Présentation C3P

...

- AVL
- Artefact

Sommaire

- Signification et utilisation
- Conception et implémentation
- Les Tests
- Méthode d’apprentissage

1. Signification et utilisation

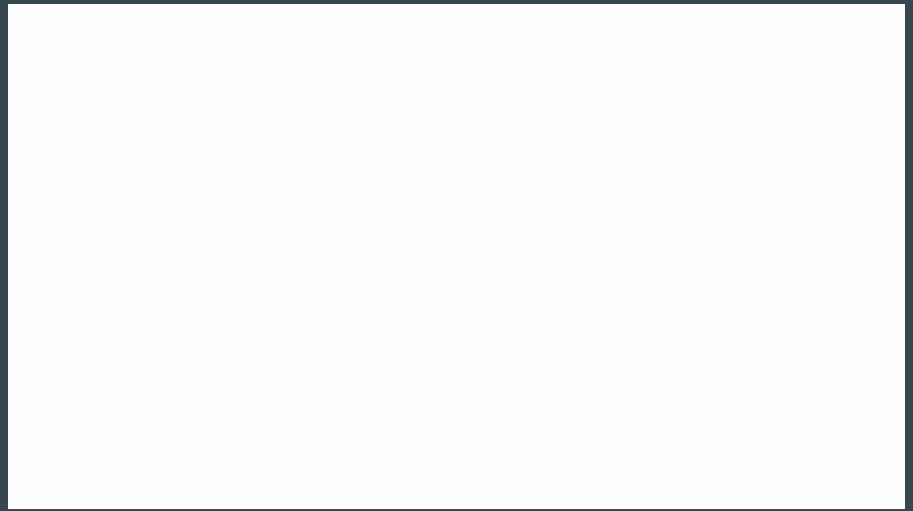
AVL tree

(Adelson-Velsky et Landis) est un arbre de recherche binaire auto-équilibré

- Les hauteurs des deux sous-arbres d'un même nœud diffèrent au plus de un

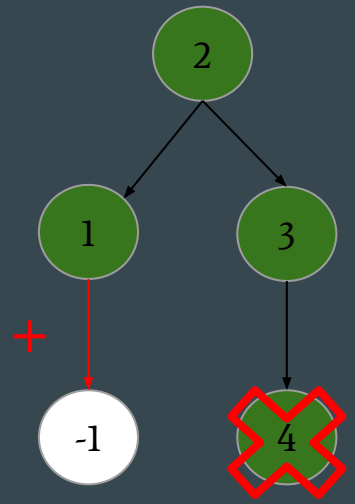
Action sur l’arbre : O(log n)

- n = nb noeuds



Utilisation AVL tree

```
tree1 := AVLTree new.
tree1 addAll: { 1. 2. 3. 4. -1 }.
tree1.
```



