



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

Podstawy Projektowania Przyrządów Wirtualnych

Wykład 5: Zmienne lokalne, globalne oraz funkcjonalne.

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



Powtórka z wykładu 4

1. Uzupełnij zdanie: W przeciwieństwie do graph, który wyświetla dane, chart aktualizuje dane cyklicznie i poprzednio wyświetlone.

- a) Zachowane w pamięci, nadpisuje
- b) Poprzez nadpisanie poprzednich danych, zachowuje w historii
- c) Poprzez dodanie nowych danych do poprzednich, nadpisuje
- d) Żadne z powyższych

Powtórka z wykładu 4

1. Uzupełnij zdanie: W przeciwieństwie do graph, który wyświetla dane, chart aktualizuje dane cyklicznie i poprzednio wyświetlone.

- a) Zachowane w pamięci, nadpisuje
- b) **Poprzez nadpisanie poprzednich danych, zachowuje w historii**
- c) Poprzez dodanie nowych danych do poprzednich, nadpisuje
- d) Żadne z powyższych

Powtórka z wykładu 4

2. Który z poniższych umożliwia Ci narysowanie zestawu punktów nierównomiernie rozłożonych?

- a) Waveform Graph
- b) Waveform Chart
- c) XY Graph
- d) Zarówno a i b

Powtórka z wykładu 4

2. Który z poniższych umożliwia Ci narysowanie zestawu punktów nierównomiernie rozłożonych?

- a) Waveform Graph
- b) Waveform Chart
- c) **XY Graph**
- d) Zarówno a i b

Powtórka z wykładu 4

3. Która z poniższych opcji umożliwia Ci narysowanie wielu wykresów na Waveform Graph?

- a) Utworzenie klastrów z dwóch tablic jednowymiarowych X i Y. Następnie zbudowanie tablicy z tych klastrów i przekazanie jej do Waveform Graph.
- b) Zbudowanie n-wymiarowej tablicy, w której poszczególne serie danych są umieszczone w osobnych kolumnach lub wierszach.
- c) Zbudowanie klastrów z punktów pomiarowych. Następnie stworzenie klastra z klastrów z seriami pomiarowymi.
- d) Zarówno opcja B i C

Powtórka z wykładu 4

3. Która z poniższych opcji umożliwia Ci narysowanie wielu wykresów na Waveform Graph?

- a) Utworzenie klastrów z dwóch tablic jednowymiarowych X i Y. Następnie zbudowanie tablicy z tych klastrów i przekazanie jej do Waveform Graph.
- b) Zbudowanie n-wymiarowej tablicy, w której poszczególne serie danych są umieszczone w osobnych kolumnach lub wierszach.
- c) Zbudowanie klastrów z punktów pomiarowych. Następnie stworzenie klastra z klastrów z seriami pomiarowymi.
- d) **Zarówno opcja B i C**

Powtórka z wykładu 4

4. Która z poniższych metod umożliwia stworzenie dwóch wykresów na XY Graph?

- a) Stworzenie jednego klastra z dwóch tablic z wartościami X i dwóch tablic z wartościami Y według wzoru X,Y,X,Y.
- b) Stworzenie oddzielnych klastrów z tablic X i Y dla każdej serii danych a następnie stworzenie tablicy z tych klastrów.
- c) Stworzenie oddzielnych klastrów z tablic X i Y dla każdej serii danych a następnie stworzenie klastra z tych klastrów.
- d) Stworzenie tablicy z tablic wartościami X i Y według wzoru X,Y,X,Y.

Powtórka z wykładu 4

4. Która z poniższych metod umożliwia stworzenie dwóch wykresów na XY Graph?

- a) Stworzenie jednego klastra z dwóch tablic z wartościami X i dwóch tablic z wartościami Y według wzoru X,Y,X,Y.
- b) **Stworzenie oddzielnych klastrów z tablic X i Y dla każdej serii danych a następnie stworzenie tablicy z tych klastrów.**
- c) Stworzenie oddzielnych klastrów z tablic X i Y dla każdej serii danych a następnie stworzenie klastra z tych klastrów.
- d) Stworzenie tablicy z tablic wartościami X i Y według wzoru X,Y,X,Y.

Powtórka z wykładu 4

5. Który rodzaj wyświetlania danych na wykresie zapewnia dodawanie punktów z lewej do prawej a następnie wyczyszczenie wykresu w celu dodania najnowszych danych?

- a) Strip Chart
- b) Scope Chart
- c) Sweep Chart
- d) Step Chart

Powtórka z wykładu 4

5. Który rodzaj wyświetlania danych na wykresie zapewnia dodawanie punktów z lewej do prawej a następnie wyczyszczenie wykresu w celu dodania najnowszych danych?

- a) Strip Chart
- b) **Scope Chart**
- c) Sweep Chart
- d) Step Chart

Powtórka z wykładu 4

**6. Który rodzaj wyświetlania danych powinien być użyty aby oddzielić poprzednie i obecne dane poprzez pionową linię?
(Analogicznie do zapisów z EKG)**

- a) Strip Chart
- b) Scope Chart
- c) Sweep Chart
- d) Step Chart

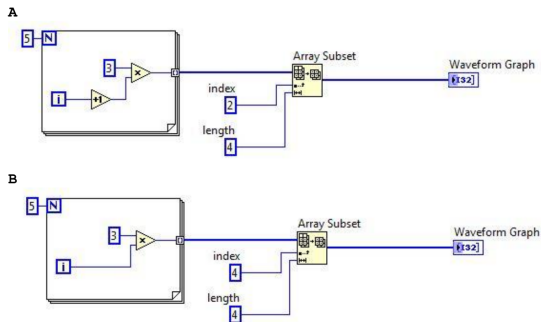
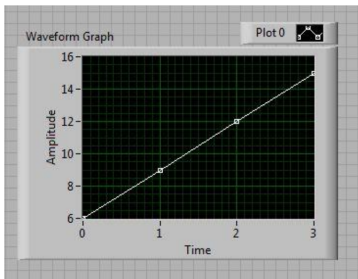
Powtórka z wykładu 4

**6. Który rodzaj wyświetlania danych powinien być użyty aby oddzielić poprzednie i obecne dane poprzez pionową linię?
(Analogicznie do zapisów z EKG)**

