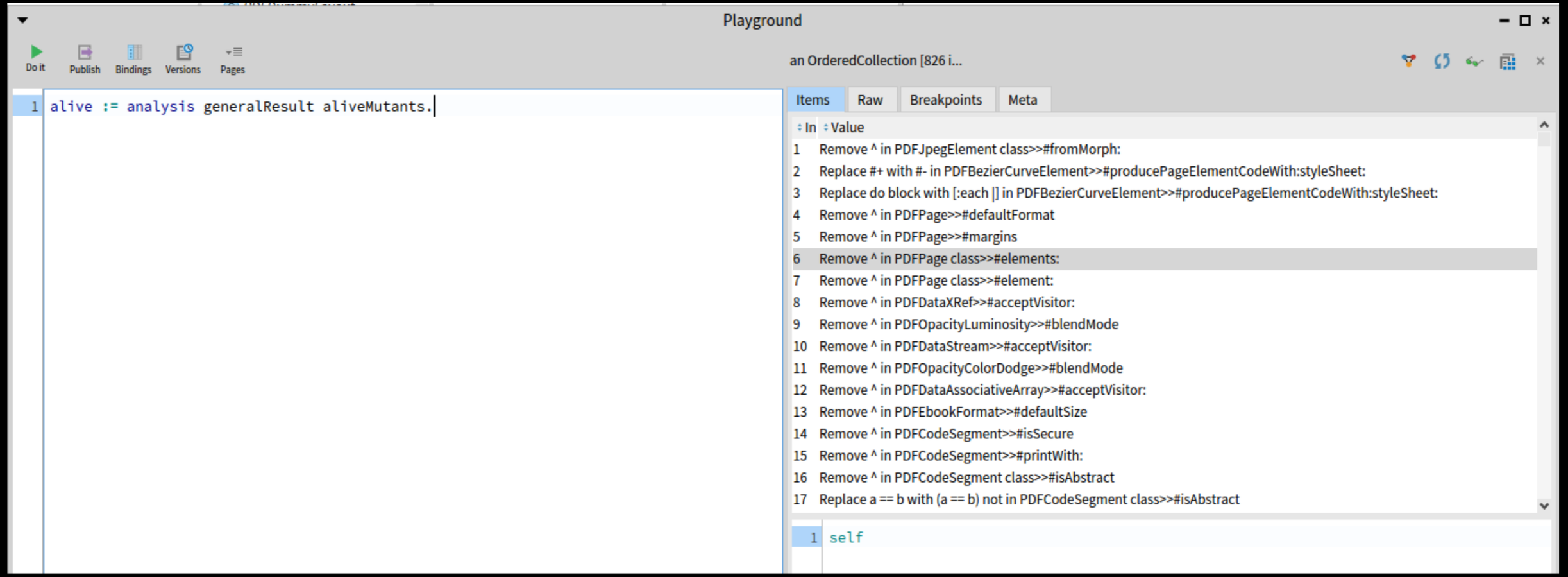
# I.Mutation Testing



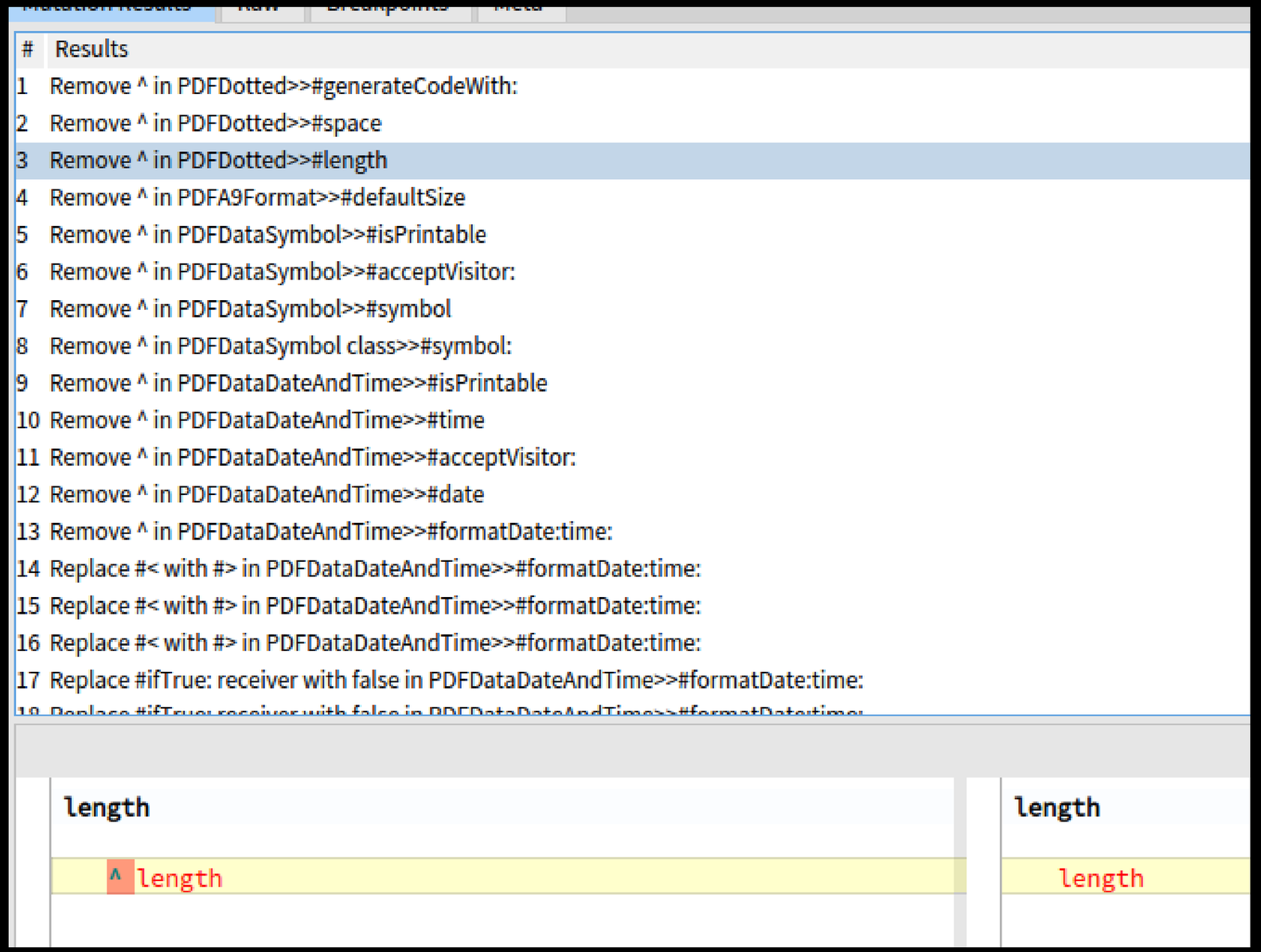
# I.Mutation Testing



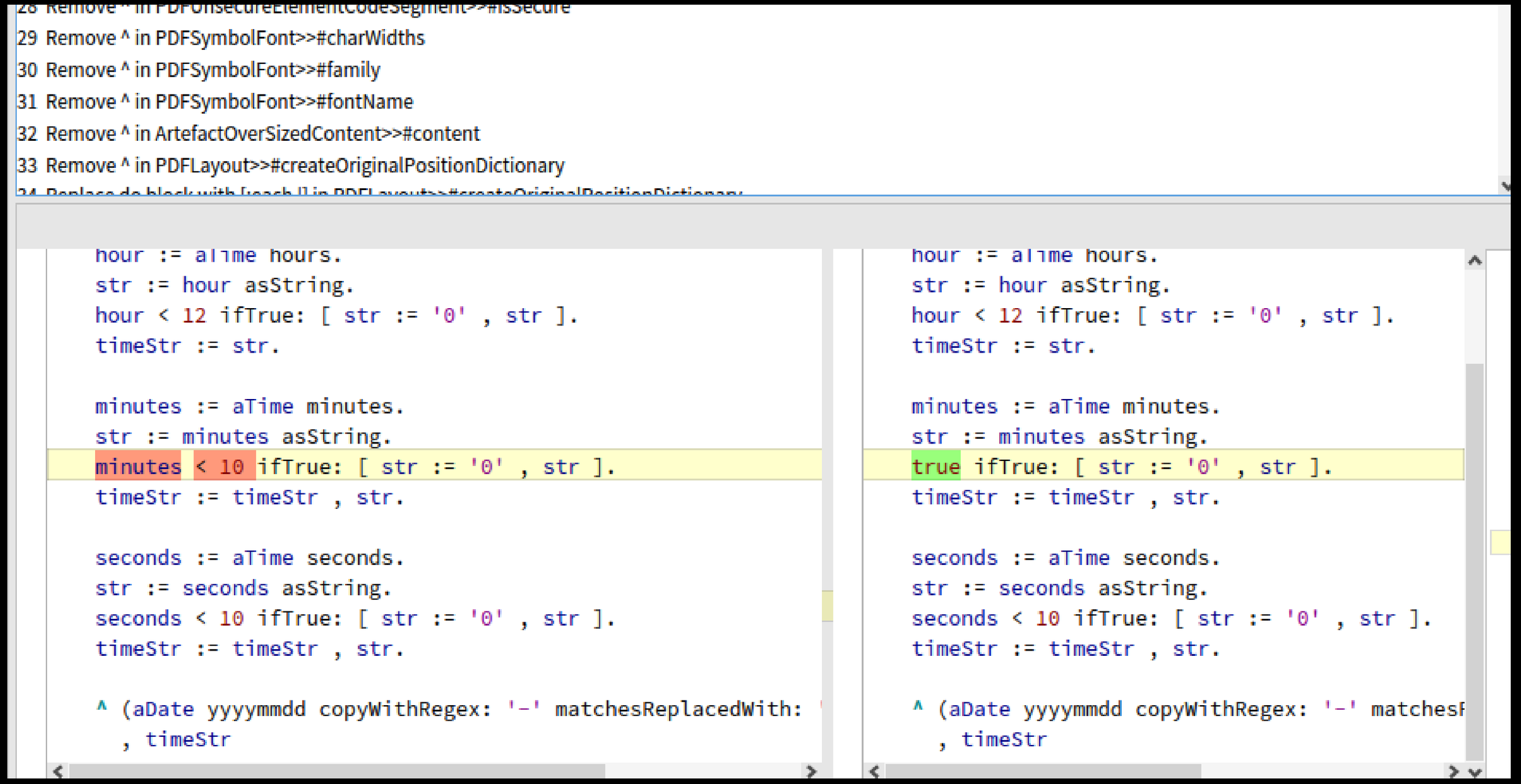
# I.Mutation Testing



# I.Mutation Testing

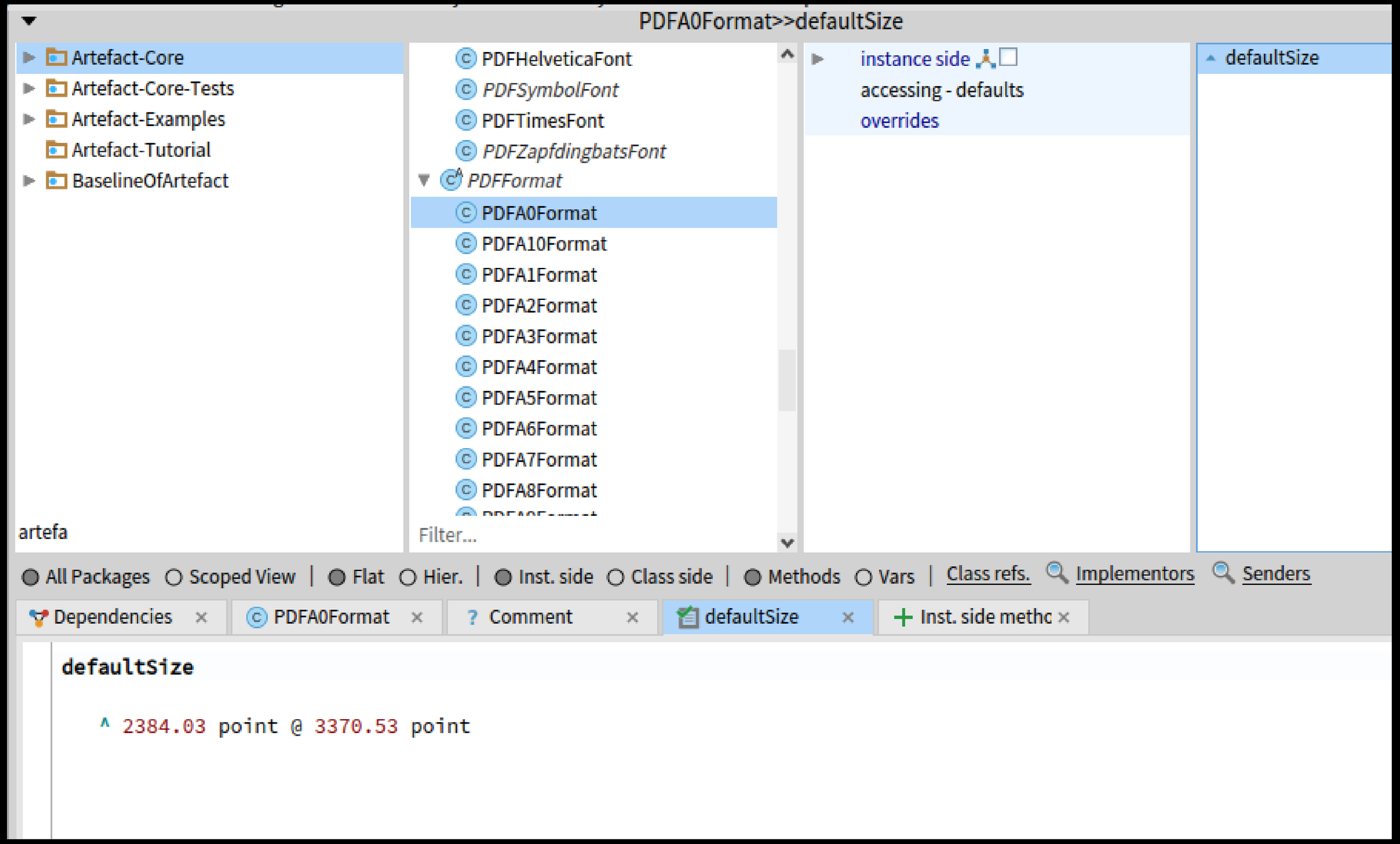


# I.Mutation Testing

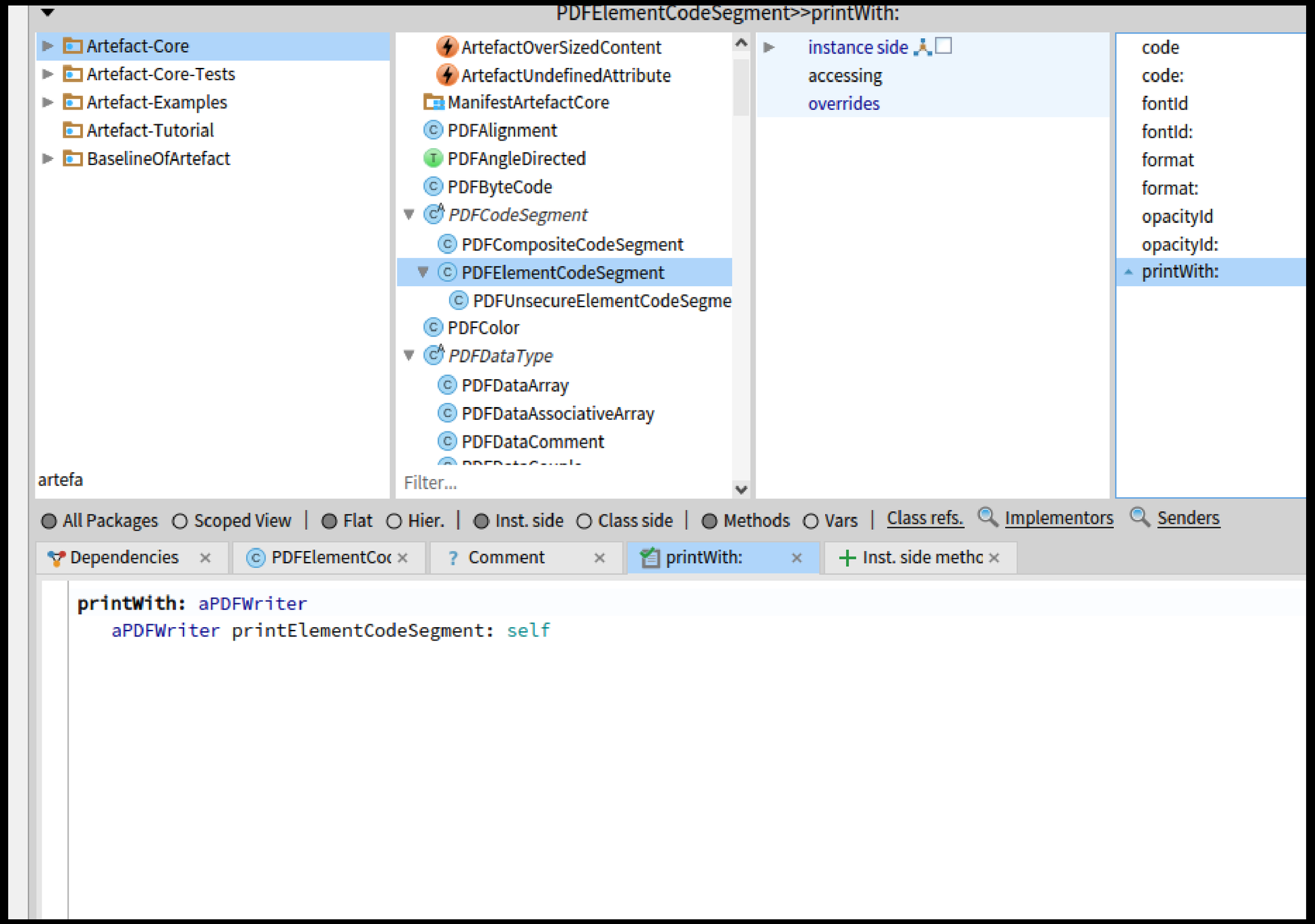


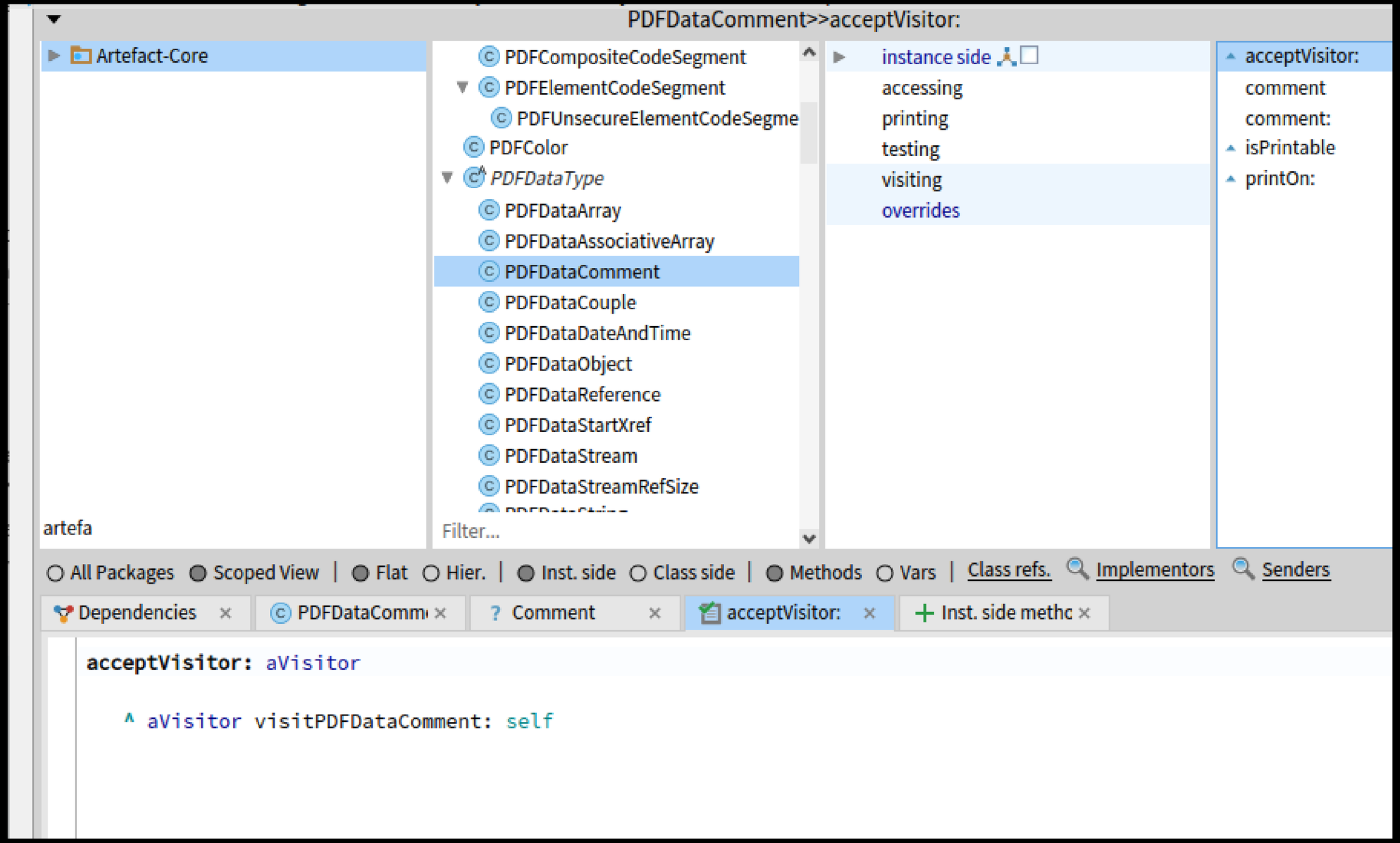
# II.DESIGN PATTERN

# Factory

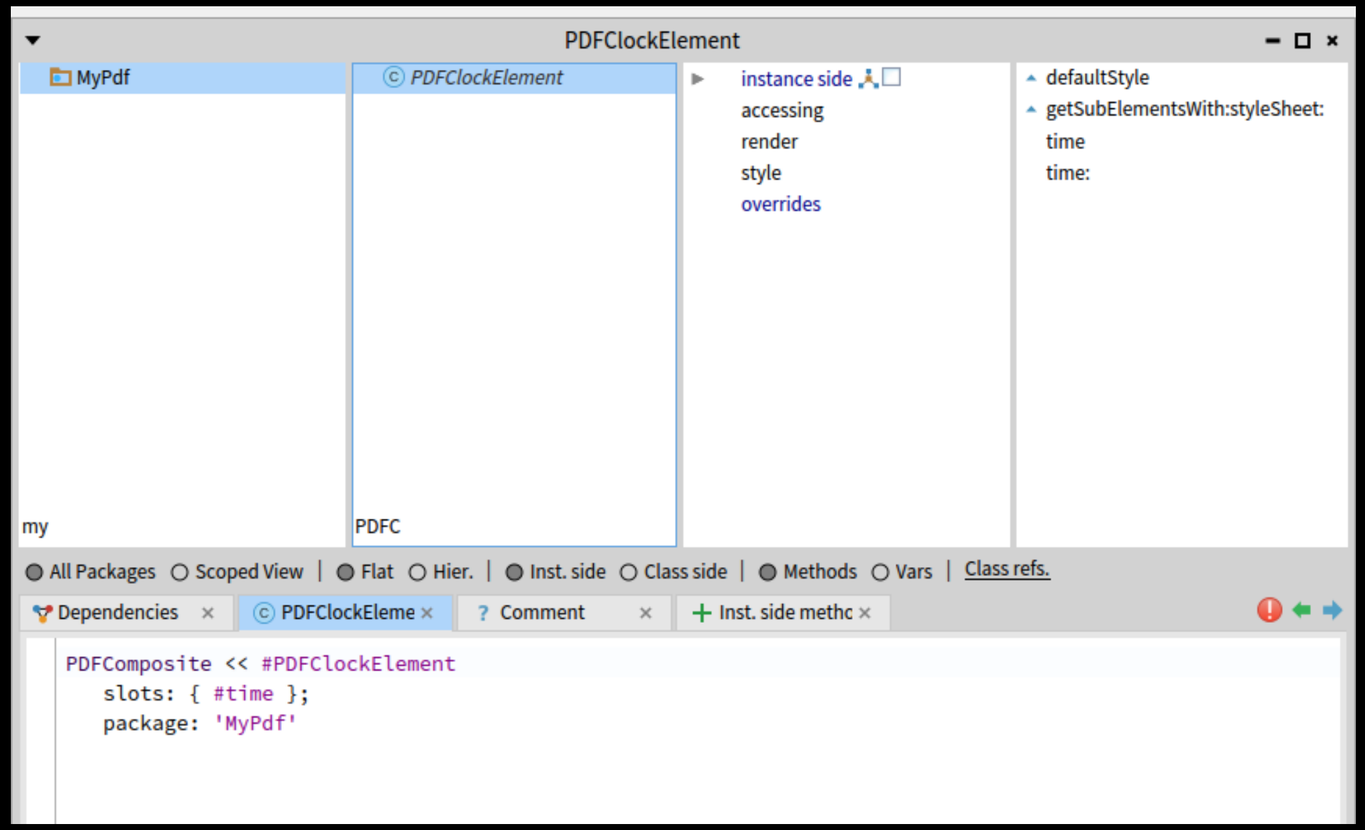

PDFA0Format>>defaultSize
Artefact-Core
Artefact-Core-Tests
Artefact-Examples
Artefact-Tutorial
