

Program kursu Mobile Developer PRO

WPROWADZENIE DO JĘZYKA KOTLIN

- Podstawy składni
- Typy bazowe
- Operatory
- Instrukcje sterujące
- Kolekcje
- Praca z tekstem
- Klasy, interfejsy, wyliczenia, rozszerzenia
- Funkcje i domknięcia
- Programowanie obiektowe
- Obsługa wyjątków
- Typy generyczne
- Zarządzanie pamięcią
- Istotne idiomy programistyczne
- Interoperacyjność z Java

PRACA Z PROJEKTEM ANDROID

- Podstawy użycia Android Studio
- Tworzenie, konfigurowanie i budowanie projektu
- Uruchamianie aplikacji na urządzeniu oraz emulatorze
- Debugowanie kodu
- Praca z dokumentacją
- Zarządzanie zależnościami w oparciu o system Gradle

PODSTAWY TWORZENIA APLIKACJI

- Struktura i najważniejsze elementy projektu
- Zarządzanie zasobami
- Komponenty aplikacji: Activities, Services, Content Providers, Broadcast receivers,
- Intents
- Wprowadzenie do Android Jetpack
- Architektura MVVM w oparciu o Architecture Components

BUDOWANIE INTERFEJSU UŻYTKOWNIKA

- Wprowadzenie do Material Design
- Aktywności i fragmenty - wykorzystanie, cykl życia, bindowanie kontrolek, obsługa zdarzeń
- Zarządzanie rozkładem elementów (layouts managers)
- Wykorzystanie dostępnych kontrolek
- Elastyczne zarządzanie rozkładem z ConstraintLayout
- Zaawansowane elementy widoku na przykładzie RecyclerView
- Programowe budowanie widoków
- Nawigacja między ekranami

PROGRAMOWANIE REAKTYWNE

- Wprowadzenie do RxJava/RxKotlin
- Publikowanie i nasłuchiwanie zdarzeń
- Operatory - transformacja, filtrowanie, kombinowanie, czas
- Sterowanie kontekstem wykonania z użyciem Schedulers
- Praca ze strumieniami

UTRWALANIE I DOSTĘP DO DANYCH W APLIKACJACH ANDROID

- User preferences
- Praca z plikami
- SQLite i biblioteka Room

KOMUNIKACJA W APLIKACJACH ANDROID

- Coroutines i praca z kodem asynchronicznym
- Integracja z REST API oraz GraphQL
- Wykorzystanie przykładowych bibliotek: Retrofit, OkHttp, Moshi, Picasso

DYSTRYBUCJA APLIKACJI ANDROID

- Wykorzystanie klas i interfejsów
- Specjalne typy klas (Data, Sealed, Nested, Enum, Inline)
- Hermetyzacja i kontrola dostępu
- Delegacja na poziomie klas (classes) i własności (properties)
- Rozszerzenia (extensions)
- Typy generyczne

ARCHITEKTURA APLIKACJI ANDROID

- › Czym jest architektura i jakie ma znaczenie
- › Spójność, niskie sprzężenie, modułowość
- › Wstrzykiwanie zależności z użyciem Dagger / Hilt
- › Czysta architektura
- › Architektura w praktyce - implementacja wzorców Model-View-ViewModel, Model-View-Intent

TESTOWANIE APLIKACJI ANDROID

- › Dlaczego testowanie jest ważne?
- › Kod niskiej jakości a testowanie
- › Architektura aplikacji a testowanie
- › Piramida testów
- › Cechy dobrych testów
- › Metryki pokrycia testami
- › Przygotowanie i konfiguracja środowiska
- › Testy jednostkowe
- › Testy integracyjne
- › Testy na poziomie UI z użyciem Espresso

ZAAWANSOWANE ASPEKTY TWORZENIA INTERFEJSU UŻYTKOWNIKA

- › Budowanie niestandardowych kontrolerek
- › Motywy aplikacji i zarządzanie stylami
- › Rysowanie z użyciem obiektu Canvas
- › Rola animacji w projektowaniu UI
- › Wykorzystanie animacji
- › Zarządzanie stanem oraz jego efektywne współdzielenie

KOTLIN COROUTINES

- › Techniki programowania asynchronicznego
- › Podstawy pracy z coroutines
- › Zasięg i kontekst wykonania
- › Komponowanie zadań
- › Obsługa wyjątków
- › Współdzielenie stanu
- › Channels
- › Flows
- › Porównanie do RxJava

