

KURS JĘZYKA C++

STOS NAPISÓW

Instytut Informatyki Uniwersytetu Wrocławskiego

Paweł Rzechonek

Zadanie.

Zdefiniuj klasę `stos`, która będzie strukturą typu LIFO — element, który został do tej struktury dodany najpóźniej, będzie z niej wyciągnięty najszybciej. Struktura ta ma służyć do przechowywania łańcuchów znakowych typu `string`.

Sam `stos` zaimplementuj w postaci tablicy tworzonej dynamicznie na stercie (za pomocą operatora `new[]`, na przykład `new string[pojemnosc]`). W destruktorze należy zwolnić pamięć przydzieloną dla tej tablicy (operatorem `delete[]`, na przykład `delete[] stos`). Pojemność `stosu` ma być niezmienna i ustalona w konstruktorze — zdefiniuj więc publiczne stałe pole `pojemnosc` typu `int`, w którym będziesz pamiętała maksymalną wielkość `stosu`. Dodatkowo będzie potrzebna informacja o liczbie elementów aktualnie włożonych na `stos` — zdefiniuj zatem prywatne pole `ile` typu `int`, w którym będziesz pamiętała liczbę elementów przechowywanych na `stosie`. Pamiętaj, aby w klasie `stos` znalazł się konstruktor z zadaną pojemnością `stosu`, konstruktor bezparametrowy (wówczas pojemność `stosu` ma być ustalona na 1) konstruktor kopiujący, przenoszący i konstruktor inicjalizujący `stos` za pomocą listy napisów (jeśli lista posiada d elementów, to wtedy ustal pojemność `stosu` na $2d + 1$).

Sama funkcjonalność `stosu` ma być bardzo prosta: kładziemy napis na `stos` (metoda `void wloz (string)`), ściągamy napis ze `stosu` (metoda `string sciagnij ()`), sprawdzamy jaki napis leży na wierzchu (metoda `string sprawdz ()`) oraz pytamy o liczbę wszystkich elementów na `stosie` (metoda `int rozmiar ()`).

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

Uwaga.

Podziel program na pliki nagłówkowe i źródłowe. Definicję klasy umieść w pliku `stos.hpp` a definicje funkcji składowych w pliku `stos.cpp`. Interaktywny program testujący umieść w pliku `main.cpp`.

Podpowiedź.

W funkcjach składowych i w konstruktorze `stosu` zgłaszaj błędy za pomocą instrukcji `throw`. Przyczyną błędów mogą być sytuacje, w których program będzie chciał włożyć element na maksymalnie zapelniony `stos`, albo ściągnąć element z pustego `stosu`, albo utworzyć `stos` o pojemności niedodatniej, itp. Niech wyjątkami będą obiekty typu `string`.