

Lateinit

Kotlin programlama dilinde birden çok değişken ve değişken tanımlama türleri vardır. Bunlar dan bazıları var,val,lateinit,lazy , getter&setters .. olarak çoğaltılabilir.

Lateinit değişkenini incelemek istersek ; Türkçe anlamından da anlamamız gerekirse geç başlangıç anlamına gelir. Geç başlangıç demek kısaca “ seni şuan tanımlıyorum, fakat ben sana bir şey verene kadar sadece bekle değerini boşta bırakıyorum 😊 ” demektir. Eğer bu değişkeni kullanacağınız garanti ise fakat vereceğiniz, atayacağınız değer hakkında emin değilseniz bu işi lateinit ile yapabiliyoruz. Programımıza dahil ettiğimiz zaman lateinit başlatılana kadar bellekte ona yer ayrılmaz. Bir lateinit her zaman var ya da NOT val değişkeni kullanmalıdır. Ufak bir örnek yapalım :

```
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
        Kontrol().lateinit()  
    }  
    class Kontrol{  
        lateinit var name :String  
        fun lateinit(){  
            name = "Buse"  
            if (this::name.isInitialized){  
                println("late init başlatıldı")  
            }else{  
                println("late init başlatılmadı")  
            }  
        }  
    }  
}
```

busedemir.basic I/System.out: late init başlatıldı

```
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
        Kontrol().lateinit()  
    }  
    class Kontrol{  
        lateinit var name :String  
        fun lateinit(){  
            if (this::name.isInitialized){  
                println("late init başlatıldı")  
            }else{  
                println("late init başlatılmadı")  
            }  
        }  
    }  
}
```

busedemir.basic I/System.out: late init başlatılmadı

Solda ki resimde lateinit tanımlandığından ve başlatıldığından emin olmamız içinde isInitialized ile if bloğunda kontrol ettiğimizde başlatıldı çıktısını alıyoruz. Fakat sağdaki resimde late init tanımlı olmadığından başlatılmadı çıktısını alıyoruz. Bu kontrolü yapmak aslında önemlidir. Çünkü lateinit başlatılmadığı takdirde bize tatlı bir “UninitializedPropertyAccessException: lateinit property courseName has not been initialized 😊” hatası verecektir. Bu yüzden çok değişkenli karışabilecek durumlarda bu kontrolün yapılması kurtarıcı olabilecektir.

- Bir argümanın eğer null gelme ihtimali varsa kesinlikle nullable kullanılmalıdır. Late init null olamaz ve sen kullanmaya başladığın ana kadar bekler. Bu açıdan database den veri çekerken daha güvenli gözükebilir. Ancak null bir değer geldiğinde late init null gelebilecek verileri engeller,backend de bekletir. Eğer bir argümanın null gelme ihtimalinin olmadığından kesin olarak eminseniz lateinit kullanmalısınız.

ToolNamespace Nedir : Araçlar Alan Adı olarak tanımlanır. Emülatör deki görünümümüze uygulayabileceğimiz işlevleri bildiren layout componentlerin çalışması için gereken bir XML layout alanıdır. Tasarım zamanında bir layoutun bileşenin hangi düzeninde gösterilmesini sağlayan bir çok XML niteliklerini destekleyen yapıdır.