



**Innovative Teaching Approaches  
in development of Software Designed  
Instrumentation and its application  
in real-time systems**

# **Podstawy Projektowania Przyrządów Wirtualnych**

## **Wykład 2: Typy danych**

Co-funded by the  
Erasmus+ Programme  
of the European Union



# Powtórka z wykładu 1

## Z jakich elementów składa się VI?

- a) Front panel
- b) Block diagram
- c) Projekt
- d) Icon and Connector Pane

# Powtórka z wykładu 1

**Z jakich elementów składa się VI?**

- a) **Front panel**
- b) **Block diagram**
- c) Projekt
- d) **Icon and Connector Pane**

# Powtórka z wykładu 1

**Jaka linia reprezentuje tablicę 2-wymiarową liczb naturalnych lub całkowitych?**

- a) Pogrubiona linia pomarańczowa
- b) Podwójna linia pomarańczowa
- c) Podwójna linia niebieska
- d) Pogrubiona linia niebieska

**Jakim kolorem są oznaczone łańcuchy znaków?**

- a) Pomarańczowy
- b) Niebieski
- c) Różowy
- d) Zielony

# Powtórka z wykładu 1

**Jaka linia reprezentuje tablicę 2-wymiarową liczb naturalnych lub całkowitych?**

- a) Pogrubiona linia pomarańczowa
- b) Podwójna linia pomarańczowa
- c) **Podwójna linia niebieska**
- d) Pogrubiona linia niebieska

**Jakim kolorem są oznaczone łańcuchy znaków?**

- a) Pomarańczowy
- b) Niebieski
- c) **Różowy**
- d) Zielony

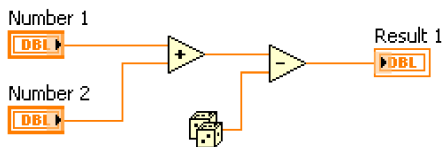
# Powtórka z wykładu 11

**Która funkcja wykona się jako pierwsza?**

- a) Dodawanie
- b) Odejmowanie
- c) Losowanie liczby
- d) Nie wiadomo

**Która funkcja wykona się jako ostatnia?**

- a) Dodawanie
- b) Odejmowanie
- c) Losowanie liczby
- d) Nie wiadomo



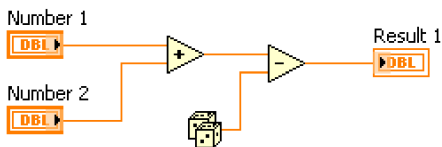
# Powtórka z wykładu 11

**Która funkcja wykona się jako pierwsza?**

- a) Dodawanie
- b) Odejmowanie
- c) Losowanie liczby
- d) **Nie wiadomo**

**Która funkcja wykona się jako ostatnia?**

- a) Dodawanie
- b) **Odejmowanie**
- c) Losowanie liczby
- d) Nie wiadomo



# Podstawowe typy danych w LabVIEW

- Numeryczne
- Logiczne
- Wyliczeniowe
- Łańcuchy znaków

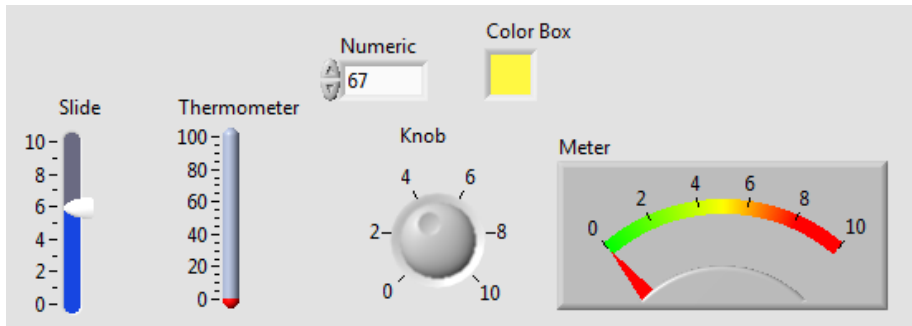
Dodatkowo mamy do dyspozycji:

- Tablice
- Klastry (odpowiednik struktur w C++ czy list w Python-ie)
- Waveform
- Dynamic data

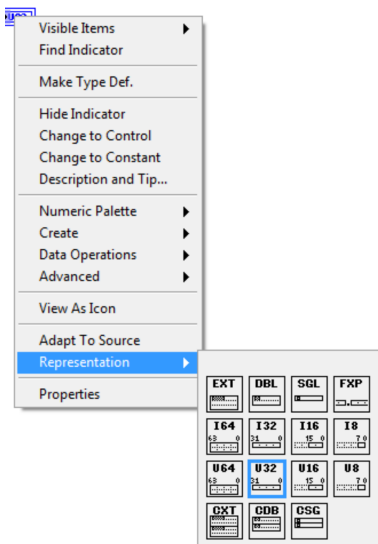


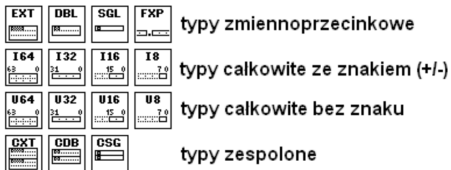
# Dane numeryczne

- Dane numeryczne mogą reprezentować różne typy danych np. liczby całkowite (*integer*) czy zmiennoprzecinkowe (*real*).



