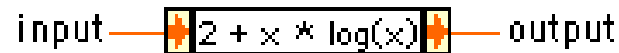
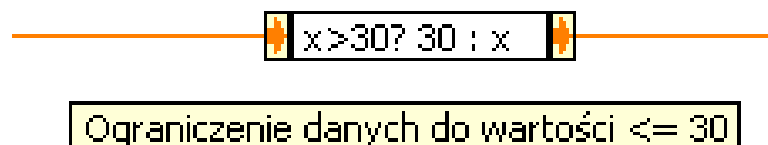


# Wyrażenia

## Expression Node



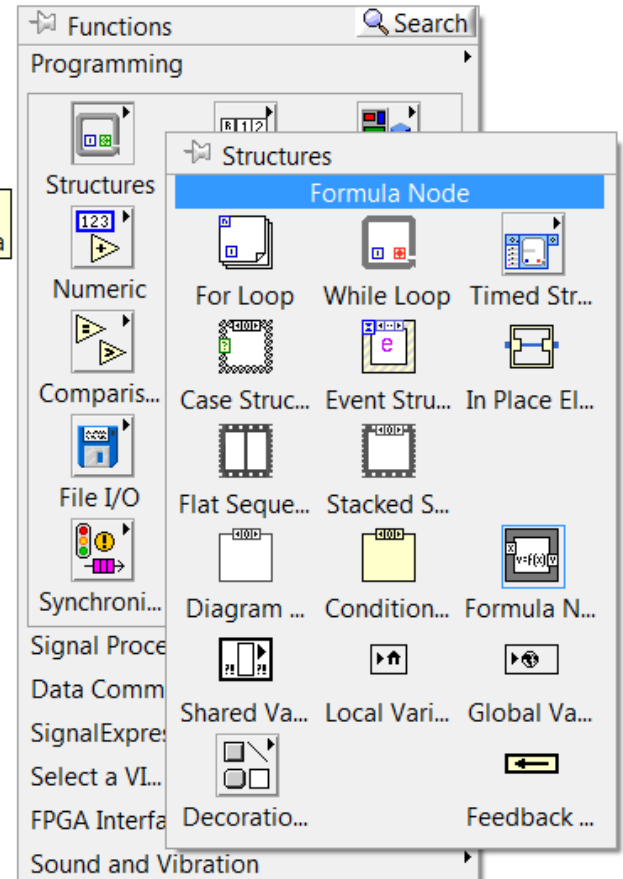
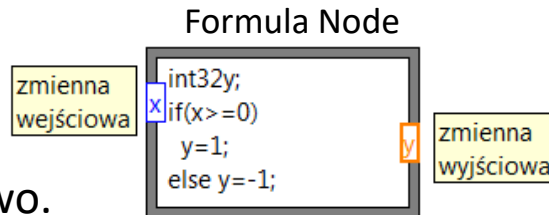
- Węzeł stosuje się do obliczenia wyrażenia zawierającego jedną zmienną.
- Wejście węzła jest zmienną wyrażenia o nazwie X.
- Wynik wyrażenia jest daną wyjściową węzła.
- Wyrażenie może korzystać z funkcji: abs, acos, acosh, asin, asinh, atan, atanh, ceil, cos, cosh, cot, csc, exp, expm1, floor, getexp, getman, int, intrz, ln, lnp1, log, log2, max, min, mod, rand, rem, sec, sign, sin, sinc, sinh, sizeOfDim, sqrt, tan, tanh.
- Przykład:



# Formuły i skrypty

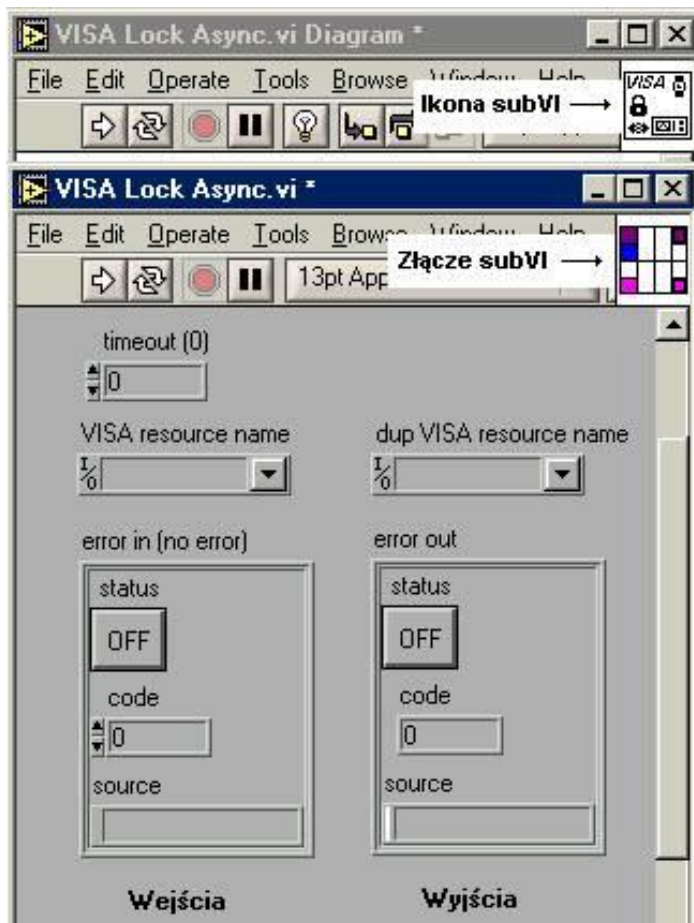
- Węzły formuł i skryptów są użyteczne do realizacji operacji zapisanych tekstowo.
- Formuła matematyczna (**Formula Node**) - blok tekstowy zawierający formuły matematyczne i wyrażenia o syntaktyce podobnej do języka C.
- W formule matematycznej dostępne są następujące funkcje:

Abs, acos, acosh, asin, atan, atan2, atanh, ceil, cos, cosh, cot, csc, exp, expm1, floor, getex, getman, int, intrz, ln, lnp1, log, log2, max, min, mod, pow, rand, rem, sec, sign, sin, sinc, sinh, sizeofDim, sqrt, tan, tanh.



