```
enum class Direction{
    UP,DOWN,UNKNOWN
}
fun main() {
    val myBoolean: Boolean = true
    val inputString : String = "u"

    val x = if (myBoolean) 3 else 5
    val direction :Direction = when (inputString) {
        "u" -> Direction.UP
        "d" -> Direction.DOWN
        else -> Direction.UNKNOWN
    }
    val number :Int = try {
        inputString.toInt()
    } catch (nfe: NumberFormatException) {
        println("Amúgy hiba történt: ${nfe.message}")
        -1
    }
    println("x = $x")
    println("direction = $direction")
    println("number = $number")
}
```

Elemzés:

---

1. `enum class Direction`

```
enum class Direction {
    UP, DOWN, UNKNOWN
}
```

- Ez egy **enum osztály** (felsorolt típus), amely három állapotot tartalmaz: `UP`, `DOWN` és `UNKNOWN`.
- Az enumokat általában olyan változókhoz használjuk, amelyek előre meghatározott, véges értékkészletből választhatnak.

---

2. `fun main()`

A program belépési pontja, ahol a változók és a logika található.

---

3. Változók

```
val myBoolean: Boolean = true
val inputString: String = "u"
```

- `myBoolean`: logikai változó, ami most `true`.

- `inputString`: szöveges változó, amely `"u"` értéket kap.

---

4. `x` értékének meghatározása

```
val x = if (myBoolean) 3 else 5
```

- Ez egy egyszerű feltételes kifejezés: ha `myBoolean` igaz (`true`), akkor `x` értéke `3`, különben `5`.
- Mivel `myBoolean` igaz, így `x = 3`.

---

5. `direction` értékének meghatározása `when`-nel

```
val direction: Direction = when (inputString) {
    "u" -> Direction.UP
    "d" -> Direction.DOWN
    else -> Direction.UNKNOWN
}
```

- Ez egy `when` szerkezet, ami az `inputString` értékét vizsgálja.
- Ha az `inputString` `"u"`, akkor `direction` értéke `Direction.UP`.
- Ha `"d"`, akkor `Direction.DOWN`.
- Minden más esetben `Direction.UNKNOWN`.
- Mivel `inputString` értéke `"u"`, a `direction` értéke `Direction.UP`.

---

6. Szám konvertálása `try-catch`-szel

```
val number: Int = try {
    inputString.toInt()
} catch (nfe: NumberFormatException) {
    println("Amúgy hiba történt: ${nfe.message}")
    -1
}
```

- Próbálja az `inputString`-et számmá konvertálni (`toInt()`).
- Mivel `inputString` `"u"`, ami nem szám, a konverzió **hibát dob** (`NumberFormatException`).
- A `catch` blokk fut le, kiírja a hibaüzenetet (`Amúgy hiba történt: ...`), és a `number` értéke `-1` lesz.
- Így a program nem áll meg, és biztonságosan kezeli a hibát.

---

7. Kiírások

```
println("x = $x")
println("direction = $direction")
println("number = $number")
```

- Kiírja a változók aktuális értékeit.

- A kimenet lesz:

```
Amúgy hiba történt: For input string: "u"
x = 3
direction = UP
number = -1
```

Összegzés

- A program alapvetően egy egyszerű példa feltételes elágazásra (`if`), enumerált típus használatára (`enum class`) és kivételkezelésre (`try-catch`).
- A `direction` változó megfelelően enum értékre állítódik az `inputString` alapján.
- A `number` változó biztonságosan kezeli azt az esetet, ha a bemenet nem szám, és ilyenkor -1 értéket ad vissza, valamint hibaüzenetet ír ki.
- A változók értékei jól láthatóak a `println` segítségével.