- a) Strip Chart
- b) Scope Chart
- c) **Sweep Chart**
- d) Step Chart

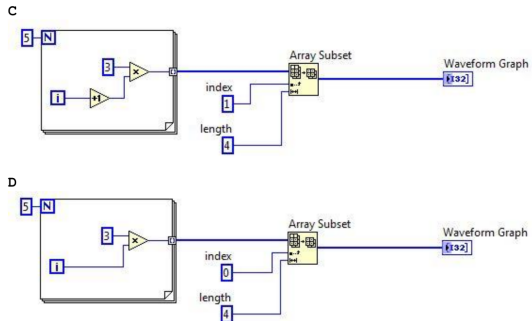
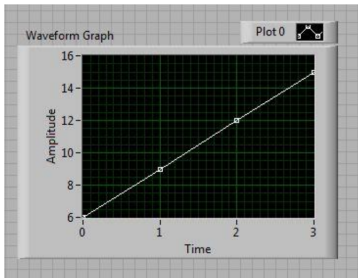
Powtórka z wykładu 4

7. Który kod programu spowodować wyświetlenie poniższego wykresu?



Powtórka z wykładu 4

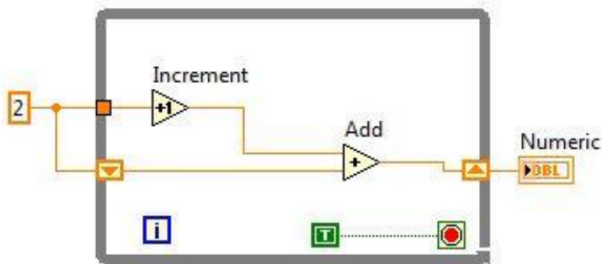
7. Który kod programu spowodować wyświetlenie poniższego wykresu?



Powtórka z wykładu 4

8. Jaka wartość zostanie wyświetlona w zmiennej Numeric?

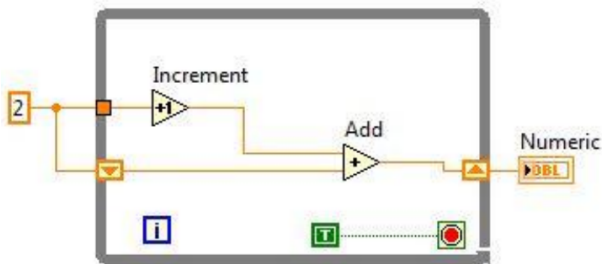
- a) 0
- b) 4
- c) 5
- d) Pętla będzie wykonywała się w nieskończoność



Powtórka z wykładu 4

8. Jaka wartość zostanie wyświetlona w zmiennej Numeric?

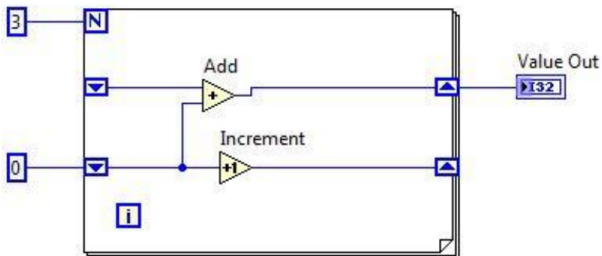
- a) 0
- b) 4
- c) 5
- d) Pętla będzie wykonywała się w nieskończoność



Powtórka z wykładu 4

9. VI został otwarty i uruchomiony dwa razy bez zamykania i dokonywania zmian. Jaka wartość będzie wyświetlona w zmiennej Value Out po drugim uruchomieniu?

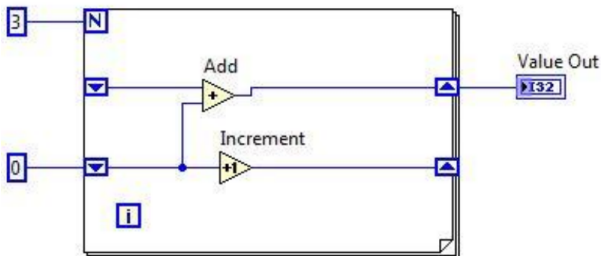
- a) 3
- b) 4
- c) 5
- d) 6



Powtórka z wykładu 4

9. VI został otwarty i uruchomiony dwa razy bez zamykania i dokonywania zmian. Jaka wartość będzie wyświetlona w zmiennej Value Out po drugim uruchomieniu?

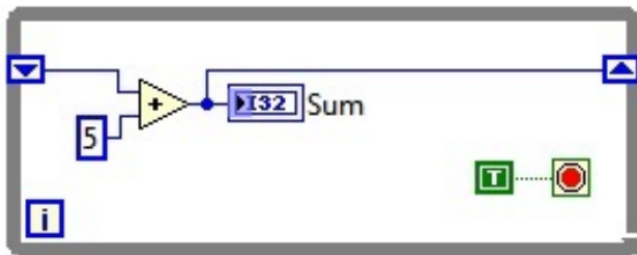
- a) 3
- b) 4
- c) 5
- d) 6



Powtórka z wykładu 4

10. Poniższy subVI został załadowany do pamięci i uruchomiony. Po pewnym czasie ten sam subVI wciąż znajduje się w pamięci i został uruchomiony drugi raz. Jaka wartość zostanie przekazana do zmiennej Sum po uruchomieniu subVI drugi raz?

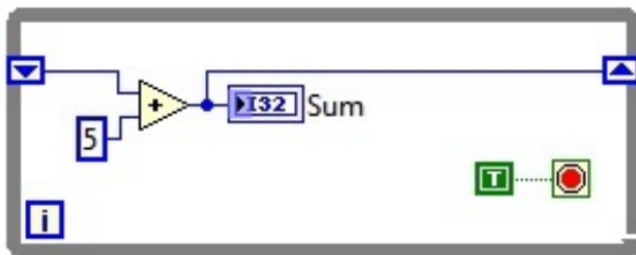
- a) 5
- b) 10
- c) Nie wiadomo
- d) 0



Powtórka z wykładu 4

10. Poniższy subVI został załadowany do pamięci i uruchomiony. Po pewnym czasie ten sam subVI wciąż znajduje się w pamięci i został uruchomiony drugi raz. Jaka wartość zostanie przekazana do zmiennej Sum po uruchomieniu subVI drugi raz?

