

Projekt: Odwrotna Notacja Polska (ONP)

Autor: Łukasz Maruszak

Program pozwala na obliczenie wartości wyrażenia w ONP oraz konwersję wyrażenia z notacji tradycyjnej do ONP.

1. Opis użytej struktury danych oraz zaimplementowanych funkcji

Zastosowana struktura danych to stos w reprezentacji tablicowej. Kolejne elementy stosu znajdują się w komórkach $S[0]$, $S[1]$... $S[TOP]$. Maksymalna wielkość stosu jest zdefiniowana 'maxlength' i wynosi 100. Klasa Stos zawiera dwa pola protected $S[\text{maxlength}]$ oraz position Top czyli szczyt stosu. Konstruktor ustawia $Top = -1$, to samo wykonuje metoda MakeNull, usuwa wszystkie elementy ze stosu. Do wstawiania wartości na Stos służy Push(elementtype x) a do zdjęcia elementu ze stosu służy metoda Pop, która zmniejsza Top o 1. Dzięki metodzie Empty możemy sprawdzić czy Stos jest pusty, gdyż zwraca ona true kiedy $Top == -1$. TopElem() zwraca nam wartość elementu na szczycie Stosu.

```
int okresl_piorytet_operatora(char znak)
```

Funkcja służąca do określania wartości priorytetu znaku. Dodawanie oraz odejmowanie mają najniższy czyli 1, 2 posiada mnożenie oraz dzielenie. Najwyższy priorytet ma potęgowanie – 3. W innym przypadku zwracana jest wartość 0.

```
void odwrotna_notacja_polska(string obliczTo)
```

Funkcja odpowiedzialna za obliczanie wartości wyrażenia podanego w ONP. Na początku tworzony jest obiekt stos, do której dodawane lub zdejmowane będą wprowadzone przez użytkownika elementy. Pętla obliczająca wartość przechodzi po każdym elemencie. Na początku sprawdzam czy jest to spacja, jeżeli tak to przechodzę do następnego elementu. Sprawdzam czy jest liczbą i dodaje ją na stos. Jeżeli znak jest jednym z sześciu operatorów to wykonuje odpowiednie dla tego operatora operacje arytmetyczne. Na początku jednak ściągam dwa elementy ze stosu i na nich wykonuje operację. Po obliczeniu wynik znów wstawiam na stos. Jeżeli natrafiam na '=' to wypisuje wynik jaki znajduje się na stosie.

```
void konwertuj_wyrazenie_do_ONP(string konwertujTo)
```

Funkcja odpowiedzialna za konwersję wyrażenia zapisanego w notacji tradycyjnej na ONP. Na początku tworzony jest obiekt stos, na którym znajdować się będą operatory. Funkcja Przechodzi po każdym elemencie w wyrażeniu i wykonuje odpowiednie dla niego instrukcje, jeżeli jest to liczba dopisuje ją do zmiennej wynik. Kiedy trafimy na '(' wstawiamy go na stos. Nawias zamykający ')' sprawia że ze stosu zdejmowane są elementy aż stos będzie pusty lub trafimy na '(' i dopisujemy te elementy do zmiennej wynik. Kiedy elementem jest operand musimy najpierw określić jego priorytet i porównać z priorytetem znaku na stosie a następnie albo dopisać znak do zmiennej wynik lub dodać go na stos. Jeżeli natrafiam na '=' to ze stosu zostają zdjęte elementy znajdujące się na nim oraz wypisany wynik.

2. Uruchomienie i działanie programu

Po uruchomieniu przez użytkownika programu wyświetlony zostaje napis powitalny oraz informacje jakie czynności może wykonać. Użytkownik ma trzy warianty wyboru 'a', 'b' lub 'q'. Po wybraniu 'a' użytkownik proszony jest o wpisanie wyrażenia w ONP zakończonego '=' program wylicza jego wartość i wyświetla użytkownikowi. Elementy oddzielone powinny być spacją.

Wybranie 'b' to konwersja wyrażenia w notacji tradycyjnej na ONP. Użytkownik podaje wyrażenie, elementy nie muszą być one oddzielone spacją lecz wyrażenie musi być zakończone '='. Po zatwierdzeniu program przekształca wyrażenie na ONP i wyświetla wynik użytkownikowi.

Wybranie 'q' to zakończenie programu, użytkownik nie może wykonać już żadnej opcji.