
C++

kolekcja z iteratorem (szablony)

studia dzienne

W przestrzeni nazw **struktury** zdefiniuj wzorzec klasy dla kolekcji elementów zadanego typu. Od elementów tych będziemy wymagali jedynie operacji porównania w sensie identyczności, co będzie potrzebne przy wyszukiwaniu. Kolekcja ta może być więc zaimplementowana w postaci *tablicy dynamicznej* albo *listy dwukierunkowej*. Kolekcja ta powinna posiadać standardowe metody operacji na zbiorach: dodawanie, usuwanie i wyszukiwanie.

Szablon kolekcji należy wyspecjalizować na przypadek wskaźników: gdy dostarczamy obiekt przez wartość (lub referencję do stałego obiektu), to kolekcja ma przechowywać obiekty (inicjalizacja przy pomocy przypisania kopiującego), ale gdy dostarczamy wskaźnik do obiektu, to kolekcja ma przechowywać wskaźniki do obiektów na stercie (tworzenie przy pomocy konstruktora kopiującego).

Jednak najważniejszym elementem tej kolekcji ma być *iterator*! Iterator to taki specjalny obiekt do pokazywania na elementy w kolekcji. Nie musimy i nie chcemy znać struktury danych użytej do zapamiętania elementów, ale możemy myśleć o nich jak o danych zapisanych na taśmie. Iterator pełni w takim modelu rolę głowicy czytającej, która może się po tej taśmie przesuwać i czytać zapisane tam wartości elementów (nie można ich modyfikować). Ma być dostępny tylko jeden iterator dla danej kolekcji. Gdy w trakcie przeglądania kolekcji za pomocą iteratora dopuścimy się modyfikacji (dodamy coś do zbioru albo usuniemy), to kasujemy iterator. Skasowanie oznacza w tym przypadku, że jego dalsze używanie ma powodować zgłaszanie wyjątków; dopiero powtórna inicjalizacja iteratora ma go ustawić na początku taśmy i pozwolić na ponowny odczyt całej kolekcji.

Następnie (w przestrzeni nazw **testy**) napisz program testujący twoją kolekcję na obiektach typu `complex<double>*` i `string`.