BaselineOfArtefact
artefa
PDFHelveticaFont
PDFSymbolFont
PDFTimesFont
PDFZapfdingbatsFont
PDFFormat
PDFA0Format
PDFA10Format
PDFA1Format
PDFA2Format
PDFA3Format
PDFA4Format
PDFA5Format
PDFA6Format
PDFA7Format
PDFA8Format
Filter...
instance side
accessing - defaults
overrides
defaultSize
All Packages
Scoped View
Flat
Hier.
Inst. side
Class side
Methods
Vars
Class refs.
Implementors
Senders
Dependencies
PDFA0Format
Comment
defaultSize
Inst. side methc
defaultSize
^ 2384.03 point @ 3370.53 point

# Double Dispatch




PDFDataComment>>acceptVisitor:
Artefact-Core
artefa
PDFCompositeCodeSegment
PDFElementCodeSegment
PDFUnsecureElementCodeSegme
PDFColor
PDFDataType
PDFDataArray
PDFDataAssociativeArray
PDFDataComment
PDFDataCouple
PDFDataDateAndTime
PDFDataObject
PDFDataReference
PDFDataStartXref
PDFDataStream
PDFDataStreamRefSize
Filter...
instance side
accessing
printing
testing
visiting
overrides
acceptVisitor:
comment
comment:
isPrintable
printOn:
All Packages
Scoped View
Flat
Hier.
Inst. side
Class side
Methods
Vars
Class refs.
Implementors
Senders
Dependencies