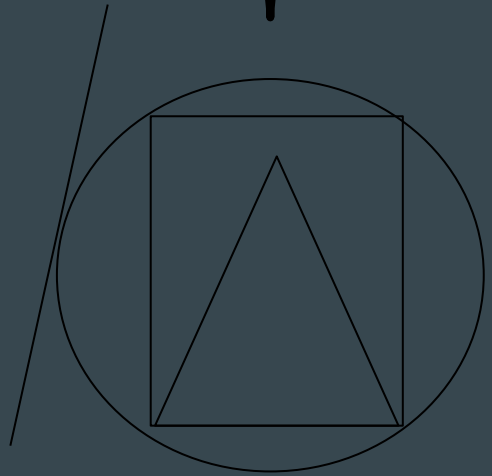
Height

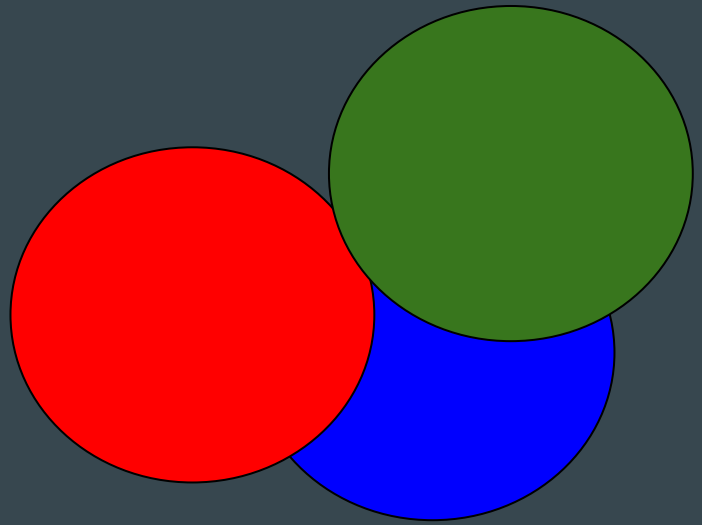
3 ? OUI

8 ? NON

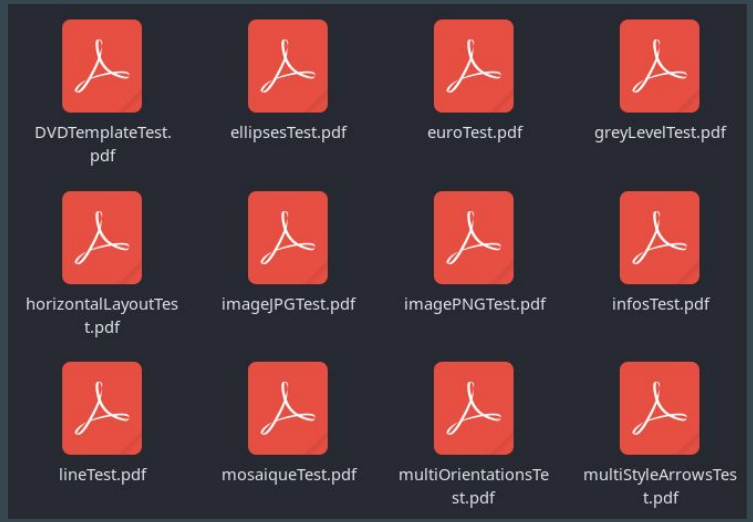
```
tree1 includes:5.   true
```

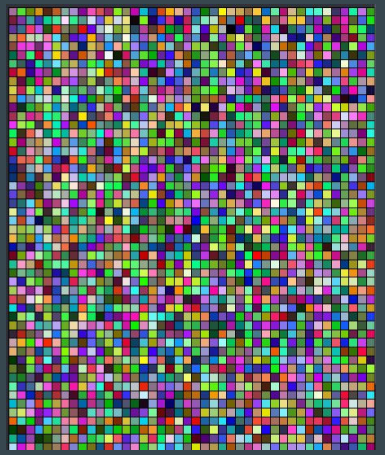
Artefact

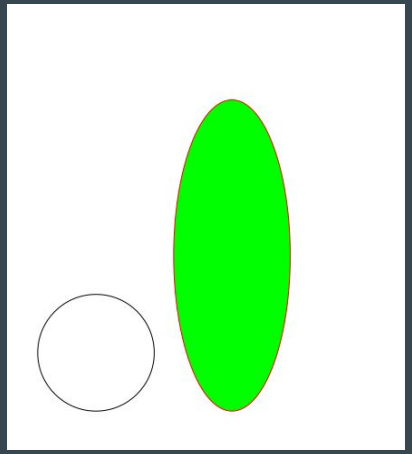


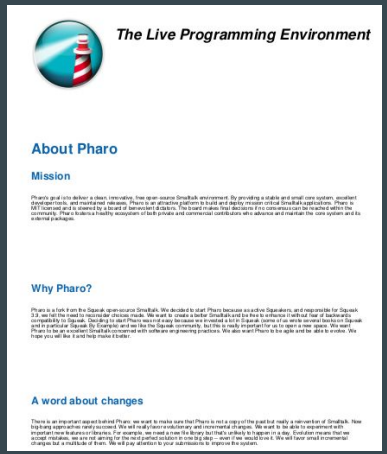


Exemple Artefact





