

Kod szkolenia: **CPP/ARCH**

Tytuł szkolenia: **Budowa i architektura złożonych systemów w języku C++**

Dni: 4

Opis:

Uczestnicy szkolenia zapoznają się z metodami budowy architektury złożonych systemów w języku C++. Na zajęciach zaznajomią się z koncepcjami definiowania architektury, wymaganiami wpływającymi na wybór poszczególnych rozwiązań oraz aspektami specyficznymi dla języka C++.

Szczególny nacisk kładziemy na zagadnienia związane z konsolidacją plików obiektów rozbudowanych architektur, przyśpieszeniem konsolidacji oraz stosowaniem podstawowych wzorców tak projektowych jak i architektonicznych. Uwypuklamy aspekty prawidłowego definiowania kluczowych wymagań co do architektury, zdefiniowania modelu prawidłowego wykonania procesu budowy, a także definiowanie elastycznych interfejsów aplikacyjnych. Szkolenie odpowiada potrzebom zaawansowanych programistów oraz architektów systemów. Jako bibliotek i narzędzi wspierających używamy oprogramowania otwartego. Szczególne znaczenie przykładamy do określenia czynników separujących elementy przetwarzania na etapie kompilacji, generowania kodu, realizacji algorytmów constexpr.

Zajęcia prowadzone są przez doświadczonych praktyków, którzy na co dzień wdrażają techniki i procedury posługując się technologiami związanymi z językiem C++ w rzeczywistych projektach. W trakcie zajęć używamy technologii i środowisk otwartych.

Zakres szkolenia

Szkolenie obejmuje:

- model architektury i procesu budowy oprogramowania w języku C++,
- przegląd wzorców architektury aplikacji dla przedsiębiorstw,
- wstępne zapoznanie się z wymaganiami procesów wytwórczych związanych z inżynierią oprogramowania,
- zapoznanie się ze standardem i składnią nowych właściwości C++ (C++17, nadchodzącym C++20).
- użyteczne techniki metaprogramowania, implementacji zasad architektur reaktywnych i funkcyjnych.

Wymagania

Od uczestników szkolenia wymaga się:

- biegłego posługiwania się wybranym środowiskiem IDE (Eclipse, Visual Studio, NetBeans, vim),
- znajomości koncepcji programowania obiektowego, systemowego i aplikacyjnego,
- biegłej znajomości narzędzi do tworzenia aplikacji z rodziny gcc i binutils,
- umiejętności posługiwania się narzędziem debuggera.
- znajomości koncepcji właściwych programowaniu w języku C++ na poziomie średnio i zaawansowanym.
- znajomości tworzenia kodu szablonowego oraz posługiwania się dowolną implementacją biblioteki standardowej.

Adresaci szkolenia

Szkolenie adresowane jest do:

- średnio zaawansowanych i zaawansowanych programistów programujących w języku C++ w środowiskach tworzenia aplikacji dla przedsiębiorstw,
- osób znających w stopniu podstawowym i zaawansowanym procesów związanych z budowaniem dokumentacji projektowej, jednocześnie biorących bezpośredni udział w tworzeniu oprogramowania,
- programistów tworzących oprogramowanie na platformie systemowej GNU/Linux lub MS Windows,
- programistów tworzących aplikacje głównego nurtu oraz aplikacje sieciowe.

Cel szkolenia

Głównymi celami w procesie szkoleniowym są:

- kształcenie umiejętności i rozwijanie wiedzy dotyczącej definiowania architektury systemu w języku C++,
- osadzenie narzędzi i procesów tworzenia oprogramowania w realiach współczesnych wymagań Inżynierii Oprogramowania,
- dokonanie konsolidacji wiedzy i uzupełnienie braków w kompetencjach w zakresie aspektów przyspieszenia budowy złożonych systemów,
- zapoznanie się z zaawansowanymi mechanizmami związanymi z warstwą ABI (ang. Application Binary Interface) i API (ang. Application Programmer Interface) tworzonych

Umiejętności zdobywane podczas szkolenia

Uczestnicy szkolenia po jego zakończeniu zdobędą następujące umiejętności:

- stosowania wzorców projektowych i architektonicznych,
- określania kluczowych wymagań wpływających na wybór architektury,
- stosowania składni bieżących standardów (C++14, C++17 i nadchodzącego C++20), szczególnie w zakresie ułatwiającym obsługę złożonego oprogramowania.