WYBRANE ELEMENTY ANDROID SDK

- Lokalizacja GPS i geokodowanie z Google Play Services
- Praca z mapami
- Kamera i biblioteka zdjęć
- Sensors API

WPROWADZENIE DO JĘZYKA SWIFT

- Podstawy składni
- Typy bazowe
- Operatory
- Instrukcje sterujące
- Kolekcje
- Praca z tekstem
- Struktury, klasy, protokoły, wyliczenia, rozszerzenia
- Funkcje i domknięcia
- Programowanie obiektowe
- Obsługa wyjątków
- Typy generyczne
- Zarządzanie pamięcią

PRACA Z PROJEKTEM XCODE

- Podstawy użycia Xcode
- Tworzenie, konfigurowanie i budowanie projektu
- Uruchamianie aplikacji na urządzeniu oraz symulatorze
- Debugowanie kodu
- Praca z dokumentacją
- Zarządzanie zależnościami w oparciu o CocoaPods i Swift Package Manager

UIKIT - KLASYCZNE BUDOWANIE INTERFEJSU

- View Controllers - wykorzystanie, cykl życia, obsługa zdarzeń
- Interface builder / Storyboards
- Wykorzystanie podstawowych kontrol
- Elastyczne zarządzanie rozkładem - Auto Layout, Trait Collections, Size Classes
- Zaawansowane elementy widoku - TableView, CollectionView, ScrollView
- Programowe budowanie widoków
- Nawigacja między ekranami

SWIFTUI - DEKLARATYWNE BUDOWANIE INTERFEJSU

- Typ View jako sposób na tworzenie reużywalnych elementów widoków
- Kontrolki wbudowane
- Zarządzanie rozkładem w oparciu o dostępne kontenery
- Bindowanie, reagowanie na zdarzenia i obsługa gestów
- Stylizacja z wykorzystaniem modyfikatorów
- Warunkowe renderowanie interfejsu
- Nawigacja między ekranami

PROGRAMOWANIE REAKTYWNE

- Wprowadzenie do Combine framework
- Publikowanie i nasłuchiwanie zdarzeń
- Operatory - transformacja, filtrowanie, kombinowanie, czas
- Sterowanie kontekstem wykonania z użyciem Schedulers
- Praca ze strumieniami

UTRWALANIE I DOSTĘP DO DANYCH W APLIKACJACH IOS

- User Defaults
- Praca z plikami
- SQLite
- Wprowadzenie do Core Data

KOMUNIKACJA W APLIKACJACH IOS

- Podstawy użycia URLSession
- GDC i praca z kodem asynchronicznym
- Integracja z REST API oraz GraphQL
- Wykorzystanie przykładowych bibliotek: Alamofire, Moya, Siesta

DYSTRYBUCJA APLIKACJI IOS

- Budowanie i podpisywanie kodu
- Publikacja z wykorzystaniem iTunes Connect
- Udostępnianie wersji testowej
- Przygotowanie do review

ARCHITEKTURA APLIKACJI IOS

- Implementacja wzorców MVC, VIPER, MVVM, Composable Architecture / Flux

TESTOWANIE APLIKACJI IOS

- Przygotowanie i konfiguracja środowiska
- Testy jednostkowe
- Testy integracyjne
- Testy na poziomie UI
- Test Driven Development

ZAAWANSOWANE ELEMENTY SWIFTUI I COMBINE

- Zarządzanie stanem oraz jego efektywne współdzielenie
- Tworzenie zaawansowanych i niestandardowych widoków
- Zastosowanie matched geometry effect
- Wykorzystanie animacji
- Integracja z UIKit
- SwiftUI na innych platformach
- Debugowanie i obsługa błędów na poziomie Combine
- Wykorzystanie / tworzenie reaktywnych rozszerzeń w kontekście istniejącego API

CORE DATA

- Architektura i główne założenia
- Konfiguracja Core Data Stack
- Modelowanie encji
- Utrwalanie i ładowanie danych
- Wersjonowanie oraz migracja danych
- Mierzenie i optymalizacja wydajności
- Dobre praktyki

WYBRANE ELEMENTY IOS SDK

- Core Location oraz praca z mapami
- Kamera i biblioteka zdjęć
- Core Graphics i Core animation