UTILISATION D'UN TYPE « OU » DANS UN SYTÈME DE TYPE ML

LINK TO THE PATH

```
let label d = match d with
    | North -> "north"
    | East -> "east"
    | South -> "south"
    (*| West -> "west" *)
```

```
Warning : this pattern-matching is not exhaustive.
Here is an example of a case that is not matched:
West
```

On peut valider dès la compilation qu'on traite TOUTES LES valeurs valides !!!

Compilation WARNING

MIAGE M2 - QUALITÉ DU SI - THOMAS HAESSLÉ

Warn par défaut mais on peut en faire une erreur bloquant la compilation en ajoutant le paramètre -warn-error -a+31+8

LINK TO THE PATH

UTILISATION D'UN TYPE « OU » DANS UN SYTÈME DE TYPE ML

```
let label d = match d with
  | North -> "north"
  | East -> "east"
  | South -> "south"
  (*| West -> "west" *)
```

Compilation WARNING

```
Warning : this pattern-matching is not exhaustive.
Here is an example of a case that is not matched:
West
```

On peut valider dès la compilation qu'on traite TOUTES LES valeurs valides !!!

Warn par défaut mais on peut en faire une erreur bloquant la compilation en ajoutant le paramètre -warn-error -a+31+8

TAKE AWAY

AVEC UN LANGAGE STATIQUEMENT (BIEN) TYPÉ

On peut valider qu'on traite uniquement les valeurs valides

On peut valider qu'on traite toutes les valeurs valides

Dès la compilation

Dans les langages ML : OCaml, Haskell, F#, …

Mais aussi les langages modernes qui s'en inspirent : Scala, Kotlin, Swift, Rust, C++17, …