laboratorium: zadanie 2 termin: 17–20 marca 2014 r.

KURS JĘZYKA C++

STOS NAPISÓW

Instytut Informatyki Uniwersytetu Wrocławskiego *Paweł Rzechonek*

Zadanie.

Zdefiniuj klasę `stos`, która będzie służyć do składowania napisów typu `string` na stosie.

Sam stos zaimplementuj w postaci prywatnej tablicy tworzonej dynamicznie (za pomocą operatora `new[]`, na przykład `new string[rozmiar]`). W destruktorze należy zwolnić pamięć przydzieloną tablicy (operatorem `delete[]`, na przykład `delete[] stos`). Rozmiar stosu ma być niezmienny i ustalony w konstruktorze — zdefiniuj publiczne stałe pole `rozmiar` typu `int`, w którym będziesz pamiętać wielkość stosu. Dodatkowo będziesz potrzebować informacji o liczbie elementów włożonych na stos — zdefiniuj prywatne pole `ile` typu `int`, w którym będziesz pamiętać liczbę elementów przechowywanych aktualnie na stosie. Pamiętaj, aby w klasie `stos` znalazł się konstruktor z podanym rozmiarem stosu, konstruktor bezparametrowy (wówczas rozmiar stosu ma być ustalony na 1) oraz kopiujący.

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 zapelnienie ()`).

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`.