C40

Jazyk C++ – moderní C++, STL, šablony a standardní knihovny

Popis:

Kurz Jazyk C++ – Moderní C++, STL, šablony a standardní knihovny je navržen pro C++ programátory, kteří se chtějí seznámit s návrhem šablonových funkcí a tříd s použitím standardních C++ knihoven (STL).

Absolvent kurzu bude umět:

- Princip generického programování
- Rozvinutí šablony, implicitní a explicitní určení parametrů šablony i jejich návrh
- Základní algoritmy copy, accumulate, find, count, min_element, replace, reverse
- Návrh a použití funkčních objektů a predikátů
- Modifikace a přizpůsobování základních algoritmů
- Používat generický kontejner
- Návrh vkládacího iterátoru
- Kontejner vector, jeho struktura a rozhraní
- Alokační strategie vektoru a invalidování iterátorů
- Generování prvků, algoritmus generace
- více viz. osnova

Požadavky pro absolvování kurzu:

Základy programování.

Kurz určen pro:

Programátory, kteří se chtějí seznámit s návrhem šablonových funkcí a tříd s použitím standardních C++ knihoven (STL).

Literatura:

Všichni účastníci školení obdrží materiály společnosti OKsystem.

Technické vybavení:

Všechny učebny jsou vybaveny nadstandardními počítači připojenými k Internetu, učebny jsou prostorné, klimatizované, bezbariérové a s připojením na Wi-Fi. V případě zájmu lze školení absolvovat online live.

Osnova:

- Princip generického programování
- Klíčové slovo template
- Rozvinutí šablony, implicitní a explicitní určení parametrů šablony
- Specializace šablony (vyjímky ze šablony)
- Návrh parametrů šablony
- Základní algoritmy copy, accumulate, find, count, min_element, replace, reverse
- Přetížení operátoru ( ) - kulaté závorky
- Návrh a použití funkčních objektů a predikátů
- Modifikace a přizpůsobování základních algoritmů for_each, transform, find_if, count_if, replace_if, min_element, accumulate
- Návrh generického kontejneru
- Základní operace s kontejnery
- Koncept iterátoru a použití kontejnerů v algoritmech
- Návrh vkládacího iterátoru
- Použití funkcí back_inserter, front_inserter, inserter
- Proudové iterátory input_stream_iterator, output_stream_iterator
- Kontejner vector, jeho struktura a rozhraní
- Alokační strategie vektoru a invalidování iterátorů
- Generování prvků, algoritmus generace
- Zpětné iterátory
- Kontejnery deque, list a jejich speciální vlastnosti
- Třídění vektoru a seznamu, duplikování STL algoritmu členskou metodou kontejneru
- Odstranění prvků z pole, vektoru či seznamu pomocí remove
- Třída basic_string, string a wstring
- Elementární řetězcové operace
- Neformátované čtení řetězců z proudu getline
- Řetězcové proudy v hlavičce
- Kontejnery set, multiset
- Určování třídícího kriteria
- Algoritmy find, count, lower_bound, upper_bound
- Množinové operace set_union, set_intersection, set_difference, set_symmetric_difference
- Kontejnery map, multimap a jejich použití
- Operátor [] u kontejneru map
- Pomocná třída pair
- Předdefinované funkční objekty less, greater, equal_to, plus, minus, multiply…
- Vázání parametrů bind1st, bind2nd
- Adaptéry členských funkcí mem_fun, mem_fun_ref
- Adaptér pro normální funkce ptr_fun
- Návrhový vzor smart pointer
- Návrh, správné a nesprávné použití třídy auto_ptr
- Novinky TR1: shared_ptr (bind, mem_fn)