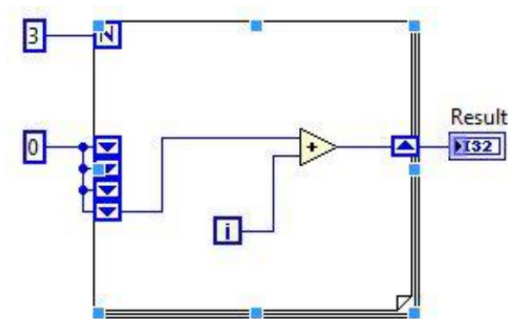
- a) 5
- b) 10
- c) Nie wiadomo
- d) 0



Powtórka z wykładu 4

11. Jaka wartość zostanie wyświetlona w zmiennej Result po zakończeniu działania VI?

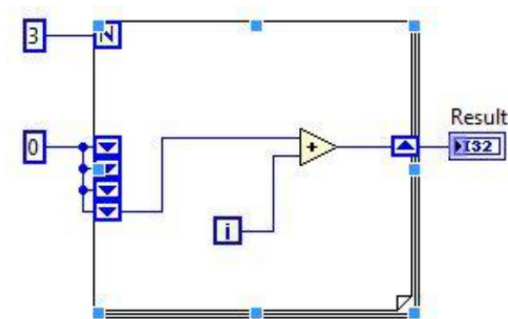
- a) 3
- b) 2
- c) 1
- d) 0



Powtórka z wykładu 4

11. Jaka wartość zostanie wyświetlona w zmiennej Result po zakończeniu działania VI?

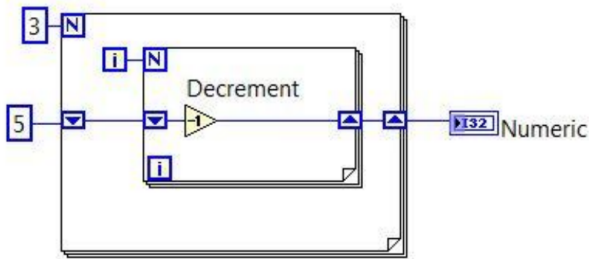
- a) 3
- b) 2
- c) 1
- d) 0



Powtórka z wykładu 4

12. Jaka wartość zostanie wyświetlona w zmiennej Numeric?

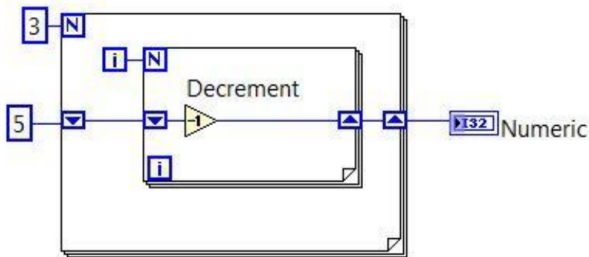
- a) 0
- b) 1
- c) 2
- d) 4



Powtórka z wykładu 4

12. Jaka wartość zostanie wyświetlona w zmiennej Numeric?

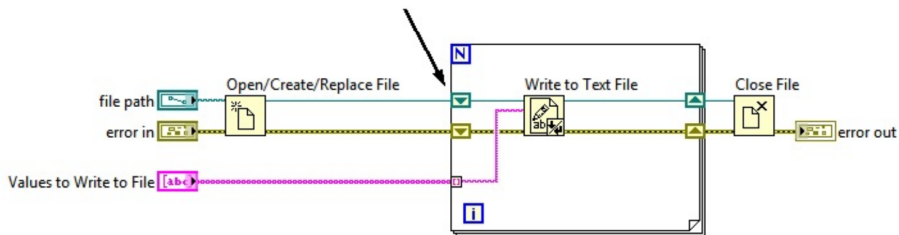
- a) 0
- b) 1
- c) 2
- d) 4



Powtórka z wykładu 4

13. Dlaczego użyto rejestru przesuwanego do numeru referencyjnego (refnum) w pętli For?

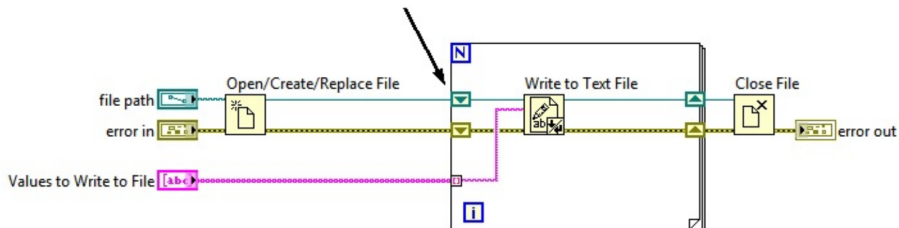
- a) Jeśli Values to Write File jest pusty, pętla For wykona się zero razy lecz numer referencyjny zostanie przekazany do wyjścia rejestru przesuwanego.
- b) Funkcje Write to File modyfikuje wartości numeru referencyjnego za każdym razem wykonania pętli For.
- c) To jest jedyny sposób aby zapobiec utworzeniu tablicy na wyjściowym terminalu.
- d) Nie ma powodu. Zwykły tunel zapewniłby taką samą funkcjonalność.



Powtórka z wykładu 4

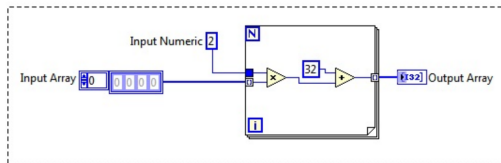
13. Dlaczego użyto rejestru przesuwanego do numeru referencyjnego (refnum) w pętli For?

- a) Jeśli Values to Write File jest pusty, pętla For wykona się zero razy **lecz numer referencyjny zostanie przekazany do wyjścia rejestru przesuwanego.**
- b) Funkcje Write to File modyfikuje wartości numeru referencyjnego za każdym razem wykonania pętli For.
- c) To jest jedyny sposób aby zapobiec utworzeniu tablicy na wyjściowym terminalu.
- d) Nie ma powodu. Zwykły tunel zapewniłby taką samą funkcjonalność.



