**INF3430 – Méthodes de test et de validation du logiciel**

TP No. 2

Groupe 1 :

1437737 – Maxime Jacob

1539437 – Dan Vatnik

1776903 – Cédrick Busque

Présenté à :

Mlouki, Ons

Polytechnique Montréal

13 février 2017

1. **Exigences du TP No.1**

Le module a été codé en java. Il était également développé entre 2 et 5 classes. Il devait y avoir des interfaces. Les opérateurs +, -, \* et / ne devaient pas être exprimé de façon usuelle.

Nous devions également fournir à l’utilisateur de méthodes de gestion de la pile tel qu’ajouter un élément a la pile(empiler(item)), retirer un élément de la pile(dépiler()), retourner la tête de la pile(getHead()), retourne un booléen indiquant si la pile est vide ou non (estVide()), retourne la taille de la pile(getSize()), et remet la pile à vide(reset())[[1]](#endnote-1).

1. **Conception de test boite noire**

Les classes d’équivalences valides :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Classes d'équivalences** | | | | |
| **Équation** | Add(+) | Soust(-) | mult(\*) | Div(/) |
| **nPair** | Nombre pair  entre  INT\_MIN et 0 exclusivement (p[INT\_MIN – 0[) | 0 | Nombre pair  entre  0 exclusivement et INT\_MAX  (p ]0-INT\_MAX]) |  |
| **nImpair** | Nombre impair  entre  INT\_MIN et 0 exclusivement  (imp[INT\_MIN – 0[) | Nombre impair  entre  0 exclusivement et INT\_MAX  (imp ]0-INT\_MAX]) |  |  |
| **Longueur** | [2-10] |  |  |  |

Suivant ces classes d’équivalence nous avons conçus des tests suivant la partition EC (Each Choice) ainsi que des tests suivants le critère AC(all combinations).

**Tests EC**

La technique de test EC comporte un nombre de test égale au nombre de classe d’équivalence dans la variable en ayant le plus. Dans notre cas, cette variable est équation qui en contient 4.

Les trame tests sont :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Équation** | **Npair** | **Nimpare** | **Longueur** | **Resultat** |
| 1 | Operateur1 | p[INT\_Min – 0[ | imp[INT\_MIN – 0[ | [2-10] | Ok |
| 2 | Operateur2 | 0 | imp[INT\_MIN – 0[ | [2-10] | Ok |
| 3 | Operateur3 | p ]0-INT\_MAX] | imp ]0-INT\_MAX] | [2-10] | Ok |
| 4 | Operateur4 | p[INT\_Min – 0[ | imp ]0-INT\_MAX] | [2-10] | Ok |

Les tests sont :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Équation** | **Npair** | **Nimpare** | **Longueur** | **Resultat** |
| 1 | Add | 2 | -1 | 4 | {2, -1, 1, 0} |
| 2 | Soust | 0 | 5 | 5 | {0, 5, -5, 10, -15} |
| 3 | Mult | -10 | 5 | 3 | {-10, 5, -50} |
| 4 | Div | 14 | 7 | 3 | {14, 7, 2} |

**Tests AC**

La technique de test AC contient toutes les combinaisons de toutes les classe d’équivalence avec toutes les variables. Dans notre cas, nous avons 24 combinaisons. Nous avons donc établi les tests suivant :