# Zmiana reprezentacji liczb





Typ	Liczba bitów	Zakres
Single-precision, floating-point	32	+1.40e-45 do +3.40e+38 oraz -1.40e-45 do -3.40e+38
Double-precision, floating-point	64	+4.94e-324 do +1.79e+308 oraz -1.79e+308 do -4.94e-324
Extended-precision, floating-point	128	+6.48e-4966 do +1.19e+4932 oraz -1.19e+4932 do -6.48e-4966
Byte signed integer	8	-128 do 127
Word signed integer	16	-32768 do 32767
Long signed integer	32	-2 147 483 648 do 2 147 483 647
Quad signed integer	64	-1e19 do 1e19
Byte unsigned integer	8	0 do 255
Word unsigned integer	16	0 do 65 535
Long unsigned integer	32	0 do 4 294 967 295
Quad unsigned integer	64	0 do 2e19

# Shortcut menu

The image displays two parts of a software interface. On the left, a 'Meter' object is shown with a scale from 0 to 10 and a yellow needle pointing to approximately 8. A context menu is open over the meter, listing various actions. On the right, the 'Knob Properties: Meter' dialog box is open, showing configuration options for the meter's appearance and behavior.

**Knob Properties: Meter**

Appearance | Data Type | Scale | Display Format | Text Labels | Document

**Label**

- Visible
- Meter

**Caption**

- Visible

**Enabled State**

- Enabled
- Disabled
- Disabled & grayed

**Size**

Height: 90 | Width: 200

Needle 1 [v] [Add] [Delete]

- Needle color
- Lock at minimum and maximum
- Snap to mouse
- Show digital display(s)
- Show radix
- Show increment/decrement buttons
- Show value tip strip

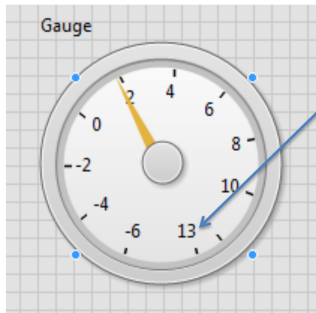
**Shortcut Menu:**

- Visible Items
- Find Terminal
- Change to Indicator
- Make Type Def.
- Description and Tip...
- Create
- Replace
- Data Operations
- Advanced
- Fit Control to Pane
- Scale Object with Pane
- Representation
- Data Entry...
- Display Format...
- Add Needle
- Scale
- Text Labels
- Properties

**Visible Items Sub-menu:**

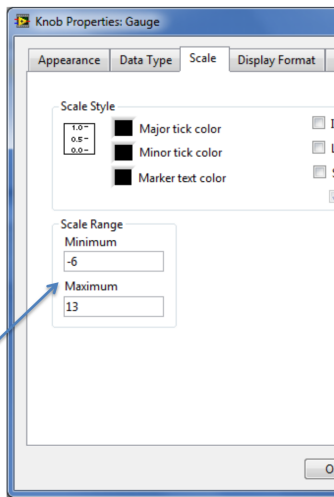
- Label
- Caption
- Unit Label
- Digital Display
- Ramp

# Zmiana zakresu wartości



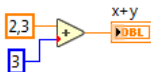
Wpisanie wartości po kliknięciu myszką na koniec skali

Okienko „properties”



# Konwersja liczb

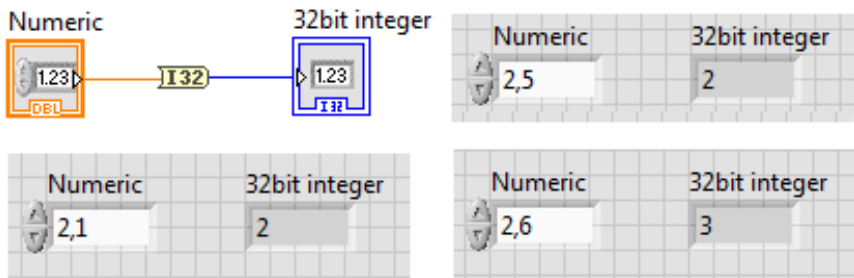
- Jeśli do funkcji numerycznej podłączymy dwie zmienne o różnej reprezentacji to funkcja zwróci nam wartość w **"większym" formacie** np:



- Jeśli podłączymy do funkcji **signed integer** oraz **unsigned integer** to na wyjściu funkcji uzyskamy **unsigned integer**.
- Wszelka konwersja liczby przez funkcję będzie zaznaczona jako ***Coercion Dot***.

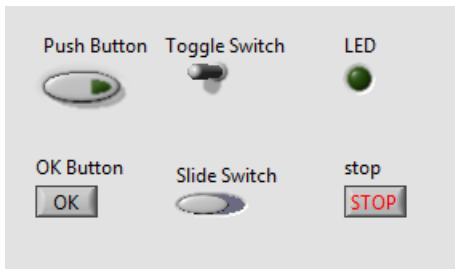
# Konwersja liczb

- W przypadku konwersji liczby zmiennoprzecinkowej na całkowite, VI zwróci najbliższą całkowitą wartość np.  $2.8 \rightarrow 3$ .
- Jeśli wartość jest idealnie pośrodku dwóch całkowitych liczb, VI zwróci parzystą liczbę całkowitą np.  $2.5 \rightarrow 2$ .



