Osnove iOS-a

• Odgovoriti na pitanja:

• Osnove iOS-a :

• Što je AppDelegate? Objasni njegovu funkcionalnost.

O: AppDelegate je root objekt aplikacije, klasa koja upravlja shared behavioursima i s UIApplication upravlja dijelom interakcija sa sistemom. Upravlja lifecycle metodama aplikacije. Funkcionalnost: inicijalizira centralne podatkovne strukture, konfigurira scene, odaziva se na notifikacije van aplikcaije i događaje namijenjene prema aplikaciji.

• Što je UIWindow? Objasni njegovu funkcionalnost.

O: UIWindow je objekt koji dispatcha i handlea evente za view. Funkcionalnost: prikazuje sadržajj aplikacije, hosta jedan ili više viewova

• Objasni ViewController. Objasni njegovu funkcionalnost. Navedi najčešće korištene ViewControllere.

O: ViewControlleri su klase koje reagiraju na korisnikovu interakciju i ažuriraju sadržaj viewova. Također, mijenjaju veličinu viewova i upravljaju layoutom interfacea te koordiniraju s drugim objektima (uključujući s view controllerima) u appu. UIViewController je root klasa, ali se uglavnom ne instancira on već pravi subklasa njega pa onda dodaju implementacije.. Često se koristi više view controllera od kojih svaki sadržio dio appovog UI-a. View controlleri najčešće korišteni su Content view controlleri, Container view controlleri, controlleri za search interface i activities interface.

• View:

• Objasni View. Navedi njegovu funkcionalnost.

O: laički, View je vizualni dio appa, ono što korisnik vidi na ekranu. View može hostat druge viewove, stvarajućući super i sub viewove.. Također, pomoću viewa možemo reagirati na touch i druge evente, drag-n-drop interakcije, focus change. UIView je root klasa svih viewova.

• Objasni frame.

O: Frame je opisuje lokaciju viewa i veličinu unutar superviewovog koordinantnog sustava.

• Objasni proces određivanje veličine view-a (constraint setup)

O: Tehnički, view se sastoji od 4 constrainta za veličinu, top bottom leading I trailing. Razmak između top i bottom je height, između leading I trailing width.

Auto Layout dinamički računa veličinu svih viewova u hijerarhiji ovisno o constraintima postavljenima na tim viewovima. Tri načina određivanja veličine view-a. 1. control-drag između viewova, pin and align alati i pustiti Interface Builder da postavi constrainte pa modificirati.

• Objasni View draw cycle.

O: Klasa UIView koristi model crtanja na zahtjev za predstavljanje sadržaja. Kada se view prvi put pojavi na zaslonu, sistem traži da nacrta njegov sadržaj. Sistem pravi snapshot sadržaja i koristi ga dokle god se sadržaj ne promijeni. Ako se promijeni, javljaš sistemu da se view promijenio, ali ne crtaju se promjene direktno nego se metodama setNeedsDisplay ili setNeedsDisplaInRect invalidira view i ponovno crtaju prvom prilikom.

• UI:

• Što je UIKit?

O: Framework koji pruža potrebnu infrastrukturu za iOS app: window i view arhitekturu za implementaciju interfacea, event handling infrastrukturu za multi-touch i druge tipove inputa i main run loop za za upravljanje interakcijom između korisnika, sistema i aplikacije.

• Navedi najćešće korištene elemente UIKit-a.

O: Elementi UIKit-a se dijele u tri kategorije: bars (navigacija), views (text, graphics, animations, interactive) I controls (buttons, witches, text fields, progress indicators)

• Objasni moguće načine konstruiranja UI-a. Navedi prednosti pojedinog načina.

O: Možemo programatski ili koristiti vizualno. Vizualno je lakše i brže, ali kod kompleksnijih projekata postaje zeznutije i neuredno, dok programatski isprva se čini kao više posla, ali kod kompleksnijih projekata bude uredniji i manje sklon nepotrebnim problemima.

• Objasni Storyboard.

O: Story board je vizualna reprezentacija user interfacea, služi da ubrza i olakša kreiranje UI-a. Može se definirati kao slijed zaslona od koji se svaki sastoji od view controllera i viewova, a tranzicija između storyboarda je tranzicija između dva view controllera (putem segue objekta)

• Objasni interface builder. Objasni povezivanje UI prikaza sa klasom.

O: Unutar Interface buildera se kreiraju user interfacei. iOS IDE Xcode dolazi s built-in interface builderom. Sastoji se od dva glavna dijela, dock i canvas. Dock je popis objekata koje se nalaze u user interfaceu, a u canvasu se radi. UI prikaz se povezuje s klasom tako što napišemo kod koji implementira ponašanje objekata user interfacea, a kod i ti objekti komuniciraju preko action i outlet konekcija.

• Objasni outlete.

O: Outleti su način na koji kod pristupa Interface Builder objektima u storyboardu te se koristi kada treba poslati poruku s koda na UI objekt.

• Objasni IBDesignable/ IBInspectable. Objasni primjenu.

O: IBDesignable pruža funkcionalnost za live rendering promjena u viewu direktno u storyboardu ili .xib. IBInspectable omogućuje kreiranje atributa u kodu koje dajemo storyboardu ili .xib fileu. Koriste se za pravljenje custom viewova.

