Kod szkolenia: **CPP/ADV**

Tytuł szkolenia: **Zaawansowane programowanie w języku C++**

Dni: 3

Opis:

Uczestnicy szkolenia zapoznają się z metodami wytwarzania oprogramowania z użyciem zaawansowanych mechanizmów języka C++ oraz szablonów STL. Na zajęciach poruszamy zagadnienia związane z koncepcjami algorytmów, iteratorów i zaawansowanych struktur danych. W przykładach wykorzystywane są elementy biblioteki standardowej oraz elementy biblioteki Boost, te które stają się kandydatami do rozwijania następnego standardu.
Uczestnicy szkoleń poznają aspekty programowania wielowątkowego, uogólnionego oraz wstępnie koncepcje leżące u podstaw metaprogramowania ze szczególnym uwzględnieniem cech typów.
Zajęcia uwzględniają także potrzeby testowania aplikacji tworzonych w języku C++ z użyciem narzędzi wspierających proces TDD oraz BDD. Jako biblioteki wspierające, wykorzystane są: Google Test, Boost.Test oraz Catch.
Szczególny nacisk w trakcie zajęć kładziemy na zrozumienie aspektów wydajności oraz gospodarowania zasobami z użyciem wskaźników inteligentnych. Zajmujemy się także zarządzaniem czasem życia obiektu w odniesieniu do mechanizmu wyjątków oraz semantyki r-referencji.
Jako techniki tworzenia aplikacji obiektowej, prezentowane są klasy wytycznych, tradycyjne wzorce projektowe (GoF) oraz nie-ortodoksyjne podejścia z kręgu programowania funkcyjnego.
Na zajęciach posługujemy się technologiami otwartymi z dostępem do kodu źródłowego.

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

Zakres szkolenia

Szkolenie obejmuje:

- konsolidację wiedzy o elementach biblioteki standardowej,
- algorytmy biblioteki standardowej,
- tworzenie i testowanie aplikacji wielowątkowych,
- inteligentne wskaźniki i ich adekwatne stosowanie,
- klasy wytycznych,
- użycie języka w kontekście funkcyjnym.

Wymagania

Od uczestników szkolenia wymaga się:

- posługiwania się wybranym środowiskiem IDE (Eclipse, NetBeans, vim),
- znajomości koncepcji programowania obiektowego, systemowego i aplikacyjnego,
- 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 podstawowym i średnio zaawansowanym.

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

Adresaci szkolenia

Szkolenie adresowane jest do:

- programistów programujących w języku C i C++ w środowisku GNU/Linux lub MS Windows,
- osób znających w podstawowym zakresie system 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 rozwijania 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++.

Umiejętności zdobywane podczas zajęć

Uczestnicy szkolenia po jego zakończeniu zdobędą następujące umiejętności:

- testowanie jednostkowe aplikacji w języku C++,
- tworzenia aplikacji wielowątkowych z użyciem mechanizmów biblioteki standardowej,
- identyfikacji i tworzenia oprogramowania z użyciem wzorców GoF oraz elementów programowania funkcyjnego,

- zapoznania się ze współczesnymi bibliotekami przetwarzającymi dane.

Parametry szkolenia

Szkolenie trwa 3 dni. Szkolenie obejmuje 60% wykładu i 40% ćwiczeń.

Program szkolenia:

1. Algorytmy i iteratory biblioteki standardowej.
    - Przegląd dostępnych algorytmów.
    - Wydajność i aspekty implementacji.
    - Praktyczna implementacja z użyciem algorytmów z biblioteki standardowej.
    - Implementacja własnych algorytmów do ponownego wykorzystania.
    - Złożoność obliczeniowa i zakres zastosowań.
    - Warsztat wykorzystania algorytmów.
    - Rodzaje iteratorów dostępnych w bibliotece standardowej.
2. Iteratory i programowanie uogólnione.
    - Aspekty implementacji iteratorów.
    - Implementacja własnych iteratorów.
    - Programowanie uogólnione – koncepcje i zastosowania.
3. Szablony STL – wydajne stosowanie i wstęp do metaprogramowania.
    - Wzorce – pułapki i zastosowania.
    - Konkretyzacja – reguły i działanie.
    - Cechy typów.
    - Metaprogramowanie – podstawy.
    - Algorytmy na poziomie meta.
4. Wzorce projektowe.
    - Wprowadzenie wzorców projektowych.
    - Strukturalne wzorce projektowe.
    - Konstrukcyjne wzorce projektowe.
    - Behawioralne wzorce projektowe.
    - Specyfika implementacji wzorców projektowych w języku C++.
5. Użyteczne biblioteki
    - Boost – przegląd użytecznych rozwiązań.
    - Biblioteki rozwijające paradygmat programowania uogólnionego.
    - Biblioteki przetwarzania danych.
    - ML i AI w języku C++ - przydatne biblioteki
6. Testowanie i metodyka testowania.
    - Proces TDD w pracy programisty.
    - Proces BDD i różnice w stosunku do TDD
    - Rodzaje bibliotek testów i dostępne asercje.
    - Projektowanie i implementowanie testów.
7. Praca z narzędziem debuggera w języku C++.
    - Dostęp do podstawowych struktur C++.
    - Automatyzacja i skrypty.
    - Wstęp do integracji z infrastrukturą testowania.

8. Programowanie wielowątkowe.
    - Wyzwania i problemy.
    - Współpraca wątków.
    - Inne biblioteki obsługi wielowątkowości (Boost Threads, Intel Threading Building Blocks)
    - Prymitywy synchronizujące.
    - Sekcje krytyczne.
    - Komunikacja i dane współdzielone w aplikacjach wielowątkowych
    - Testowanie aplikacji wielowątkowych.
9. Wzorce wielowątkowości.
    - Wzorce wielowątkowości – przegląd.
    - Implementacja najbardziej użytecznych wzorców wielowątkowości.
    - Zaawansowane wzorce wielowątkowe.
10. Nowe standardy C++.
    - Przegląd nowości wprowadzonych w nowych standardach C++.
    - Zarządzanie pamięcią (wskaźniki inteligentne).
    - Nowe aspekty składni.
    - Obiekty funkcyjne (typ std::function, funkcje lambda, łączenie obiektów)
    - Zarządzanie zakresami.
    - Ujednolicona obsługa systemu plików i czasu.
    - Cechy typów oraz rozszerzenia wielowątkowości (implementacja typów atomowych).
11. Wykładnie i nowoczesne projektowanie aplikacji.
    - Zasady S.O.L.I.D. , GRASP, YAGNI - przegląd i przypomnienie.
    - Wykładnie w programowaniu obiektowym - aspekt praktyczny
    - Implementacja wykładni (ang. policy) we własnych aplikacjach.
    - Rola polimorfizmu statycznego i technik wykonania na etapie kompilacji.
12. Tworzenie aplikacji niezawodnych.
    - Aspekty wydajności obsługi wyjątków.
    - Tworzenie aplikacji odpornej na błędy.
    - Programowanie aspektowe.

Sages sp. z o.o., ul. Nowogrodzka 62c, 02-002 Warszawa
tel. +48 22 203 56 00 fax: +48 22 203 56 01
e-mail: biuro@sages.com.pl web: www.sages.com.pl