# Typ logiczny

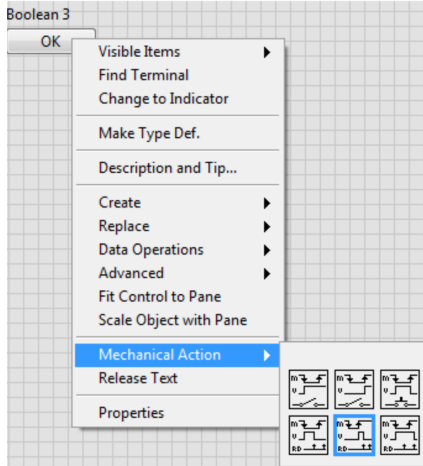
- Typ logiczny (*Boolean*) reprezentuje dane mające tylko dwa stany np. TRUE i FALSE lub ON i OFF.



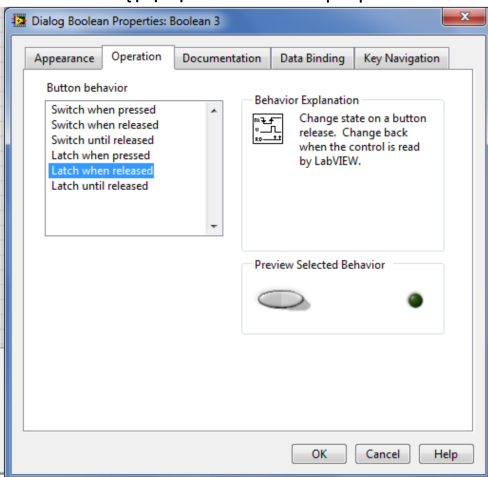


# Akcja mechaniczna przycisków

Dostęp poprzez shortcut menu



Dostęp poprzez okienko properties



# Akcja mechaniczna przycisków

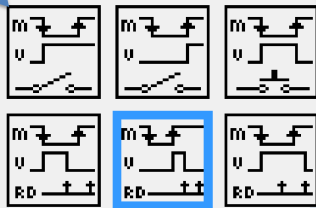
Switch when pressed



Switch until released



Switch when released



Latch when pressed



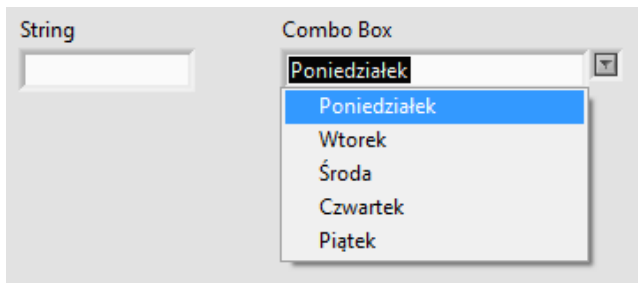
Latch when released



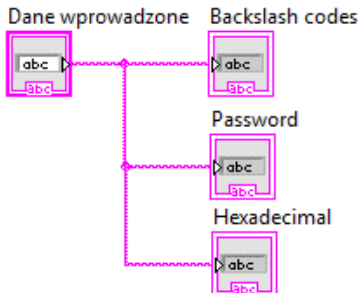
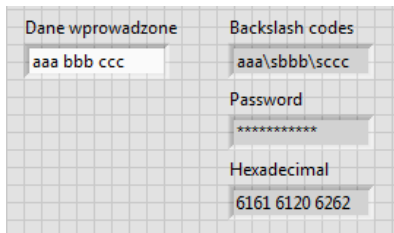
Latch until released

# Łańcuch znaków

- Łańcuch znaków (*String*) to sekwencja ciągów znaków, zakodowanych przy użyciu ASCII.

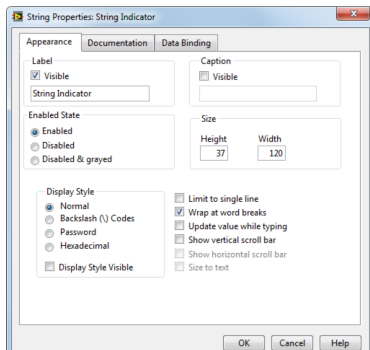


# Style wyświetlania łańcuchów znaków

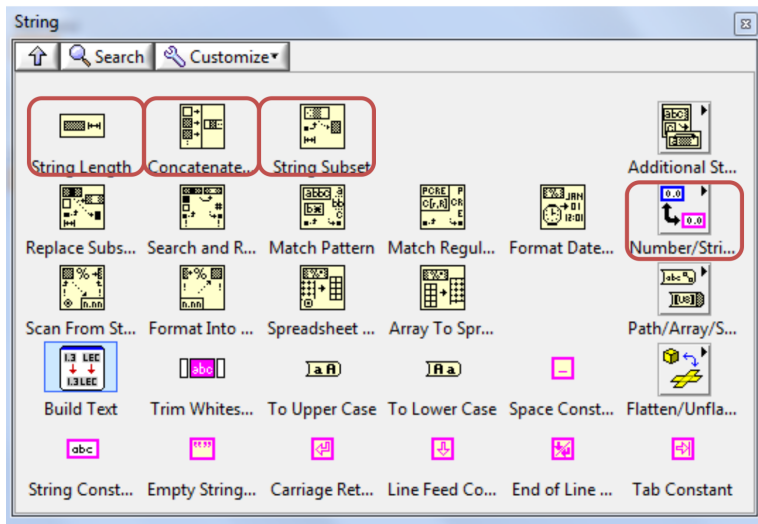


Kontrolki typu string mają 4 tryby wyświetlania:

