**Мобайлын асуултууд хариултууд 2**

**Дараах код нь number хувьсагч 25-тай тэнцүү байвал "Divisible by 5" гэж хэвлэнэ.**

if (number % 10 == 0) {

println("Divisible by 10")

} else if (number == 5) {

println("Divisible by 5")

}  
**false**

**x = 5 үед дараах нөхцлүүдийн аль нь биелэх вэ?**  
Олон зөв хариултыг сонгож болно.

* x == 5 ✅
* x in 1..5 ✅
* x is Int✅
* x % 5

**Дараах ойлголтуудаас алийг нь объект хандлагат програмчлалын үндсэн ойлголт гэж үзэхгүй вэ?**

* **Абстракци (Abstraction)**
* **Унших чадвар (Readability)** ✅
* **Өвлөх (Inheritance)**
* **Полиморфизм (Polymorphism)**

**Kotlin хэлний дөрвөн харагдах байдлын (visibility) тодорхойлогч аль нь вэ?**

* public, private, protected, abstract
* static, override, internal, external
* **private, protected, public, internal** ✅ (Зөв хариулт)
* public, protected, static, internal

**Өвлөгч ангийн (parent class) функцыг дуудахдаа ямар түлхүүр үг ашигладаг вэ?**

* this
* **super** ✅ (Зөв хариулт)
* parent
* self

**\_\_\_ нь анги (class)-д заавал хэрэгжүүлэх ёстой шинж чанар болон функцүүдийг тодорхойлдог.**

* Delegate
* Generic type
* **Interface** ✅ (Зөв хариулт)
* Subclass

**Дараах сонголтуудаас nullable төрөлд хамгийн сайн тохирох нь аль вэ?**

* **Нийгмийн сүлжээний апп дээрх дагагчдын тоо (0 эсвэл түүнээс олон).**
* **Сонголтын дагуу оруулж болох профайл зураг.** ✅ (Зөв хариулт)
* **Хэрэглэгчийн нэр, хамгийн багадаа нэг тэмдэгт байх ёстой.**
* **Бүх хэрэглэгчдэд өгөгдсөн дахин давтагдахгүй ID.**

**\_\_\_ операторыг ашигласнаар объект null биш бол тухайн объектын функцийг дуудах боломжтой.**

* .
* !!
* ?:
* **?.** ✅ (Зөв хариулт)

**Kotlin хэлний функцийн тухай аль нь үнэн биш вэ?**

* **Функцийг өөр өгөгдлийн төрөл болгон хувиргаж болно.** ❌ (Зөв хариулт, буруу ойлголт)
* Функц өөр нэг функцээс буцаж ирж болно.
* Функц өөр нэг функцыг параметр болгон авч болно.
* Функц тодорхой өгөгдлийн төрөлтэй байдаг, жишээ нь (Int) -> Unit.

**Функцийн literal гэдэг нь \_\_\_ гэсэн өөр нэртэй.**

* Function type
* **Lambda expression** ✅ (Зөв хариулт)
* Function reference
* Trailing lambda

1️.**Зураг харуулахын тулд ямар Composable ашиглах вэ?**

* ✅ **Image**

2️.**Alignment.Center нь UI элементүүдийг хөндлөн ба босоо тэнхлэгт төвлөрүүлдэг.**

* ✅ **Үнэн (True)**

3️.**Composable функцүүд объект хадгалахын тулд ямар Composable ашигладаг вэ?**

* ✅ **remember**

4️.**Debugger нь код гүйцэтгэх явцыг зогсоосон үед хувьсагчдыг шалгах боломж олгодог.**

* ✅ **Үнэн (True)**

5️.**Composable функцэд \_\_\_ ашигласнаар хувьсагчдыг утга өөрчлөгдөхөд дахин зурах (recomposition) үйлдэл хийх байдлаар ажиглагдах (observable) болгодог.**

* ✅ **mutableStateOf**

6️.**\_\_\_ composable нь доторх элементүүдийг босоо дарааллаар байрлуулдаг.**

* ✅ **Column**

7️.**\_\_ debugger-ийн функц нь дуудагдсан функцүүдийн дарааллыг ухрах боломжийг олгодог.**

* ✅ **Step out**

Jetpack Compose нь таны composable-уудыг анх удаа ажиллуулах үед, \_\_\_ үе шатанд таны UI-г тодорхойлсон composable-уудыг хянах болно.

* **Анхны зохиомж (Initial composition)** ✅
* Дахин зохиомж (Recomposition)
* Төлөвийн өөрчлөлт (State change)
* Апп хаагдах (App termination)

Зохиомжийг өөрчлөх цорын ганц арга нь дахин зохиомж хийх явдал юм.

* **Үнэн (True)** ✅
* Худал (False)

\_\_\_ нь Jetpack Compose өгөгдлийн өөрчлөлтөд хариу үйлдэл үзүүлж, өөрчлөгдөж болзошгүй composable-уудыг дахин гүйцэтгэх үйл явц юм.

* **Дахин зохиомж (Recomposition)** ✅
* Анхны зохиомж (Initial composition)
* Төлөвийн өөрчлөлт (State change)
* Апп хаагдах (App termination)

\_\_\_ нь апп-д цаг хугацааны явцад өөрчлөгдөж болох аливаа утга юм.

* **Төлөв (State)** ✅
* Утга (value)
* Утга өөрчлөгдөх (valueChange)
* Төлөв утга (StateValue)

\_\_\_ нь компонентыг төлөвгүй (stateless) болгохын тулд төлвийг дээш нь шилжүүлэх загвар юм.

* **Төлөв өргөлт (State hoisting)** ✅
* Төлөвийн өөрчлөлт (State change)
* Зохиомж өргөх (Hoist composition)
* Дахин зохиомж (Recomposition)

Ямар **KeyboardAction** шинж чанарыг дараагийн composable руу анхаарлыг шилжүүлэхэд ашигладаг вэ?

* **onNext**✅
* onDone
* onGo
* onSend

Kotlin-д Double эсвэл Float тоог дээш нь бүхэл тоо руу тоймлоход ашигладаг функц аль нь вэ?

* **kotlin.math.ceil()**✅
* kotlin.math.ceilUp()
* kotlin.math.roundDown()
* kotlin.math.roundUp()

**Layout Inspector** нь Jetpack Compose дахь байршлыг эмулятор эсвэл физик төхөөрөмж дээр ажиллаж буй апп-аас шалгах боломжтой хэрэгсэл юм.

* **Үнэн (True)** ✅
* Худал (False)\*\*

UI тестүүд \_\_\_ сан дахь файлуудад хадгалагддаг.

* **androidTest**✅
* main
* test
* res

Локал тест болон UI тестүүдийг \_\_\_ аннотациар тодорхойлох ёстой.

* **@Test**✅
* @VisibleForTesting
* @Preview
* @Composable