

TP6 : Bataille Navale, deuxième partie

Suite du TP5. Dans cette deuxième partie, nous allons réaliser la classe `GrilleNavale` représentant une grille avec un ensemble de navires. Cette classe permet de gérer le placement des navires et la gestion des tir reçus.

Le squelette de la classe à compléter est le suivant :

```
/**
 * Classe représentant une grille (carrée) de bataille navale contenant
 * des navires et pouvant gérer la réception de tirs
 */
public class GrilleNavale {

    /**
     * Le tableau des navires contenus dans la grille
     */
    private Navire[] navires;

    /**
     * Le nombre de navires contenus dans navires
     */
    private int nbNavires;

    /**
     * Taille de la grille, la grille est carrée
     */
    private int tailleGrille;

    /**
     * Tableau des tirs recus
     */
    private Coordonnee[] tirsRecus;

    /**
     * Nombre de tirs contenus dans tirRecus
     */
    private int nbTirsRecus;

    /**
     * Initialise une grille navale de taille donnée
     * Tous les attributs doivent être initialisés
     * @param taille la taille de la grille
     */
    public GrilleNavale(int taille) {...}
```

```

/**
 * Ajoute un navire dans la grille.
 * Un navire n'est pas ajouté si ses dimensions font qu'il sort de la
 * grille ou si il touche ou chevauche un navire déjà présent dans la
 * grille
 * L'attribut navire est agrandi automatiquement de 5 éléments si besoin
 * @param n le navire à ajouter
 * @return true si le navire a été ajouté.
 */
public boolean ajouteNavire(Navire n) {...}

/**
 * Place automatiquement tailleNavires.length navires
 * dont les tailles sont données dans tailleNavires
 * @param taillesNavires tableau donnant les tailles des navires à créer
 * et à placer dans la grille
 */
public void placementAuto(int[] taillesNavires) {...}

/**
 * Teste si la coordonnée passée en paramètre est contenue dans la grille
 * @param c référence non nulle vers la coordonnée à tester
 * @return true si la coordonnée est dans la grille
 */
public boolean estDansGrille(Coordonnee c) {...}

/**
 * Teste si la coordonnée passée en paramètre est dans
 * l'ensemble des tirs reçus sur la grille
 * @param c référence non nulle vers la coordonnée à tester
 * @return true si la coordonnée est dans l'ensemble des tirs reçus
 * sur la grille
 */
private boolean estDansTirsRecus(Coordonnee c) {...}

/**
 * Ajoute un tir dans le tableau tirsRecus.
 * Pour être ajouté un tir doit être dans la grille et ne doit pas
 * être déjà présent dans tirsRecus
 * @param c la coordonnée à ajouter
 * @return true si la coordonnée a été ajoutée dans tirsRecus
 */
private boolean ajouteDansTirsRecus(Coordonnee c) {...}

/**
 * Gère la réception d'un tir sur la grille.
 * Si le tir est ajouté dans les tir recus alors ce tir est
 * également envoyé
 * à chaque navire contenu dans la grille
 * @param c une référence vers la coordonnée du tir à réceptionner
 * @return true si le tir a touché un navire de la grille
 */
public boolean recoitTir(Coordonnee c) {...}

```

```

/**
 * Teste si l'un des navires présent dans la grille est touché
 * au niveau de la coordonnée passée en paramètre
 * @param c ue référence vers la coordonnée à tester
 * @return true si l'un des navire de la grille est touché
 * sur cette coordonnée
 */
public boolean estTouche(Coordonnee c) {...}

/**
 * Teste si un tir est dans les tirs recus mais n'a touché aucun navire
 * @param c ue référence vers la coordonnée à tester
 * @return true si la coordonnée est dans tirs reçus mais aucun bateau n'a
 * été touché
 */

public boolean estALEau(Coordonnee c) {...}

/**
 * Teste si tous les navires de la grille sont coulés
 * @return true si tous les navires contenus dans la grille sont coulés
 */
public boolean perdu() {...}

/**
 * Donne la taille de la grille
 * @return la taille de la grille
 */
public int getTaille() {...}
/**
 * Si l'un des navires est présent sur la coordonnée c,
 * la méthode teste si ce navire est est coulé
 * @param c une référence vers la coordonnée à tester
 * @return true si un navire est présent sur cette coordonnée et
 * si il est coulé
 */
public boolean estCoule(Coordonnee c) {...}
}

```

Vous penserez à tester les fonctionnalités au fur et à mesure