

Kod szkolenia: **C/BOOST**

Tytuł szkolenia: **Praktyczne zastosowanie bibliotek Boost oraz standardów C++11, C++14 i C++17**

Dni: 4

Opis:

Uczestnicy podczas szkolenia zapoznają się z metodami wytwarzania oprogramowania z użyciem zaawansowanych mechanizmów języka C++, szablonów STL oraz standardami C++11, C++14 i C++17. Na zajęciach poruszane są zagadnienia związane z koncepcjami algorytmów, iteratorów i zaawansowanych struktur danych dostępnych w ramach biblioteki standardowej (zgodnej ze standardem C++17) oraz biblioteki Boost. Szkolenie obejmuje także zagadnienia programowania uogólnionego oraz cechy typów. Szczególny nacisk kładziemy na nowe właściwości C++, które niwelują obecne do tej pory bolączki starszych standardów języka. Zajęcia uwzględniają także potrzeby testowania aplikacji tworzonych w języku C++ z użyciem narzędzi wspierających proces TDD. Jako biblioteki wspierające, wykorzystane są: cppunit oraz Boost Test. Duży nacisk w trakcie zajęć kładziemy na zrozumienie aspektów wydajności oraz gospodarowania zasobami z użyciem inteligentnych wskaźników. Jako techniki tworzenia aplikacji obiektowej, prezentowane są klasy wytycznych wspierające tradycyjne wzorce projektowe (GoF). Podczas szkolenia posługujemy się technologiami otwartymi z dostępem do kodu źródłowego.

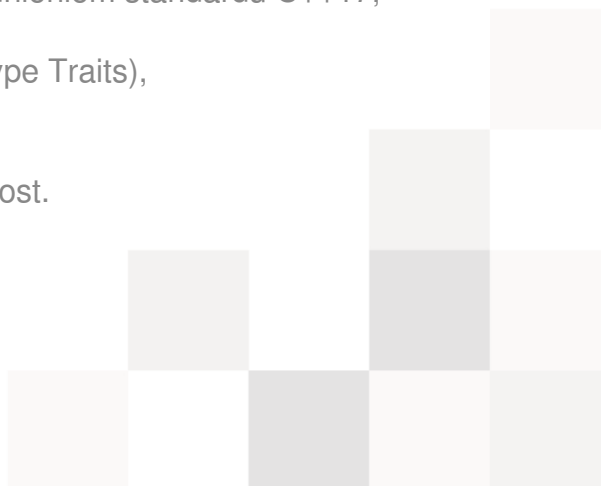
Zajęcia prowadzone są przez doświadczonych praktyków, którzy na co dzień wdrażają techniki i procedury związane z technologiami związanymi z językiem C++ w rzeczywistych projektach. W trakcie zajęć używamy technologii i środowisk otwartych.

Zakres szkolenia

Szkolenie obejmuje:

- omówienie zmian wprowadzonych wraz ze standardami C++11 i C++14,
- nowe elementy języka i biblioteki standardowej wprowadzone wraz z C++17,
- algorytmy biblioteki standardowej z uwzględnieniem standardu C++17,
- kontenery i iteratory biblioteki standardowej,
- cechy typów i ich praktyczne użycie (ang. Type Traits),
- inteligentne wskaźniki i ich stosowanie,
- wyrażenia lambda,
- przegląd przydatnych bibliotek w ramach Boost.

Wymagania



Od uczestników szkolenia wymaga się:

- biegłego posługiwania się wybranym środowiskiem IDE (Eclipse, 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 związanych z programowaniem w języku C++ na poziomie średnio i zaawansowanym.

Dodatkowo mile widziana jest znajomość aspektów sprzętowych platformy x86 (przydzielanie i zarządzanie pamięcią, alokacja rejestrów, konwencje wywołań, optymalizacje).

