

Kod szkolenia: **ANDROID/PRACTICAL**

Tytuł szkolenia: **Android w praktyce**

Dni: 5

## Opis:

### Cele szkolenia:

- Wprowadzenie do programowania na platformie Android
- Zapoznanie z dostępnymi narzędziami developerskimi
- Zapoznanie z narzędziami oraz bibliotekami najczęściej wykorzystywanymi przy tworzeniu aplikacji na platformę Android

### Mocne strony:

- Szkolenie w maksymalnym stopniu nastawione jest na aspekt praktyczny
- Jego głównym elementem jest rozwinięta część warsztatowa obejmująca zarówno budowę złożonego projektu jak i niezależne ćwiczenia pomagające zrozumieć i utrwalić poruszane zagadnienia
- Uczestnicy poznają dobre praktyki oraz sposoby rozwiązywania typowych problemów, które mogą pojawić się przy tworzenia aplikacji Android

### Adresaci szkolenia:

- Szkolenie przeznaczone dla osób rozpoczynających naukę tworzenia aplikacji mobilnych na platformie Android
- Programiści chcący powtórzyć i uporządkować posiadaną wiedzę

### Wymagania:

- Od uczestników szkolenia wymagana jest praktyczna umiejętność programowania w języku Kotlin lub Java

### Parametry szkolenia:

5\*8 godzin (5\*7 godzin netto) wykładów i warsztatów (z wyraźną przewagą warsztatów).

### Program szkolenia:

- Wprowadzenie do Kotlin



- Charakterystyka języka
- Typy
- Podstawy składni
- Instrukcje sterujące
- Kolekcje
- Wyrażenia lambda
- Extension functions
- Pakiety
- Obsługa wyjątków
- Programowanie obiektowe
- Programowanie funkcyjne
- Idiomy programistyczne
- Interoperacyjność z Java
- Wprowadzenie do Androida
  - Architektura
  - Kompatybilność
  - Bezpieczeństwo i system uprawnień
  - Dokumentacja
- Środowisko pracy / Narzędzia
  - Android Studio + Kotlin plugin
  - Tworzenie projektu
  - Konfiguracja Gradle
  - Budowanie i uruchamianie
  - Emulatory
  - Debuggowanie
  - Profilowanie
- Podstawy tworzenia aplikacji
  - Activities
  - Services
  - Content Providers
  - Broadcast receivers
  - Intencje
  - Plik manifestu
  - Zarządzanie zasobami
  - Interfejs użytkownika
  - Praca z danymi
- Material Design and Theming
  - Czym jest Material Design?
  - Material Design library
  - Tworzenie widoków w zgodzie z MD
  - Theming w Androidzie
- Najczęściej używane biblioteki
  - Dependency Injection z Dagger
  - REST API z Retrofit, OkHttp
  - Parsowanie JSON z Moshi
  - Wygodna obsługa dat (JSR-310)



- Asynchroniczność z RxJava
- Asynchroniczność z Coroutines
- SQLite z Room
- Wyświetlanie obrazków z Glide
- Firebase
- Architektura
  - Struktura projektu i modularyzacja
  - Czysta architektura
  - Model View View Model
  - Wstrzykiwanie zależności
  - JetPack Architecture Components
- Publikowanie aplikacji
  - Budowanie i podpisywanie
  - Procedura publikacji
  - Aktualizacje
- Testowanie
  - Unit Tests - testy jednostkowe
  - Podmiana zachowań z Mockk
  - Dependency Injection w testach
  - Testowanie kodu asynchronicznego
  - Espresso i testy komponentów
  - Wzorce przydatne w testowaniu