• Na primjeru objasni dodavanje click eventa na određeni view. Obavezno objasni programatski način, IB način je bonus :)

O: Primjer se odnosi na dodavanje klika na button. U funkciju viewDidLoad() dodati instancu UIButton (ili u viewDidLoad() dodati funkciju initUI() ili tako nešto pa u nju UIButton), postaviti atribute i dodat ga u subview.

override func viewDidLoad() {

super.viewDidLoad()

let btn = UIButton()

btn.setTitle(“Button”, forState: .Normal)

self.view.addSubview(btn)

}

Onda napraviti button action metodu:

func onClick(sender: UIButton) {

print(“Button click”)

}

Pozvati UIButton addTarget metodu i povezati button s onClick eventom (to unutar viewDidLoad()):

btn.addTarget(self, action: #selector(self.onClick(\_:)), forControlEvents: .TouchUpInside)

A za IB, možemo klik+ctrl i dragat button iz storyboarda u kod i iskočiti će popup u kojem možemo nazvati metodu, odabrati event, konekciju i argument. Na kraju kliknemo Connect.

• Što je first responder? Što on kontrolira? Zbog čega se prati ovaj status?

O: First responder je obično prvi objekt u nekom responder lancu koji primi event ili action, a obično first responder buse view objekt koji user klikne ili aktivira. Kontrolira što će imati fokus.

• Objasni constraintse.

O; Constraintsi opisuju odnos između dva itema user interfacea ili dva atributa istog itema. Služe da se objekti mogu prilagođavati viewu prilikom promjene.

• Za slijedeći view opiši kako bi postavio costrainse. Za opis možeš koristiti programsku sintaksu sa anchorima ili navesti kako bi to napravio unutar interfacebuilder-a.

• Primjer: Potrebno je pozicionirati dva view-a, jedan pokraj drugoga. Viewovi moraju centrirani unutar parenta te moraju biti jednako široki.

O: Unutar interfacea: dodam dva buttona unutar istog viewa, oznacim oba, napravim stack view, kliknem na Add new alignment constraint i označim oba Vertically i Horizontally in Container.

• Viewcontroller:

• Navedi lifecycle metode ViewControllera. Opiši što pojedina metoda radi te kada će biti pozvana.

O:

⁃ loadView(): koristi se kada je view controller kreiran iz koda, kada je current view nil. Najbolje ništa ne raditi unutar te metode.

⁃ viewDidLoad(): pozvan kad je view spreman (loadan). U njoj raditi neke uobičajene taskove (network call, UI, jednokratne callove)

⁃ viewWillAppear(): pozvano svaki put prije nego što je view visible. Može se overrideat i utjecat na kako će se view prikazat, tipa sakriti neke stvari

⁃ viewWillLayoutSubviews(), viewDidLayoutSubviews()

⁃ viewDidAppear(): kad se pojavi view na zaslonu. Spremanje podataka, animacije, rad s networkom itd.

⁃ viewWillDisappear(): netom prije nego što se view makne iz view hijerarhije. unloadanje animacija, cancelanje network zahtjeva, saveanje stanja…

⁃ viewDidDisappear(): kada je maknut iz hijerarhije. Micanje listenera.

⁃ deinit(): prije nego što se view controller obriše iz memorije, deinicijalizira se. Koristi se da se očiste resurse koje ARC ne očisti sam.

⁃ didReceiveMemoryWarning(): upozorenje za punu memoriju

⁃ viewWillTransition(to:with:): kada se promijeni interface orijentacija.

• Opiši proces izmjene kontrolera. Objasni programatski način. Dodatno : objasni seque.

O:

• Objasni NavigationController. Objasni njegovu ulogu.

O: Container view controller koji upravlja jednim ili više view controllerima u navigacijskom interfaceu. Upravlja tim view controllerima koristeći ordered array (navigation stack). Prvi view controller u arrayu je root view controller, a zadnji je na vrhu stacka i predstavlja view controller koji je trenutno prikazan. Navigation controller upravlja navigation barom na interfaceu i ažurira navigation bar koristeći sadržaj od child view controllera, a koordinira sve preko delegata.

• Objasni TableViewController. Na primjeru objasni prednosti njegovog korištenja.

O: TableViewController upravlja table viewovima i sadrži protokole potrebne za upravljanje sadržajem i respondanje na promjene. Ako sam dobro shvatio, TableView je ekvivalent Androidovog RecyclerViewa, onda bi

TableViewController bio Adapter?

• Objasni kako funkcionira TableViewController.

O: Stvori se podklasa UITableViewController za svaki table view. Kada se inicijalizira, mora se navesti stil prikaza tablice i overrideat data source i delegate metode da bi napunili tablicu s podacima.

• Objasni reusanje cellova.

O: Reusanje cellova je dobro radi performance managementa. Ako je neki table view cell reusable (a to se zna po identifieru), UITableView objekt održava queue ili listu reusable cellova i čini ih dostupnima delegatu. Metoda prepareForReuse() se potom koristi da delegat pripremi cell za reuse, a dequeueReusableCell(withIdentifier:) vraća reusable table-view cell objekt.

• Što je delegat?

O: Delegat je objekt koji djeluje u ime drugog objekta ili u koordinaciji s drugim objektom kada taj objekt naiđe na event u programu. Delegat prima poruke od objekta koji delegira i potom ažurira izgled ili stanje sebe ili drugih objekata u aplikaciji, ili vraća neku vrijednost.