Les trames de tests sont :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Équation | Npair | Nimpare | Longueur | Resultat |
| 1 | Opération1 | p[INT\_Min – 0[ | imp[INT\_MIN – 0[ | [2-10] | Ok |
| 2 | Opération1 | p[INT\_Min – 0[ | imp ]0-INT\_MAX] | [2-10] | Ok |
| 3 | Opération1 | 0 | imp[INT\_MIN – 0[ | [2-10] | Ok |
| 4 | Opération1 | 0 | imp ]0-INT\_MAX] | [2-10] | Ok |
| 5 | Opération1 | p ]0-INT\_MAX] | imp[INT\_MIN – 0[ | [2-10] | Ok |
| 6 | Opération1 | p ]0-INT\_MAX] | imp ]0-INT\_MAX] | [2-10] | Ok |
| 7 | Opération2 | p[INT\_Min – 0[ | imp[INT\_MIN – 0[ | [2-10] | Ok |
| 8 | Opération2 | p[INT\_Min – 0[ | imp ]0-INT\_MAX] | [2-10] | Ok |
| 9 | Opération2 | 0 | imp[INT\_MIN – 0[ | [2-10] | Ok |
| 10 | Opération2 | 0 | imp ]0-INT\_MAX] | [2-10] | Ok |
| 11 | Opération2 | p ]0-INT\_MAX] | imp[INT\_MIN – 0[ | [2-10] | Ok |
| 12 | Opération2 | p ]0-INT\_MAX] | imp ]0-INT\_MAX] | [2-10] | Ok |
| 13 | Opération3 | p[INT\_Min – 0[ | imp[INT\_MIN – 0[ | [2-10] | Ok |
| 14 | Opération3 | p[INT\_Min – 0[ | imp ]0-INT\_MAX] | [2-10] | Ok |
| 15 | Opération3 | 0 | imp[INT\_MIN – 0[ | [2-10] | Ok |
| 16 | Opération3 | 0 | imp ]0-INT\_MAX] | [2-10] | Ok |
| 17 | Opération3 | p ]0-INT\_MAX] | imp[INT\_MIN – 0[ | [2-10] | Ok |
| 18 | Opération3 | p ]0-INT\_MAX] | imp ]0-INT\_MAX] | [2-10] | Ok |
| 19 | Opération4 | p[INT\_Min – 0[ | imp[INT\_MIN – 0[ | [2-10] | Ok |
| 20 | Opération4 | p[INT\_Min – 0[ | imp ]0-INT\_MAX] | [2-10] | Ok |
| 21 | Opération4 | 0 | imp[INT\_MIN – 0[ | [2-10] | Ok |
| 22 | Opération4 | 0 | imp ]0-INT\_MAX] | [2-10] | Ok |
| 23 | Opération4 | p ]0-INT\_MAX] | imp[INT\_MIN – 0[ | [2-10] | Ok |
| 24 | Opération4 | p ]0-INT\_MAX] | imp ]0-INT\_MAX] | [2-10] | Ok |

Les tests sont :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Équation** | **Npair** | **Nimpare** | **Longueur** | **Resultat** |
| 1 | Add | -2 | -3 | 2 | {-2, -3} |
| 2 | Add | -2 | 5 | 3 | {-2, 5, 3} |
| 3 | Add | 0 | -3 | 4 | {0, -3, -3, -6} |
| 4 | Add | 0 | 5 | 5 | {0, 5, 5, 10, 15} |
| 5 | Add | 4 | -3 | 2 | {4, -3} |
| 6 | Add | 4 | 5 | 3 | {4, 5, 9} |
| 7 | Soust | -2 | -3 | 4 | {-2, -3, 1, -4} |
| 8 | Soust | -2 | 5 | 5 | {-2, 5, -7, 12, -19} |
| 9 | Soust | 0 | -3 | 2 | {0, -3} |
| 10 | Soust | 0 | 5 | 3 | {0, 5, -5} |
| 11 | Soust | 4 | -3 | 4 | {4, -3, 7, -10} |
| 12 | Soust | 4 | 5 | 5 | {4, 5, -1, 6, -7} |
| 13 | Mult | -2 | -3 | 2 | {-2, -3} |
| 14 | Mult | -2 | 5 | 3 | {-2, 5, -10} |
| 15 | Mult | 0 | -3 | 4 | {0, -3, 0, 0} |
| 16 | Mult | 0 | 5 | 5 | {0, 5, 0, 0, 0} |
| 17 | Mult | 4 | -3 | 2 | {4, -3} |
| 18 | Mult | 4 | 5 | 3 | {4, 5, 20} |
| 19 | Div | -2 | -3 | 4 | {-2, -3, 0} |
| 20 | Div | -2 | 5 | 5 | {-2, 5, 0} |
| 21 | Div | 0 | -3 | 2 | {0, -3} |
| 22 | Div | 0 | 5 | 3 | {0, 5, 0} |
| 23 | Div | 4 | -3 | 4 | {4, -3, -1, 3} |
| 24 | Div | 4 | 5 | 5 | {4, 5, 0} |

1. Information pris directement sur https://moodle.polymtl.ca/pluginfile.php/372649/mod\_resource/content/2/TP1\_log3430%20%281%29.pdf [↑](#endnote-ref-1)