

## Program szkolenia:

# Kotlin dla programistów Java i Android

### Informacje:

<b>Nazwa:</b>	<b>Kotlin dla programistów Java i Android</b>
<b>Kod:</b>	<b>android-kotlin</b>
<b>Kategoria:</b>	Android
<b>Odbiorcy:</b>	developerzy, architekci
<b>Czas trwania:</b>	2 dni
<b>Forma:</b>	20% wykłady / 80% warsztaty

---

W trakcie szkolenia uczestnik nauczy się używać konstrukcji składniowych Kotlina oraz pozna idiomy i wzorce ich wykorzystania. Dodatkowo, zapozna się z podstawami programowania funkcyjnego i sposobem użycia tego paradygmatu w kontekście tworzenia aplikacji androidowych. Zostaną również omówione zagadnienia związane z działaniem Kotlina razem z najpopularniejszymi narzędziami i bibliotekami oraz wpływu użycia języka na wydajność aplikacji.

Kotlin jest językiem coraz mocniej zyskującym na popularności, głównie za sprawą wygodnych konstrukcji składniowych i wysunięcia na pierwszy plan paradygmatu programowania funkcyjnego. Cechy te znacznie ułatwiają pisanie dobrego kodu zawierającego mniej błędów. Znajdując szerokie zastosowanie na różnych platformach, Kotlin staje się językiem uniwersalnym. Jest oficjalnie wspierany przez Google do tworzenia aplikacji androidowych, z powodzeniem stosowany do tworzenia rozwiązań backendowych, webowych (kotlin-js) oraz natywnych, w tym iOS (kotlin-native). Twórcą Kotlina jest firma JetBrains, co skutkuje doskonałym wsparciem dla języka w środowiskach programistycznych (IntelliJ oraz Android Studio).

### Zalety szkolenia:

- Paradygmat funkcyjny
- Zastosowanie w kontekście Android
- Łatwe przejście z Javy poprzez wskazywanie analogii

## Szczegółowy program:

### 1. Składnia języka

1.1. Podstawy składni i różnice względem Javy

1.2. Deklaracje zmiennych i stałych

1.3. Typy podstawowe

1.4. Kontrola przepływu

1.4.1. if/when/try jako wyrażenia

1.4.2. for i while

1.5. Zwracanie wartości funkcji/lambdy i skoki

1.6. Null-safety na poziomie systemu typów

1.7. Równość obiektów w Kotlinie

1.8. Obsługa wyjątków

### 2. Konstrukcje języka do programowania obiektowego

2.1. Interface, Class and amp; Object (Singleton)

2.2. Companion Objects

2.3. Konstruktory i atrybuty

2.4. Enkapsulacja (modyfikatory dostępu)

2.5. Data Classes i niemutowalny stan

2.6. Sealed Classes jako narzędzie do modelowania stanu

2.7. Rozszerzanie możliwości klas przez extension functions

2.8. Delegacja na poziomie klasy jako alternatywa do dziedziczenia

2.9. Delegowane atrybuty

2.9.1. Lazy

2.9.2. Observable

2.10. Implementacja własnych delegatów

2.11. Aliasy typów

2.12. Nadpisywanie operatorów

### 3. Biblioteka standardowa

3.1. Zakresy

3.2. Tablice i kolekcje

3.3. Przydatne rozszerzenia do biblioteki standardowej Javy

### 4. Programowanie funkcyjne

4.1. Założenia paradygmatu funkcyjnego

4.2. Funkcje w Kotlinie

4.2.1. Parametry domyślne

4.2.2. Nazwane argumenty

4.2.3. Deklaracja typów zwracanych

4.2.4. Notacja infiksowa

4.2.5. Funkcje wyższego rzędu

4.2.6. Wyrażenia lambda

4.3. Wprowadzenie w świat monad

4.4. Optymalizacja funkcji wyższego rzędu przez inlinowanie wykonania

4.5. Funkcyjne przetwarzanie kolekcji

4.6. Programowanie bez zmiennych - funkcje let/use/with/apply

### 5. Programowanie współbieżne z wykorzystaniem Coroutines

### 6. Generyczne klasy i funkcje

6.1. Składnia i różnice względem Javy

6.2. Zapobieganie wymazywaniu typów dzięki reifikowanym typom generycznym

### 7. REPL (read-eval- print loop)

## 8. Współpraca Kotlin z Javą

## 9. Kotlin w środowisku Android

9.1. Bindowanie widoków za pomocą kotlin-android- extension

9.2. Konfiguracja procesora adnotacji w Kotlinie

9.3. Kotlin Android DSL - Biblioteka Anko

## 10. Popularne narzędzia i biblioteki a Kotlin

10.1. Dagger 2

10.2. RxJava

10.3. Testy automatyczne

10.4. Mockito

## 11. Kotlin i jego wpływ na performance aplikacji

11.1. Porównanie bytecode Javy i Kotlin dla podstawowych konstrukcji języka