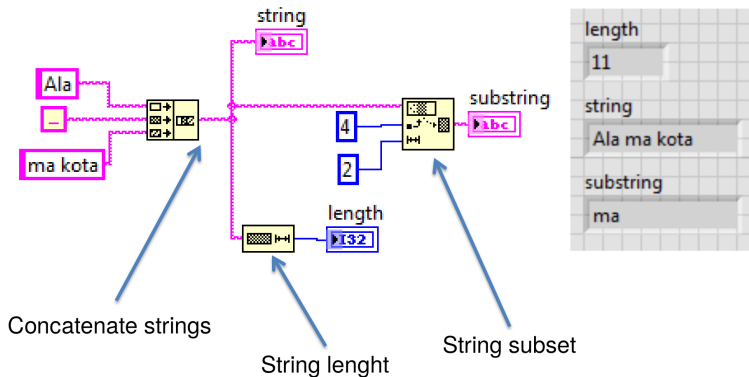
- Normal
- Backslash codes
- Password
- Hexadecimal



# Podstawowe operacje na łańcuchach znaków

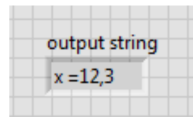
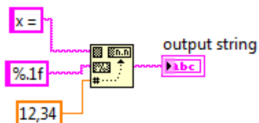


# Podstawowe operacje na łańcuchach znaków

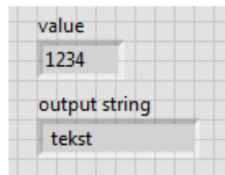
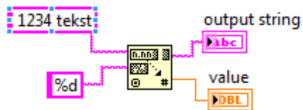


# Konwersja łańcuchów znaków na typy numeryczne

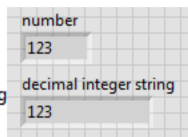
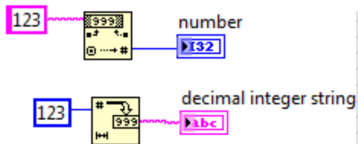
Format value



Scan value

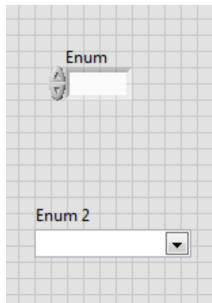


Number to (decimal) string  
(decimal) String to number

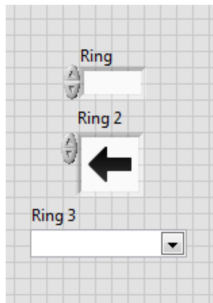


# Typy wyliczeniowe

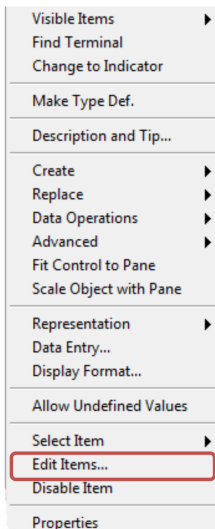
## Enum



## Ring

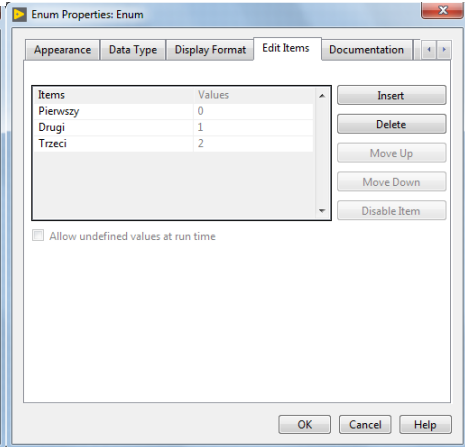
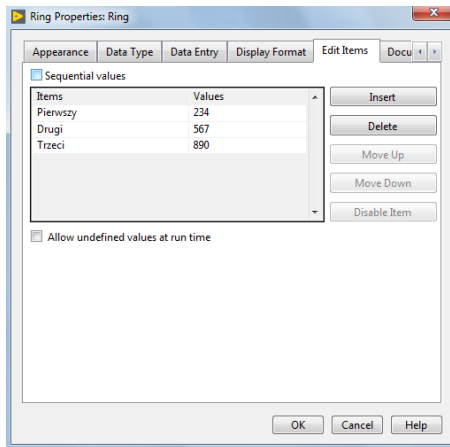


Obie kontrolki przyporządkowują każdej pozycji na liście wartość liczbową - unsigned int 16 (kolor niebieski).



