

Program szkolenia:

Software Craftsmanship z wykorzystaniem Generatywnej Sztucznej Inteligencji - warsztaty praktyczne na bazie TypeScript lub Java+SpringBoot lub React

Informacje:

Nazwa:	Software Craftsmanship z wykorzystaniem Generatywnej Sztucznej Inteligencji - warsztaty praktyczne na bazie TypeScript lub Java+SpringBoot lub React
Kod:	ai-craft
Kategoria:	AI
Odbiorcy:	architekci, developerzy
Czas trwania:	3 dni
Forma:	20% wykłady / 80% warsztaty

Sztuczna inteligencja przestaje być jedynie narzędziem do generowania kodu a staje się prawdziwym partnerem w codziennej pracy programisty.

W trakcie tych 3-dniowych, intensywnych warsztatów nauczysz się, jak skutecznie wykorzystywać Generatywną Sztuczną Inteligencję w duchu Software Craftsmanship.

Warianty szkolenia:

- backend w Java i Spring Boot
- backend w TypeScript
- frontend w React + TypeScript

Poprzez serię praktycznych ćwiczeń, sesji live codingowych i analiz rzeczywistych przypadków użycia:

- Nauczysz się współpracować z AI przy tworzeniu czystego, łatwego w utrzymaniu i dobrze zaprojektowanego kodu
- Zastosujesz podejście Test Driven Development (TDD) oraz cykl Red-Green-Refactor przy wsparciu AI
- Dowiesz się, jak AI może wspomagać tworzenie testów, refaktoryzację oraz podejmowanie decyzji projektowych
- Zrozumiesz, jak maksymalizować produktywność, zachowując pełną kontrolę nad bazą kodu
- Poznasz typowe pułapki pracy z AI i nauczysz się, jak ich unikać

Zgłębisz również wzorce architektoniczne i dobre praktyki, które nie tylko wspierają budowę skalowalnych i łatwych w utrzymaniu systemów, ale też znacząco ułatwiają współpracę z Generatywną Sztuczną Inteligencją:

- **Domain-Driven Design (DDD)** – pozwala na zorganizowany modelu dziedzinowego tak aby był zrozumiały dla AI
- **Ports and Adapters (Hexagonal Architecture)** – wspiera separację odpowiedzialności, co pozwala AI operać na wyizolowanych komponentach

BO·TT·EGA

IT minds

- **Modularization** – pozwala na generowanie i refaktoring w dobrze zdefiniowanych kontekstach
- **Microservices Architecture** – umożliwia rozproszone wytwarzanie na poziomie serwisu co pozwala AI na niezależną ewolucję i wdrożenia.

Zalety szkolenia:

- Domain-Driven Design (DDD) – pozwala uporządkować model domenowy w sposób zrozumiały dla AI, co sprawia, że interakcje z nią stają się bardziej kontekstowe i trafne
- Ports and Adapters (Architektura Heksagonalna) – wspiera czysty podział odpowiedzialności, umożliwiając AI bezpieczne działanie na odseparowanych komponentach bez ryzyka powiązań
- Modularyzacja – zapewnia wyraźne granice i odpowiedzialności w bazie kodu, co pozwala AI generować i refaktoryzować kod w dobrze zdefiniowanych kontekstach
- Architektura mikrousługowa – umożliwia rozproszony rozwój i precyzyjne wsparcie AI na poziomie pojedynczych usług, wspierając ich niezależną ewolucję i wdrażanie

Szczegółowy program:

1. Fundamenty współpracy z Generatywną AI w codziennej pracy programisty

1.1. Wprowadzenie do Generatywnej AI w kontekście Software Craftsmanship

1.1.1. Czym jest Generatywna AI i jak zmienia sposób pracy programistów

1.1.2. Przegląd narzędzi AI dla programistów Java/Spring

1.1.3. Etyka i odpowiedzialność przy korzystaniu z AI w kodowaniu

1.2. Podstawy efektywnej współpracy z AI

1.2.1. Pisanie skutecznych promptów i komunikacja z AI

1.2.2. Rola kontekstu, intencji i struktury w pracy z AI

1.2.3. Omówienie typowych błędów i nieporozumień

1.3. Ćwiczenia praktyczne: AI jako partner w codziennej pracy

1.3.1. Tworzenie kodu biznesowego z pomocą AI

1.3.2. Korekta i poprawianie kodu z udziałem AI

1.3.3. Weryfikacja rezultatów i jakość wygenerowanego kodu

1.3.4. Implementacja elementów technicznych i integracyjnych

2. Zapewnienie jakości - testowanie i architektura

2.1. Test-Driven Development (TDD) z udziałem AI

2.1.1. Zasady i wartości TDD

2.1.2. Red-Green-Refactor z pomocą AI

2.1.3. Generowanie testów jednostkowych i integracyjnych

2.1.4. Generowanie testy, a poprawność systemu

2.2. Ćwiczenia praktyczne: AI jako pomocnik w utrzymaniu jakości

2.2.1. Refaktoryzacja nieczytelnego kodu z AI

2.2.2. Rozszerzanie funkcjonalności z wykorzystaniem testów

2.2.3. Testy łatwe w utrzymaniu i chroniące przed regresją

3. Wzorce architektoniczne i AI w złożonych systemach

3.1. Domain-Driven Design i kontekstowe zrozumienie przez AI

3.1.1. Wprowadzenie do DDD i jego znaczenie w pracy z AI

3.1.2. Budowanie modeli domenowych z uwzględnieniem zrozumiałości dla AI

3.1.3. Współtworzenie agregatów, value obiektów i usług domenowych z AI

3.2. Architektura heksagonalna, modularność i mikrouslugi

3.2.1. Korzyści z Ports and Adapters w kontekście AI

3.2.2. Tworzenie i refaktoryzacja modułów w dużych systemach

3.2.3. Skalowanie rozwiązań z pomocą AI na poziomie mikrouslug

3.3. Zastosowania praktyczne i strategia wdrożeniowa

3.3.1. Studium przypadku: rozbudowa istniejącej aplikacji z AI

3.3.2. Studium przypadku: budowa nowych aplikacji z AI

3.3.3. Wyznaczanie granic odpowiedzialności między człowiekiem a AI

3.3.4. Plan wdrożenia AI do zespołu produkcyjnego