

PROJET AVL

I. Introduction

II. Structure du projet

III. Code coverage &
Mutation testing

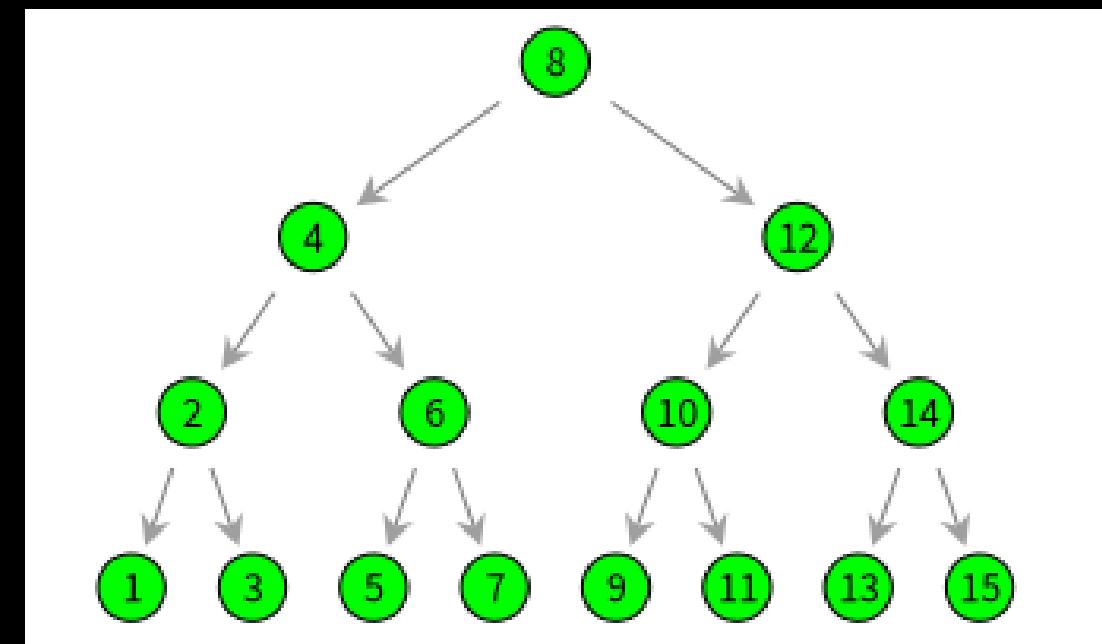
IV. Conclusion

Besbas Mélik
Leulmi Mohamed

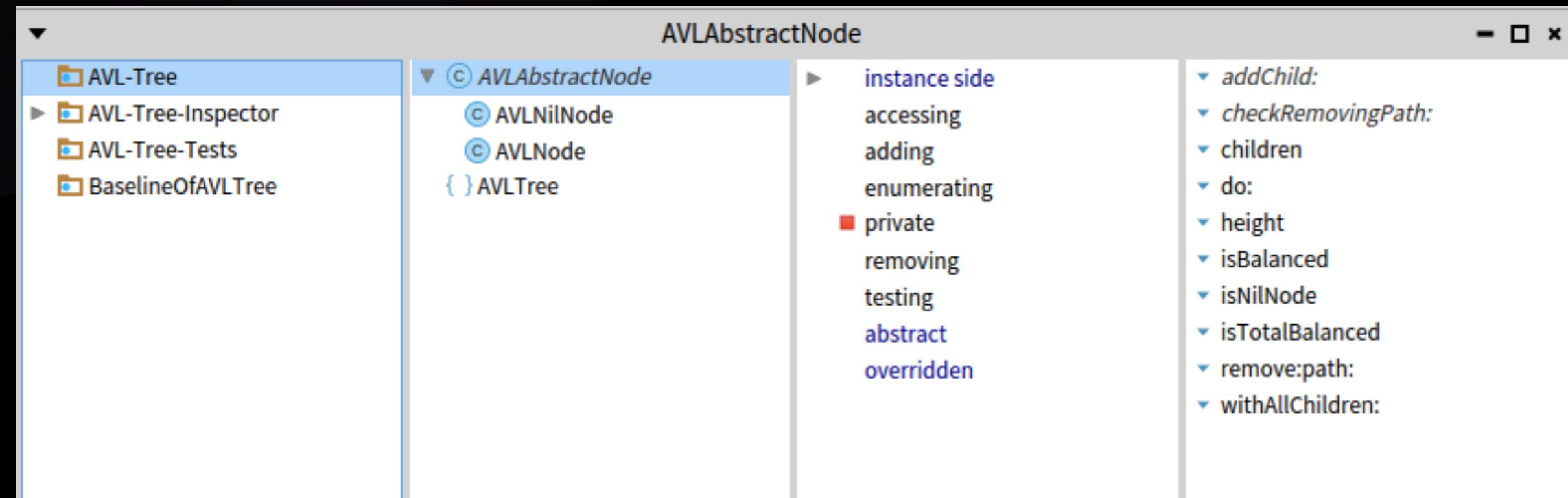
I. INTRODUCTION

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.
- BaselineOfAVLTree : un package de configuration et de gestion des dépendances.