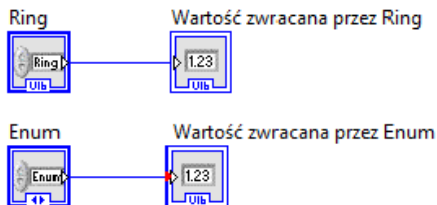


# Typy wyliczeniowe



# Typy wyliczeniowe

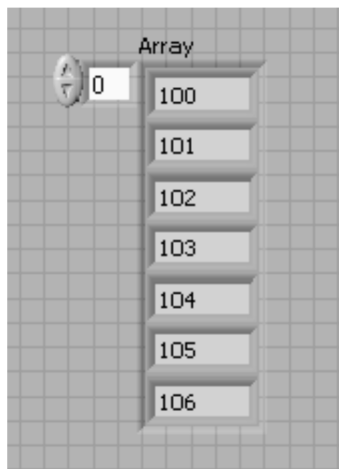
Ring	Wartość zwracana przez Ring
Pierwszy	234
Enum	Wartość zwracana przez Enum
Pierwszy	0



# Tablice

Tablica:

- Jest to zbiór kilku elementów tego samego typu
- Ma 1 lub więcej wymiarów
- Zawiera do  $(2^{31})-1$  elementów na wymiar, jeżeli pozwala na to pamięć.
- Dostęp do elementów odbywa się poprzez ich indeks.
- Nie można zrobić tablicy tablic



**Pierwszy indeks w tablicy jest równy 0!**

# Tablice

## Tablica 1D

0	1	2	3	4	5	6	7	8	9

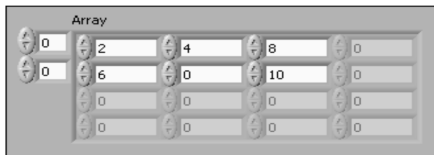
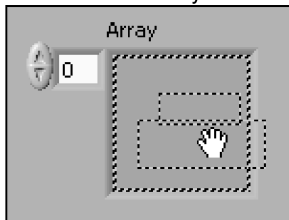
## Tablica 2D

	0	1	2	3	4	5	6
0							
1							
2							
3							
4							

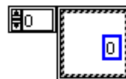
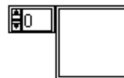
indeksy

# Tablice

### Kontrolka/indykator

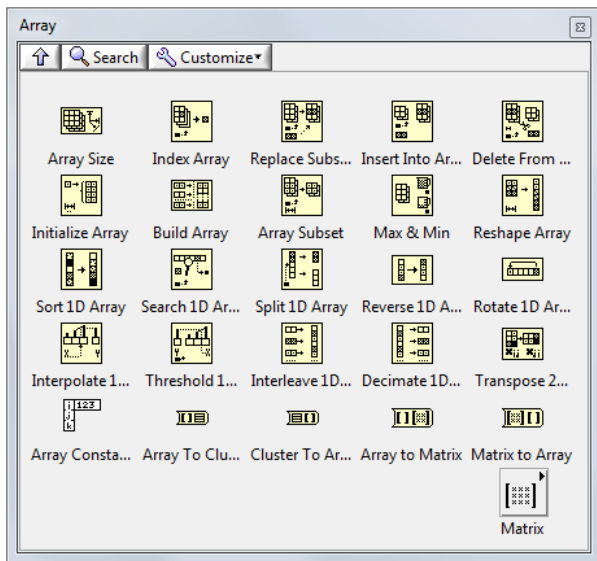


### stała



Część elementów w tablicy może zostać niezainicjalizowana. W takim wypadku są one zamalowane na szaro

# Funkcje związane z tablicami



# Rozmiar tablicy

1-D Array



Array Size



1-D Array Size

6 [32]

1-D Array Size

4

2-D Array



Array Size



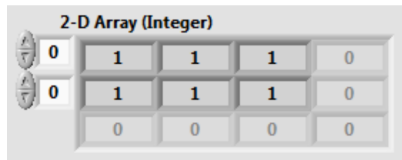
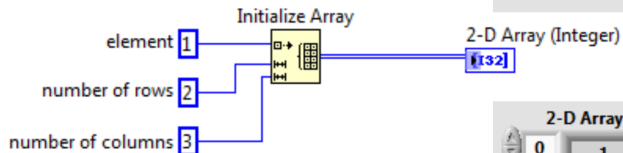
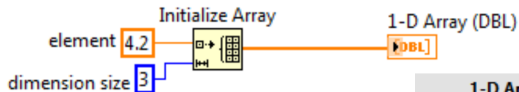
2-D Array Size

3 [32]

2-D Array Size

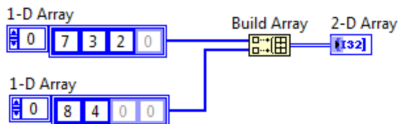
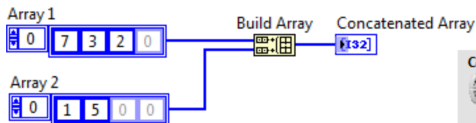
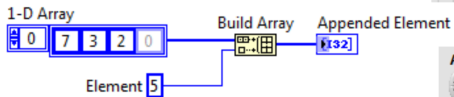
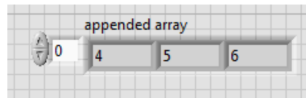
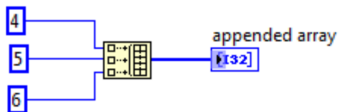
0 2 3 0

