MOBILFEJLESZTÉS

1. forduló

autsoft

A kategória támogatója: AutSoft Zrt.

Ismertető a feladathoz

Kérjük, hogy a feladatlap indítása előtt mindenképp olvasd el az alábbi útmutatót:

- MINDEN kérdésre **van helyes válasz**.
- Olyan kérdés **NINCS**, amire az összes válasz helyes, ha mégis az összes választ bejelölöd, arra a feladatra automatikusan 0 pont jár.
- A **radio button-os kérdésekre egy helyes válasz van**.
- **Ha lejár a feladatlap ideje, a rendszer AUTOMATIKUSAN** beküldi azt az addig megjelölt válaszokkal.
- Azokat a feladatlapokat, amelyekhez **csatolmány** tartozik, javasoljuk **NEM mobilon** elindítani, erre az érintett feladatlapok előtt külön felhívjuk a figyelmet.
- Az **adatbekérős feladatokra NEM jár részpontszám**, csak a feleletválasztósakra.
- **Helyezéseket a 4. forduló után mutatunk**, százalékos formában: adott kategóriában a TOP 20-40-60%-hoz tartozol.
- **Badge-ket** szintén a 4.forduló után kapsz majd először.
- Ha egyszerre több böngészőből, több ablakban vagy több eszközről megnyitod ugyanazt a feladatlapot, **nem tudjuk vállalni** az adatmentéssel kapcsolatban esetlegesen felmerülő anomáliákért a felelősséget!
- A hét forduló során az egyes kategóriákban (de nem feltétlenül mindegyikben) **könnyű-közepes-nehéz kérdésekkel** egyaránt találkozhatsz majd.

Jó versenyzést kívánunk!

Felhasznált idő: 00:00/20:00 **Elért pontszám:** 0/15

1. feladat 0/1 pont

Az alábbiak közül melyik **NEM** Android alkalmazás komponens?

Válasz

- (●) Intent
  Ez a válasz helyes, de nem jelölted meg.
- ( ) Service
- ( ) ContentProvider
- ( ) Activity

Magyarázat

Az Intent egy végrehajtandó művelet absztrakt leírása. Használható a startActivity egy Activity indítására, a broadcastIntent, hogy elküldje a művelet leírását bármely figyelő BroadcastReceiver komponensnek és a Context.startService(Intent) vagy Context.bindService(Intent, ServiceConnection, int), hogy kommunikáljon a háttérszolgáltatásokkal.

2. feladat 0/2 pont

Az Apple ökoszisztémában mit nevezünk “Complication”-nek?

Válasz

- ○ Amikor két alkalmazás komponens egymást negatívan befolyásolja
- ◉ Az Apple Watchon elérhető “widgeteket”

  Ez a válasz helyes, de nem jelölted meg.
- ○ Ha az adott csomagkezelővel nem sikerül feloldani minden függőséget
- ○ Ez egy szinonima a szemantikai hibákra

Magyarázat

Az Apple Watch Complication az alkalmazásokból származó apró információk, amelyek az óra számlapján jelennek meg. A különböző óralapok, Apple Watch-modellek és a watchOS verziói különféle komplikációkat támogatnak, az alkalmazásfejlesztők pedig egyedi specifikációk alapján építik fel.

3. feladat 0/6 pont

Melyik kódrészlet írja ki a következőt?

```
true false
```

Válasz

- ○
```
fun main() {
    val a = BigDecimal( val: "12.00")
    val b = BigDecimal( val: "12.0")
    println((a <= b) && (a >= b))
    println(a != b)
}
```
- ○
```
fun main() {
    val a = BigDecimal( val: "12.00")
    val b = BigDecimal( val: "12.0")
    println((a >= b) && (a <= b))
    println(a != b)
}
```

```
fun main() {
    val a = BigDecimal( val: "12.00")
    val b = BigDecimal( val: "12.00")
    println((a >= b) && (a >= b))
    println(a == b)
}
```

```
fun main() {
    val a = BigDecimal( val: "12.00")
    val b = BigDecimal( val: "12.0")
    println((a >= b) && (a <= b))
    println(a == b)
}
```

Ez a válasz helyes, de nem jelölted meg.

Magyarázat

A >= és <= operátorok compareTo() operátor szimbólumai, míg a == az equals() operátoré. Az equals() operátor figyelembe veszi a tizedesjegyek számosságát is, míg a compareTo() nem.

4. feladat 0/2 pont

Mi a különbség a témák és stílusok között Android platformon?

Válasz

- A témák nem származhatnak más témákból, de a stílusok igen más stílusokból
- A témák a teljes alkalmazásra vonatkoznak, a stílusok csak adott felületelemre
  Ez a válasz helyes, de nem jelölted meg.
- A stílusok nem származhatnak más stílusokból, de a témák igen más témákból
- A témákat az Android biztosítja, míg a stílusokat a fejlesztő határozza meg

Magyarázat

A stílus attribútumok gyűjteménye, amelyek egyetlen felületelem megjelenését határozzák meg. A téma attribútumok gyűjteménye, amely egy teljes alkalmazásra, Activity-re vagy nézethierarchiára vonatkozik – nem csak egy egyedi felületelemre.

5. feladat 0/2 pont

Hogyan lehet kiíratni Swiftben a megadott érték típusát?

```
var x = 128
```

A print() függvény segítségével írassuk ki az x dinamikus típusát. Az output a következő:

```
Int
```

(Megjegyzés: amennyiben a felhasznált függvény paramétert várna, akkor a paraméter neve után azonnal kettőspont, ezt követően pontosan 1 space majd a paraméter értéke következzen. Zárójelek előtt, illetve után nem szabad space-t rakni.)

```
.frame(width: defaultWidth)
```

Válaszok

A helyes válasz:

print(type(of: x))

type(of: x)

Magyarázat

A megoldáshoz a type(of:) függvényt használjuk, amely egy érték dinamikus típusát adja vissza.

6. feladat 0/2 pont

Vannak-e Kotlin nyelvben primitív típusok, mint int, float, long?

Válasz

- (●) Nyelvi szinten nem. De a compiler használja őket a jobb teljesítményért
  Ez a válasz helyes, de nem jelölted meg.
- ( ) Igen, a Kotlin ebben a tekintetben hasonlít a Java-ra
- ( ) Nem, nincsenek a Kotlin nyelvben és nem is használ primitív típusokat
- ( ) Igen, de legbelül mindig a nem-primitív megfelelőjükre fordítja őket

Magyarázat

Kotlinnak nincs primitív típusa. Olyan osztályokat használ, mint az Int, a Float a primitívek objektum wrappereként. Amikor a Kotlin kódot JVM kódra alakítják, amikor csak lehetséges, a "primitív objektum" Java primitívvé alakul a jobb teljesítmény érdekében.