Zadanie 3 *21–24 marca 2017 r.*

JĘZYK PROGRAMOWANIA C++

KOLEJKA ZBUDOWANA NA TABLICY

Instytut Informatyki Uniwersytetu Wrocławskiego *Paweł Rzechonek*

Kolejka danych to bufor, do którego można w dowolnym momencie wkładać oraz wyciągać pewne elementy. Cechą charakterystyczną dla kolejki jest to, że w danym momencie można z niej wyciągnąć tylko ten element, który został do niej najwcześniej włożony.

Odpowiednikiem tej strukruty w codziennym życiu może być na przykład kolejka do kasy w sklepie — osoba, która ustawiła się w kolejce jako pierwsza zostanie jako pierwsza obsłużona. Stąd nazwa takiego bufora: FIFO (ang. *First In, First Out*).

Zadanie.

Zdefiniuj klasę `kolejka`, która będzie buforem 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: wtawiamymy napis do kolejki (metoda `void wstaw(string)`), pobieramy napis z kolejki (metoda `string pobierz()`), sprawdzamy jaki napis znajduje się na początku kolejki (metoda `string sprawdz()`) oraz pytamy o liczbę wszystkich napisów 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`).

Pojemność kolejki ma być ustalona w konstruktorze — zdefiniuj więc prywatne pole `pojemnosc` typu `int`, w którym będzie pamiętany maksymalny rozmiar kolejki. Będziesz też potrzebować informacji o liczbie elementów aktualnie włożonych do kolejki i o pozycji na której znajduje się pierwszy element w kolejce — zdefiniuj zatem prywatne pola `ile` oraz `poczatek` typu `int`, w których będą pamiętane te informacje.

Kolejka ma posiadać pięć konstruktorów: konstruktor z zadaną pojemnością, konstruktor bezparametrowy i jednocześnie delegatowy (domyślna pojemność kolejki niech wynosi 4), konstruktor inicjalizyjący stos za pomocą listy napisów (za pomocą `initializer_list<string>`), konstruktor kopiujący i przenoszący. Aby uzupełnić semantykę kopiowania i przenoszenia zdefiniuj opowiednie operatory przypisania (przypisanie kopijące 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!

Uwaga.

Podziel program na pliki nagłówkowe i źródłowe.

Elementy w programie, na które należy zwrócić szczególną uwagę.

- Podział programu na plik nagłówkowy (np. `kolejka.hpp`) z definicją kasy reprezentującej kolejkę zbudowaną na tablicy, plik źródłowy (np. `kolejka.cpp`) z definicją funkcji składowych z klasy modelującej kolejkę oraz plik źródłowy (np. `main.cpp`) z funkcją `main()`.
- 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).
- W funkcji `main()` należy zaprogramować interakcję programu z użytkownikiem — użytkownik wydaje polecenie dotyczące kolejki, program je interpretuje i realizuje (wstawienie, uzunięcie, odpytanie o olość elementów, odpytanie o pojemność kolejki, wypisanie zawartości całej kolejki).