Parametry szkolenia

4 dni. Szkolenie obejmuje 60% wykładu oraz 40% ćwiczeń.

Program szkolenia:

1. Wprowadzenie do zasad budowy architektury.
 - Przypomnienie pojęć dotyczących analizy obiektowej.
 - Główne elementy i zasady modelowania.
 - Perspektywy rozwiązania informatycznego.
 - Proces wytwórczy, a model.
2. Analiza procesów biznesowych.
 - Pojęcie procesu biznesowego.
 - Modelowanie procesów biznesowych – przegląd metod i narzędzi z punktu widzenia architekta.
 - Procesy biznesowe, a architektura aplikacji.
 - Pojęcie domeny rozwiązania.
3. Analiza wymagań z punktu widzenia architekta.
 - Wymagania wpływające na wybór architektury oprogramowania w języku C++
 - Wzorce architektury i ich aspekty właściwe dla języka C++.
4. DDD (ang. Domain Driven Design) – przegląd zagadnień.
 - Elementy i wzorce DDD.
 - DDD w praktyce projektu w języku C++.
5. C++, a rozbudowana architektura.
 - Przegląd istotnych zagadnień związanych z konsolidacją plików obiektów.
 - Reguły separacji warstw i komponentów w języku C++.
 - Szkielety aplikacyjne wspierające budowę złożonych architektur (ang. Frameworks).
6. Architektura logiczna.
 - Projektowanie komponentów.
 - Projektowanie obiektów i interfejsów.

- Projektowanie funkcji.
 - Wzorce projektowe GoF – przegląd.
 - Wzorce aplikacji dla przedsiębiorstw – przegląd.
7. Projekt fizyczny.
 - Komponent – zakres obiektów fizycznych.
 - Hierarchia fizyczna.
 - Aspekty testowalności.
 - Izolacja fizyczna – techniki izolacji (częściowe i całkowite).
 - Interfejsy proceduralne.
 - Pakiety w warstwie fizycznej.
 8. Typowe problemy związane z rozwojem aplikacji.
 - Koszt i konieczność refaktoryzacji.
 - Wzorce refaktoryzacji.
 - Zarządzanie pamięcią w złożonych aplikacjach w języku C++.
 - Zagadnienia aspektowe w aplikacjach w języku C++ i ich implementacja.
 - Problem zależności cyklicznych.
 - Rodzaje odwrócenia kontroli IoC (ang. Inversion of Control) i konsekwencje stosowania w języku C++.
 9. Cykl życia oprogramowania i wpływ na architekturę i projekt.
 - Rodzaje cyklu życia aplikacji – wpływ na projekt i aspekty jakości aplikacji.
 - Techniczne aspekty wspierające implementację cyklu życia aplikacji.
 - Narzędzia komercyjne.
 - Narzędzia na licencjach wolnych.
 - Programowanie obiektowe – czynniki wpływające na proces analizy i implementacji.
 - Programowanie wielowątkowe – wady i zalety.
 - DSL (ang. Domain Specific Language).
 - Maszyna stanu.
 - Programowanie funkcyjne (metaprogramowanie w języku C++).
 - Przetwarzanie w czasie kompilacji (constexpr)
 - Nowoczesne metody zarządzania zasobami w języku C++.
 - Przetwarzanie w czasie kompilacji na poziomie szablonów i funkcji constexpr.
 - Aspekty praktyczne budowy architektury korporacyjnej.
 - Proces budowy aplikacji w języku C++.
 10. Przegląd narzędzi umożliwiających budowanie architektury.
 11. Dodatkowe paradygmaty i modele – aspekty projektowe.
 12. Nowe standardy języka C++ - na co zwrócić uwagę

