

**CFW(X) avant suppression**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | TOTAL |
| 1 |  | x | x | x | x | x | x | x | x | x | 9 |
| 2 |  |  | x | x | x | x | x |  | x |  | 6 |
| 3 |  |  |  | x | x |  |  |  |  |  | 2 |
| 4 |  |  |  |  |  |  |  |  |  |  | 0 |
| 5 |  |  |  |  |  |  |  |  |  |  | 0 |
| 6 |  |  | x | x | x |  | x |  | x |  | 5 |
| 7 |  | x | x | x | x | x |  |  | x |  | 6 |
| 8 |  | x | x | x | x | x | x |  | x |  | 7 |
| 9 |  |  |  |  |  |  |  |  |  |  | 0 |
| 10 |  |  |  |  |  |  |  |  |  |  | 0 |

**DOIT ON SUPPRIMER DES RELATIONS ?** oui car on a une boucle entre 2 et 7, il faut supprimer dans l’ordre As / Ag / I… donc la seule relation As est visitDepartment entre 2 et 6 permet de supprimer cette boucle.

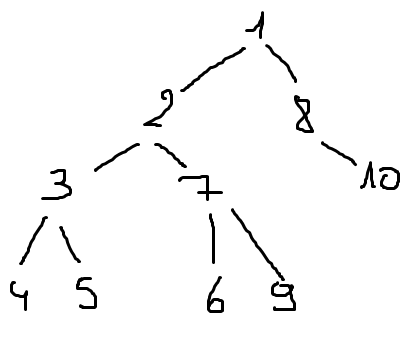
**CFW(X) après suppression**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | TOTAL |
| 1 |  | x | x | x | x | x | x | x | x | x | 9 |
| 2 |  |  | x | x | x |  | x |  | x |  | 5 |
| 3 |  |  |  | x | x |  |  |  |  |  | 2 |
| 4 |  |  |  |  |  |  |  |  |  |  | 0 |
| 5 |  |  |  |  |  |  |  |  |  |  | 0 |
| 6 |  |  |  |  |  |  |  |  |  |  | 0 |
| 7 |  |  |  |  |  | x |  |  | x |  | 2 |
| 8 |  |  |  |  |  | x |  |  | x | x | 3 |
| 9 |  |  |  |  |  |  |  |  |  |  | 0 |
| 10 |  |  |  |  |  |  |  |  |  |  | 0 |

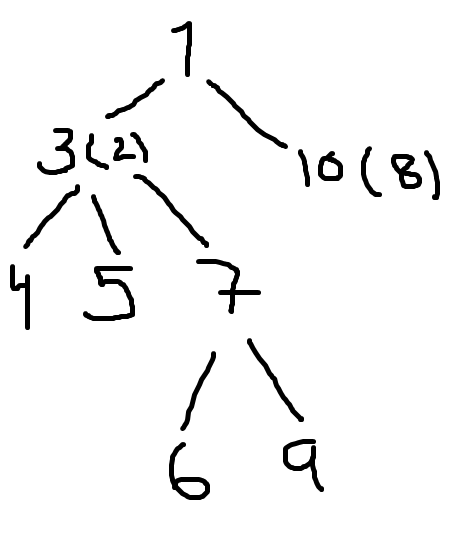
|  |  |
| --- | --- |
| NIVEAU DE TEST | CLASSES |
| 1 | 1 |
| 2 | 2/8 |
| 3 | 3/7/10 |
| 4 | 4/5/6/9 |
| 5 | 6\*, 7\* |

6\* : Il faut tester l’interface polymorphique de 6 avec 8 et ses enfants (9 et 10).  
6/(9,10)  
6/7/(3,4,5)

7\* : Il faut tester l’interface polymorphique avec 2 et ses enfants   
7/(3,4,5)



Supposition que 8 et 2 devient abstrait :



**PARTIE 2**



|  |  |
| --- | --- |
| méthodes | Séquences |
| Cm1 | (cm1, sm1, pS)  (cm1, sm0, as1)  (cm1, sm2, as2) |
| Cm3 | (cm3, sm2.sm1,anS)  (cm3, sm2,anS)  **(cm3, sm3.sm0,as1)**  **(cm3, sm0.sm3,as1)**  (cm3, sm2,as2)  (cm3, sm2.sm1.sm1,as2) |
| Cm1.cm3 | (cm1.cm3, sm1.sm1, pS)  **(cm1.cm3, sm0. sm3.sm0, as1)**  **(cm1.cm3, sm0. sm0. sm3, as1)**  (cm1.cm3, sm2. sm2, as2)  (cm1.cm3, sm2. sm2.sm1.sm1, as2)  **(cm3, sm2.sm1,anS)**  **(cm3, sm2,anS)** |

**Criteres :**

**1 - pas de distinction entre les serveurs attributs / paramètre =** retenir une séquence si possible

**2 – distinction entre les serveurs attributs / paramètre =** retenir une séquence pour att / para

**3 – distinction entre tous =** retenir autant de séquence que de serveurs

Critère 1 :

|  |  |
| --- | --- |
| Cm1.cm3 | ~~(cm1.cm3, sm1.sm1, pS~~)  (cm1.cm3, sm0. sm3.sm0, as1)  (cm1.cm3, sm0. sm0. sm3, as1)  ~~(cm1.cm3, sm2. sm2, as2)~~  (cm1.cm3, sm2. sm2.sm1.sm1, as2)  ~~(cm3, sm2.sm1,anS)~~  ~~(cm3, sm2,anS)~~ |

Critère 2 :

|  |  |
| --- | --- |
| Cm1.cm3 | (cm1.cm3, sm1.sm1, pS)  (cm1.cm3, sm0. sm3.sm0, as1)  (cm1.cm3, sm0. sm0. sm3, as1)  ~~(cm1.cm3, sm2. sm2, as2)~~  (cm1.cm3, sm2. sm2.sm1.sm1, as2)  (cm3, sm2.sm1,anS)  ~~(cm3, sm2,anS)~~ |

Critère 3 :

|  |  |
| --- | --- |
| Cm1.cm3 | (cm1.cm3, sm1.sm1, pS)  (cm1.cm3, sm0. sm3.sm0, as1)  (cm1.cm3, sm0. sm0. sm3, as1)  ~~(cm1.cm3, sm2. sm2, as2)~~  (cm1.cm3, sm2. sm2.sm1.sm1, as2)  (cm3, sm2.sm1,anS)  ~~(cm3, sm2,anS)~~ |