

C++

kalkulator ONP (szablony)

studia wieczorowe

Napisz program *interaktywnego kalkulatora*. Kalkulator ten powinien interpretować i obliczać wyrażenia zapisane w systemie ONP. Program powinien odczytywać polecenia ze standardowego wejścia `cin`, wykonywać obliczenia i ewentualnie wypisywać wyniki na standardowe wyjście `cout`. Wszelkie komentarze i uwagi czynione przez program mają być wysyłane na standardowe wyjście dla błędów `clog`.

Program powinien rozpoznawać trzy rodzaje poleceń:

- **set** *zm = wyrażenieONP*
Utworzenie nowej zmiennej *zm* i przypisanie jej wartości obliczonego wyrażenia *wyrażenieONP*. Jeśli zmienna *zm* była zdefiniowana już wcześniej, to należy tylko zmodyfikować przypisaną do niej wartość. Zmienne pamiętaj w zaprogramowanej przez siebie liście asocjacyjnej.
- **print** *wyrażenieONP*
Obliczenie wartości wyrażenia *wyrażenieONP* i wypisanie jej na standardowym wyjściu. Wyrażenie będzie zapisane w postaci postfiksowej (*Odwrotna Notacja Polska*). Czytając kolejne tokeny wyrażenia program powinien je zamieniać na elementy obliczalne i umieszczać w kolejce. Przy obliczaniu wartości wyrażenia należy się posłużyć stosem.
- **clear**
Usunięcie wszystkich zmiennych zapamiętanych do tej pory w liście asocjacyjnej. Do tablicy mogą trafiać tylko zmienne o nazwach różnych od nazw zaprogramowanych przez siebie funkcji (`pi`, `e`, `abs`, `floor`, `ceil`, `frac`, `min`, `max`, `sin`, `cos`, `atan`, `acot`, `log`, `ln`, `power`, `ex`, i innych).

Po wydaniu każdego polecenia, jeśli było ono poprawnie zapisane, to należy je powtórzyć na standardowym wyjściu dla błędów, a jeśli było niepoprawne (nieznana komenda, źle sformułowane wyrażenie, błędna nazwa, błędny literał stałopozycyjny, czy nierozpoznany operator, funkcja lub zmienna) to należy wypisać stosowny komunikat o błędzie.

W programie wykorzystaj zaprogramowane przez siebie wcześniej struktury danych (lista asocjacyjna, kolejka, stos) i całą hierarchię klas dla elementów dających się policzyć. Program należy napisać zgodnie z kunsztem sztuki programistycznej, dzieląc kod na części nagłówkowe i źródłowe i umieszczając je w osobnych plikach.

Powodzenia!