

-----

L'objectif des exercices est de mettre en place les **débarquements et embarquements de troupes** sur les îles et finaliser ainsi les classes *Ile* et *Navire*.

### Exercice 1. Classe *Ile* – V 1.4.0 / Evolutions fonctionnelles majeures

Installer avec succès la version V 1.4.0 de la classe *Ile* et vérifier sa conformité par rapport aux spécifications techniques détaillées de cette classe également fournies en annexe.

### Exercice 2. Classe *Ile* – V 1.4.0 / Accesseur *ajouterTroupes*

L'accesseur *ajouterTroupes* met à jour l'effectif courant des troupes d'une flotte cible en un point d'une île support, en augmentant ses effectifs locaux. La méthode sera invoquée après chaque débarquement de nouvelles troupes. Elle n'intègre pas les calculs des pertes subies par les belligérants au point de débarquement.

Exploiter le squelette fourni en annexe pour développer la méthode *ajouterTroupes* et exécuter avec succès le module de tests unitaires également fourni en annexe.

Aurait-il été possible de remplacer cette méthode par une surcharge de l'opérateur += ?

### Exercice 3. Classe *Ile* – V 1.4.0 / Accesseur *retirerTroupes*

L'accesseur *retirerTroupes* met à jour l'effectif courant des troupes d'une flotte cible en un point d'une île support, en diminuant ses effectifs locaux. La méthode sera invoquée après chaque embarquement de troupes. Elle n'intègre pas les calculs des pertes subies par les belligérants au point d'embarquement.

Par analogie avec le squelette fourni pour la méthode précédente, développer la méthode *retirerTroupes*.

Exécuter avec succès le module de tests unitaires fourni en annexe.

### Exercice 4. Classe *Ile* – V 1.4.0 / Méthode *calculerPertes*

Les pertes subies par tous les belligérants lors d'un débarquement de troupes sur une île seront calculées par la méthode privée *calculerPertes* de la classe *Ile*. Proposer un algorithme de calcul des pertes conforme à l'ensemble des exigences fonctionnelles.

### Exercice 5. Classe *Ile* – V 1.4.0 / Service *debarquement*

Le service *debarquement* de la classe *Ile* permet de mettre à jour les effectifs de toutes les troupes présentes au voisinage d'un point de débarquement, en intégrant les calculs des pertes subies par chaque belligérant présent.

Développer le service et exécuter avec succès le module de tests unitaires associé.

### Exercice 6. Classe *Ile* – V 1.4.0 / Service *embarquement*

Le service *embarquement* de la classe *Ile* permet de mettre à jour les effectifs de toutes les troupes présentes au voisinage d'un point d'embarquement, en intégrant les calculs des pertes subies par chaque belligérant présent.

Développer le service et exécuter avec succès le module de tests unitaires associé.

### Exercice 7. Classe *Navire* – V 1.4.0 / Service *debarquer*

Le service *debarquer* permet à un navire de débarquer des troupes sur une île cible. La méthode reçoit en paramètres : l'île de débarquement (objet de la classe *Ile*), le point cible et l'effectif des troupes à débarquer.

Développer le service et le module de tests unitaires associé.

**Exercice 8. Classe *Navire* – V 1.4.0 / Service *embarquer***

Le service *embarquer* permet à un navire de prendre à bord des troupes de sa propre flotte, déjà stationnées sur une île cible. La méthode reçoit en paramètres : l'île d'embarquement (objet de la classe *Ile*), le point cible et l'effectif des troupes à embarquer.

Développer le service et le module de tests unitaires associé.