Powtórka z wykładu 4

14. Po wybraniu opcji **Create SubVI** z menu Edycji, jaki rezultat otrzymasz?



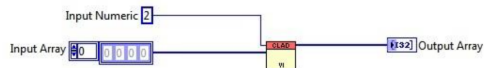
A



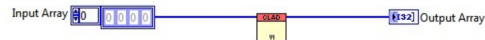
B



C



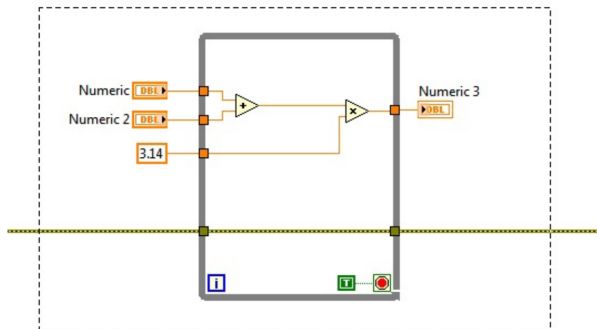
D



Powtórka z wykładu 4

15. SubVI został stworzony z zaznaczonego obszaru kodu. Ile terminali posiada subVI?

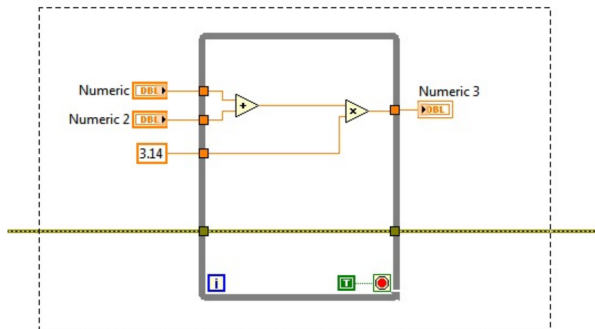
- a) 3
- b) 4
- c) 5
- d) 6



Powtórka z wykładu 4

15. SubVI został stworzony z zaznaczonego obszaru kodu. Ile terminali posiada subVI?

- a) 3
- b) 4
- c) 5
- d) 6



Powtórka z wykładu 4

16. Jeśli nazwa danej wejściowej jest oznaczona pogrubioną czcionką w Show Context Help, to oznacza to że:

- a) Dana wejściowa musi być skalarem.
- b) Dana wejściowa jest zalecana ale nie wymagana.
- c) Dana wejściowa jest wymagana.
- d) Strzałka run zostanie przekreślona jeśli dana wejściowa nie będzie podłączona.

Powtórka z wykładu 4

16. Jeśli nazwa danej wejściowej jest oznaczona pogrubioną czcionką w Show Context Help, to oznacza to że:

- a) Dana wejściowa musi być skalarem.
- b) Dana wejściowa jest zalecana ale nie wymagana.
- c) **Dana wejściowa jest wymagana.**
- d) **Strzałka run zostanie przekreślona jeśli dana wejściowa nie będzie podłączona.**

Powtórka z wykładu 4

17. Jak udokumentować VI aby opis pojawił się w Show Context Help?

- a) Używając okno VI Properties → Documentation
- b) Tworząc komentarze na front panelu
- c) Edytując plik LabVIEW help

Powtórka z wykładu 4

17. Jak udokumentować VI aby opis pojawił się w Show Context Help?

- a) **Używając okno VI Properties → Documentation**
- b) Tworząc komentarze na front panelu
- c) Edytując plik LabVIEW help

Powtórka z wykładu 4

18. Które z poniższych zdań dotyczących obsługi plików binarnych lub TDMS nie jest prawdziwe?

- a) Pliki binarne akceptują wszystkie typy danych dostępnych w LabVIEW.
- b) TDMS akceptuje wszystkie typy danych dostępne w LabVIEW.
- c) TDMS jest szybsze w zapisanie danych niż pliki binarne.
- d) Pliki binarne mogą być obsługiwane pod wszystkimi wersjami LabVIEW.

Powtórka z wykładu 4

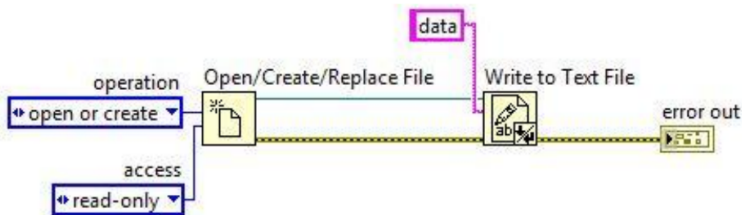
18. Które z poniższych zdań dotyczących obsługi plików binarnych lub TDMS nie jest prawdziwe?

- a) Pliki binarne akceptują wszystkie typy danych dostępnych w LabVIEW.
- b) TDMS akceptuje wszystkie typy danych dostępne w LabVIEW.**
- c) TDMS jest szybsze w zapisanie danych niż pliki binarne.
- d) Pliki binarne mogą być obsługiwane pod wszystkimi wersjami LabVIEW.

Powtórka z wykładu 4

19. Dlaczego pojawia się błąd podczas wykonywania poniższego kodu?

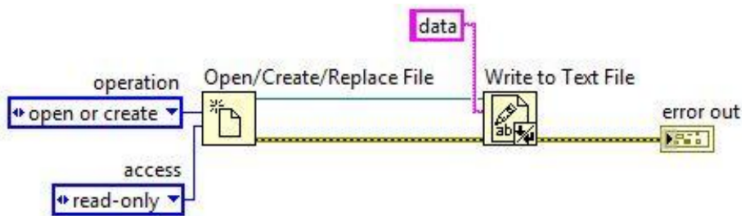
- a) Ścieżka do pliku nie jest podłączona do wejścia funkcji Open/Create/Replace.
- b) Plik jest otwierany z nieprawidłowymi uprawnieniami.
- c) Referencja do pliku nie jest zamykana.
- d) Format pliku jest nieprawidłowy.



