



UTILISANT EN MANIÈRE NON CONFORME

LINK TO THE PATH

```
enum Direction { North, East, South, West }

public class Main {
    public static final String label(Direction d) {
        switch (d) {
            case North : return "north";
            case East : return "east";
            case South : return "south";
            //case West : return "west";
        }
        throw new IllegalArgumentException("not a valid direction");
    }

    public static void main(String args[]) {
        System.out.println(label(Direction.East));
        System.out.println(label(Direction.West));
    }
}
```



Exception in thread 'main' java.lang.IllegalArgumentException: not a valid direction



RuntimError

Le campilateur ne sait pas décompiler si nous avons traité tous les cas!!!

MAGNET 2-DUALS - TRANSLES

Indice sur le problème



LINK TO THE PATH

UTILISATION DES ENUM EN JAVA ... MAIS EN MODIFIANT LE CODE



Indice sur le problème



```
enum Direction { North, East, South, West }

public class Main {
    public static final String label(Direction d) {
        switch (d) {
            case North : return "north";
            case East : return "east";
            case South : return "south";
            //case West : return "west";
        }
        throw new IllegalArgumentException("not a valid direction");
    }

    public static void main(String args[]) {
        System.out.println(label(Direction.East));
        System.out.println(label(Direction.West));
    }
}
```

Runtime ERROR

Exception in thread "main" java.lang.IllegalArgumentException: not a valid direction

Le compilateur ne sait pas détecter si nous avons traité tous les cas !!!

LINK TO THE PATH

ON EST EN OOP : ON A DU POLYMORPHISME PAR HÉRITAGE



```
interface Direction{
    public int toInt();
    public String toString();
};
class North implements Direction{
    public int toInt(){ return 1;};
    public String toString(){ return "north";};
};
class East implements Direction{
    public int toInt(){ return 2;};
    public String toString(){ return "east";};
};
class South implements Direction{
    public int toInt(){ return 3;};
    public String toString(){ return "south";};
};
class West implements Direction{
    public int toInt(){ return 4;};
    public String toString(){ return "west";};
};

public class Main {
    public static final String label(Direction d) {
        return d.toString();
    }

    public static void main(String args[]) {
        System.out.println(label(new East()));
        System.out.println(label(new West()));
    }
}
```