

KURS JĘZYKA C++

KOLEJKA ZBUDOWANA NA TABLICY

Instytut Informatyki Uniwersytetu Wrocławskiego

Paweł Rzechonek

Prolog.

Kolejka to bufor na dane, do którego można wkładać nowe elementy oraz wyciągać elementy, ale tylko w kolejności zgodnej z czasem ich włożenia — z kolejki można usunąć tylko ten element, który został do niej włożony najwcześniej.

Odpowiednikiem tej struktury w codziennym życiu może być na przykład kolejka klientów do kasy w sklepie — klient, który ustawił się w kolejce jako pierwszy zostanie jako pierwszy obsłużony. Stąd się wzięła nazwa takiego bufora: FIFO (ang. *First In, First Out*), czyli „kto pierwszy przychodzi ten pierwszy wychodzi”.

Zadanie.

Zdefiniuj klasę `kolejka`, która będzie strukturą typu FIFO — element, który został do tej struktury dodany najwcześniej, będzie z niej wyciągnięty najszybciej. Struktura ta ma służyć do przechowywania napisów typu `string`.

Sama funkcjonalność kolejki ma być bardzo prosta: wkładamy napis do kolejki (metoda `void wloz(string)`), wyciągamy napis z kolejki (metoda `string wyciagnij()`), sprawdzamy jaki napis jest na początku kolejki (metoda `string naprzodzie()`) oraz pytamy o liczbę wszystkich elementów w kolejce (metoda `int rozmiar()`).

Kolejkę zaimplementuj na tablicy utworzonej dynamicznie na stercie (za pomocą operatora `new[]`, na przykład `kol = new string[pojemnosc]`). W destruktorze należy zwolnić pamięć przydzieloną dla tej tablicy (za pomocą operatora `delete[]`, na przykład `delete[] kol`). Programując kolejkę wykorzystaj zawinięcie w tablicy (następny po ostatnim elemencie jest pierwszy element), co ułatwi obsługę tej struktury.

Pojemność kolejki ma zostać określona w konstruktorze — zdefiniuj więc prywatne pole `pojemnosc` typu `int`, w którym będzie pamiętany maksymalny rozmiar kolejki. Będziesz też potrzebować informacji o miejscu gdzie kolejka się zaczyna oraz o liczbie elementów aktualnie przechowywanych w kolejce — zdefiniuj zatem prywatne pola `pocz` oraz `ile` typu `int`, w których będziesz pamiętała odpowiednio miejsce pierwszego elementu oraz liczbę elementów przechowywanych w kolejce.

Kolejka ma posiadać pięć konstruktorów: konstruktor z zadaną pojemnością, konstruktor bezparametrowy i jednocześnie delegatowy (domyślna pojemność kolejki niech wynosi 1), konstruktor inicjalizujący kolejkę za pomocą listy napisów (za pomocą `initializer_list<string>`), konstruktor kopiujący i przenoszący. Aby uzupełnić semantykę kopiowania i przenoszenia zdefiniuj odpowiednie operatory przypisania (przypisanie kopiujące i przenoszące).

Napisz też interaktywny program testujący działanie kolejki (interpretuj i wykonuj polecenia wydawane z klawiatury). Obiekt kolejki, który będziesz testować stwórz na stercie operatorem `new` i nie zapomnij zlikwidować go operatorem `delete` przed zakończeniem programu!

Istotne elementy programu.

- Podział programu na plik nagłówkowy (np. `kolejka.hpp`) z definicją klasy reprezentującej kolejkę, plik źródłowy (np. `kolejka.cpp`) z definicją funkcji składowych dla kolejki oraz plik źródłowy z funkcją `main()` (np. `main.cpp`).
- Obiekt kolejki ma być inicjalizowany na kilka różnych sposobów: konkretną pojemnością, domyślnie (konstruktor delegatowy), przez skopiowanie z innej kolejki (konstruktor przenoszący), za pomocą listy wartości początkowych (lista wartości inicjalizujących zawartość kolejki), za pomocą przeniesienia zawartości z kolejki tymczasowej (konstruktor przenoszący).
- Obiekt kolejki ma być kopiowalny (przypisanie kopiujące i przenoszące).
- Zgłaszanie wyjątków w konstruktorach i funkcjach składowych.
- W funkcji `main()` należy zaprogramować interakcję programu z użytkownikiem — użytkownik wydaje polecenie dotyczące kolejki, program je interpretuje i realizuje (wstawienie, uzunięcie, odpytanie o ilość elementów, odpytanie o pojemność kolejki, wypisanie zawartości całej kolejki).