

# WPROWADZENIE DO JĘZYKA SCALA – PROGRAMOWANIE FUNKCYJNE

KOD: SCALA

# Profil uczestnika

Programista:

- chce poznać i zastosować język Scala w projektach komercyjnych;
- chce wprowadzić narzędzie usprawniające tworzenie aplikacji na platformie Java;
- chce poszerzyć umiejętności programistyczne o zastosowanie języków funkcyjnych w codziennej pracy;
- posiada przynajmniej roczne doświadczenia w programowaniu.

## Korzyści ze szkolenia

1. **Wzrasta efektywność pracy programisty** – dzięki zastosowaniu języka skryptowego do realizacji części zadań w projekcie, kod powstaje szybciej i jest prostszy w analizie.
2. **Programista poprawia swój warsztat pracy** – ucząc się i używając języka funkcyjnego potrafi wyjść poza utarte schematy działania.
3. **Zwiększa się jakość kodu tworzonego przez programistę** – dzięki odpowiedniemu zastosowaniu skryptowego języka funkcyjnego, łatwiej utrzymać i przetestować kod.

## Parametry szkolenia

CZAS TRWANIA: 2 dni - 16 godzin.

FORMA ZAJĘĆ: Laboratorium Scala - 70%, wykład – 30%.

WIELKOŚĆ GRUPY: do 10 osób.

# Szczegółowy program

Moduły szkoleniowe	Nabyte wiedza i umiejętności, poruszane zagadnienia
<b>Wprowadzenie do języka Scala</b>	<ul style="list-style-type: none"><li>• Cele postawione przed językiem Scala</li><li>• Narzędzia do pracy w języku Scala</li><li>• Przegląd języka:<ul style="list-style-type: none"><li>○ zmienne,</li><li>○ funkcje,</li><li>○ pętle,</li><li>○ instrukcje warunkowe,</li><li>○ iterowanie,</li><li>○ tablice,</li><li>○ listy,</li><li>○ krotki,</li><li>○ zbiory,</li><li>○ mapy.</li></ul></li></ul>
<b>Podstawowe elementy języka</b>	<ul style="list-style-type: none"><li>• Klasy i obiekty</li><li>• Typy i operacje</li><li>• Obiekty funkcyjne</li><li>• Instrukcje sterujące</li><li>• Funkcje i domknięcia</li><li>• Tworzenie abstrakcji</li><li>• Kompozycja i dziedziczenie</li></ul>
<b>Zaawansowane elementy języka Scala</b>	<ul style="list-style-type: none"><li>• Hierarchia klas języka Scala</li><li>• Cechy (ang. traits)</li><li>• Pakiety i importy</li><li>• Listy</li><li>• Kolekcje</li></ul>

- Asercje i testy jednostkowe
- Obiekty stanowe
- Parametryzacja typu
- Konwersje niejawne
- Parametry
- Wyrażenia typu for
- Wyłuskiwacze (ang. extractors)
- Adnotacje
- Praca z XML
- Parsery

#### Interfejs użytkownika

- Programowanie GUI
- Panele
- Obsługa zdarzeń