

Bataille navale

- Modèle (-10String pour debug)
 - Contrôle
-

boolean detruit

boolean []
touché (taille)

Bateau (abs
int taille (final ou privée avec access))

BateauSurface
(Constructeur
← taille
+ super

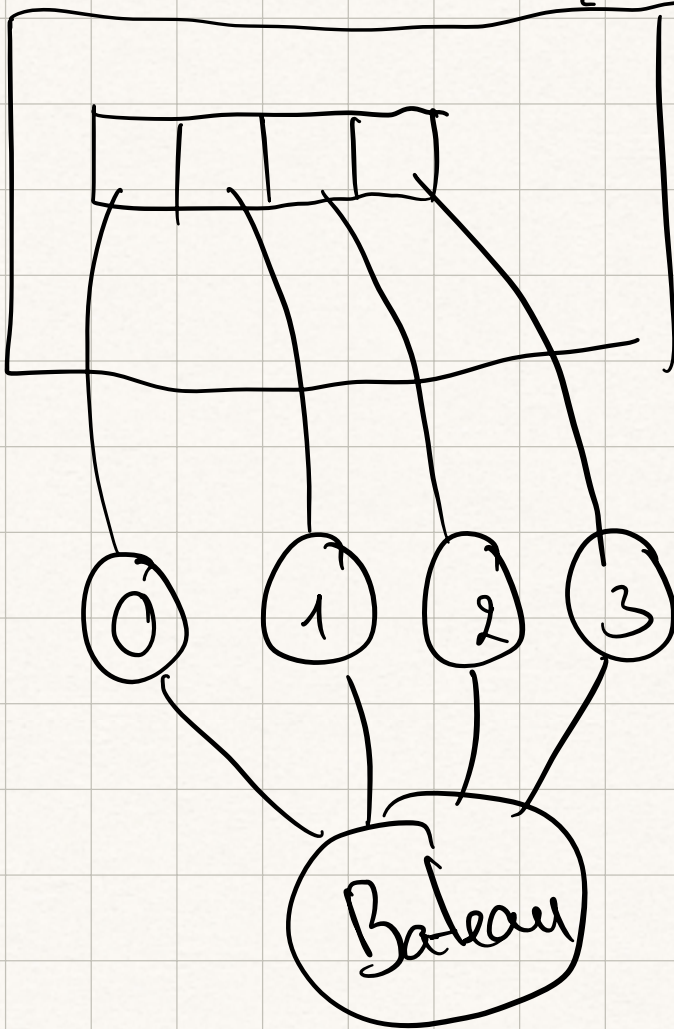
SousMarin
(%)

Tranche Bateau

Bateau

int numéro (de section)

→ Plateau

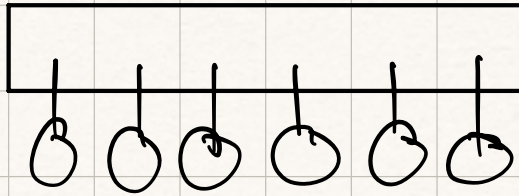


TrancheBateau[][]
Case vide : null

Faneur

Plateau

Bolean[]



Bolean

indice

indice - adresse

Fee

Faneur[]

int courant

print / scan

Bolean

int get Taille()

boolean detruit()

void detruire (int num)

String toString() dans les
classes héritière "S", "M"

Tranche Bateau

- void détruire()
 - boolean détruit()
- (deux Bateau)

Plateau → réussite, échec d'ajouter
le bateau

boolean ajouter

(Bateau b, int i, int j, boolean
horizontal)

boolean attaquer(int i, int j)

