

KURS JĘZYKA C++

SZABLON LISTY JEDNOKIERUNKOWEJ

Instytut Informatyki Uniwersytetu Wrocławskiego

Paweł Rzechonek

Prolog.

Szablony w C++ umożliwiają programowanie uogólnione, czyli tworzenie abstrakcyjnych algorytmów oraz struktur danych niezależnych od konkretnych typów, na których one pracują. Typ ten jest określany (w sposób jawny bądź niejawny) w miejscu użycia szablonu — wówczas kompilator na podstawie szablonu wygeneruje odpowiednią definicję funkcji czy klasy z określonym już typem.

Zadanie.

Zdefiniuj szablon klasy `list<T>` dla listy jednokierunkowej w przestrzeni nazw `calc`. Klasa reprezentująca listę ma być napisana zgodnie ze sztuką programowania dynamicznych struktur danych — w pełni funkcjonalny węzeł listy `node<T>` zdefiniuj jako prywatną klasę zagnieżdżoną w opakowaniu `list<T>`, posiadającym wygodny dla programisty interfejs z operacjami słownikowymi (wstawienie elementu na zadaną pozycję, usunięcie elementu o podanej wartości, wyszukiwanie elementu, zliczenie wszystkich elementów). Obiekty `list` mają być kopiowalne (konstruktor i przypisanie kopiujące i przenoszące). Uzupełnij definicję szablonu o inicjalizację za pomocą listy wartości `initializer_list<T>`. Nie zapomnij o operatorze strumieniowym do wypisania zawartości listy.

Następnie zdefiniuj szablony dwóch funkcji do pracy z listami: funkcja `issorted()` ma sprawdzić, czy lista jest uporządkowana oraz funkcja `sort()` do sortowania listy. Definicje te także umieść w przestrzeni nazw `calc`. Szablony tych funkcji powinny posiadać dwa parametry: typ danych przechowywanych w liście oraz trejta implementującego operację porównywania elementów wybranego typu. Trejt ma być parametrem domyślnym w szablonie ustawionym na obiekt zawierający operację tradycyjnego porównywania za pomocą operatora `≤`. Zdefiniuj też innego trejta implementującego porównywanie za pomocą `≥`. W trejcie zadbaj o specjalizację dla wskaźników a w szczególności dla wskaźnika typu `const char*`.

Na koniec napisz interaktywny program testujący działanie listy jednokierunkowej i funkcji operujących na takiej liście — program ma interpretować i wykonywać polecenia wydawane z klawiatury. Obiekt listy, który będziesz testować utwórz na sterce operatorem `new`; nie zapomnij zlikwidować go operatorem `delete` przed zakończeniem programu!

Istotne elementy programu.

- Podział programu na pliki nagłówkowe i pliki źródłowe (osobny plik z funkcją `main()`).
- Użycie przestrzeni nazw `calc`.
- Definicja szablonu klasy dla listy jednokierunkowej.
- Zagnieżdżona definicja węzła.
- Szablony funkcji do sortowania danych na liście i do weryfikacji posortowania.
- Definicja trejta realizującego porównania.
- Realizacja specjalizacji dla wskaźników i w szczególności dla `const char*` w trejtach.
- Implementacja kopiowania i przenoszenia list.
- Inicjalizacja stanu początkowego listy za pomocą `initializer_list<T>`.
- Destrukcja listy.
- W funkcji `main()` należy przetestować całą słownikową funkcjonalność listy za pomocą interaktywnego programu.