II. STRUCTURE DU PROJET

Implémentation du code

Classe abstraite

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

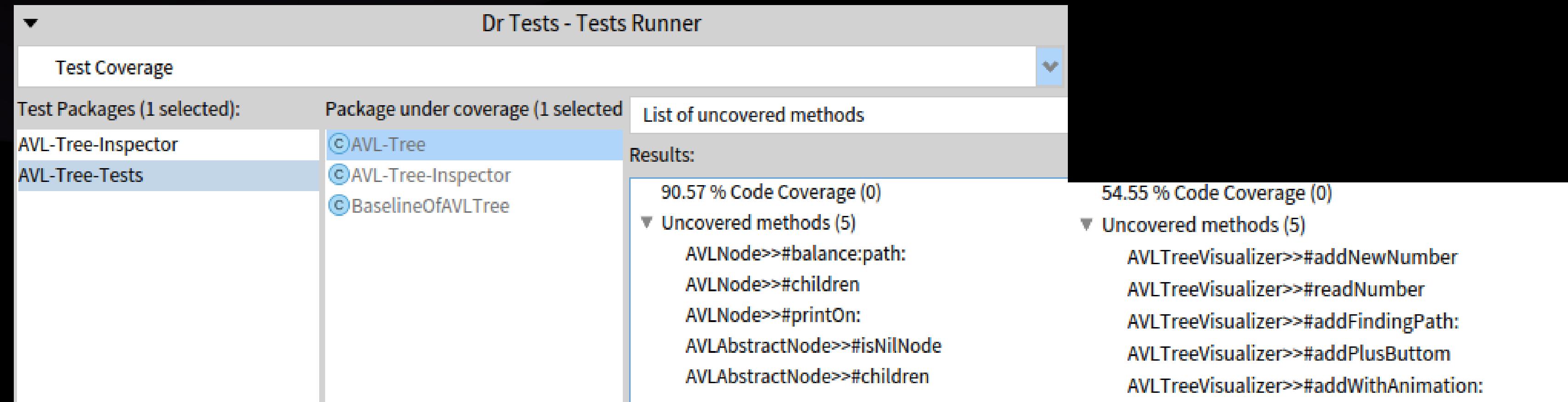
Classe représentant un noeud vide

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

Classe représentant un noeud non vide

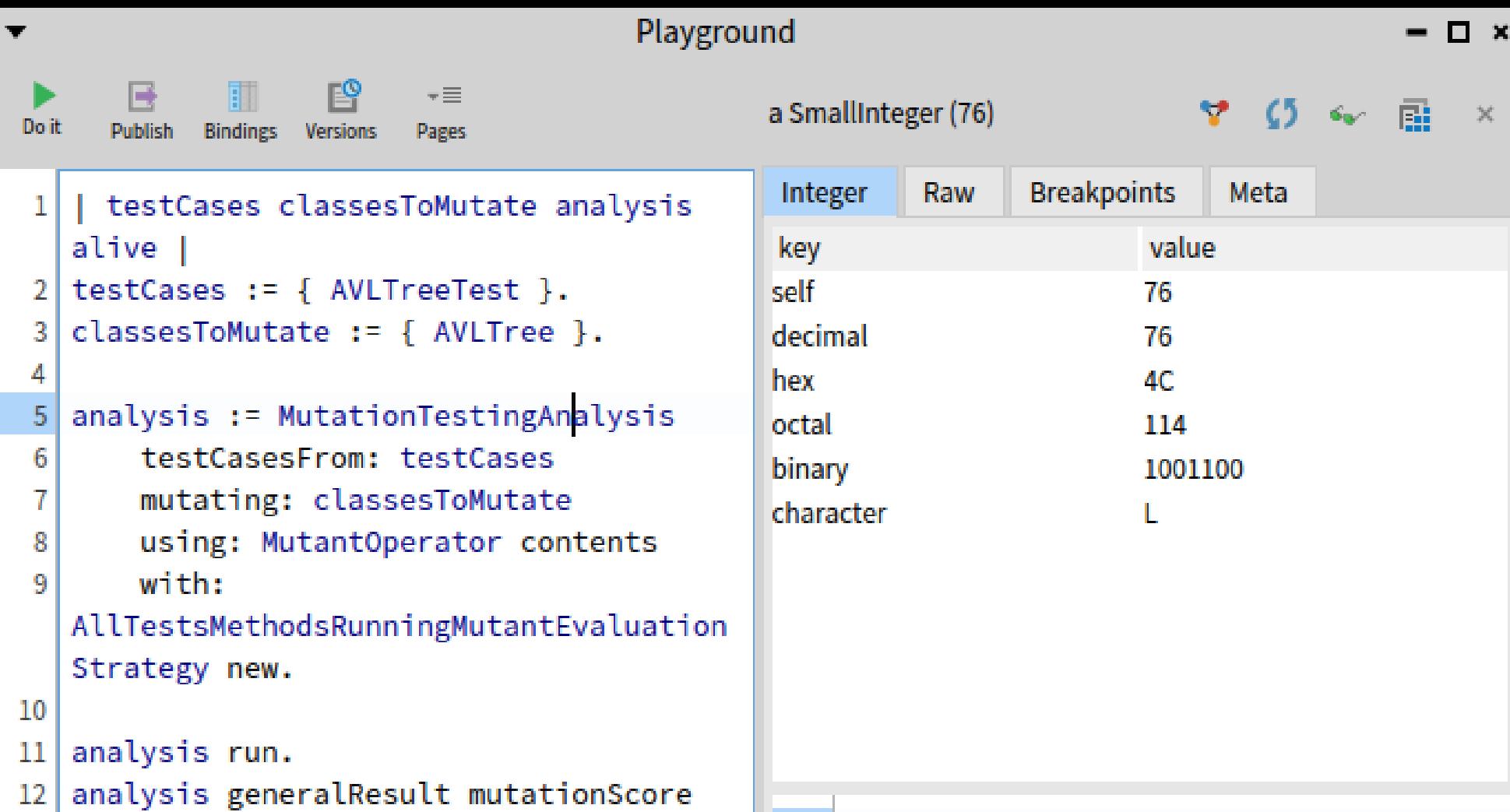
```
AVLAbstractNode subclass: #AVLNode  
instanceVariableNames: 'left contents right'  
classVariableNames: ''  
package: 'AVL-Tree'
```

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



The screenshot shows a 'Playground' window with a toolbar at the top featuring 'Do it', 'Publish', 'Bindings', 'Versions', and 'Pages' buttons. The main area has a title 'a SmallInteger (76)' above a table. The table has columns 'Integer', 'Raw', 'Breakpoints', and 'Meta'. It contains the following data:

key	value
self	76
decimal	76
hex	4C
octal	114
binary	1001100
character	L

The code editor on the left contains the following code:

```
1 | testCases classesToMutate analysis
  alive |
2 testCases := { AVLTreeTest }.
3 classesToMutate := { AVLTree }.
4
5 analysis := MutationTestingAnalysis
  testCasesFrom: testCases
  mutating: classesToMutate
  using: MutantOperator contents
  with:
    AllTestsMethodsRunningMutantEvaluation
    Strategy new.
10
11 analysis run.
12 analysis generalResult mutationScore
```

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

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 [...])
Σ 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
- Mais le code devient non fonctionnel

IV. CONCLUSION

- Pas de Design Pattern évident
- Coverage de test améliorable
- Très peu de documentation
- Utilisation du reverse engineering intéressante

ARTEFACT

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

I.MUTATION TESTING

I. Mutation Testing

Dr Tests - Tests Runner

Test Coverage

Minimize

Test Packages (1 selected): Artefact-Core-Tests

Package under coverage (1 selected)

- AI-Algorithms-Graph
- AI-Algorithms-Graph-Components
- AI-Algorithms-Graph-Tests
- AST-Core
- AST-Core-Tests
- AST-Core-Traits
- Announcements-Core
- Announcements-Core-Tests
- Artefact-Core
- Artefact-Core-Tests
- Artefact-Examples
- Athens-Balloon
- Athens-Cairo