# Inicjalizacja tablicy

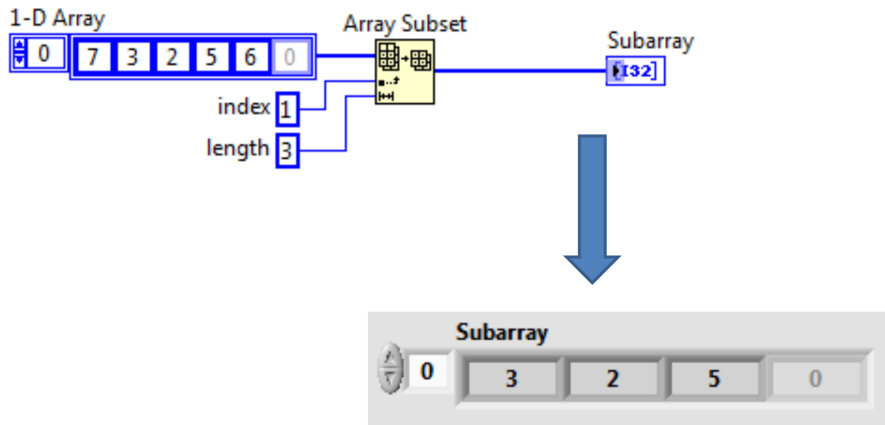




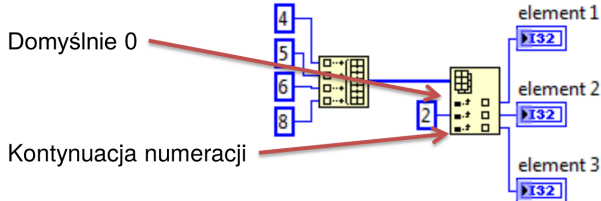
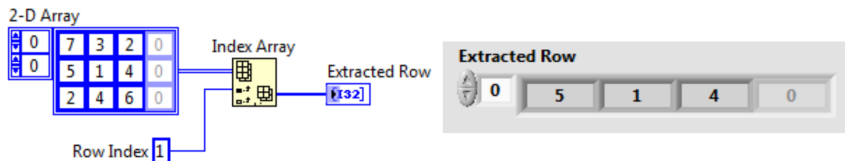
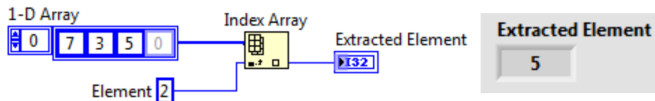
# Tworzenie tablicy



# Wyodrębnienie fragmentu tablicy



# Dostęp do elementów tablicy



element 1

4

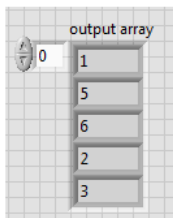
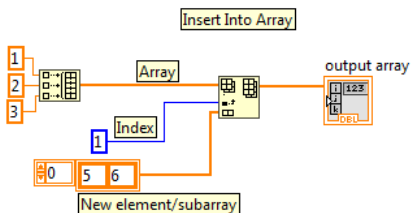
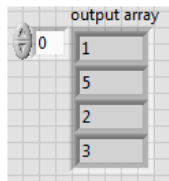
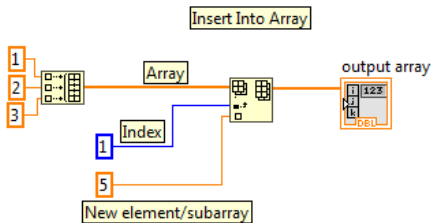
element 2

6

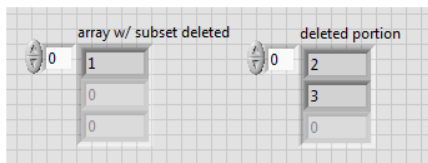
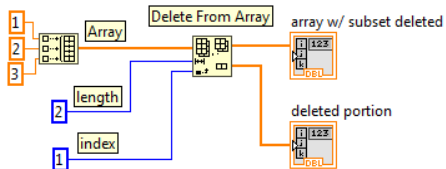
element 3

8

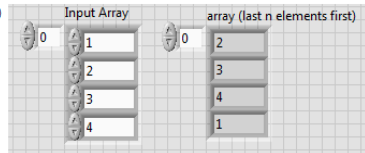
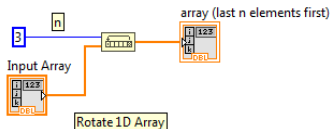
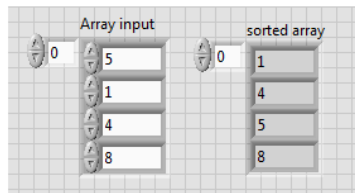
# Dodanie elementu do tablicy



# Usunięcie elementu z tablicy

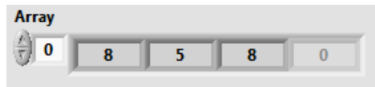
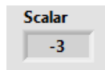
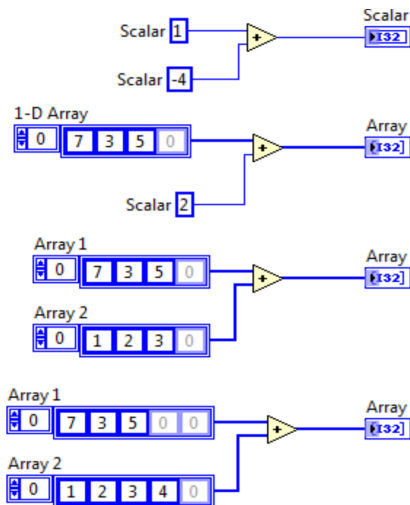


