

Az `enum class` a Kotlin nyelvben egy **enumerációs osztály** – vagyis egy olyan speciális osztálytípus, amely **egy előre meghatározott, fix számú konstans értéket** képvisel. Ez hasonló a Java `enum` típusához, de Kotlinban több lehetőséget nyújt: például metódusokat is írhatasz az enum osztályokba, vagy minden értékhez különböző viselkedést rendelhetsz.

◆ Mire jó az `enum class`?

- Állapotok, opciók, típusok fix listájának reprezentálására.
 - Olvashatóbb és biztonságosabb kód (nem szabad formátumú `String` vagy `Int` értékekkel dolgozol).
 - Könnyen használható `when` ágban (switch-szerű vezérlésnél).
-

◆ Egyszerű példa

```
enum class Day {  
    MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, SUNDAY  
}  
  
fun main() {  
    val today = Day.FRIDAY  
  
    when (today) {  
        Day.MONDAY -> println("Hétfő van!")  
        Day.FRIDAY -> println("Péntek van, majdnem hétvége!")  
        Day.SATURDAY, Day.SUNDAY -> println("Hétvége!")  
        else -> println("Munkanap.")  
    }  
}
```

◆ Enum adattagokkal (property + metódus):

```
enum class Animal(val sound: String) {  
    DOG("Woof"),  
    CAT("Meow"),  
    COW("Moo");  
  
    fun speak() {  
        println("The $name says $sound")  
    }  
}  
  
fun main() {  
    Animal.DOG.speak() // The DOG says woof  
}
```

◆ Minden elemhez külön működés (override-olható metódusokkal):

```
enum class Operation {  
    ADD {  
        override fun apply(x: Int, y: Int) = x + y  
    },  
    SUBTRACT {  
        override fun apply(x: Int, y: Int) = x - y  
    };  
  
    abstract fun apply(x: Int, y: Int): Int  
}
```

Összefoglalva

Az `enum class`:

- előre definiált értékek halmaza (konstansok),
 - erősen típusos,
 - bővíthető metódusokkal, property-kkel,
 - jól használható vezérlésben (`when`),
 - támogatott `name`, `ordinal`, és egyéb beépített tulajdonságokkal.
-