

Java Enum

<< Java

Enum Constructor

- <https://jenkov.com/tutorials/java/enums.html#enum-fields>
- <https://www.programiz.com/java-programming/enum-constructor>

Az Enum-nak készíthetünk konstruktort is, ami kizárólag privát típusú lehet:

```
public enum AdamTest {  
  
    LALI(2),  
    ADAM(3)  
    ;  
  
    private AdamTest(int a) {  
    }  
}
```

Ha nem gyártunk konstruktort az enum-hoz akkor is van neki, ez a default constructor, ami nem vár paramétert. Ha paraméteres konstruktort hozunk létre, akkor minden ENUM elemnek () között meg kell adni a paramétereket. Létrehozhatunk konstruktort üres paraméter listával is. Ekkor az alábbi helyes:

```
public enum AdamTest2 {  
  
    LALI,  
    ADAM()  
    ;  
  
    private AdamTest2() {  
    }  
}
```

Láthatjuk, hogy megadhatjuk az enum felsorolást zárójellel és anélkül is.

Enum field

Az enumnak tetszőleges mezőket is definiálhatunk, de kötelezően private mezőknek kell lenni.

Ezeket a mezőket az enum konstruktor segítségével szokásos feltölteni.

```
public enum AdamTest {  
  
    LALI(2),  
    ADAM(3)  
    ;  
  
    private int enumInt;  
  
    private AdamTest(int a) {  
        this.enumInt=a;  
    }  
}
```

Tehát minden egyes enum példányban ami vagy LALI vagy ADAM típus lesz, benne lesz az az int érték, amit az enum példányosító konstruktor hozzárendel.

- LALI -> 2
- ADAM -> 3

Enum methods

Az enum-nak készíthetünk publikus metódusokat is, amik már vissza tudják adni azokat az értékeket, amiket a konstruktorral hozzárendeltünk az enum példányokhoz.

```
public enum AdamTest {  
  
    LALI(2),  
    ADAM(3)  
    ;  
  
    private int enumInt;  
  
    private AdamTest(int a) {  
        this.enumInt=a;  
    }  
  
    public int getEnumInt(){  
        return enumInt;  
    }  
}
```

Ha példányosítjuk a fenti enumot, akkor az alábbi fogjuk látni:

```
AdamTest adamTest1 = AdamTest.LALI;  
AdamTest adamTest2 = AdamTest.ADAM;  
  
System.out.print(adamTest1.getEnumInt()); --> 2
```

```
System.out.print(adamTest2.getEnumInt()); --> 3
```