Adresaci szkolenia

Szkolenie adresowane jest do:

- średnio zaawansowanych i zaawansowanych programistów programujących w języku C i C++ w środowisku GNU/Linux,
- osób znających w stopniu podstawowym i zaawansowanym zakres systemu szablonów STL,
- 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:

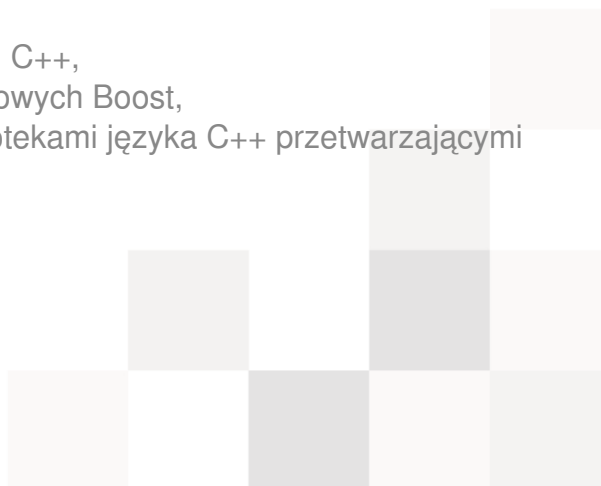
- kształcenie umiejętności oraz rozwijanie wiedzy dotyczącej tworzenia zaawansowanych rozwiązań w języku C++,
- osadzenie narzędzi i procesów tworzenia oprogramowania w realiach współczesnych wymagań Inżynierii Oprogramowania,
- dokonanie konsolidacji wiedzy i uzupełnienie braków w kompetencjach w kontekście standardu C++17.

Umiejętności zdobywane podczas zajęć

Uczestnicy szkolenia po jego zakończeniu zdobędą umiejętności:

- testowania jednostkowego aplikacji w języku C++,
- tworzenia aplikacji z użyciem bibliotek składowych Boost,
- zapoznają się także ze współczesnymi bibliotekami języka C++ przetwarzającymi dane w różnorodnych formatach.

Parametry szkolenia

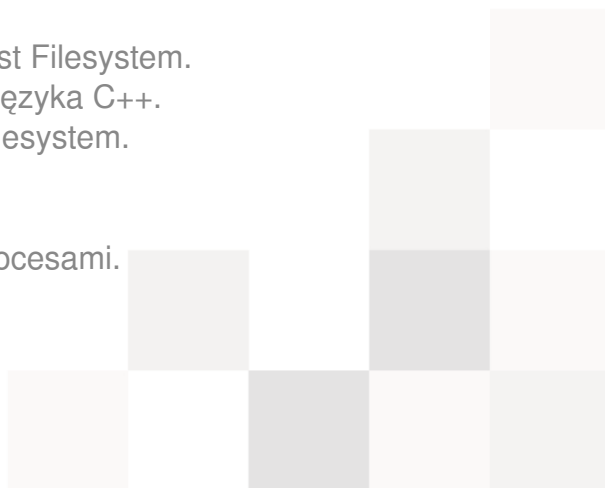


4 dni. Szkolenie obejmuje 60% wykładu oraz 40% ćwiczeń.

Program szkolenia:

1. Algorytmy i iteratory biblioteki standardowej – uzupełnienia związane ze standardami C++11, C++14 i C++17
 - I. Wydajność algorytmów standardowych.
 - II. Stosowanie algorytmów i iteratorów.
 - III. Algorytmy i kontenery dostępne w ramach standardów C++11, C++14 i C++17
 - IV. Modyfikacja działania algorytmów standardowych.
2. STL – uzupełnienie i przypomnienie wiedzy.
 - I. Szablony – stosowanie i reguły dopasowań.
 - II. Reguły SFINAE (ang. substitution failure is not an error).
 - III. Wzorzec „zadziwiająco często powracającego szablonu” CRTP (ang. curiously recurring template pattern).
 - IV. Trik Bartona-Nackmana (ang. Barton-Nackman trick).
 - V. Kruczki i sztuczki związane ze stosowaniem szablonów.
 - VI. Szablon o zmiennej liczbie argumentów (ang. variadic templates).
 - VII. Szablony zagnieżdżone.
3. Mechanizmy i elementy standardu C++17.
 - I. Manipulacja typem.
 - II. Nowe słowa kluczowe.
 - III. Mechanizmy kontroli definiowania i instancji klas.
 - IV. Wyjątki i obsługa w środowisku wielowątkowym.
 - V. Rozszerzenia biblioteki standardowej.
 - VI. Wyrażenia lambda.
4. Cechy typów (ang. type traits).
 - I. Pojęcie cechy typu i manipulowanie nim.
 - II. Cechy typów w bibliotece standardowej.
 - III. Rozwiązywanie najczęstszych problemów z użyciem cechy typów.
 - IV. Własne cechy typów – tworzenie i stosowanie.
 - V. Rozszerzenia C++17.
5. Wykładnie – praktyczne stosowanie.
 - I. Pojęcie wykładni (ang. class policy).
 - II. Wykładnie – wady i zalety.
 - III. Tworzenie i stosowanie własnych wykładni.
 - IV. Implementacja popularnych wzorców projektowych z użyciem wykładni.
 - V. Programowanie aspektowe, a wykładnie.
6. Inteligentne wskaźniki.
 - I. Rodzaje inteligentnych wskaźników.
 - II. Problem własności/przynależności.
 - III. Stosowanie inteligentnych wskaźników w codziennej pracy.
7. Testowanie aplikacji w języku C++.

