

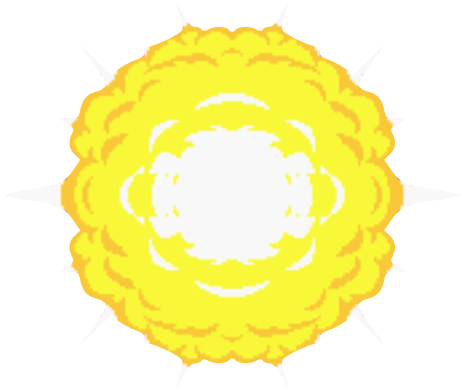


LINK TO THE PATH

```
/** lib tierce */
interface Direction{
    public int toInt();
};
class North implements Direction{
    public int toInt(){ return 1;};
};
class East implements Direction{
    public int toInt(){ return 2;};
};
class South implements Direction{
    public int toInt(){ return 3;};
};
class West implements Direction{
    public int toInt(){ return 4;};
};

/** notre code */
interface DirectionImprove{
    public String toString();
};
class NorthImprove extends North implements DirectionImprove{
    public String toString(){ return "north";};
};
class EastImprove extends North implements DirectionImprove{
    public String toString(){ return "east";};
};
class SouthImprove extends North implements DirectionImprove{
    public String toString(){ return "south";};
};
class WestImprove extends North implements DirectionImprove{
    public String toString(){ return "west";};
};
```





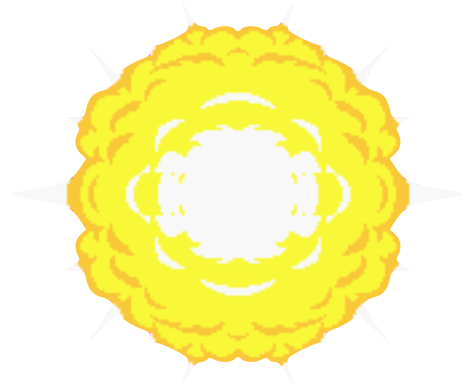
MAGNET 2-DUALS - THOUGHTS

Les encodages initiaux sont complexes à maintenir!

*Composition over inheritance: "Favor 'object composition' over 'class inheritance'."
(Gang of Four 1995:20)*

LINK TO THE PATH

QUE SE PASSE-T-IL SI ON RÉCUPÈRE LES DIRECTIONS D'UNE LIB



Les encodages initiaux sont complexes à maintenir!

Composition over inheritance: "Favor 'object composition' over 'class inheritance'."
(Gang of Four 1995:20)

```
/** lib tierce */
interface Direction{
    public int toInt();
};
class North implements Direction{
    public int toInt(){ return 1;};
};
class East implements Direction{
    public int toInt(){ return 2;};
};
class South implements Direction{
    public int toInt(){ return 3;};
};
class West implements Direction{
    public int toInt(){ return 4;};
};

/** notre code */
interface DirectionImprove{
    public String toString();
};
class NorthImprove extends North implements DirectionImprove{
    public String toString(){ return "north";};
};
class EastImprove extends North implements DirectionImprove{
    public String toString(){ return "east";};
};
class SouthImprove extends North implements DirectionImprove{
    public String toString(){ return "south";};
};
class WestImprove extends North implements DirectionImprove{
    public String toString(){ return "west";};
};
```

LINK TO THE PATH

EN PASSANT PAR JAVA 15 (SEALED), JAVA 14 (RECORD) ET JAVA 17 (EXHAUSTIVITÉ)

```
sealed interface Direction {  
    record North() implements Direction {}  
    record East() implements Direction {}  
    record South() implements Direction {}  
    record West() implements Direction {}  
}  
public static final String label(Direction d) {  
    return switch(d){  
        case Direction.North n -> "north";  
        case Direction.East e -> "east";  
        case Direction.South s -> "south";  
        case Direction.West w -> "west";  
    }  
}
```