

SZKOLENIE ZAAWANSOWANE

Android - zaawansowane tworzenie aplikacji

ANDROID/ADVANCED

Czas trwania: 5 dni (40h)

Zaawansowane tworzenie aplikacji na platformie Android

Cele szkolenia

- Rozszerzenie wiedzy o zagadnienia zaawansowane m.in. poprawna architektura aplikacji, testowanie, kotlin coroutines
- Poznanie istotnych elementów SDK takich jak Google Play Services czy Sensors API
- Nabycie wiedzy niezbędnej do tworzenia atrakcyjnego interfejsu użytkownika

Zalety

- Rozbudowana część warsztatowa pomagająca zrozumieć i przyswoić omawiane zagadnienia
- Duża ilość kompletnych i życiowych przykładów, gotowych do wykorzystania podczas realizacji własnych aplikacji
- Praktyka przed teorią - wszystkie szkolenia technologiczne prowadzone są w formie warsztatowej. Konieczna teoria jest wyjaśniana na przykładzie praktycznych zadań
- Konkretność umiejętności - w ramach każdego szkolenia rozwijamy praktyczne umiejętności związane z daną technologią i tematyką
- Nauka z praktykami - wszyscy trenerzy na co dzień pracują w projektach, gwarantuje to dostęp do eksperckiej wiedzy i praktycznego know-how

Dla kogo?

- Szkolenie dedykowane dla osób znających platformę Android, chcących poznać bardziej zaawansowane aspekty, dotyczące między innymi architektury, dobrych praktyk, wykorzystania dostępnych narzędzi oraz SDK

Wymagania

- Od uczestników szkolenia wymagana jest praktyczna umiejętność programowania w języku Kotlin, a także podstawowa znajomość platformy Android

Program

1. Architektura aplikacji Android



- a. Czym jest architektura i jakie ma znaczenie?
 - b. Spójność, niskie sprzężenie, programowanie z użyciem interfejsów
 - c. Moduły i ich granice, mapowanie modelu danych między warstwami
 - d. Poprawna organizacja kodu i komponentów w projekcie
 - e. Wstrzykiwanie zależności z użyciem Dagger / Hilt
 - f. Czysta architektura w praktyce (Clean architecture + MVVM)
 - g. Przegląd podejść alternatywnych: Model-View-Presenter, Model-View-Intent
2. Kotlin coroutines
 - a. Podstawy pracy z coroutines
 - b. Zasięg i kontekst wykonania
 - c. Structured concurrency
 - d. Obsługa wyjątków
 - e. Współdzielenie stanu
 - f. Asynchronous Flows
 - g. Channels
 - h. Testowanie
3. Zaawansowane aspekty tworzenia interfejsu użytkownika
 - a. Budowanie niestandardowych kontrolerek
 - b. Rysowanie z użyciem obiektu Canvas
 - c. Wykorzystanie animacji
4. Testowanie
 - a. Dlaczego testowanie jest ważne?
 - b. Piramida testów
 - c. Metryki pokrycia testami
 - d. Definicja i rola testów jednostkowych
 - e. Czym jest jednostka kodu i co testować (podejście classic vs. mockist)?
 - f. Unit tests vs. Instrumented unit tests
 - g. Testowanie w praktyce z wykorzystaniem bibliotek JUnit i Kotest
 - h. Obiekty zastępcze / izolacja zależności z wykorzystaniem Mockito oraz MockWebServer
 - i. Praca z kodem asynchronicznym
 - j. Definicja i rola testów end2end
 - k. Wprowadzenie do testów opartych o Espresso - zasada działania, matchers, actions, assertions
 - l. Espresso praktyce
 - m. Espresso vs. inne rozwiązania: Barista, Kaspresso, BusyBee, Repeato, Waldo