Powtórka z wykładu 4

19. Dlaczego pojawia się błąd podczas wykonywania poniższego kodu?

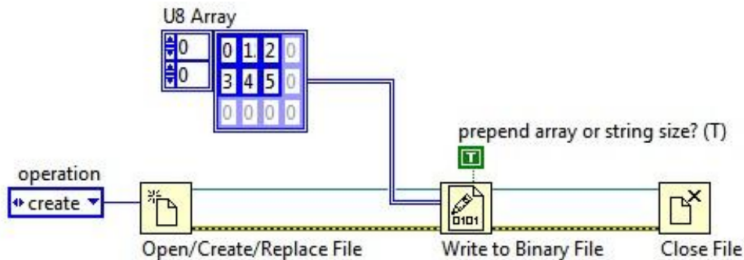
- a) Ścieżka do pliku nie jest podłączona do wejścia funkcji Open/Create/Replace.
- b) **Plik jest otwierany z nieprawidłowymi uprawnieniami.**
- c) Referencja do pliku nie jest zamykana.
- d) Format pliku jest nieprawidłowy.



Powtórka z wykładu 4

20. Jak dużo bajtów danych LabVIEW zapisze do pliku po wykonaniu poniższego kodu?

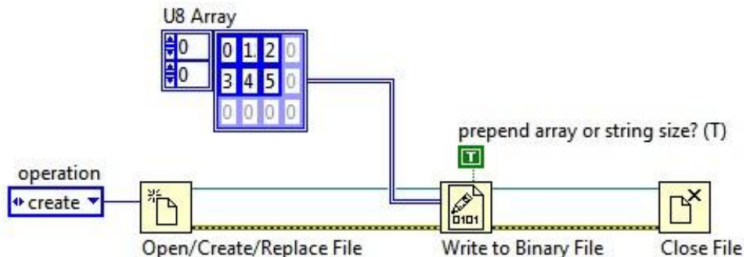
- a) 5 bajtów
- b) 6 bajtów
- c) 12 bajtów
- d) 14 bajtów



Powtórka z wykładu 4

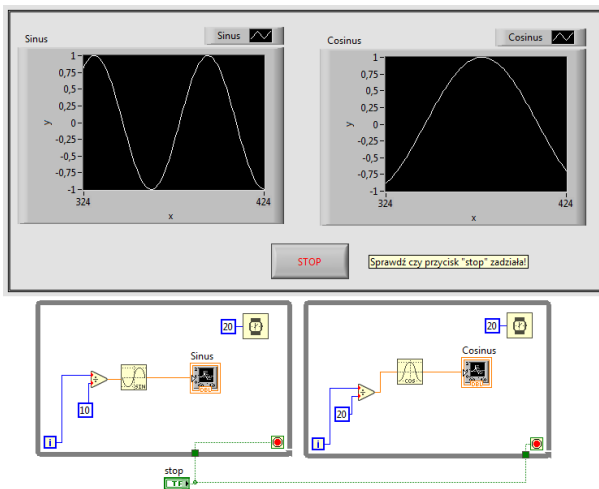
20. Jak dużo bajtów danych LabVIEW zapisze do pliku po wykonaniu poniższego kodu?

- a) 5 bajtów
- b) 6 bajtów
- c) 12 bajtów
- d) 14 bajtów = 6 bajtów danych + 4 bajty na rozmiar(ilość wierszy) + 4 bajty na rozmiar (ilość kolumn)



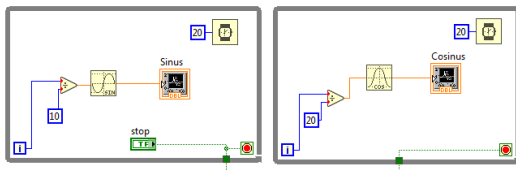
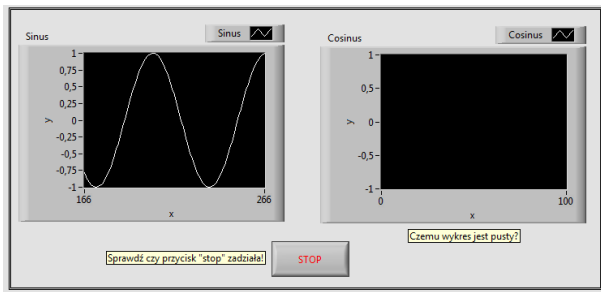
Komunikacja między dwiema pętlami

W jaki sposób zatrzymać dwie pętle Loop w tym samym czasie?
(Sposób niepoprawny)



Komunikacja między dwiema pętlami

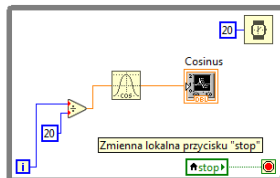
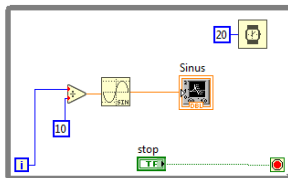
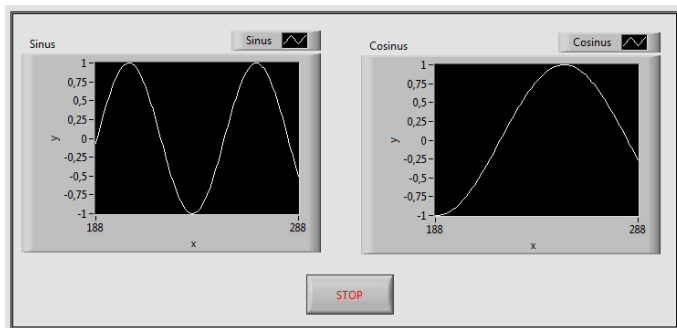
W jaki sposób zatrzymać dwie pętle Loop w tym samym czasie?
(Sposób niepoprawny)



Zmienne lokalne

- Zmienne lokalne są stosowane do przekazywania danych w ramach jednego VI bez konieczności używania bezpośrednich połączeń. Dane są zapisywane w kontrolkach/indykatorach na front panel-u.
- Standardowo z kontrolki możemy tylko odczytywać dane a do indykatorów zapisywać. Zmienne lokalne umożliwiają zapis i odczyt do/z kontrolki jak i indykatora.
- Wykorzystanie: komunikacja między równoległymi pętlami, zapis danych w zależności od sytuacji (np. struktura case) oraz inicjalizacja kontrolki/indykatorów.
- Zmienne lokalne mogą być używane jedynie w ramach jednego subVI.
- Ich działanie jest wolniejsze niż terminali.
- Kontrolki logiczne muszą być typu switch.

