Kotlin Újdonságok

Kotlin 1.3-M2 Early Access Preview

Balogh Tamás
Tech lead @ AutSoft

Újdonságok a Kotlin 1.3 előzetes verziójában (2018.Augusztus)

Stabilitás

- Coroutine – Végre fix!

Új nyelvi elemek (Experimental)

- Unsigned típusok
- Inline classok
- Contractok

Unsigned Types

```
@ExperimentalUnsignedTypes
fun tryUnsigned() {
    val unsignedInteger=42u
    val unsignedLong=11112uL
    val unsignedByte=0xFFu
    val unsignedBinary=0b011001100u
    val unsignedShort=65535u

    val serialNumber : ULong =123_456_789L.toULong()
    val magicBytes : UByteArray = ubyteArrayOf(0x00u,0xF0u,0x0Fu,0xFFu)

    Log.d("Kotlin1.3","$unsignedInteger $unsignedLong $unsignedByte " +
            "$unsignedBinary $unsignedShort $serialNumber $magicBytes")
}
```

Inline Classok

- Inline function – Ilyen már létezik Kotlinban
  - A függvény hívás helyre a függvény tartalma kerül

```
inline fun logTime(block: () -> Unit) {
    Log.d("Start", "Started at ${Date(System.currentTimeMillis())}")
    block.invoke()
    Log.d("End", "Ended at ${Date(System.currentTimeMillis())}")
}
```

Inline Classok

- Inline function – Ilyen már létezik Kotlinban
  - A függvény hívás helyre a függvény tartalma kerül

```
class HeavyComputation{

    init {
        logTime{
            doHeavyStuff()
        }

    }

    private fun doHeavyStuff(){
        //...
    }

}
```

Inline Classok

- Inline function – Ilyen már létezik Kotlinban
  - A függvény hívás helyre a függvény tartalma kerül

```
public HeavyComputation() {
    Log.d( tag: "Start",  msg: "Started at " + new Date(System.currentTimeMillis()));
    this.doHeavyStuff();
    Log.d( tag: "End",  msg: "Ended at " + new Date(System.currentTimeMillis()));
}
```

Inline Classok

- Inline class
  - Egy másik osztályt burkol, ami a helyére kerül fordításkor
  - Nem kell feleslegesen egyparaméteres osztályokat létrehozni
  - pl: API kommunikáció során saját transzparens wrapper osztályok

```
@Suppress("EXPERIMENTAL_FEATURE_WARNING")
inline class Token(val value:String)

@Suppress("EXPERIMENTAL_FEATURE_WARNING")
inline class DeviceID(val value:String)

@Suppress("EXPERIMENTAL_FEATURE_WARNING")
inline class NickName(val value:String)
```

```
data class PushRequest(
        val fcmToken: Token,
        val deviceID: DeviceID,
        val deviceNickName: NickName)
```

```
PushRequest(Token( value: "APAv2..f23"), DeviceID( value: "e32f..a87"),NickName( value: "Pixel 3 XL"))
```

Inline Classok

- Inline class
  - Egy másik osztályt burkol, ami a helyére kerül fordításkor
  - Nem kell feleslegesen egyparaméteres osztályokat létrehozni
  - pl: API kommunikáció során saját transzparens wrapper osztályok

```
public final class PushRequest {
    @NotNull
    private final String fcmToken;
    @NotNull
    private final String deviceID;
    @NotNull
    private final String deviceNickName;
```

```
new PushRequest( fcmToken: "APAv2..f23",  deviceID: "e32f..a87",  deviceNickName: "Pixel 3 XL");
```

Inline Classok

- Inline class
  - Egy másik osztályt burkol, ami a helyére kerül fordításkor
  - Nem kell feleslegesen egy paraméteres osztályokat létrehozni
  - API kommunikáció során saját transzparens wrapper osztályok
  - Probléma: Függvény szignatúra ütközések!

```
data class PushRequest(
        var fcmToken: Token,
        var deviceID: DeviceID,
        var deviceNickName: NickName) {

    fun save(token: Token)  { this.fcmToken = token }
    fun save(deviceId: DeviceID)  { this.deviceID=deviceId }
    fun save(deviceNickName: NickName) { this.deviceNickName=deviceNickName }
}
```

Inline Classok

- Inline class
  - Egy másik osztályt burkol, ami a helyére kerül fordításkor
  - Nem kell feleslegesen egy paraméteres osztályokat létrehozni
  - API kommunikáció során saját transzparens wrapper osztályok
  - Probléma: Függvény szignatúra ütközések!

```
public final void save_dmiujwd0/* $FF was: save-dmiujwd0*/(@NotNull String token) {
   Intrinsics.checkParameterIsNotNull(token, paramName: "token");
   this.fcmToken = token;
}

public final void save_c83uf5c8/* $FF was: save-c83uf5c8*/(@NotNull String deviceId) {
   Intrinsics.checkParameterIsNotNull(deviceId, paramName: "deviceId");
   this.deviceID = deviceId;
}

public final void save_d1voxx53/* $FF was: save-d1voxx53*/(@NotNull String deviceNickName) {
   Intrinsics.checkParameterIsNotNull(deviceNickName, paramName: "deviceNickName");
   this.deviceNickName = deviceNickName;
}
```

