

## Program szkolenia:

# Corba w C++

### Informacje:

<b>Nazwa:</b>	<b>Corba w C++</b>
<b>Kod:</b>	<b>ccpp-Corba</b>
<b>Kategoria:</b>	C i C++
<b>Odbiorcy:</b>	developerzy
<b>Czas trwania:</b>	3-4 dni
<b>Forma:</b>	25% wykłady / 75% warsztaty

---

Szkolenie w nieszablony sposób przedstawia szeroko stosowany obiektowy standard RPC (Remote Procedure Call) w kontekście języka C++ wychodząc od tematów zupełnie podstawowych i kończąc na najbardziej wyszukanych funkcjonalnościach.

Uczestnicy zdobywają wiedzę w oparciu o zadania, które w pełni pozwalają zrozumieć i obyć się z architekturą CORBA i zasadami rządzącymi wykorzystaniem jej w C++.

Szkolenie przeznaczone jest dla programistów C++, architektów i liderów projektów pracujących w projektach zorientowanych na rozproszone przetwarzanie, mediację lub inne pokrewne. Zdobyta wiedza umożliwi płynne posługiwanie się CORBA oraz w znacznym stopniu ułatwi pracę w „kodzie odziedziczonym”, który już ją wykorzystuje.

### Zalety szkolenia:

- Szkolenie prowadzone w trybie „od zera do bohatera” (od zupełnych podstaw do najbardziej zaawansowanych mechanizmów)
- Nauka w oparciu o przykłady z naciskiem na wykorzystanie wzorców projektowych
- Weryfikacja i oszlifowanie zdobytej wiedzy w egzaminie teoretyczno-praktycznym

## Szczegółowy program:

### 1. CORBA – Common ORB Architecture

#### 1.1. Wprowadzenie

1.1.1. Rozwój CORBA w ujęciu historycznym

1.1.2. Funkcjonalność z wysokiego poziomu

1.1.3. Architektura

1.1.4. Modele komunikacyjne

#### 1.2. Pierwsze kroki

1.2.1. Przykład IDL

1.2.2. Przykłady implementacji serwera w dwóch modelach komunikacji

1.2.3. Przykłady implementacji klienta w dwóch modelach komunikacji

#### 1.3. Podsumowanie

1.3.1. Najczęściej popełniane błędy

1.3.2. Wskazówki dotyczące zarządzania pamięcią dynamicznie alokowaną

### 2. IDL, obiekty, referencje...

#### 2.1. Systematyzacja

2.1.1. Motywacja inżynierii zorientowanej na rozproszenie

2.1.2. Złożoność aplikacji rozproszonych

2.1.3. Motywacja powstania CORBA'y

2.2. IDL – Interface Description Language – jego rozumienie w kontekście obiektów implementowanych i używanych

2.2.1. Najważniejsze reguły mapowania do C++

2.2.2. CORBA::Object – źródło wszystkich obiektów

2.2.3. Referencje do obiektów i ich rozumienie

2.2.4. Typy podstawowe i złożone

2.2.5. Obiekty klienckie i serwerowe od środka

2.3. Naming Service

2.3.1. Overview – czyli „z czym to się je”

2.3.2. Przykład wykorzystania

2.4. Podsumowanie

2.4.1. Poruszanie się wzdłuż hierarchii dziedziczenia

2.4.2. Zamykanie aplikacji „na czysto”

2.4.3. Obiekty i obiekty referencyjne

2.4.4. Struktury, sekwencje i typ „Any”

### 3. Wszystko o POA

3.1. Systematyzacja

3.1.1. CORBA „success stories”

3.1.2. Cele CORBA’y

3.1.3. Protokół z niskopoziomowego punktu widzenia

3.1.4. Modele komunikacyjne

3.2. POA – Portable Object Adaptor – definiowanie QoS dla serwisów CORBA’owych

3.2.1. Architektura POA

3.2.2. POA Manager i jego właściwości

3.2.3. Root POA

3.2.4. Cykl życia obiektu

3.2.5. Podstawowe polityki POA

3.2.6. Servant Managers

3.2.7. Interceptory

3.3. Podsumowanie

3.3.1. Kompatybilność polityk POA

3.3.2. Czynniki doboru polityk POA

## 4. Zakończenie

4.1. Podsumowanie

4.1.1. Inżynieria rozproszona w generalnym ujęciu

4.1.2. CORBA w kontekście inżynierii rozproszonej

4.1.3. Zestawienie alternatyw dla CORBA'y

4.1.4. Część dedykowana

4.1.4.1. Możliwość oceny i analizy użycia CORBA w istniejących projektach

4.1.4.2. Pomoc w zrozumieniu rozwiązań w kodzie odziedziczonym