List of uncovered methods

Results:

- 60.00 % Code Coverage (0)
- Uncovered methods (2)
- Partially covered methods (0)

I. Mutation Testing

Playground

a Duration (0:00:18:58.769)

```
1 testCases := {PDFBasicTest . PDFColorTest . PDFDataTypeTest  
 .PDFDemosTest .PDFElementTest . PDFFontTest . PDFGeneratorTest  
 .PDFHorizontalLayoutTest .PDFParagraphTest .PDFStreamPrinterTest } .|  
2 classesToMutate := 'Artefact-Core'asPackage definedClasses.  
3  
4 analysis := MutationTestingAnalysis  
5   testCasesFrom: testCases  
6   mutating: classesToMutate  
7   using: MutantOperator contents  
8   with: AllTestsMethodsRunningMutantEvaluationStrategy new.  
9  
10 analysis run.  
11 testSelection := [analysis run.] timeToRun. "0:00:01:19.115"
```

Details Raw Breakpoints Meta

key	value
self	0:00:18:58.769
human readable	18 minutes 58 seconds 769 milliseconds
days	0
hours	0
minutes	18
seconds	58
nanoseconds	769000000

1 self

+L

Line: 1:26

I. Mutation Testing

Playground

a SmallInteger (31)

Do it Publish Bindings Versions Pages

```
1 analysis generalResult mutationScore
```

	Integer	Raw	Breakpoints	Meta
key				value
self				31
decimal				31
hex				1F
octal				37
binary				11111
character				

I. Mutation Testing

Playground

an OrderedCollection [826 i...]

Items Raw Breakpoints Meta

: In : Value

- 1 Remove ^ in PDFJpegElement class>>#fromMorph:
- 2 Replace #+ with #- in PDFBezierCurveElement>>#producePageElementCodeWith:styleSheet:
- 3 Replace do block with [:each |] in PDFBezierCurveElement>>#producePageElementCodeWith:styleSheet:
- 4 Remove ^ in PDFPage>>#defaultFormat
- 5 Remove ^ in PDFPage>>#margins
- 6 Remove ^ in PDFPage class>>#elements: XXXXXXXXXX
- 7 Remove ^ in PDFPage class>>#element:
- 8 Remove ^ in PDFDataXRef>>#acceptVisitor:
- 9 Remove ^ in PDFOpacityLuminosity>>#blendMode
- 10 Remove ^ in PDFDataStream>>#acceptVisitor:
- 11 Remove ^ in PDFOpacityColorDodge>>#blendMode
- 12 Remove ^ in PDFDataAssociativeArray>>#acceptVisitor:
- 13 Remove ^ in PDFFbookFormat>>#defaultSize
- 14 Remove ^ in PDFCodeSegment>>#isSecure
- 15 Remove ^ in PDFCodeSegment>>#printWith:
- 16 Remove ^ in PDFCodeSegment class>>#isAbstract
- 17 Replace a == b with (a == b) not in PDFCodeSegment class>>#isAbstract

1 self

Do it Publish Bindings Versions Pages

I. Mutation Testing

The screenshot shows a mutation testing interface with a toolbar at the top and a main window divided into two panes.

Toolbar: Contains tabs for "Preference", "Resources", "Run", "Breakpoints", and "Profile".

Results Pane: A list of 18 mutation operators, each numbered from 1 to 18. The first 17 are highlighted in blue, while the last one is partially visible in grey.

- 1 Remove ^ in PDFDotted>>#generateCodeWith:
- 2 Remove ^ in PDFDotted>>#space
- 3 Remove ^ in PDFDotted>>#length
- 4 Remove ^ in PDFA9Format>>#defaultSize
- 5 Remove ^ in PDFDataSymbol>>#isPrintable
- 6 Remove ^ in PDFDataSymbol>>#acceptVisitor:
- 7 Remove ^ in PDFDataSymbol>>#symbol
- 8 Remove ^ in PDFDataSymbol class>>#symbol:
- 9 Remove ^ in PDFDataDateAndTime>>#isPrintable
- 10 Remove ^ in PDFDataDateAndTime>>#time
- 11 Remove ^ in PDFDataDateAndTime>>#acceptVisitor:
- 12 Remove ^ in PDFDataDateAndTime>>#date
- 13 Remove ^ in PDFDataDateAndTime>>#formatDate:time:
- 14 Replace #< with #> in PDFDataDateAndTime>>#formatDate:time:
- 15 Replace #< with #> in PDFDataDateAndTime>>#formatDate:time:
- 16 Replace #< with #> in PDFDataDateAndTime>>#formatDate:time:
- 17 Replace #ifTrue: receiver with false in PDFDataDateAndTime>>#formatDate:time:
- 18 Replace #ifTrue: receiver with false in PDFDataDateAndTime>>#formatDate:time:

Code Editor: Displays two side-by-side code snippets. Both snippets contain the word "length" in blue. In the left snippet, the character '^' is highlighted in red and positioned above the word "length". In the right snippet, the character '^' is also present but is not highlighted.

```
length  
^ length
```

```
length
```

I. Mutation Testing

```
28 Remove ^ in PDFUnsecureElementCodeSegment>>#isSecure  
29 Remove ^ in PDFSymbolFont>>#charWidths  
30 Remove ^ in PDFSymbolFont>>#family  
31 Remove ^ in PDFSymbolFont>>#fontName  
32 Remove ^ in ArtefactOverSizedContent>>#content  
33 Remove ^ in PDFLayout>>#createOriginalPositionDictionary  
34 Replace do block with branch if in PDFLayout>>#createOriginalPositionDictionary
```

The screenshot shows a mutation testing interface with two columns of code. The left column contains the original code, and the right column contains a mutated version where the condition in the ifTrue block of the first if statement has been changed.

```
hour := aTime hours.  
str := hour asString.  
hour < 12 ifTrue: [ str := '0' , str ].  
timeStr := str.  
  
minutes := aTime minutes.  
str := minutes asString.  
minutes < 10 ifTrue: [ str := '0' , str ].  
timeStr := timeStr , str.  
  
seconds := aTime seconds.  
str := seconds asString.  
seconds < 10 ifTrue: [ str := '0' , str ].  
timeStr := timeStr , str.  
  
^ (aDate yyyyymmdd copyWithRegex: '-' matchesReplacedWith:  
    , timeStr
```

hour := aTime hours.
str := hour asString.
hour < 12 ifTrue: [str := '0' , str].
timeStr := str.

minutes := aTime minutes.
str := minutes asString.
true ifTrue: [str := '0' , str].
timeStr := timeStr , str.

seconds := aTime seconds.
str := seconds asString.
seconds < 10 ifTrue: [str := '0' , str].
timeStr := timeStr , str.

^ (aDate yyyyymmdd copyWithRegex: '-' matchesReplacedWith:
 , timeStr

I. Mutation Testing

The screenshot shows a Java IDE interface with the following details:

- Title Bar:** PDFColorTest>>testEqualDifferents
- Left Panel (Artefact View):** Shows project structure with Artefact-Core, Artefact-Core-Tests (selected), Artefact-Examples, and BaselineOfArtefact.
- Middle Panel (Method List):** Shows a list of tests under ManifestArtefactCoreTest:
 - PDFBasicTest
 - PDFColorTest (selected)
 - PDFDataTypeTest
 - PDFDemosTest
 - PDFDummyBasic
 - PDFDummyLayout
 - PDFElementTest
 - PDFFontTest
 - PDFGeneratorTest
- Right Panel (Mutation Results):** Shows mutation results for testEqualDifferents:
 - testEqual (highlighted)
 - testEqualDifferents
- Bottom Navigation:** Includes tabs for All Packages, Scoped View, Flat, Hier., Inst. side (selected), Class side, Methods, Vars, Class refs, Implementations, and several search/filter buttons.
- Bottom Content:** Displays the code for testEqualDifferents:

```
self deny: (PDFColor r: 54 g: 98 b: 12) equals: (PDFColor r: 12 g: 48 b: 79)
```

II.DESIGN PATTERN

Factory

PDFA0Format>>defaultSize

The screenshot shows a Java IDE interface with the following details:

- Project Tree:** On the left, under the package `artefact`, are the projects: `Artefact-Core`, `Artefact-Core-Tests`, `Artefact-Examples`, `Artefact-Tutorial`, and `BaselineOfArtefact`.
- Search Bar:** The search bar at the top contains the text `defaultSize`. Below it are buttons for `All Packages`, `Scoped View`, `Flat`, `Hier.`, `Inst. side` (which is selected), `Class side`, `Methods`, `Vars`, `Class refs.`, `Implementors`, and `Senders`.
- Search Results:** The results pane shows the class `PDFA0Format` highlighted in blue. Other classes listed include `PDFHelveticaFont`, `PDFSymbolFont`, `PDFTimesFont`, `PDFZapfdingbatsFont`, `PDFFormat`, `PDFA10Format`, `PDFA1Format`, `PDFA2Format`, `PDFA3Format`, `PDFA4Format`, `PDFA5Format`, `PDFA6Format`, `PDFA7Format`, and `PDFA8Format`.
- Annotations:** Annotations on the right side of the results pane include `instance side`, `accessing - defaults`, and `overrides`.
- Bottom Navigation:** The bottom navigation bar includes tabs for `Dependencies`, `PDFA0Format`, `Comment`, `defaultSize` (which is selected and highlighted in blue), and `Inst. side methc`.
- Result Preview:** The preview pane at the bottom shows the `defaultSize` method with the signature `^ 2384.03 point @ 3370.53 point`.

DoubleDispatch

PDFFElementCodeSegment>>printWith:

The screenshot shows a Java documentation search interface. On the left, there's a sidebar with project navigation. The main area displays the class hierarchy for `PDFFElementCodeSegment`. The `printWith:` method is selected, and its implementation is shown in the bottom pane. The search bar at the top contains the query `PDFFElementCodeSegment>>printWith:`.

Artefact-Core

Artefact-Core-Tests

Artefact-Examples

Artefact-Tutorial

BaselineOfArtefact

Artefact

ArtefactOverSizedContent

ArtefactUndefinedAttribute

ManifestArtefactCore

PDFAlignment

PDFAngleDirected

PDFByteCode

PDFFCodeSegment

PDFCompositeCodeSegment

PDFFElementCodeSegment

PDFUnsecureElementCodeSegment

PDFColor

PDFDataType

PDFdataArray

PDFDataAssociativeArray

PDFDataComment

PDFDataString

Filter...

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

Dependencies PDFFElementCodeSegment Comment printWith: Inst. side methods

printWith: aPDFWriter
aPDFWriter printElementCodeSegment: self

instance side ▲□
accessing
overrides

code
code:
fontId
fontId:
format
format:
opacityId
opacityId:
printWith:

PDFDataComment>>acceptVisitor:

The diagram shows a UML class hierarchy. At the top level, there is a class named 'Artefact-Core'. Below it, under the package 'Artefact-Core', is a class 'PDFDataComment'. This class has several subclasses listed below it: 'PDFCompositeCodeSegment', 'PDFElementCodeSegment', 'PDFUnsecureElementCodeSegment', 'PDFColor', 'PDFDataType', 'PDFdataArray', 'PDFDataAssociativeArray', and 'PDFDataComment' (which is highlighted with a blue selection bar). Further down the hierarchy are 'PDFDataCouple', 'PDFDataDateAndTime', 'PDFDataObject', 'PDFDataReference', 'PDFDataStartXref', 'PDFDataStream', and 'PDFDataStreamRefSize'. To the right of the class list, there is a tree view labeled 'instance side' with nodes: 'accessing', 'printing', 'testing', 'visiting', and 'overrides'. On the far right, there is a vertical list of methods: 'acceptVisitor:', 'comment', 'comment:', 'isPrintable', and 'printOn:'. At the bottom of the interface, there are several tabs: 'Dependencies', 'PDFDataComment', 'Comment', 'acceptVisitor:' (which is selected and highlighted in blue), and 'Inst. side methc'. The text area at the bottom contains the code for the 'acceptVisitor:' method.

Artefact-Core

PDFCompositeCodeSegment

PDFElementCodeSegment

PDFUnsecureElementCodeSegment

PDFColor

PDFDataType

PDFdataArray

PDFDataAssociativeArray

PDFDataComment

PDFDataCouple

PDFDataDateAndTime

PDFDataObject

PDFDataReference

PDFDataStartXref

PDFDataStream

PDFDataStreamRefSize

PDFDataComment

instance side

accessing

printing

testing

visiting

overrides

acceptVisitor:

comment

comment:

isPrintable

printOn:

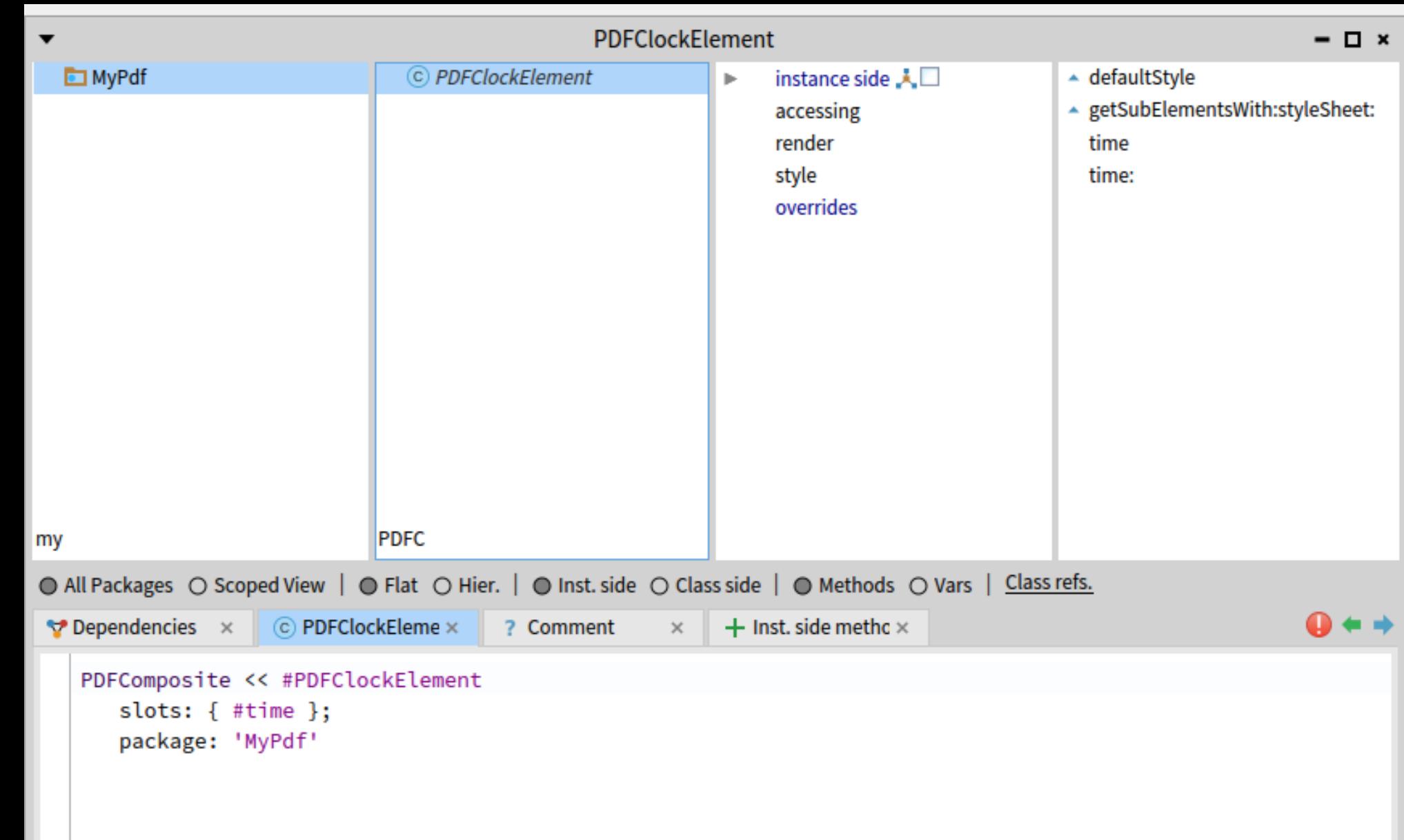
All Packages Scoped View Flat Hier. Inst. side Class side Methods Vars Class refs. Implementors Senders

Dependencies PDFDataComment Comment acceptVisitor: Inst. side methc

acceptVisitor: aVisitor

 aVisitor visitPDFDataComment: self

Visitor



III. EST CE FACILEMENT UTILISABLE ?

PDFClockElement>>getSubElementsWith:styleSheet:

MyPdf

PDFClockElement

instance side A □

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. 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

- Ne pas plonger tête baissée dans le code et savoir cibler les portions de code importantes
- Les mutations testing et le coverage donnent des indications importantes
- Ne pas se contenter d'observer mais aussi DEDUIRE
- Un coverage élevé n'est pas synonyme de qualité de test