# Przesuwanie elementów tablicy



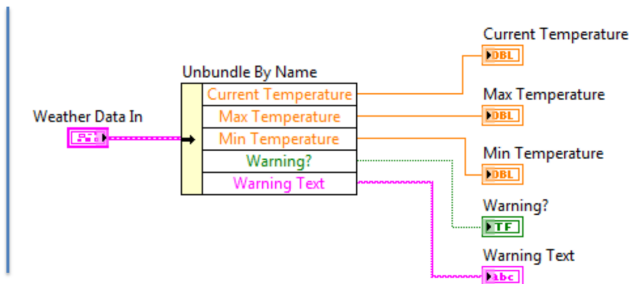
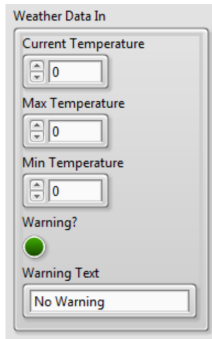
# Polimorfizm funkcji

Funkcje arytmetyczne dostosowują swoje działanie do typu danych wejściowych.



# Klastry

- Klaster grupuje dane różnego typu - jak struktury w C++ czy listy w Python-ie
- **Ograniczenie:** wszystkie elementy w klastrze muszą być albo indykatorami albo kontrolkami.



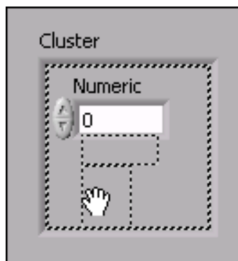


# Dlaczego używamy klastrów?

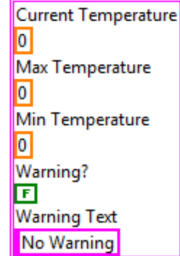
- Grupuje dane w sposób logiczny
- Zapewnia lepszą czytelność kodu - mniej połączeń na diagramie
- Zapewnia opis połączeń
- Ułatwia rozbudowę programu

# Tworzenie klastrów

Tak samo jak w przypadku tablic

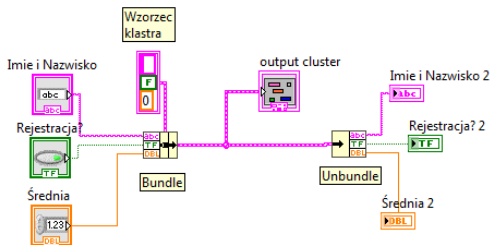


Weather Data



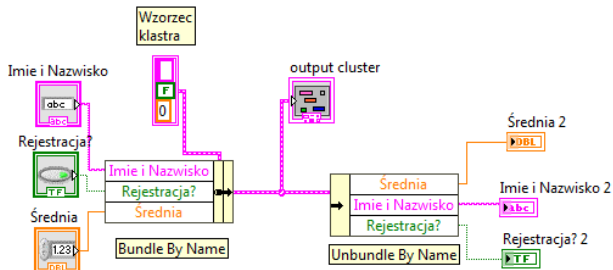
# Tworzenie klastrów

Funkcja *Bundle* tworzy klaster na podstawie kolejności elementów, natomiast *Bundle by name* na podstawie nazw.



Dane wejściowe	Klaster	Dane wyjściowe z klastra
Imie i Nazwisko Jan Kowalski	output cluster Imie i Nazwisko Jan Kowalski	Imie i Nazwisko 2 Jan Kowalski
Średnia 4,5	Rejestracja? <input checked="" type="checkbox"/>	Rejestracja? 2 <input checked="" type="checkbox"/>
Rejestracja? <input checked="" type="checkbox"/>	Średnia 4,5	Średnia 2 4,5

# Tworzenie klastrów

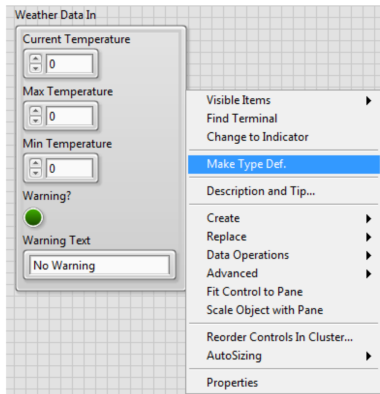


Dane wejściowe	Klaster	Dane wyjściowe z klastra
Imie i Nazwisko Jan Kowalski	output cluster Imie i Nazwisko Jan Kowalski	Imie i Nazwisko 2 Jan Kowalski
Średnia 4,5	Rejestracja? ●	Rejestracja? 2 ●
Rejestracja? ●	Średnia 4,5	Średnia 2 4,5

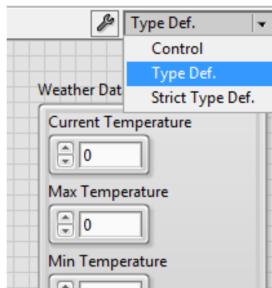
# Definicja typu

- Definicja typu zawiera kopię typu danych (control, indicator, constant).
- Przy zmianie typu danych zmieniają się wszystkie kontrolki/indykatory z nim związane.
- Definicja typu obejmuje typ danych oraz wygląd kontrolki.
- Każda użyta definicja typu (*Type Definition*) może mieć własne: *caption, label, description, tip strip, default value, size, color, style of control/indicator*.
- Każda użyta definicja typu (*Strict Type Definition*) może mieć własne: *caption, label, description, tip strip, default value*.

