

## Program szkolenia:

# Nowoczesny C++ Dla zaawansowanych

### Informacje:

<b>Nazwa:</b>	<b>Nowoczesny C++ Dla zaawansowanych</b>
<b>Kod:</b>	<b>ccpp-C++ Pro</b>
<b>Kategoria:</b>	C i C++
<b>Grupa docelowa:</b>	developerzy
<b>Czas trwania:</b>	2-3 dni
<b>Forma:</b>	60% wykłady / 40% warsztaty

Opanowanie zaawansowanych elementów języka C++ stanowi nie lada wyzwanie, ale jest niezbędne, kiedy w grę wchodzi tworzenie oprogramowania systemowego o najwyższej jakości. Poprzez położenie szczególnego nacisku na programowanie za pomocą szablonów, niniejsze szkolenie pozwoli na oderwanie się od dominującego paradygmatu OOP i tworzenie nowoczesnych, wydajnych bibliotek, które z powodzeniem można stosować w wielu aplikacjach.

Szkolenie przeznaczone jest dla programistów, którzy w codziennej pracy korzystają z języka C++. Najwięcej korzyści ze szkolenia wyniosą osoby, które ukończyły również kurs Nowoczesny C++. Dobry start.

### Zalety szkolenia:

- Programowanie ogólne (generic programming)
- Metaprogramowanie za pomocą szablonów
- Kod odporny na błędy
- Nacisk na pisanie poprawnego, czystego i wydajnego kodu

## Szczegółowy program:

### 1. Szablony – potęga programowania ogólnego (generic programming)

- 1.1. Szablony funkcji
- 1.2. Szablony klas
- 1.3. Szablony zmienne (C++11)
- 1.4. Specjalizacja
- 1.5. Argumenty szablonów, które nie są typami
- 1.6. Statyczne asercje

### 2. Biblioteka standardowa – nie tylko OOP

- 2.1. Iteratory
- 2.2. Kontenery (C++11)
- 2.3. Algorytmy (C++11)
- 2.4. Programowanie współbieżne (C++11)
- 2.5. Przetwarzanie tekstu: zestawy znaków oraz wyrażenia regularne (C++11)

### 3. Metaprogramowanie – przenosimy obliczenia na czas kompilacji

- 3.1. Funkcje operujące na typach
- 3.2. Struktury sterujące
- 3.3. Reguła SFINAE
- 3.4. Definicje warunkowe: enable\_if (C++11)
- 3.5. Praktyczny przykład (std::chrono) (C++11)

### 4. Wyjątki - jak poprawnie obsługiwać błędy?

- 4.1. Problemy z tradycyjną obsługą błędów
- 4.2. Mechanizm wyjątków
- 4.3. Gwarancje wyjątków

