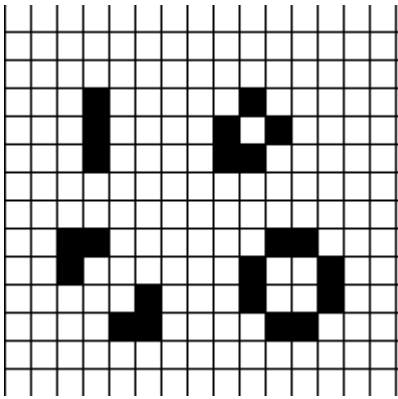


Algorithmique et programmation

Le monde du jeu de la vie de Conway est un plan infini quadrillé dont chaque case est occupée par une cellule morte ou vivante. Chaque case possède huit voisines et, d'une génération à l'autre, des naissances et des décès s'y déroulent mécaniquement selon la règle simple suivante :

- si une case est occupée par une cellule morte (en blanc sur la figure) et qu'elle a exactement trois de ses voisines occupées par des cellules vivantes, alors la cellule renaît ;
- si une case est occupée par une cellule vivante (en noir sur la figure) la survie n'y est possible que si deux ou trois cases voisines sont occupées par des cellules vivantes ; dans tous les autres cas, la cellule meurt.

Nous nous intéressons à la simulation de ce jeu dans un plan fini.



Question 0. (1 points)

Quel est l'état du monde ci-contre à la génération suivante.

Question 1. (2 points)

Ecrire la définition du constructeur et des méthodes *setVivant* et

```
public class Cellule {
    private boolean vivant;
    public Cellule() {...}
    public void setVivant(boolean b) {...}
    public boolean estVivant() {...}
}
```

estVivant de la classe *Cellule*.

Le monde du jeu de la vie est représenté par un tableau de *nbLignes* lignes et *nbColonnes* colonnes.

```
public abstract class MondeAbstrait {
    protected Cellule [][]rep;

    public MondeAbstrait (int l, int c) {...}

    public int getLignes(){...}
    public int getColonnes(){...}
    public void genererAlea(){...}

    public abstract int nbVoisinsVivants( int i, int j);

    public void generationSuiivante(){...}
}
```

Question 3. (3 points)

Ecrire la définition du constructeur de la classe *MondeAbstrait*. On construira un monde où toutes les cellules sont mortes.

Question 4. (2 points)

Ecrire la définition des méthodes *getLignes()* et *getColonnes()*.

Question 5. (3 points)

Ecrire la définition de la méthode *genererAlea()* qui remplit le monde de façon aléatoire : en moyenne une cellule sur 3 sera vivante.

Question 6. (3 points)

Ecrire la définition de la méthode *generationSuivante()* qui calcule le monde à la génération suivante.

Il y a deux façons d'envisager le voisinage des cellules des bords du monde :

- Un monde plat : les cellules du bord ont soit 5 voisins soit 3 voisins pour les coins.
- Un monde « torique » : les cellules du bord ont toutes 8 voisins (chaque coin a aussi pour voisin le coin opposé suivant la diagonale) : on considère la première ligne et la dernière lignes voisines, et la première colonne et la dernière colonne voisines.

Ces deux mondes sont représentés par les classes suivantes :

```
public class MondePlat extends MondeAbstrait {
    public MondePlat (int l, int c){...}
    public int nbVoisinsVivants( int i, int j){...}
}
```

```
public class MondeTorique extends MondeAbstrait {
    public Monde MondeTorique (int l, int c){...}
    public int nbVoisinsVivants( int i, int j){...}
}
```

Question 7. (2 points)

Ecrire la définition des constructeurs des deux classes.

Question 8. (4 points)

Ecrire les définitions des méthodes *nbVoisinsVivants(i, j)* des deux classes qui renvoient le nombre de voisins vivants de la cellule située en (i, j) .