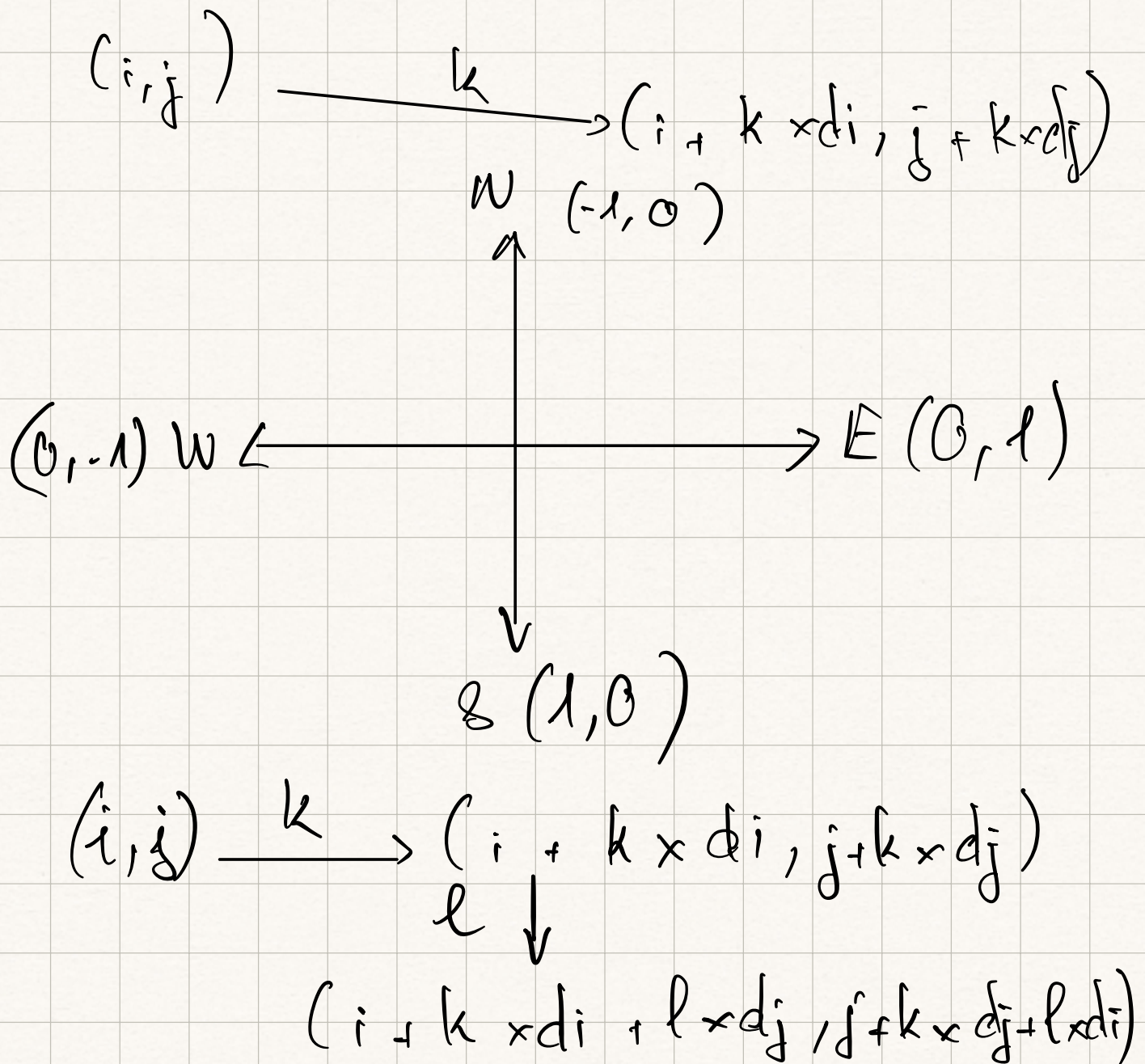
↑
touché ou non

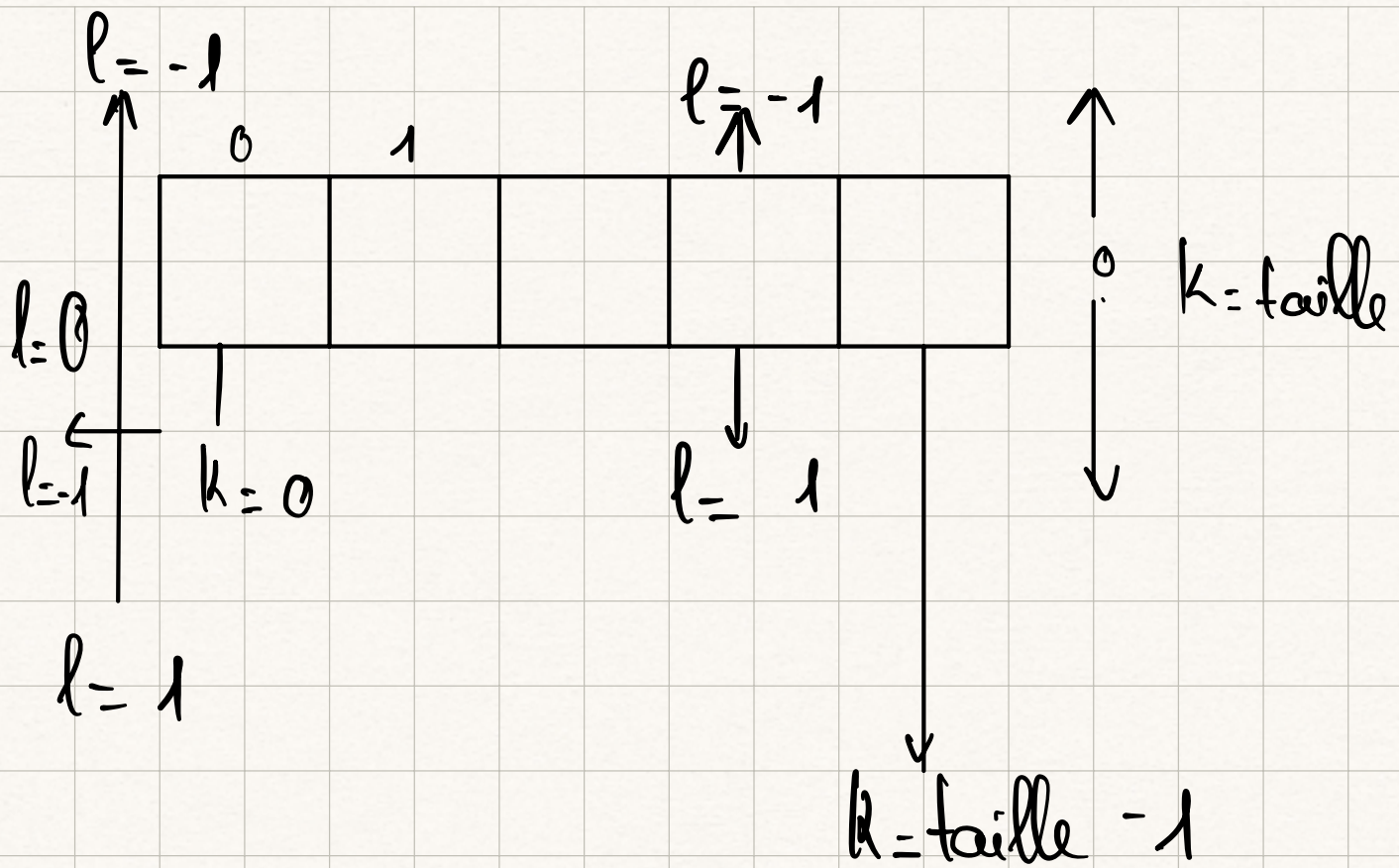
Bateau coulé (int i, int j)

> null si (vide ou) non coulé
le bateau coulé sinon

int di = (horizontal)? 1 : 0;

int dj = (horizontal)? 0 : 1;





Bateau

abstract boolean peutToucher (Bateau b);

abstract boolean peutEtreTouché (Bateau Surface bs);

abstract boolean peutEtreTouché (Sous Marin sm);

Bateau Surface

```
boolean peutEtreTouche (Bateau Surface b){  
    return false;  
}
```

```
boolean peutEtreTouche (Sousmarin sm){  
    return true;  
}
```

```
boolean peutToucher (Bateau b){  
    return b.peutEtreTouche (this);  
}
```

Sousmarin {

```
boolean peutEtreTouche (Bateau Surface b){  
    return true;  
}
```

```
boolean peutEtreTouche (Sousmarin sm){  
    return true;  
}
```

```
boolean peutToucher (Bateau b){  
    return b.peutEtreTouche (this);  
}
```