Contractok (Szerződések)

- Szerződés a fejlesztő és a fordító között
- A fejlesztő vállal megkötéseket
- A fordító így okosabb (~belelát a belső működésbe)

```
fun logTime(block: () -> Unit) {
    Log.d("Start", "Started at ${Date(System.currentTimeMillis())}")
    block.invoke()
    Log.d("End", "Ended at ${Date(System.currentTimeMillis())}")
}
```

Contractok (Szerződések)

- Szerződés a fejlesztő és a fordító között
- A fejlesztő vállal megkötéseket
- A fordító így okosabb (~belelát a belső működésbe)

```
class KontractExample {

    init {
        val magicNumber:Int
        logTime {
            magicNumber=calculateMagicNumber()
        }
        Log.d("Contract", "Magic number is $magicNumber")
    }

    private fun calculateMagicNumber() = 42
}
```

Contractok (Szerződések)

- Szerződés a fejlesztő és a fordító között
- A fejlesztő vállal megkötéseket
- A fordító így okosabb (~belelát a belső működésbe)

```
class KontractExample {

    init {
        val magicNumber:Int

        btnCalculate.setOnClickListener {
            magicNumber=calculateMagicNumber()
        }
        Log.d("Contract", "Magic number is $magicNumber")
    }

    private fun calculateMagicNumber() = 42
}
```

Contractok (Szerződések)

- Szerződés a fejlesztő és a fordító között
- A fejlesztő vállal megkötéseket
- A fordító így okosabb (~belelát a belső működésbe)
- callsInPlace contract

```
@ExperimentalContracts
fun logTime(block: () -> Unit) {
    contract { callsInPlace(block, EXACTLY_ONCE) }
    Log.d("Start", "Started at ${Date(System.currentTimeMillis())}")
    block.invoke()
    Log.d("End", "Ended at ${Date(System.currentTimeMillis())}")
}
```

Contractok (Szerződések)

- Szerződés a fejlesztő és a fordító között
- A fejlesztő vállal megkötéseket
- A fordító így okosabb (~belelát a belső működésbe)
- callsInPlace contract

```
@ExperimentalContracts
class KontractExample {

    init {
        val magicNumber:Int

        logTime {
            magicNumber=calculateMagicNumber()
        }
        Log.d("Contract", "Magic number is $magicNumber")
    }

    private fun calculateMagicNumber() = 42
}
```

Contractok (Szerződések)

- Szerződés a fejlesztő és a fordító között
- A fejlesztő vállal megkötéseket
- A fordító így okosabb (~belelát a belső működésbe)
- Returns -> Implies contract

```
@Throws(IllegalStateException::class)
private fun checkIfNotZero(input:Int?){
    if(input==null) throw IllegalStateException("Input is null")
    else if(input==0) throw IllegalStateException("Input is zero")
}
```

Contractok (Szerződések)

- Returns -> Implies contract

```
@ExperimentalContracts
class ReturnsExample {

    init {
        var state:Int?=null

        //Do something

        state=getState()
        checkIfNotZero(state)

        val uState : UInt =state.toUInt()
    }

    private fun getState(): Int?=2
```

Only safe (?.) or non-null asserted (!!.) calls are allowed on a nullable receiver of type Int?

Contractok (Szerződések)

- Returns -> Implies contract

```
@ExperimentalContracts
@Throws(IllegalStateException::class)
fun checkIfNotZero(input:Int?){
    contract { returns() implies(input !=null)}
    if(input==null) throw IllegalStateException("Input is null")
    else if(input==0) throw IllegalStateException("Input is zero")
}
```

Contractok (Szerződések)

- Returns -> Implies contract

Kedvet kaptam – Hogyan próbálhatom ki?

- Android Studio – Preferences
  - Languages & Frameworks
  - Kotlin updates
  - Update Channel = Early Access Preview 1.3
  - Install
- Project Gradle
  - ext.kotlin_version = **'1.3-M2'**
  - maven { url "http://dl.bintray.com/kotlin/kotlin-eap"}

További olvasnivaló

https://blog.jetbrains.com/kotlin/2018/08/kotlin-1-3-m2/

https://github.com/Kotlin/KEEP/blob/master/proposals/kotlin-contracts.md


github.com/BaloghTamas


balogh.tamas@autsoft.hu

Kérdések? (Majd inkább sörözés közben :D)

Köszönöm a figyelmet!

github.com/BaloghTamas

balogh.tamas@autsoft.hu