
L'objectif des exercices est de mettre en place la **gestion des troupes** stationnées sur les îles. Cette feuille met en oeuvre la librairie **STL** (classes *string* et *map*).

Exercice 1. La classe *Tests* - Version 2.0.0 / Compatibilité **STL**

Les codes sources de la classe *Tests* fournis en annexe contiennent une mise à niveau d'un des fichiers "header" (**iostream**) exploité par la classe *Tests*. Cette mise à niveau est nécessaire pour utiliser la classe *Tests* dans un contexte **STL**.

Exercice 2. Classes **STL** *string* et *map* / Premiers exemples de mise en oeuvre

Analyser en détail les codes sources du module de tests unitaires fourni en annexe et exécuter ce module avec succès.

Peut on inverser le dictionnaire des abonnés ? Est ce le cas pour celui des points du plan ? Pourquoi ?

Exercice 3. Classes **STL** *string* et *map* / Autres exemples de mise en oeuvre

Analyser en détail les codes sources des modules de tests unitaires fournis en annexe et exécuter ces modules avec succès.

Peut on comparer deux dictionnaires dont les associés ne sont pas de même type mais dont le transtypage dynamique implicite est autorisé ?

Exercice 4. Classe *_Trafalgar* - V 1.1.0 / Compatibilité **STL** et nouveaux littéraux

Installer avec succès la nouvelle version de la classe abstraite.

Exercice 5. Classe *Ile* – V 1.2.0 / Modélisation des troupes débarquées

Les troupes de plusieurs flottes peuvent être présentes simultanément sur une île. Cette situation peut se produire en cours de confrontation, mais également au démarrage du jeu (situation initiale). Pour simplifier le modèle, toutes les troupes présentes sur une île sont rassemblées autour du port. Leur localisation sur l'île n'est donc pas mémorisée dans cette version de la classe.

Pour modéliser cette règle, la propriété *m_effectifs* de la classe *Ile* est représentée par un dictionnaire (classe **STL** *map*) dont la clé est l'identificateur de flotte (classe **STL** *string*) et dont l'associé est l'effectif courant des troupes débarquées sur l'île par cette flotte.

L'accessor de modification *ajouterTroupes* permet la mise en jour dynamique du dictionnaire.

Installer les codes sources de la classe *Ile* fournis en annexe, développer les codes sources des accesseurs *effectifs* et *ajouterTroupes* et exécuter avec succès tous les modules de tests unitaires fournis en annexe.

Exercice 6. Classes **STL** *list* / Exemples de mise en oeuvre

Analyser en détail les codes sources des modules de tests unitaires fournis en annexe et exécuter ces modules avec succès.

Exercice 7. Classe *Ile* – V 1.3.0 / Multi localisation des troupes débarquées

La contrainte de rassemblement de toutes les troupes autour du port de l'île est supprimée.

Proposer et implémenter l'ajout d'une propriété permettant de s'affranchir de cette contrainte. Le corollaire de cette nouvelle règle est qu'une flotte pourra débarquer et stationner des troupes en plusieurs points d'une île.

Exercice 8. Classe Ile – V 1.3.0 / Mise à niveau de l'accesseur *effectifs*

Mettre à niveau l'accesseur *effectifs* et exécuter avec succès le module de tests unitaires d'origine.

Exercice 9. Classe Ile – V 1.3.0 / Mise à niveau de l'accesseur *effectifs*

Mettre à niveau l'accesseur *ajouterTroupes* et le module de tests unitaires associé.