PDFDataComm
Comment
acceptVisitor:
Inst. side methc
acceptVisitor: aVisitor
^ aVisitor visitPDFDataComment: self

# Visitor


PDFClockElement
MyPdf
PDFClockElement
instance side
accessing
render
style
overrides
defaultStyle
getSubElementsWith:styleSheet:
time
time:
my
PDFC
All Packages
Scoped View
Flat
Hier.
Inst. side
Class side
Methods
Vars
Class refs.
Dependencies
PDFClockEleme
Comment
Inst. side methc
PDFComposite << #PDFClockElement
slots: { #time };
package: 'MyPdf'

# III. EST CE FACILEMENT UTILISABLE ?

PDFClockElement>>getSubElementsWith:styleSheet:

MyPdf | PDFClockElement | instance side | defaultStyle

accessing | getSubElementsWith:styleSheet:

render | time

style | time:

overrides

my | PDFC

All Packages | Scoped View | Flat | Hier. | Inst. side | Class side | Methods | Vars | Class refs. | Implementors | Senders

Dependencies | PDFClockEleme | Comment | getSubElement | Inst. side methc

```
getSubElementsWith: aGenerator styleSheet: aStyleSheet
   | hourAngle minuteAngle |
   hourAngle := Float pi / 2 - (time hour12 * 2 * Float pi / 12).