Komunikacja między dwiema pętlami - rozwiązanie

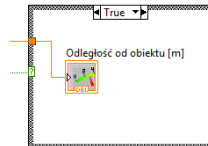
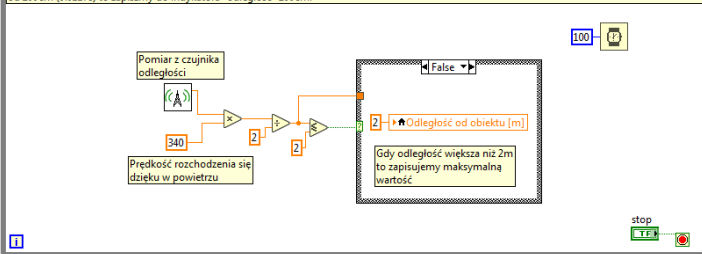


Zmiana wartości w różnych sytuacjach

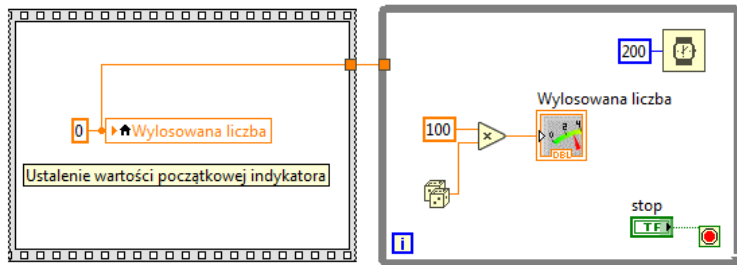
Problem: Wyobraźmy sobie, że chcemy zmierzyć odległość od przeszkody z wykorzystaniem ultradźwiękowego czujnika odległości np. HC-SR04., który emituje falę dźwiękową a następnie ją nasłuchuje na wejściu ECHO. Czas pomiędzy emisją fali dźwiękowej a odebraniem fali dźwiękowej na wejściu ECHO jest proporcjonalny do odległości od przeszkody: $d = (v \cdot t) / 2$ gdzie: d - odległość od przeszkody, t - czas pomiędzy emisją a odebraniem fali dźwiękowej, v - prędkość rozchodzenia się dźwięku w powietrzu (340m/s).

Wspomniany czujnik mierzy odległość w zakresie 2-200cm. Gdy obiekt jest oddalony o odległość większą niż 200cm to czujnik zwraca losowo dużą liczbę.

Rozwiązanie: Napiszmy program, który zasymuluje odczyt czasu z czujnika za pomocą funkcji random. Jeśli odczytana wielkość będzie większa od 200cm (0.0117s) to zapiszmy do indykatora "odległość" 200cm.



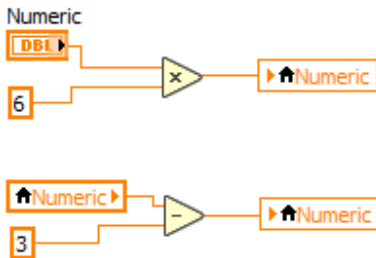
Inicjalizacja kontroltek



Pętla while "czeka" aż przyjdzie wartość na wejściowy tunel od stałej 0.
W taki sposób wymuszono aby najpierw wykonała się struktura Flat Sequence
a potem pętla while

Do czego nie używać zmiennych lokalnych?

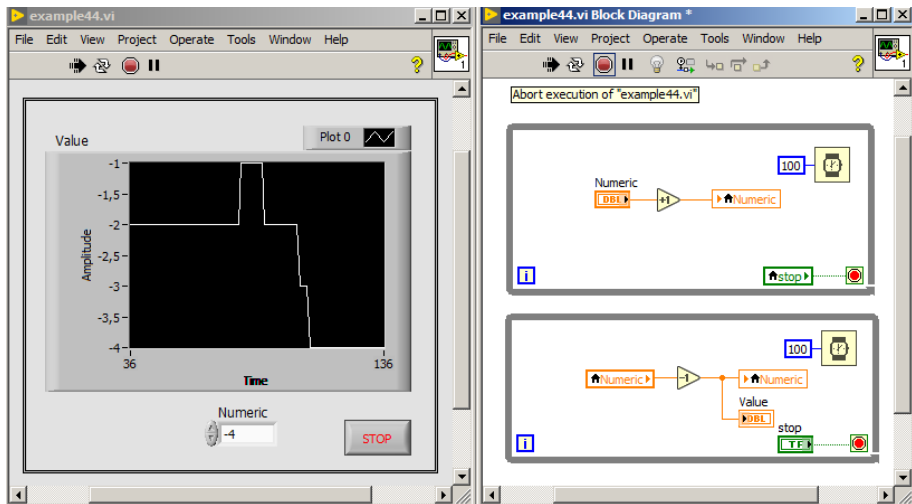
- Race condition - zjawisko polegające na braku możliwości określenia końcowego wyniku z powodu braku kontroli nad przepływem danych.



Możliwe wyniki:

1. $\text{Numeric} = \text{Numeric} * 6 - 3$
2. $\text{Numeric} = (\text{Numeric} - 3) * 6$
3. $\text{Numeric} = \text{Numeric} * 6$
4. $\text{Numeric} = \text{Numeric} - 3$

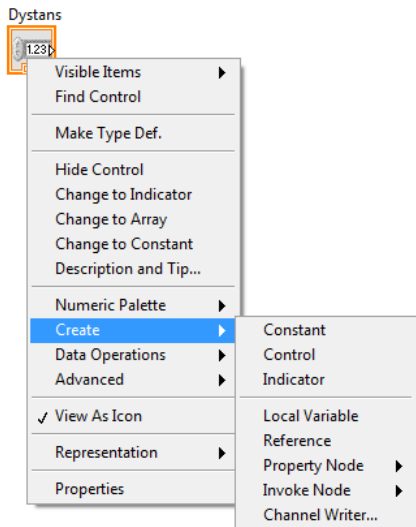
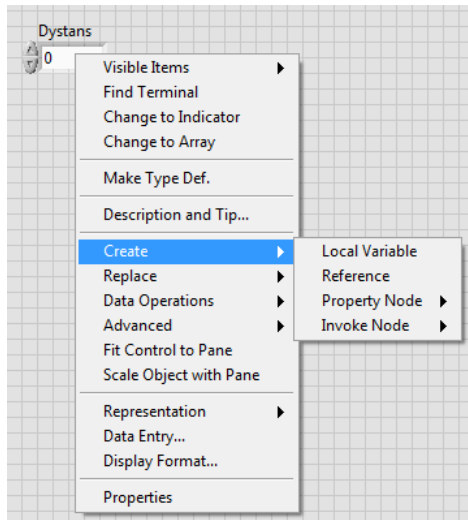
Race Condition