# Definicja typu



Plik .ctl może być kopią zwykłej kontrolki, ale może być również definicją typu.



# Definicja typu na Block Diagram-ie

- Definicje typu są oznaczone przez trójkąt w rogu ikony
- Pomoc kontekstowa pokazuje zawartość typu danych

Weather Data In



Weather Data In



Context Help

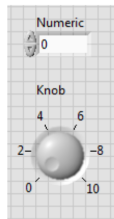
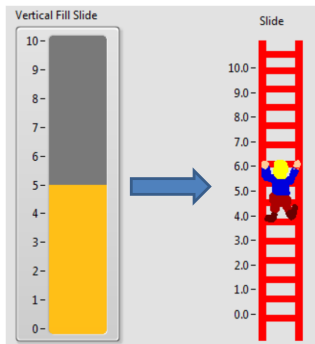
Weather Data In

No description available.

- Weather Data In (typedef 'Typedef Sample.ctl'[non-strict])
  - Weather Data In (cluster of 5 elements)
    - Current Temperature (double [64-bit real (~15 digit precision)])
    - Max Temperature (double [64-bit real (~15 digit precision)])
    - Min Temperature (double [64-bit real (~15 digit precision)])
    - Warning? (boolean (TRUE or FALSE))
    - Warning Text (string)

[Detailed help](#)

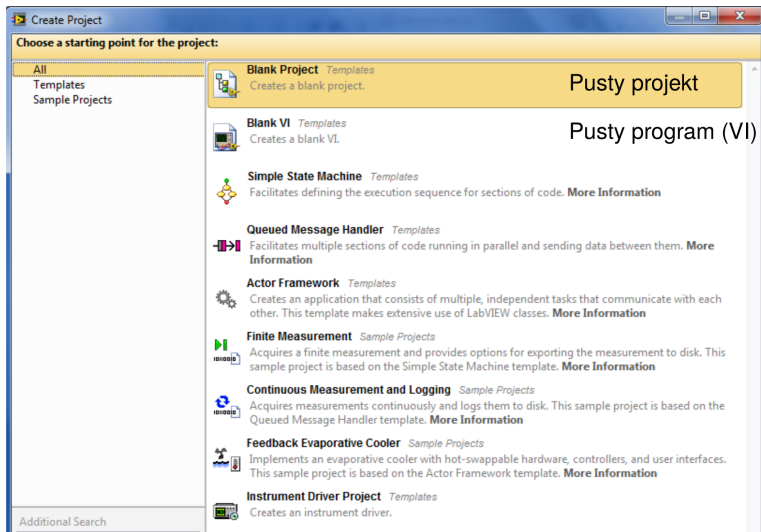
# Dopasowywanie wyglądu kontrolki



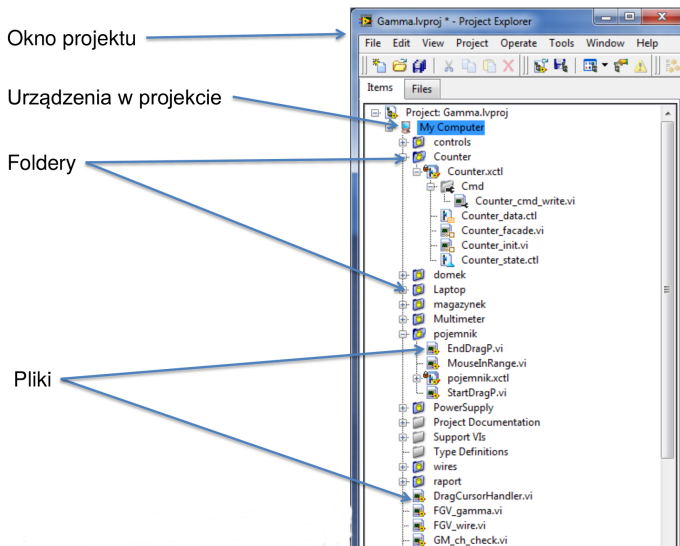


# Tworzenie projektu

Aby stworzyć projekt wybieramy: **File** → **Create Project**



# Project Explorer



# Typy plików w LabVIEW

W LabVIEW mamy pliki o rozszerzeniach:

- LabVIEW project plik.lvproj
- Virtual instrument (VI) plik.vi
- Custom control plik.ctl



**Dziękuję za uwagę!**

Wsparcie Komisji Europejskiej dla produkcji tej publikacji nie stanowi poparcia dla treści, które odzwierciedlają jedynie poglądy autorów, a Komisja nie może zostać pociągnięta do odpowiedzialności za jakiegokolwiek wykorzystanie informacji w niej zawartych.

Wykład został opracowany w oparciu o materiały: "LabVIEW Core 1 Course Manual", "LabVIEW Core 2 Course Manual", pierwotną wersję wykładu: mgr. inż. Marcina Biedy oraz przykładowe egzaminy CLAD opublikowane na stronie [www.ni.com](http://www.ni.com).