```
data class Variables(  //változok
    //Mivel data class használtam nem kell a tooString annotáció felülírása
    //és {} helyett ()
    //változok deklarálása
    var age : Int = 26, //kor
    var weight : Double = 65.5, //súly
    var isStudent : Boolean = false,// tanuló -e
    //a boolena-nak két értéke van true(igaz) False (hamis)
    var Name : String = "Alice", //név
    val Pi : Double = 3.14, //pi értéke
    //azért val mert nem változtatható
    var byte : Byte = 8,
    var grade: Char = 'A',
    var temperature: Float = 36.6f, //Hőmérséglet
    var population: Long = 7800000000, //népeesség
    var Numbers : List<Int> = listOf(1, 2, 3, 4, 5, 6, 7),
    var Names : List<String> = listOf("Alice", "Alice1", "Bob", "Bob2"),
    var Names2: Map<String, Int> = mapOf("Alice" to 1, "Alice1" to 2, "Bob" to
            3, "Bob2" to 4)
)

    //A kotlin fordító alapértelmezeten kitalálja de azért jobb beírni
    //ha var akkor változtatható
    //felülírjuk a Toostringet
    /*override fun toString() : String { //felülírás override fun
        return "Variables age=$age; weight=$weight; isStudent=$isStudent;" +
                "Name=$Name;Pi=$Pi ;byte = ${byte};grade=$grade"+
                "temperature=$temperature;
        temperature=$temperature;population=$population; " +
        "Numbers=$Numbers;Names=$Names;Names2=${Names2}"
        //hogy ne kapjunk Variables@6bc7c054 ilyen kiirast
        // return visszadja az értéket a függvényben
     */


fun main(args : Array<String>) { //nem kötelező (args : Array<String>)
    val variables = Variables() //példányosítjuk az osztályt
    println("Kiirjuk az osztály tartalmát")
    println(variables.toString())//egybe a toString kiirja
    println("\nKülön:")
    variables.toString().split(",").forEach {
        println(it.trim())}
//.split (;) külön szedi az overide felülírásból ez; alapján
//.forEach végig megy a listán
//it.trim() trim levágja a felesleges szóközöket az it az aktuális elem
        println("\nKülön még egyszer : ")
        println("\nKor : ${variables.age}")
        println("Súly : ${variables.weight}")
        println("Tanuló vagy nem ${variables.isStudent}")
        println("Név: ${variables.Name}")
        println("Pi : ${variables.Pi}\nBájt :${variables.byte}")// \n egy uj
sort adtam hozzá
```

```
        print("EgyKarakter : ${variables.grade}")
        println("népeesség: ${variables.population}")
        println("Hőmérséglet: ${variables.temperature}")
        println("Nevek: ${variables.Names}")
        println("Számok: ${variables.Numbers}")
        println("Nevek2 : ${variables.Names2}")
}
```

Figyelem! a data class csak konstruktor lehet tehát () ezt kell használni nem ezt {}

Jel	Neve	Mire használjuk
`()`	**Kerek zárójel** (*round brackets / parentheses*)	**Függvényhívásoknál**, **konstruktor paramétereknél**, **feltételeknél**
`{}`	**Kapcsos zárójel** (*curly braces*)	**Kódtömbök**, pl. `if`, `for`, `fun`, lambda kifejezések
`[]`	**Szögletes zárójel** (*square brackets*)	Listák indexeléséhez, például `list[0]`