

Mobil- és webes szoftverek - 1. kotlin előadás

Első Android projekt

- Felhasználói felület → erőforrás állományként van leírva (pl: XML)
- Kotlin kód → ez írja le a viselkedést, eseményeket
 - ```
setContentView(R.layout.activity_main) //felület beállítása
btnDemo.setOnClickListener { //esemény rendelése a gombhoz
 tvHello.text = "Demo"
};
```
- AVD manager → Android virtuális eszközöket kezeli

## Projekt felépítése

- Java mappa → package → MainActivity.kt = *ez a kotlin fájlunk*
- res → layout → activity\_main.xml = *ez felel a megjelenésért*

## Kotlin nyelv bevezető

---

### Kotlin-ról röviden

- egy szigetről kapta a nevét, 2011-ben jelent meg először
- **JetBrains** fejleszti, 2017 → hivatalos támogatás Android-ra
- funkcionális programozást is támogat
- Java alapú programozási nyelv → JVM byte kódra fordul
- automatikus konverzió Java-ról Kotlinra meg vissza
- **Null-safety** → nem lesz annyi bajunk a NullPointerException-nal
- könnyen olvasható a kód
- Tools → Kotlin → **Kotlin REPL**: Kotlin fordító/futtató, gyors tesztelésre praktikus
- nem kell pontosvessző a sorok végére

### Változók és konstansok

- Változó létrehozása → **var** kulcsszó
- konstans létrehozása → **val** kulcsszó
  - ha meg akarjuk változtatni, hibát dob
- nem kell megadni a típust, magától kitalálja

### Típusok

- Minden valójában egy objektum → függvényekkel és lekérhető tulajdonságokkal rendelkeznek
  - pl: `var a = 10; a.toString();`
- némely típus futási időben primitív típusként van reprezentálva, de osztálynak látszik
- Használhatunk alul vonást a számok jobb érthetősége végett: `val a = 1_000_000;`

- hexa szám írása: *val a = 0xFF*, bináris szám írása: *val a = 0b11*
- float-ként tárolás: *val b = 1.233\_333F*

## Kasztolás (típusok váltása)

- implicit kasztolás nem engedélyezett
- minden típus saját függvényével lehetséges ez

- ```
var a: Byte = 4
var b: Int = a.toInt()
```

Tömbök

- Sima tömbök

```
var myIntArray = intArrayOf(1, 2, 3)
var myIntArray2 = Array<Int>(3, {it * 1}) //3 eleme legyen, amik a második
paraméter szerint számolódnak ki, ahol it = elem indexe

println(Arrays.toString(myIntArray)) //[1, 2, 3]
println(Arrays.toString(myIntArray)) //[0, 1, 2]
println(myIntArray[2]) //3 (a tömb 2. eleme 0-tól számozva)
```

- Vegyes típusokat tartalmazó tömb

```
var mixedArray = arrayOf(1, "something")
println(Arrays.toString(mixedArray)) //[1, something]
```

- String = karaktertömb

```
var someString: String = String(charArrayOf('m', 'y', ' ', 'a', 'b', 'c'))
var someString2: String = "my abc"
```

- String minták (template) → \$ jellel

```
val numOfChar = 10
print("There are $numOfChar in this word")

val numOfMan = 2
val numOfWoman = 4
print("There are ${numOfA + numOfB} people in this room")
```

- String speciális karakterek
 - \t (tab), \n (új sor),
 - Nyers stringek

```
var rawString = """Hello!
    |This is me!
    """.trimMargin() //a trimMargin() letörli a sorok előtt és mögött levő
szóközöket
```

- Stringek összehasonlítása → ==

```
var a = "alma"  
var b = "alma"  
println(a == b) //true
```

Műveletek tömbökkel

- Rendezés, sorrend megkavarása, bináris keresés, stb

```
val numbers: ArrayList<Int> = ArrayList()  
numbers.add(4)  
numbers.add(2)  
numbers.add(5)  
numbers.add(7)  
numbers.add(3)  
println("Rendezetlen tömb: $numbers") //[4, 2, 5, 7, 3]  
  
numbers.sort() //sorba rendezi a tömböt pl [2, 3, 4, 5, 7]  
numbers.shuffle() //valami random sorrendbe keveri a tömböt pl [7, 5, 3, 2, 4]  
numbers.reverse() //visszafordítja a tömböt  
Collections.swap(numbers, 0, 1) //megcseréli a numbers tömb 0. és 1. elemét  
  
val newNumbers: ArrayList<Int?> = ArrayList()  
newNumbers.addAll(numbers) //hozzáadja a newNumbers tömb végéhez a numbers összes elemét  
newNumbers.fill(1) //felülírja az összes értéket a megadottal (pl mind 1-es lesz)  
  
val pos = Collections.binarySearch(numbers, 3) //bináris keresés, CSAK RENDEZETT TÖMBÖN MEGY, visszaadja az adott elem pozícióját, ha nincs benne -1-et ad vissza  
  
val count = Collections.frequency(numbers, 2) //hányszor tartalmazza az adott számot
```

- Diszjunkt-ság ellenőrzése → Megmondja, hogy van e közös elem a két tömbben

```
val value = Collections.disjoint(numbers, numbers2)  
println("Two lists are disjoint: $value")
```