Race Condition

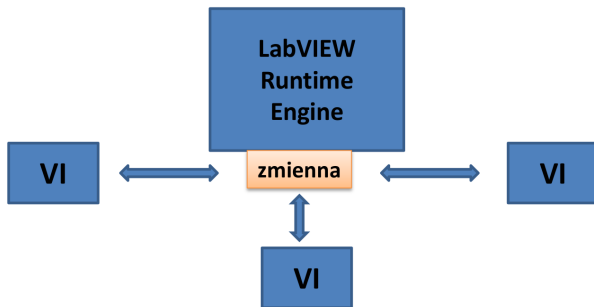
- Race Condition trudno zidentyfikować w kodzie.
- Bardzo często przy testowaniu race condition otrzymujemy ten sam wyniki wiele razy. Natomiast wciąż istnieje prawdopodobieństwo, że za n-tą iteracją uzyskamy inny wynik.
- Aby uniknąć race condition:
 - Kontroluj dzielone kontrolki/indykatory - stosuj jedną zmienną, która będzie zapisywać dane oraz wiele zmiennych, które będą odczytywać dane.
 - Stwórz tak program aby możliwie jak najwięcej czynności wykonywało się sekwencyjnie (kontroluj przepływ danych)
 - Zredukuj do minimum użycie zmiennych lokalnych.

Tworzenie zmiennej lokalnej



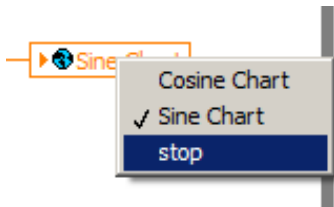
Zmienne globalne

- Zmienne globalne są stosowane do przekazywania danych między różnymi VI. Dane są przechowywane w specjalnym repozytorium.
- Działa w architekturze serwer-klient.
- Nie wymaga tworzenia projektu.

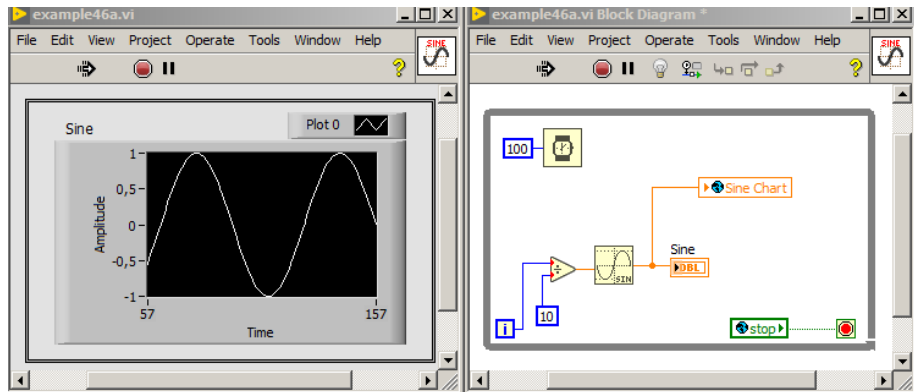


Zmienne globalne

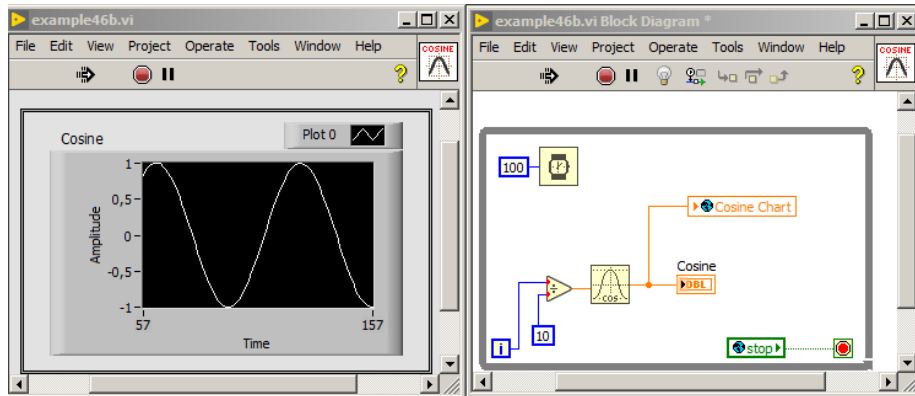
- Aby stworzyć zmienną globalną należy wybrać Functions palette → Structures → Global Variable.
- Na powstały terminal należy kliknąć co spowoduje otworenie front panelu zmiennej globalnej.
- Obiekty do zmiennej globalnej dodaje się z controls palette umieszczając je na front panelu zmiennej globalnej.
- Dana zmienna może składać się z jednego obiektu lub wielu.



