BO·TT·EGA
IT minds

Program szkolenia:

Nowoczesny C++ Dobry start

Informacje:

Nazwa:	**Nowoczesny C++ Dobry start**
Kod:	**ccpp-C++ new**
Kategoria:	C i C++
Grupa docelowa:	developerzy
Czas trwania:	3 dni
Forma:	60% wykłady / 40% warsztaty

Z naszego doświadczenia wynika, że ogromna większość oprogramowania pisanego w języku C++ jest tworzona z zastosowaniem nieaktualnych standardów języka (C++98) oraz przestarzałych technik programowania. U podłoża takiego stanu rzeczy tkwi przekonanie, że tworząc oprogramowanie o nietrywialnych wymaganiach dotyczących wydajności nie można pozwolić sobie na stosowanie nowoczesnych metod i narzędzi, które pozwalają pisać kod na wysokim poziomie abstrakcji.

To unikalne szkolenie pozwoli programistom C++ nadążyć za szybko zmieniającym się standardem języka i szybko ewoluującymi najlepszymi praktykami jego wykorzystania. Przykłady i warsztaty oparte o rzeczywisty kod pozwolą na łatwiejsze zastosowanie nabytej wiedzy w praktyce. Szczególny nacisk będzie położony na poprawę produktywności programistów poprzez zastosowanie nowych elementów składni języka.

Szkolenie przeznaczone jest dla programistów, którzy w codziennej pracy korzystają z języka C++.

Zalety szkolenia:

- Wprowadzenie do standardów C++11 oraz C++14
- Poprawa produktywności programisty
- Nacisk na pisanie poprawnego, czystego i wydajnego kodu

BO·TT·EGA
IT minds

Szczegółowy program:

1. Nowe konstrukcje – większa produktywność

- 1.1. Zakresowe instrukcje for (C++11)
- 1.2. Poprawione typy wyliczeniowe (C++11)
- 1.3. Ujednolicona składnia inicjalizacyjna (C++11)
- 1.4. Listy inicjalizacyjne (C++11)

2. Przenoszenie – w trosce o wydajność

- 2.1. Semantyka przenoszenia (C++11)
- 2.2. Referencje prawostronne (C++11)
- 2.3. Referencje uniwersalne (C++11)
- 2.4. Operacje przenoszenia (C++11)

3. Dedukcja typów – usuwamy z kodu zbędne informacje

- 3.1. Argumenty szablonów
- 3.2. auto (C++11)
- 3.3. decltype (C++11)
- 3.4. decltype (auto) (C++14)
- 3.5. auto dla wartości zwracanych (C++14)

4. Wyrażenia lambda – programowanie funkcyjne w C++

- 4.1. Funktory
- 4.2. Podstawy (C++11)
- 4.3. Domknięcia (C+11)
- 4.4. Ogólne wyrażenia lambda (C++14)