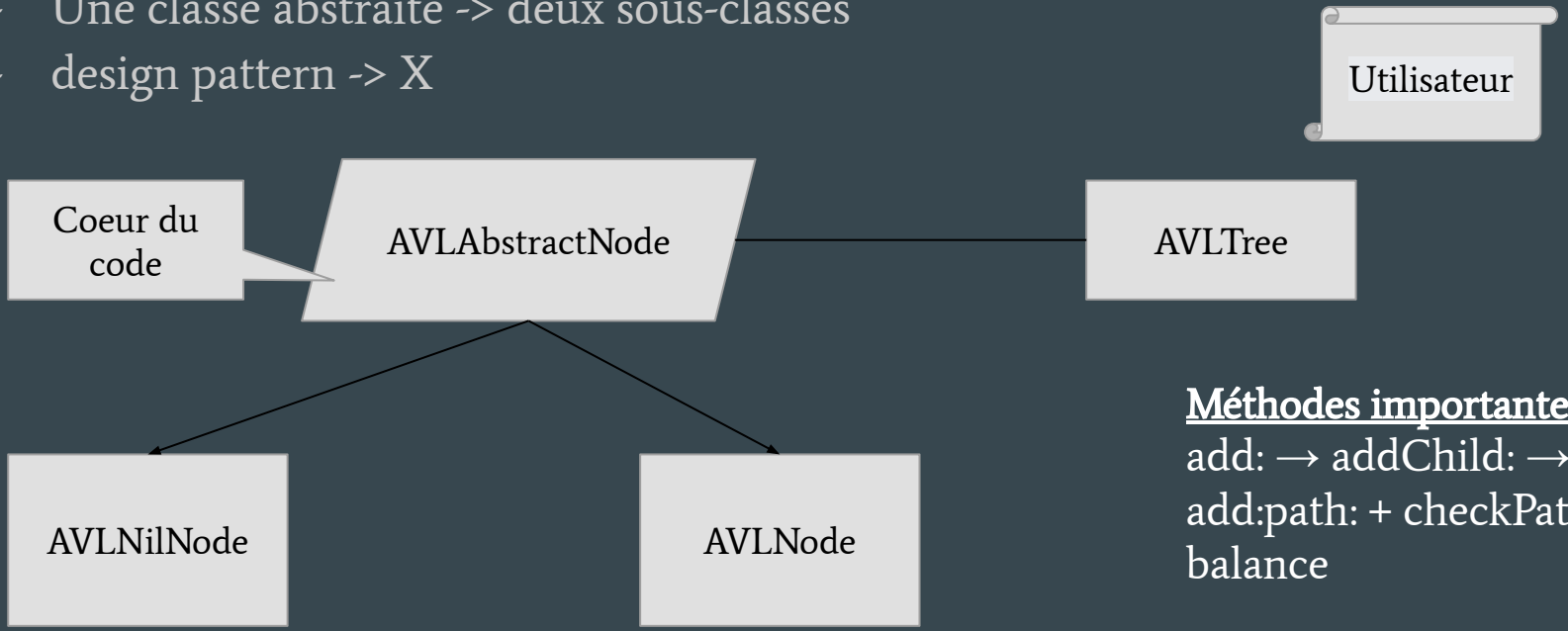




2. Conception et implémentation

AVL tree

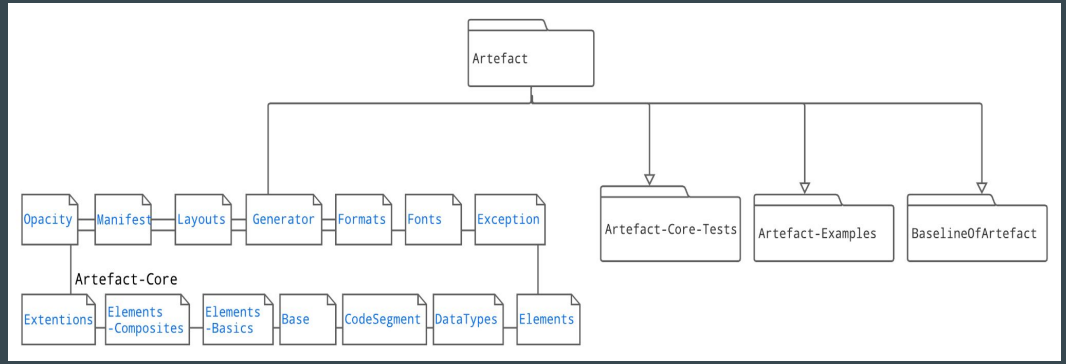
- Package principal : AVL-Tree
- AVLTree -> Utilisateur
- Une classe abstraite -> deux sous-classes
- design pattern -> X



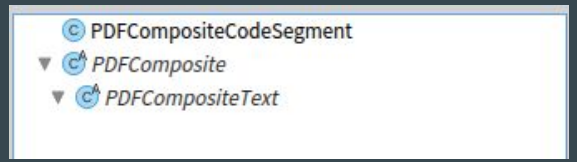
Méthodes importantes:
add: → addChild: →
add:path: + checkPath: →
balance