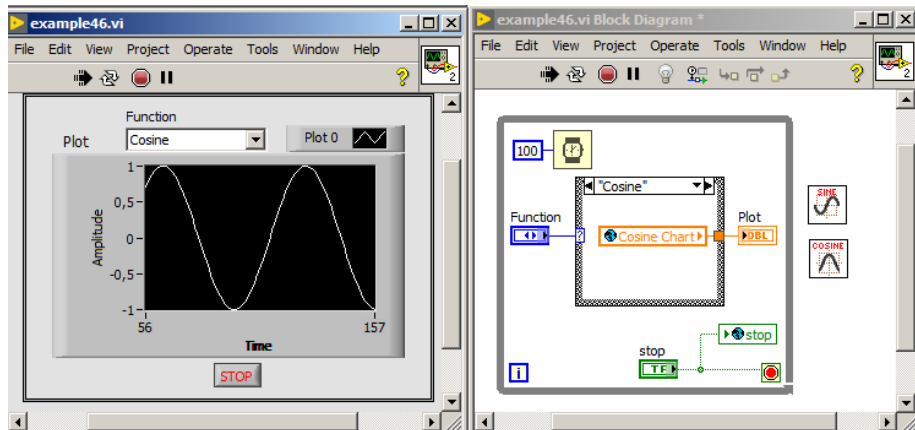
Przykład użycia zmiennej globalnej



Przykład użycia zmiennej globalnej

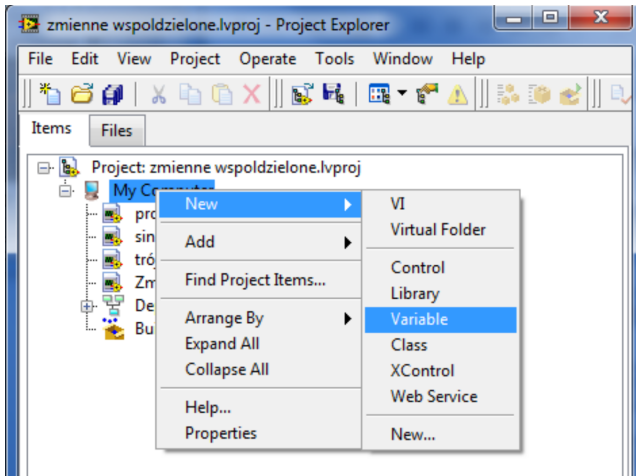


Przykład użycia zmiennej globalnej

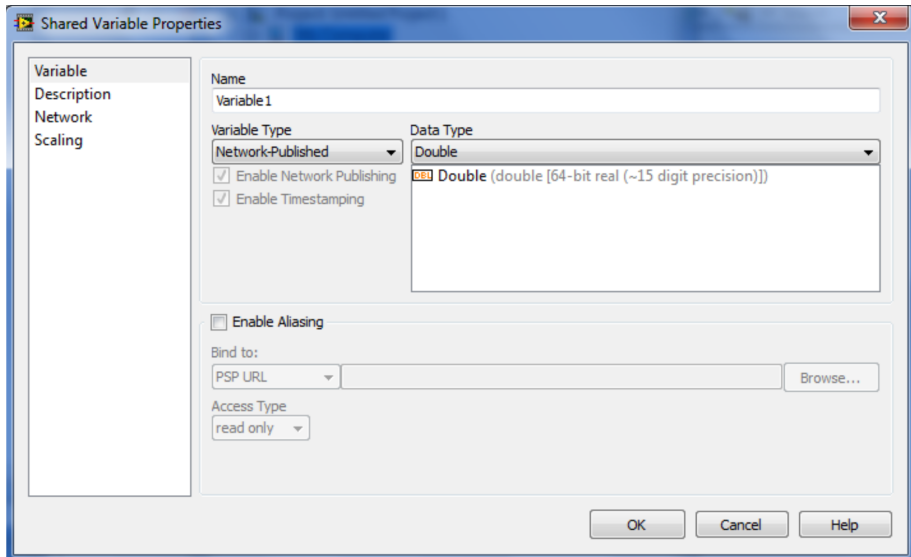


Zmienna współdzielona

- Zmienne współdzielone są stosowane do przekazywanie danych pomiędzy równoległymi VI oraz poprzez sieć.

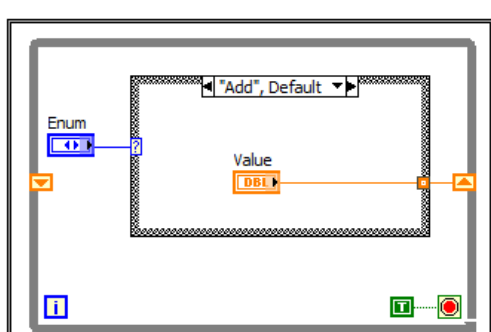


Zmienna współdzielona



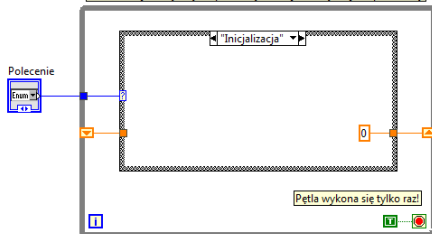
Zmienna funkcjonalna

- Zmienna funkcyjna przechowuje dane w rejestrze przesunym pętli Loop. Wykorzystuje ona fakt, że rejestr przechowuje wartości z ostatniego uruchomienia programu.
- Składa się z pętli while z rejestrem przesunym, strukturą case, enum/ring.
- Pętla while ma na stałe podłączony warunek zatrzymania - w ten sposób kod w pętli uruchamia się tylko raz przy każdym uruchomieniu subVI ze zmienną.
- Zmienna funkcjonalna może oprócz przechowywania wartości wykonywać operacje na danych, albo zmieniać ich postać.

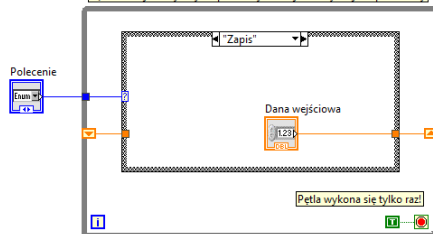


Zmienna funkcjonalna

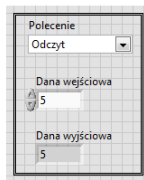
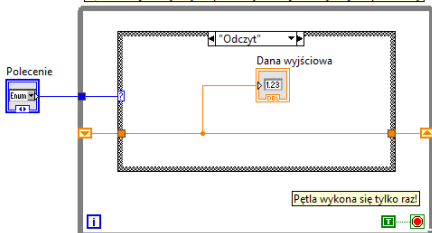
Pętla while jest użyta tylko po to aby na niej stworzyć rejestr przesuwny



Pętla while jest użyta tylko po to aby na niej stworzyć rejestr przesuwny



Pętla while jest użyta tylko po to aby na niej stworzyć rejestr przesuwny



Zmienna funkcjonalna

- Z zmiennej funkcjonalnej korzysta się analogicznie jak ze standardowego subVI.
- Może być wykorzystywana jako zmienna lokalna w jednym VI bądź jako zmienna globalna do przesyłania informacji pomiędzy VI.
- **Zmienna FGV jest jednym z najpopularniejszych sposobów chronienia krytycznego fragmentu diagramu przed zjawiskiem race conditions**, gdyż zmienna FGV to non-reentrant VI. W praktyce oznacza to, że jest tylko jedno miejsce w pamięci gdzie przechowywane są dane z subVI. Taki subVI nie może być wywoływany równolegle. Inne wątki muszą w takim przypadku "poczekać".



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.