# Ikona i złącze programu VI

Gdy program VI ma służyć jako podprogram dla innych aplikacji VI konieczne jest zbudowanie ikony stanowiącej jego graficzną reprezentację oraz zdefiniowanie złącza.

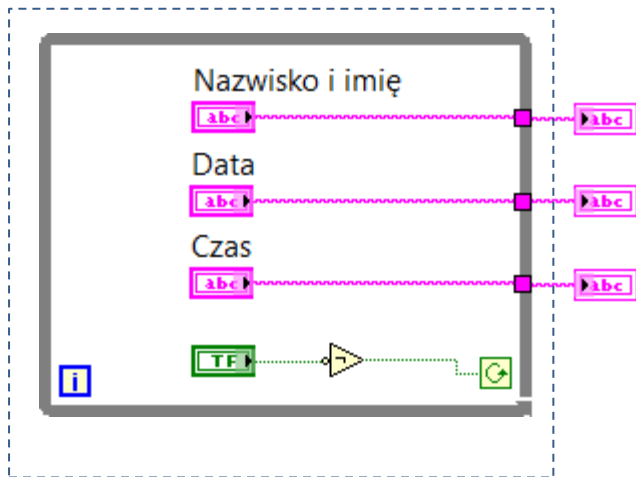


**Ikona** podprogramu jest jego graficzną reprezentacją w diagramach aplikacji wykorzystujących taki podprogram. Stanowi węzeł budowanego diagramu.

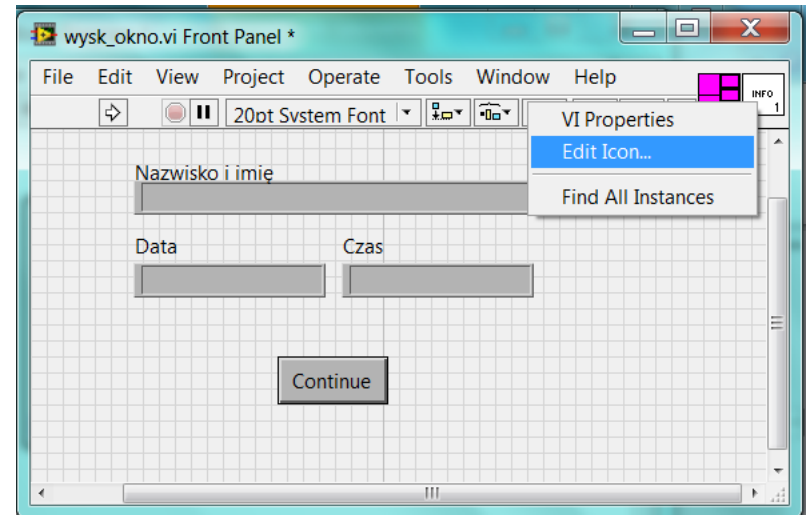
**Złącze** definiuje wejścia i wyjścia podprogramu i tym samym umożliwia wykonanie odpowiednich połączeń w diagramie programu wykorzystującego go jako podprogram.

- Złącze jest zestawem końcówek odpowiadających określonym elementom kontrolnym i prezentacyjnym pulpitu danego podprogramu.
- Zaciski wejściowe złącza przekazują dane do diagramu podprogramu za pośrednictwem elementów kontrolnych pulpitu podprogramu.
- Zaciski wyjściowe złącza otrzymują dane z diagramu za pośrednictwem elementów prezentacyjnych pulpitu podprogramu.

# Tworzenie podprogramów w środowisku programistycznym LabVIEW.



lub



- 1a. Po zamarkowaniu fragmentu diagramu wybieramy opcję: [Edit: Create SubVI](#).
- 1b. Na panelu sterującym łączymy złącza (*connectors*) z poszczególnymi wejściami i wyjściami podprogramu (podobnie jak przy tworzeniu diagramu).
2. Wchodzimy do edytora icon ([Edit Icon...](#)).
3. Po zdefiniowaniu własnej ikony zapisujemy podprogram w katalogu [LabVIEWxxx\user.lib](#) nadając mu nową nazwę.
4. Tak przygotowany podprogram będzie dostępny z biblioteki dla innych aplikacji po wybraniu opcji: [Functions: User Libraries](#).

### Zad\_5.1

Napisz program pełniący funkcję kalkulatora czterodziałaniowego.

Wskazówka: można skorzystać z następujących opcji: ...Slide, ...Ring itp...

### Zad\_5.2

Prostym elementem służącym do pomiaru temperatury jest opornik platynowy. Pomiar temperatury polega na pomiarze jego rezystancji a następnie jej zamianie na temperaturę. Przygotuj podprogram pozwalający wykonać tę operację dla rezystora Pt 100.

$R \leq 100, T = 31.06093 + 2.2276 * R + 0.00252 * R^2 - 5.85379e-6 * R^3;$

$R > 100, T = 25.26938 + 2.40754 * R + 6.04325e-4 * R^2 + 1.06776e-6 * R^3.$

### Zad\_5.3

Napisz program który w pierwszej fazie pobiera i wyświetla bieżącą datę i godzinę, następnie po zalogowaniu się użytkownika losuje 6 liczb z zakresu <1, 49>.