



# Exercice 3

**Question 3.1** Compléter le code de la classe Dedale et Conversion pour obtenir ce résultat.

## Méthodes Dédale

```
public Dedale()
{
    this.NbColonne = 5;
    this.NbLigne = 5;

    this.tabPiece = this.initPiece();
}
```

## Méthodes Piece getPieceAdj (lig : entier , col : entier , dir : caractère)

```
private Piece getPieceAdj(int lig, int col, char dir)
{
    switch (dir)
    {
        case 'N':
            if (lig > 0)
                return this.tabPiece[lig - 1][col];
            break;
        case 'O':
            if (col > 0)
                return this.tabPiece[lig][col - 1];
            break;
        case 'S':
            if (lig < this.getNbLigne() - 1)
```

```

        return this.tabPiece[lig + 1][col];
        break;
    case 'E':
        if (col < this.getNbColonne() - 1)
            return this.tabPiece[lig][col + 1];
        break;
    }
    return null;
}

```

### Méthodes getNbLigne() :

```

public int getNbLigne()
{
    return this.NbLigne;
}

```

### Méthode getNbColonne() :

```

public int getNbColonne()
{
    return this.NbColonne;
}

```

### Méthode Piece getPiece (lig : entier, col : entier)

```

public Piece getPiece(int lig, int col)
{
    if (lig > this.NbLigne || lig < 0 ) return null;
    if (col > this.NbColonne || col < 0 ) return null;

    return tabPiece[lig][col];
}

```

### Méthode Piece [][] initPiece() :

```

private Piece[][] initPiece()
{
    Piece[][] grillePiece;

    grillePiece = new Piece[][] {
        {new Piece(0, "A"), new Piece(1, "A"), new Piece(2, "A"), new Piece(3, "A"), new Piece(4, "A"), new Piece(5, "A"), new Piece(6, "A"), new Piece(7, "A"), new Piece(8, "A"), new Piece(9, "A")},
        {new Piece(0, "F"), new Piece(1, "F"), new Piece(2, "F"), new Piece(3, "F"), new Piece(4, "F"), new Piece(5, "F"), new Piece(6, "F"), new Piece(7, "F"), new Piece(8, "F"), new Piece(9, "F")},
        {new Piece(0, "K"), new Piece(1, "K"), new Piece(2, "K"), new Piece(3, "K"), new Piece(4, "K"), new Piece(5, "K"), new Piece(6, "K"), new Piece(7, "K"), new Piece(8, "K"), new Piece(9, "K")},
        {new Piece(0, "P"), new Piece(1, "P"), new Piece(2, "P"), new Piece(3, "P"), new Piece(4, "P"), new Piece(5, "P"), new Piece(6, "P"), new Piece(7, "P"), new Piece(8, "P"), new Piece(9, "P")},
        {new Piece(0, "U"), new Piece(1, "U"), new Piece(2, "U"), new Piece(3, "U"), new Piece(4, "U"), new Piece(5, "U"), new Piece(6, "U"), new Piece(7, "U"), new Piece(8, "U"), new Piece(9, "U")},
    };

    return grillePiece;
}

```

**Voici le code Main + un exemple de ce qu'il renvoie :**

```

public class TestDedale
{
    public static void main(String[] a)
    {
        /*-----*/
        /*  Donnes  */
        /*-----*/
        Dedale dedale;
        String separateur;

        /*-----*/
        /* Instructions */
        /*-----*/
        dedale = new Dedale();
    }
}

```

```

// Affichage du plan
System.out.println ( "-----" );
System.out.println ( " Grille" );
System.out.println ( "-----\n" );

System.out.println ( Conversion.grille ( dedale ) );

System.out.println();

// Affichage du Détail
System.out.println ( "-----" );
System.out.println ( " Detail" );
System.out.println ( "-----\n" );

System.out.println ( Conversion.detail ( dedale ) );

System.out.println();

}

}

```

```

-----
Grille
-----

```

```

+---+---+---+---+---+
|  0 | 12 | 14 |  6 |  0 |
+---+---+---+---+---+
|  0 |  1 |  5 |  1 |  0 |
+---+---+---+---+---+
|  8 | 10 | 15 | 10 |  2 |
+---+---+---+---+---+
|  0 |  4 | 13 |  4 |  0 |
+---+---+---+---+---+

```

```
| 0 | 9 | 11 | 3 | 0 |
+---+---+---+---+---+
```

```
-----
Detail
-----
```

```
== Ligne 0 =====
 0 (A          ) ==> Nord(0) :false   Ouest(1) :false Sud
12 (B          ) ==> Nord(0) :false   Ouest(1) :false Sud
14 (C          ) ==> Nord(0) :false   Ouest(1) :true  Sud
 6 (D          ) ==> Nord(0) :false   Ouest(1) :true  Sud
 0 (E          ) ==> Nord(0) :false   Ouest(1) :false Sud
== Ligne 1 =====
 0 (F          ) ==> Nord(0) :false   Ouest(1) :false Sud
 1 (G          ) ==> Nord(0) :true    Ouest(1) :false Sud
 5 (H          ) ==> Nord(0) :true    Ouest(1) :false Sud
 1 (I          ) ==> Nord(0) :true    Ouest(1) :false Sud
 0 (J          ) ==> Nord(0) :false   Ouest(1) :false Sud
== Ligne 2 =====
 8 (K          ) ==> Nord(0) :false   Ouest(1) :false Sud
10 (L          ) ==> Nord(0) :false   Ouest(1) :true  Sud
15 (M          ) ==> Nord(0) :true    Ouest(1) :true  Sud
10 (N          ) ==> Nord(0) :false   Ouest(1) :true  Sud
 2 (O          ) ==> Nord(0) :false   Ouest(1) :true  Sud
== Ligne 3 =====
 0 (P          ) ==> Nord(0) :false   Ouest(1) :false Sud
 4 (Q          ) ==> Nord(0) :false   Ouest(1) :false Sud
13 (R          ) ==> Nord(0) :true    Ouest(1) :false Sud
 4 (S          ) ==> Nord(0) :false   Ouest(1) :false Sud
 0 (T          ) ==> Nord(0) :false   Ouest(1) :false Sud
== Ligne 4 =====
 0 (U          ) ==> Nord(0) :false   Ouest(1) :false Sud
 9 (V          ) ==> Nord(0) :true    Ouest(1) :false Sud
11 (W          ) ==> Nord(0) :true    Ouest(1) :true  Sud
 3 (X          ) ==> Nord(0) :true    Ouest(1) :true  Sud
 0 (Y          ) ==> Nord(0) :false   Ouest(1) :false Sud
=====
```