- I. TDD w języku C++ - zasady i stosowanie.
 - II. xUnit – elementy i stosowanie.
 - III. Prawidłowe tworzenie testów.
 - IV. Zasady TDD a biblioteka cppunit.
 - V. Warsztat praktyczny testowania.
8. Boost Test – biblioteka wspierająca testowanie.
 - I. Przegląd elementów Boost Test.
 - II. Implementacja testów w bibliotece Boost Test.
 - III. Testy rozszerzone.
9. Boost – przegląd przydatnych bibliotek.
 - I. Zakresy zastosowań bibliotek Boost.
 - II. Specyfika implementacji i stosowania.
 - III. Przykładowe zastosowania bibliotek.
 - IV. Biblioteki Boost a nowe standardy języka C++.
10. Boost Regex.
 - I. Zakres stosowania Boost Regex.
 - II. Typowe operacje związane z analizą wyrażeń regularnych.
 - III. Kruczki i sztuczki zastosowania Boost Regex.
 - IV. Optymalizacja mechanizmów biblioteki we własnych aplikacjach.
11. Boost Property Tree.
 - I. Przeznaczenie biblioteki Boost Property Tree.
 - II. Zakres wydajnego stosowania Boost Property Tree.
12. Boost Meta State Machine/ Boost Statechart.
 - I. Paradygmat programowania maszyny stanu.
 - II. Wsparcie dla programowania maszyny stanu dostarczane wraz z bibliotekami z rodziny Boost.
 - III. Warsztat implementacji maszyny stanu.
13. Boost Algorithm.
 - I. Boost Algorithm jako uzupełnienie biblioteki standardowej.
 - II. Algorytmy przeszukiwania.
 - III. Algorytmy pracujące na sekwencjach.
 - IV. Warsztat implementacji.
14. Boost Asio.
 - I. Asynchroniczny dostęp do danych z użyciem Boost Asio.
 - II. Wsparcie komunikacji sieciowej.
 - III. Asio a przesyłanie danych sieciowych.
 - IV. Sygnały.
 - V. Warsztat implementacji.
15. Boost Filesystem.
 - I. Przenośność zapewniana przez Boost Filesystem.
 - II. Boost Filesystem a nowe standardy języka C++.
 - III. Przechwytywanie błędów w Boost Filesystem.
 - IV. Warsztat implementacji.
16. Boost Interprocess.
 - I. Współdzielenie struktur pomiędzy procesami.
 - II. Synchronizacja pracy procesów.



- III. Alokatory i zarządzanie pamięcią.
- IV. Boost Interprocess a inteligentne wskaźniki.
- 17. Boost Intrusive.
 - I. Wydajność i zalety struktur intruzywnych.
 - II. Intruzywne wersje kontenerów.
 - III. Analiza wydajności tradycyjnych rozwiązań i rozwiązań Boost Intrusive.
- 18. Boost MPL.
 - I. Metaprogramowanie – pojęcia, algorytmy i warsztat.
 - II. Szybkie wprowadzenie do metaprogramowania.
 - III. Metaprogramowanie a rzeczywiste wyzwania.
 - IV. Warsztat implementacji.
- 19. Profilowanie aplikacji w języku C++.
 - I. Pojęcie wąskich gardeł w aplikacji.
 - II. Mechanizmy i rodzaje profilowania aplikacji w C++.
 - III. Warsztat implementacji.

