

# Odwrótna notacja polska

## Programowanie funkcyjne

Matejko Marek, Mazur Krzysztof, Paszkot Dawid

Fizyka Techniczna  
Politechnika Krakowska

# Plan Prezentacji

Odwrócona  
notacja polska

Matejko, Mazur,  
Paszko

Historia

Historia

Odwrócona  
Notacja Polska

Odwrócona Notacja Polska

Obliczanie  
wyrażeń RPN

Obliczanie wyrażeń RPN

Interpretacja  
kodu

Interpretacja kodu

Bibliografia

Bibliografia

Historia

Historia

Odwrócona  
Notacja Polska

Odwrócona Notacja Polska

Obliczanie  
wyrażeń RPN

Obliczanie wyrażeń RPN

Interpretacja  
kodu

Interpretacja kodu

Bibliografia

Bibliografia

Odwrotna notacja polska ONP (ang. RPN – Reverse Polish Notation ), często jest również zwana notacją Postfix, została zaproponowana w 1954 r. i wprowadzona we wczesnych latach 60. Powstała na podstawie notacji polskiej, czyli prefixowej, która została stworzona przez polskiego matematyka w 1924 przez Jana Łukasiewicza.



Jan Łukasiewicz 1878 - 1956

RPN pozwala na całkowitą rezygnację z używania nawiasów w wyrażeniach, jako że jednoznacznie określona jest kolejność wykonywanych działań, co pozwala na zmniejszenie dostępu do pamięci komputera oraz na używanie stosu do oceny wyrażeń.

3 4 +

W latach 70. i 80. firma Hewlett-Packard używała RPN we wszystkich swoich kalkulatorach biurkowych i ręcznych, a w niektórych modelach używała go do lat 2020. W informatyce odwrócona notacja polska jest używana w językach programowania zorientowanych na stos, takich jak Forth, STOIC, PostScript, RPL i Joy.

Historia

Odwrócona Notacja Polska

Obliczanie wyrażeń RPN

Interpretacja kodu

Bibliografia

Historia

Odwrócona  
Notacja Polska

Obliczanie  
wyrażeń RPN

Interpretacja  
kodu

Bibliografia

Zwykle, gdy pracujemy z wyrażeniami algebraicznymi w szkole, piszemy je w sposób wrostkowy. Na przykład piszemy  $10 - (4 + 3) \cdot 2$ . Dodawanie (+), mnożenie (\*) i odejmowanie (-) to operatory wrostkowe, podobnie jak funkcje infix w Haskell (+ 'elem' itd.). Jako ludzie możemy to przeanalizować łatwo tworząc się w naszych umysłach. Minusem jest to, że musimy używać nawiasów do oznaczenia pierwszeństwa.

Innym sposobem pisania wyrażeń algebraicznych jest użycie notacji odwrotnej, czyli RPN. W RPN operator występuje po liczbach, a nie między nimi. Więc zamiast pisać  $4 + 3$ , piszemy  $4 3 +$ . Ale jak piszemy wyrażenia zawierające kilka operatorów? Na przykład, jak napisalibyśmy wyrażenie, które dodaje 4 i 3, a następnie mnoży że o 10? To proste:  $4 3 + 10 *$ . Ponieważ  $4 3 +$  jest równoważne 7, ta całość wyrażenie jest takie samo jak  $7 10 *$ .



Historia

Odwrócona Notacja Polska

Obliczanie wyrażeń RPN

Interpretacja kodu

Bibliografia

Historia

Odwrócona  
Notacja Polska

Obliczanie  
wyrażeń RPN

Interpretacja  
kodu

Bibliografia

Aby dowiedzieć się, jak obliczać wyrażenia RPN, pomyślmy o stosie liczb. Przechodzimy przez wyrażenie od lewej do prawej. Za każdym razem, gdy liczba jest napotkana, jest kładzona na wierzch stosu. Kiedy spotykamy operator, bierzemy dwie liczby znajdujące się na szczycie stosu, używamy operatora z tymi dwoma liczbami, a następnie wynik kładziemy na stos. Kiedy dochodzimy do końca wyrażenia, powinna zostać jedna liczba, która reprezentuje wynik (zakładając, że ekspresja była dobrze uformowana).

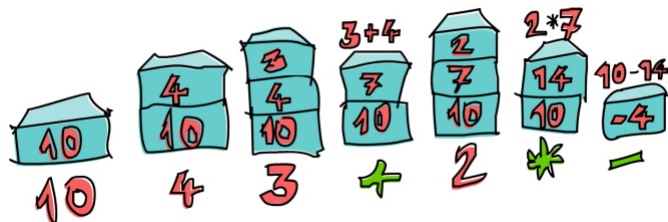
## Zobaczmy, jak obliczylibyśmy wyrażenie

RPN  $10\ 4\ 3\ +\ 2\ * - :$

1. Wkładamy 10 na stos, więc stos składa się z 10.
2. Następna pozycja to 4, więc ją również wciskamy na stos. Stos jest teraz 10, 4.
3. Robimy to samo z 3, a stos wynosi teraz 10, 4, 3.
4. Spotykamy operator: +. Wyciągamy dwie liczby ze szczytu stosu (więc teraz stos wynosi tylko 10), dodaj te liczby razem i naciśnijten wynik do stosu. Stos wynosi teraz 10, 7.
5. Odkładamy 2 na stos, więc stos wynosi 10, 7, 2.
6. Spotykamy inny operator. Zrzucamy 7 i 2 ze stosu, mnożymy je i odłładamywynik na stos. Mnożenie 7 i 2 daje 14, więc stos wynosi teraz 10, 14.

7. Wreszcie jest -. Zrzucamy 10 i 14 ze stosu, odejmujemy 14 od 10, i wrzucamy to stos.

8. Liczba na stosie wynosi teraz -4. Bo nie ma już liczb lub operatorów w naszym wyrażeniu, to nasz wynik!



Rysunek: Reprezentacja graficzna

Historia

Historia

Odwrócona  
Notacja Polska

Odwrócona Notacja Polska

Obliczanie  
wyrażeń RPN

Obliczanie wyrażeń RPN

Interpretacja  
kodu

Interpretacja kodu

Bibliografia

Bibliografia

# Interpretacja kodu

Odwrócona  
notacja polska

Matejko, Mazur,  
Paszkot

Historia

Odwrócona  
Notacja Polska

Obliczanie  
wrażen RPN

Interpretacja  
kodu

Bibliografia

```
~/Documents/haskell/RPN.hs - Mousepad
File Edit Search View Document Help
1 solveRPN :: String -> Double
2 solveRPN = head . foldl foldingFunction [] . words
3     where foldingFunction (x:y:ys) "*" = (y * x):ys
4           foldingFunction (x:y:ys) "+" = (y + x):ys
5           foldingFunction (x:y:ys) "-" = (y - x):ys
6           foldingFunction (x:y:ys) "/" = (y / x):ys
7           foldingFunction (x:y:ys) "^" = (y ** x):ys
8           foldingFunction (x:xs) "ln" = log x:xs
9           foldingFunction xs "sum" = [sum xs]
10          foldingFunction xs numberString = read numberString:xs
```

Historia

Historia

Odwrócona  
Notacja Polska

Odwrócona Notacja Polska

Obliczanie  
wyrażeń RPN

Obliczanie wyrażeń RPN

Interpretacja  
kodu

Interpretacja kodu

Bibliografia

Bibliografia

- [1] [https://en.wikipedia.org/wiki/Reverse\\_Polish\\_notation](https://en.wikipedia.org/wiki/Reverse_Polish_notation)
- [2] 1ReversePolishNotationHaskell.pdf