Artefact

- Package principal : Artefact-Core
- Projet complexe -> nb package
- package -> ++ classes
- classe = 1 utilité



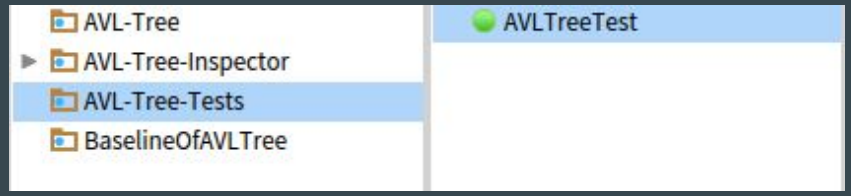
- Utilisation de design patterns (exemples = composite,visitor)



3. Les Tests

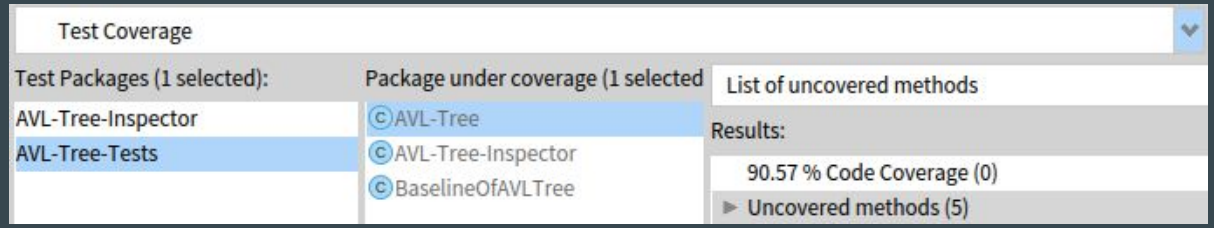
AVL tree

- Package : AVL-Tree-Tests -> deux classe



Test-coverage :

- AVL-Tree-Inspector 45.45%
- AVL-Tree-Tests 90,57%
- 5 méthodes non testés



Mutation score :

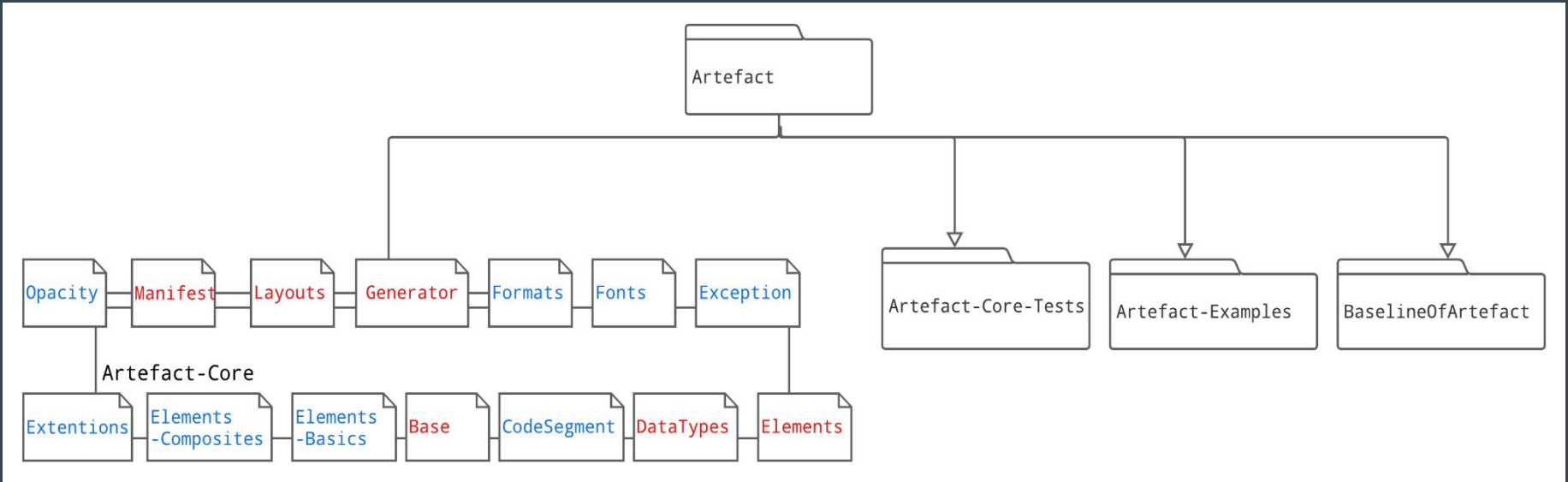
- AVLNilNode : 78%
- AVLTree : 76%
- AVLNode : 63%

