

Nevezze meg az iOS architektúra részeit!

Cocoa Touch, Media, Core Services, Core OS.

Ismertesse a Core OS réteg feladatait!

Hardverkezelés, biztonság, memória, fájlrendszer, low-level funkciók.

Ismertesse a Core Services réteg feladatait!

Hálózat, helymeghatározás, adatbázis, iCloud, fájlkezelés.

Ismertesse a Media réteg feladatait!

Grafika, animáció, hang, videó, képfeldolgozás.

Ismertesse a Cocoa Touch réteg feladatait!

UI-elemek, érintéskezelés, app életciklus.

Milyen programtervezési mintát használnak a UIKit-tel készült iOS alkalmazások?

MVC.

Nevezze meg a macOS operációs rendszer felhasználói felület részeit!

Menüsor, Dock, Finder, ablakok, Mission Control.

Xcode fejlesztőkörnyezettel milyen alkalmazások fejlesztése lehetséges?

iOS, macOS, watchOS, tvOS alkalmazások.

Xcode projekt létrehozásakor a Bundle Identifier miből jön létre?

Fordított domain + projekt neve.

iOS alkalmazások fejlesztéséhez milyen programozási nyelveket támogat az Xcode?

Swift, Objective-C.

Nevezze meg az Xcode felület részeit!

Navigator, Editor, Utilities/Inspector, Toolbar, Debug area.

Ismertesse az Xcode Navigator Pane feladatát!

Projektfájlok, hibák, keresés, tesztek, jelentések kezelése.

Ismertesse az Xcode Navigator Pane részeit!

Project, Symbol, Find, Issue, Test, Debug, Breakpoint, Report.

Ismertesse az Xcode Utilities Pane feladatát!

Kijelölt elem tulajdonságainak megjelenítése és módosítása.

Ismertesse az Xcode Editor feladatát!

Kód és UI szerkesztése.

Sorolja fel a SwiftUI tulajdonságait!

Deklaratív, állapotvezérelt, komponens-alapú, élő előnézet.

Mire használjuk a módosítókat SwiftUI-ban?

Nézetek stílusának és viselkedésének módosítására.

Mit jelent a „some View” SwiftUI-ban?

A visszatérő view típusa elrejtethető, csak azt jelzi: egy View-t adok vissza.

Mire használjuk a trailing closure szintaxist SwiftUI-ban?

Rövidebb kód írására függvény utolsó closure paraméterénél.

Nevezze meg az elrendezésekhez használt Stack View típusokat SwiftUI-ban!

HStack, VStack, ZStack.

Nyomógomb vezérlő milyen részekből áll SwiftUI-ban?

Action + Label.

Mire szolgál a @State property wrapper SwiftUI-ban?

A View saját helyi állapotának tárolására.

Mire szolgál a @FocusState property wrapper SwiftUI-ban?

Beviteli fókuszt (pl. TextField) állapotának kezelésére.

Mire szolgál a Form SwiftUI-ban?

Űrlapok mezőinek csoportosított megjelenítésére.

Melyik az a vezérlő, amivel egyszerre lehet címkét és tartalmat megjeleníteni?

Label.

Összetett View-kat van lehetőség kisebb részekre szétbontani SwiftUI-ban?

Igen, külön struct-okba.

Milyen alakzatok érhetők el SwiftUI-ban?

Circle, Rectangle, RoundedRectangle, Ellipse, Capsule.

Swiftben a függvények visszatérési értékét hogyan adjuk meg?

-> Típus.

Mit jelent Swiftben, hogy a függvények paramétereinek külső és belső nevük van?

Külső a hívónak, belső a függvényen belül használt név.

Swiftben mivel lehet a külső paraméternevet helyettesíteni?

_ (aláhúzás).

Mit jelent az UUID?

Egyedi univerzális azonosító.

Mit jelent az Identifiable?

Olyan típus, amely rendelkezik egyedi id mezővel.

Mire használjuk a ForEach-et SwiftUI-ban?

Elemek ismételt megjelenítésére.

Mit jelent az id: .self SwiftUI-ban?

Saját érték szolgál azonosítóként.

Mire használjuk a List-et SwiftUI-ban?

Görgethető lista készítésére.

Melyik vezérlőelemmel lehet többoszlopos listát létrehozni SwiftUI-ban?

LazyVGrid / LazyHGrid.

Mire szolgál a LazyVStack, LazyHStack SwiftUI-ban?

Nagy listák lusta (késeleltetett) betöltésére.

Mire használjuk a NavigationStack-et SwiftUI-ban?

Navigációs útvonalak, új képernyők kezelése.

Melyik módosítóval lehet a menüsávon szöveget megjeleníteni?

navigationTitle().

Melyik módosítóval lehet menüsávhoz gombokat hozzáadni?

toolbar().

Mire szolgál a megfigyelő tervezési minta?

UI automatikusan frissüljön állapotváltozáskor.

Melyik típustól kell örökölni, hogy megfigyelhető legyen?

ObservableObject.

Melyik property wrapper jelzi a figyelt tulajdonságokat?

@Published.

A megfigyelt objektumot melyik két wrapper-rel jelöljük a nézetben?

@ObservedObject vagy @StateObject.

Melyik típussal lehet felugró új felületet megjeleníteni SwiftUI-ban?

.sheet().

Mi a View definíciója?

Egy megjelenítendő UI-elem, amely a képernyőn látható tartalmat adja.

Milyen hatása van a View-k megjelenítésére, ha hierarchiába rendezzük őket?

A gyermek-view a szülő tulajdonságait örökli (pozíció, méret, átlátszóság), és együtt mozognak.

Lista megjelenítéséhez milyen vezérlőket használhatunk UIKit rendszerben?

UITableView, UICollectionView.

Alkalmazással érkező képi erőforrásoknak hány különböző felbontásban kell rendelkezésre állnia?

3 méretben: @1x, @2x, @3x.

Mit jelent az Outlet?

Kapcsolat a storyboard-on lévő UI-elem és a Swift kód között.

Mit jelent az Outlet Collection?

Több azonos típusú UI-elemet egy tömbbe kötünk össze.

Mit jelent az Action?

Eseménykezelő függvény (pl. gombnyomás).

Hogyan hozunk létre konstanst és változót Swift nyelvben?

Konstans: let, változó: var.

Ismertesse a String-ek és változók összefűzését Swift nyelvben!

"Szöveg \(változó)" vagy + operátorral.

Melyik függvénnyel írhatunk ki az Output panelra?

print().

Hogyan nevezzük a Table View egy sorát?

Cell (UITableViewCell).

Lehetséges a sorok csoportosítása Table View-ban?

Igen, szekciókkal (sections).

Table View adatforrás beállításához milyen protokollt kell implementálni?

UITableViewDataSource.

UITableViewDataSource protokoll mely két függvényét kötelező implementálni?

numberOfRowsInSection, cellForRowAt.

Table View interakciók eléréséhez milyen protokollt kell implementálni?

UITableViewDelegate.

UITableViewDelegate melyik függvénye reagál a sor kiválasztására?

didSelectRowAt.

Melyik típus vehet fel nil értéket?

Optional típus.

Ismertesse az Optional típus kicsomagolásának lehetőségeit!

Force unwrap !, optional binding if let, guard let, nil-koalescing ??.

Milyen típusú intervallumokat lehet létrehozni?

Zárt: a...b, fél-nyílt: a...<b.