Rapport LOG3430 TP3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | first | last | n | Full | Empty |
| \_\_init\_\_ | C | C | C | C | C |
| hasOne |  |  | R |  |  |
| isEmpty |  |  |  |  | R |
| isFull |  |  |  | R |  |
| size |  |  | R |  |  |
| checkFirst | R |  |  |  | O |
| checkLast |  | R | O |  | O |
| enqueue | T | T | T | T | T |
| dequeue | T | T | T | T | T |

Partie 5

1. La queue ne pourrait accepter plus de 2 éléments. Il suffirait simplement d’enlever la variable Max et tout le code qui est impliqué avec cette variable, notamment la méthode isFull, l’attribut full ainsi que le code vérifiant la taille de queue dans la méthode enqueue.
2. Une mauvaise pratique en programmation est d’imposer une limite arbitraire à la taille d’une structure de données. Non seulement cette limite est inutile, mais cela force aussi le programmeur à limiter l’utilisation de la taille de la queue selon la taille maximale définie, ce qui peut rendre le codage très désagréable. Il suffirait tous simplement d’enlever le code relié à la limite tel que expliqué dans en 1.