Exemple :

```
testCases :=  { AVLTreeTest  }.
classesToMutate := { AVLTree  . AVLNode   }.

analysis := MutationTestingAnalysis
    testCasesFrom: testCases
    mutating: classesToMutate
    using: MutantOperator contents
    with: AllTestsMethodsRunningMutantEvaluationStrategy new.

analysis run.

"To retrieve the alive mutations"
analysis generalResult mutationScore  63
```

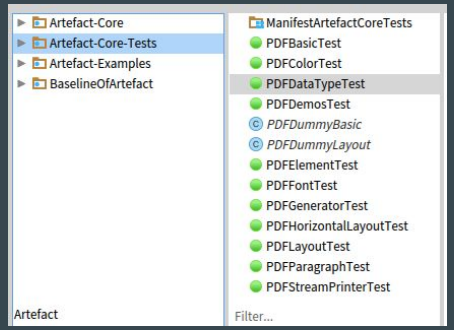
Artefact

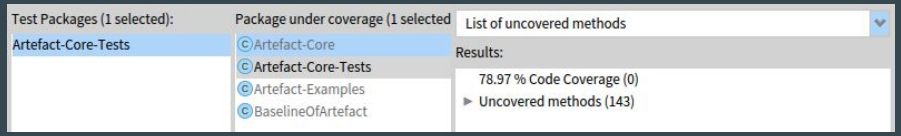
Les packages en rouge sont testées



Artefact

- Package : Artefact–Core-Tests -> + classes
- Test-coverage : 78,97%
- 143 méthodes non testés





- Mutation score : Beaucoup de classe à tester

exemple :

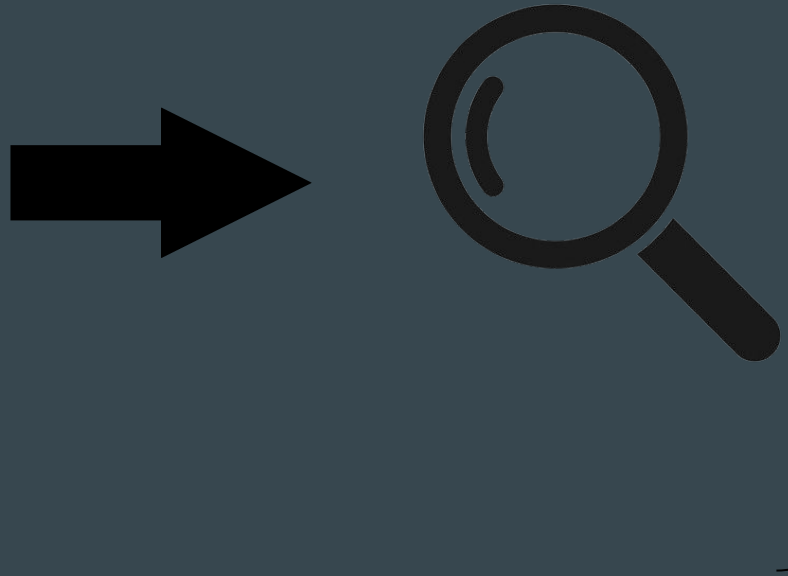
```
testCases :=  { PDFDataTypeTest   }.
classesToMutate := { PDFDataType    . PDFDataComment      }.

analysis := MutationTestingAnalysis
    testCasesFrom: testCases
    mutating: classesToMutate
    using: MutantOperator contents
    with: AllTestsMethodsRunningMutantEvaluationStrategy new.

analysis run.
```

```
"To retrieve the alive mutations"
analysis generalResult mutationScore  12
```

4. Méthodes d’apprentissage

Benjamin

Projet en générale



- Codes
- Tests
- Etc ...

Yanis

- Flux : Vue haut niveau => Utilisation (User perspective API) => Mise en œuvre
- Ignorer les éléments qui n'étaient pas dans notre champ de concentration
- Rechercher les termes nouveaux
- Avoir une vue d’ensemble
- Faire des schéma

Merci !

des questions...