

Partiel Module POO3

IUP NTIE - Durée 2h

Jeudi 15 avril 2004

Question de cours

Expliquer précisément la différence de sémantique entre **extends** et **implements**.

Bataille Navale

Remarque générale : cet exercice sera noté en fonction de la qualité de la conception objet du logiciel, pas uniquement en fonction des astuces de programmation.

On voudrait réaliser un jeu de bataille navale. Un jeu de bataille navale se compose d'une grille et d'un ensemble de bateaux, chaque bateau se compose d'un ensemble de taille fixe d'éléments. Un croiseur comprend 3 éléments, un escorteur 2 et un sous marin un seul élément. Chaque élément est caractérisé par sa position et son état : sain ou touché. Les sous-marins ont la possibilité de plonger. Lorsqu'ils plongent, ils ne peuvent pas être touchés.

Un grille contient un ensemble de bateaux (stockés dans une ArrayList). Un bateau est caractérisé par l'ensemble de ses éléments (stockés dans un tableau).

Voici comment on instancie une flotte de bateaux (qui correspond à la figure)

```
//Un croiseur horizontal dont le premier élément est en 1,1
// (Les coordonnées ont leur origine en 0,0)
Bateau b1 = new Croiseur (1,1,true);
//un escorteur vertical dont le premier élément est 2,5
Bateau b2 = new Escorteur (2,5,false);
//un sous-marin en 4,2 et en surface (false = en plongée)
Bateau b3 = new SousMarin (4,2,true);

//le plateau de jeu
Grille g1 = new Grille (7,9)
g1.ajouterBateau (b1);
g1.ajouterBateau (b2);
g1.ajouterBateau (b3);
```

	0	1	2	3	4	5	6	7	8
0									
1		X	X	X					
2						X			
3						X			
4			X						
5									
6									

1. Donnez le diagramme UML (diagramme de classe)
2. Donnez le code des classes ainsi définies (définition avec les attributs, les constructeurs et les méthodes `ajouterBateau` et `enleverBateau` dans `Grille`).
3. Donnez le code de la méthode `int estTouche (int px, int py)` définie dans `Bateau` (et éventuellement dans ses sous-classes) qui retournent 0 lorsque le coup est dans l'eau, 1 si le coup tombe sur un élément touché, 2 si un élément tombe sur un élément touché pour la première fois, et 3 si le bateau est coulé (tous les éléments sont touchés). Si c'est un sous-marin en plongée, la méthode retourne toujours 0.
Note : vous pouvez ajouter des méthodes auxiliaires si nécessaire.
4. Donnez le code de la méthode `int coup (int px, int py)` définie dans `Grille`, qui retourne -1 si le coup est en dehors de la grille. 0 lorsque le coup est dans l'eau, 1 si le coup tombe sur un élément déjà touché, 2 si un le coup tombe sur un élément touché pour la première fois, et 3 si le bateau est coulé. Si le bateau est coulé, il est alors supprimé de la grille.
5. On suppose que les bateaux peuvent avancer, sauf lorsque les sous-marins sont en plongée. Ecrire la méthode `boolean avancer (int dx, dy)` dans `Bateau` (et éventuellement dans ses sous-classes) qui fait avancer le bateau d'une case dans une direction définie par `dx` et `dy` (la valeur de `dx` et `dy` est seulement -1, 1 ou 0, ce qui implique que les bateaux peuvent avancer en diagonale).

Annexe : méthodes principales de la classe ArrayList

```

boolean add (Object o)
boolean contains (Object o)
Object get (int index )
boolean remove (Object o)
int size ()

```