   minuteAngle := Float pi / 2 - (time minute * 2 * Float pi / 60).

   ^ {(PDFCircleElement from: self from to: self to).
      (PDFCircleElement center: self center radius: self dimension x * 0.05).
      (PDFArrowElement from: self center angle: hourAngle length: dimension x * 0.25).
      (PDFArrowElement from: self center angle: minuteAngle length: dimension x * 0.45)}
```

```
colorTest: aStream
   "generate a sample document with colors"

   | pdfdoc aPage |
   pdfdoc := PDFDocument new.

   aPage := PDFPage new.
   aPage add: (PDFCellElement new
      font: (PDFTimesFont new fontSize: 32pt);
      from: 10mm@10mm;
      dimension: 100 mm @ 20 mm;
      text: 'Hello World!';
      textColor: (PDFColor r: 255 g: 0 b: 0);
      fillColor: (PDFColor r: 0 g: 255 b: 0)
      ).
   aPage add: (PDFRectElement new
      from: 10 mm @ 50 mm;
      dimension: 50 mm @ 50 mm;
      thickness: 5pt;
      drawColor: (PDFColor r: 0 g: 0 b: 255);
      fillColor: (PDFColor r: 0 g: 255 b: 0)
      ).

   pdfdoc add: aPage.

pdfdoc exportTo: aStream
```

# IV. CONCLUSION