

Projekt zaliczeniowy nr 1: Odwrócona notacja Polska

Autor: Piotr Piechowicz

Program implementujący ONP składa się z:

- 1) Implementacji tablicowej stosu
- 2) Funkcji obliczającej podaną wartość w notacji ONP
- 3) Funkcji konwertującej wyrażenie na wyrażenie ONP
- 4) Funkcji pomocniczych

1. Implementacja stosu składa się z następujących metod:

- `Makenull()` – czyści stos z elementów
- `TopElem()` – zwraca element znajdujący się na wierzchołku stosu
- `Pop()` – Usuwa element znajdujący się na wierzchołku stosu
- `Push(elementtype x)` – umieszcza element `x` na wierzchołku stosu. Element, który znajdował się poprzednio na wierzchołku stosu staje się wtedy następnym po `x`, itd.
- `Empty()` – zwraca `true`, jeśli stos jest pusty, w przeciwnym razie zwraca `false`
- `Print()` – wypisuje w konsoli wszystkie elementy znajdujące się na stosie w kolejności od elementu na szczycie stosu

2. Funkcja obliczająca wartość z notacji ONP składa się z:

- Zapisu danych wpisanych do konsoli i podziału ich na pojedyncze operandy/operators, które są zapisywane w strukturze `vector`, dzięki czemu nie ma ograniczenia na długość wpisywanych danych
- Odczytu kolejnych znaków z `vectora` i wykonaniu odpowiednich działań :
 - jeśli odczytano operand to umieszczamy go na stosie
 - jeśli odczytano operator to ściągamy ze stosu odpowiednią ilość argumentów, dokonujemy działania i zwracamy wynik na stos
- Po odczytaniu wszystkich wartości z `vectora` program wypisuje do konsoli wynik (ostatnią wartość ze stosu)

3. Funkcja konwertująca wyrażenie na ONP składa się z:

- Zapisu danych wpisanych do konsoli i podziału ich na pojedyncze operandy/operators, które są zapisywane w strukturze `vector`, dzięki czemu nie ma ograniczenia na długość wpisywanych danych
- Odczytu kolejnych znaków z `vectora` i wykonaniu odpowiednich działań :
 - jeśli odczytano operand to wypisujemy go w konsoli
 - jeśli odczytano operator i stos jest pusty to umieszczamy go na stosie
 - jeśli odczytano operator i stos nie jest pusty to sprawdzamy czy operator na szczycie stosu ma priorytet mniejszy lub równy aktualnemu elementowi: jeśli tak to ściągamy operator ze

stosu i wypisujemy go w konsoli i powtarzamy procedurę , jeśli nie to dodajemy aktualny element na szczyt stosu

- jeśli odczytano nawias "(" to wstawiamy go na stos, jeśli odczytano nawias ")" to ściągamy ze stosu i wypisujemy w konsoli wszystkie operatory ze stosu aż trafimy na "(", który następnie zdejmujemy ze stosu

- Po odczytaniu wszystkich danych z vectora ściągamy elementy ze stosu i wypisujemy do konsoli dopóki stos nie jest pusty

4.Funkcje pomocnicze:

- operand(x) – zwraca true jeśli element x jest operandem, false w przeciwnym wypadku
- operacja(x, y, operator) – wykonuje odpowiednie działanie na x i y zależnie od operatora
- priorytet(operator) – zwraca priorytet dla danego operatora:

3 dla operacji potęgowania

2 dla dzielenia i mnożenia

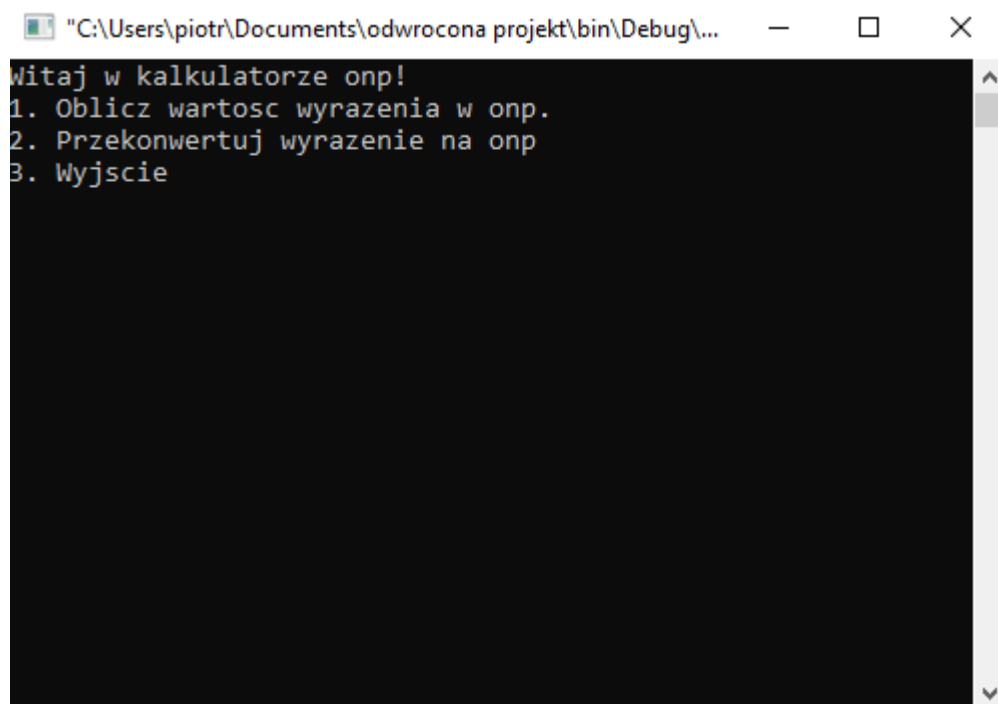
1 dla dodawania i odejmowania

Obsługa programu:

Po uruchomieniu program wyświetla menu, po wybraniu odpowiedniego numeru i zatwierdzeniu go enterem możemy wpisywać wyrażenie (poszczególne elementy należy oddzielić spacją) a następnie po zatwierdzeniu enterem program zwraca wynik i powraca do menu.

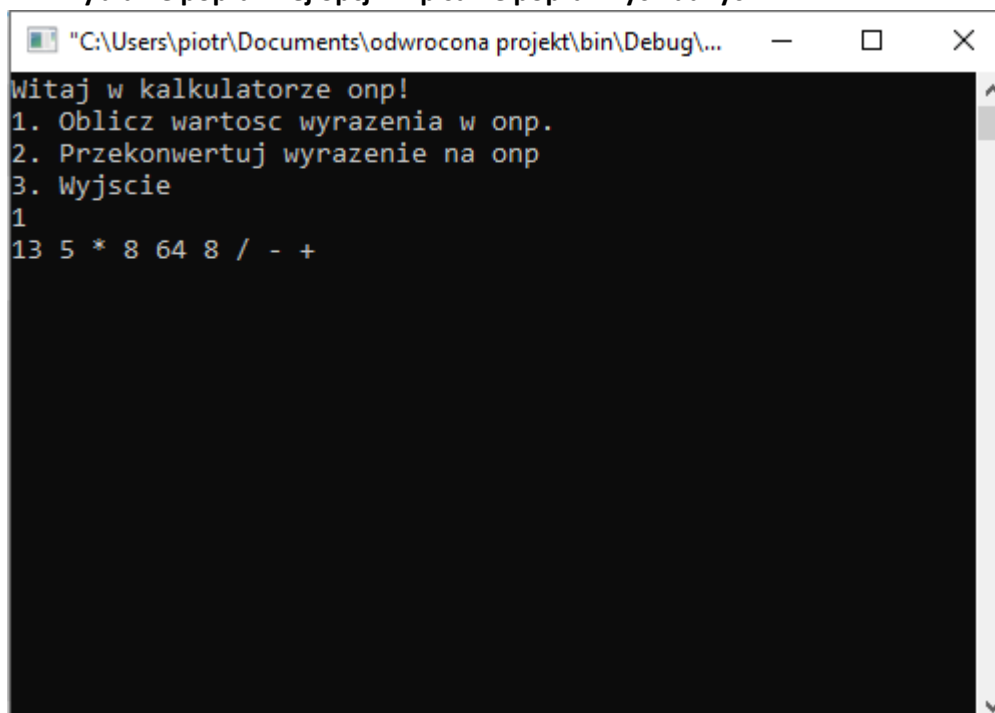
Przykład obsługi programu:

1. Menu:



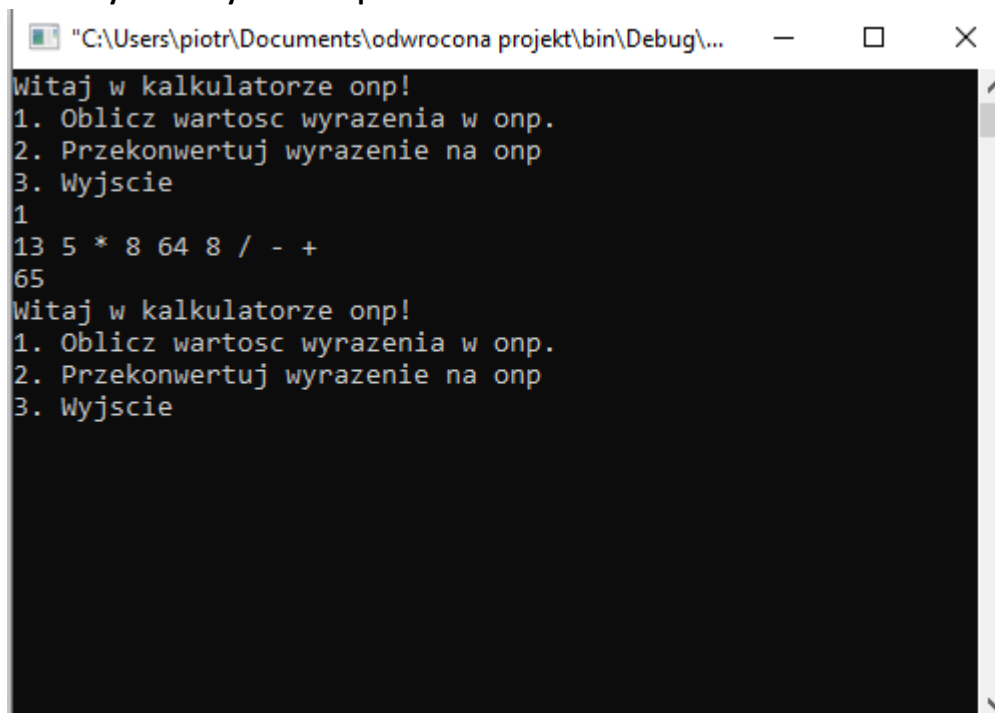
```
"C:\Users\piotr\Documents\odwrocona projekt\bin\Debug\..."
Witaj w kalkulatorze onp!
1. Oblicz wartosc wyrazenia w onp.
2. Przekonwertuj wyrazenie na onp
3. Wyjście
```

2. Wybranie poprawnej opcji i wpisanie poprawnych danych:



```
"C:\Users\piotr\Documents\odwrocona projekt\bin\Debug\..."
Witaj w kalkulatorze onp!
1. Oblicz wartosc wyrazenia w onp.
2. Przekonwertuj wyrazenie na onp
3. Wyjście
1
13 5 * 8 64 8 / - +
```

3. Otrzymanie wyniku oraz powrót do menu:



```
"C:\Users\piotr\Documents\odwrocona projekt\bin\Debug\..."
Witaj w kalkulatorze onp!
1. Oblicz wartosc wyrazenia w onp.
2. Przekonwertuj wyrazenie na onp
3. Wyjście
1
13 5 * 8 64 8 / - +
65
Witaj w kalkulatorze onp!
1. Oblicz wartosc wyrazenia w onp.
2. Przekonwertuj wyrazenie na onp
3. Wyjście
```