Structuri Repetitive si Conditionale

Tipul de date enum (enumerație – pereche cheie text-valoare numerică)

While, For, transmiterea datelor de la o iteratie la alta

Case Structure – functia de IF/ELSE sau de SWITCH

Slideuri preluate din trainingurile oficiale NI disponibile pe ni.com/upb

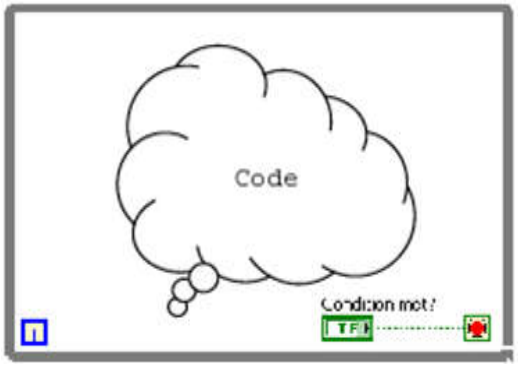
D. While Loops (RECAPITULARE)

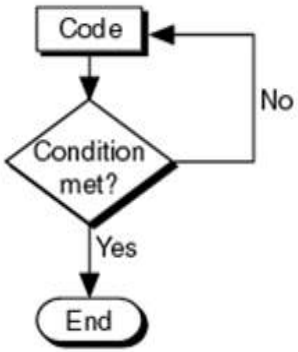
Iteration and Conditional Terminals

Tunnels

Error Checking

While Loops





Repeat (code);

Until Condition met;

End;

LabVIEW While Loop

Flowchart

Pseudo Code

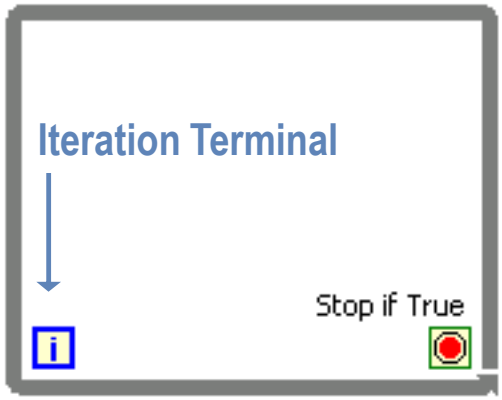
While Loops

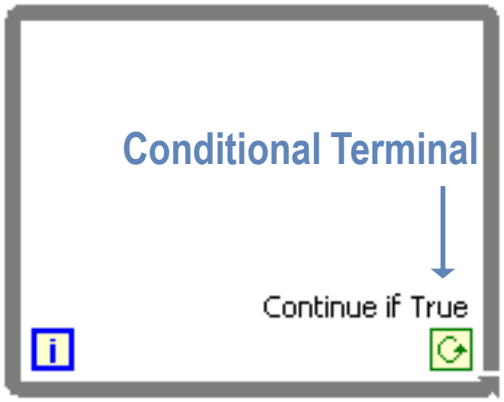
Iteration terminal

- Returns number of times loop has executed.
- Is zero-indexed.

Conditional terminal

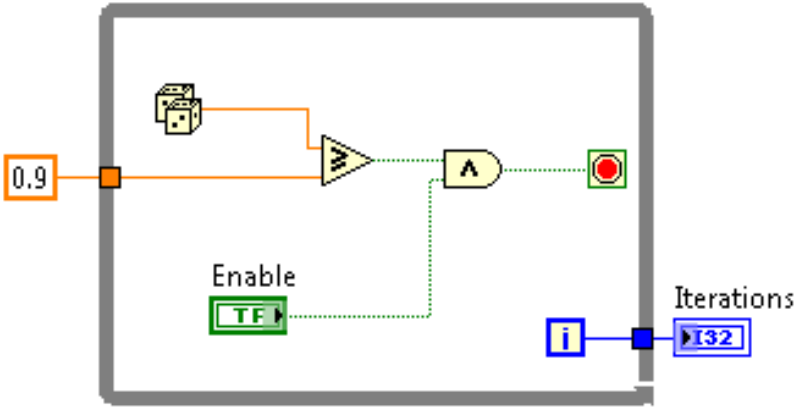
- Defines when the loop stops.
- Has two options.
  - Stop if True
  - Continue if True





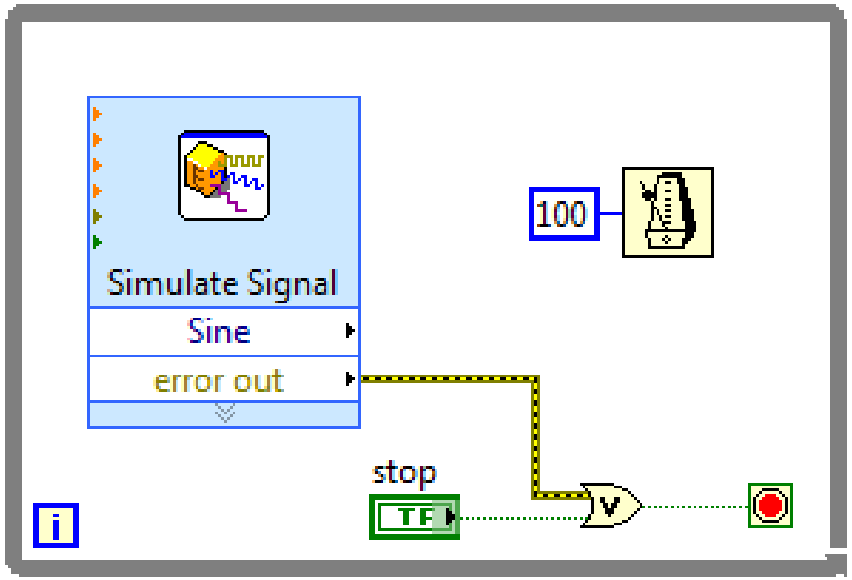
While Loops – Tunnels

- Tunnels transfer data into and out of structures.
- Data pass out of a loop after the loop terminates.
- When a tunnel passes data into a loop, the loop executes only after data arrive at the tunnel.



While Loops – Error Checking and Error Handling

Use an error cluster in a While Loop to stop the While Loop if an error occurs.



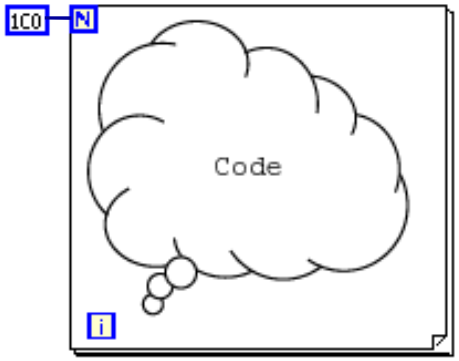
E. For Loops

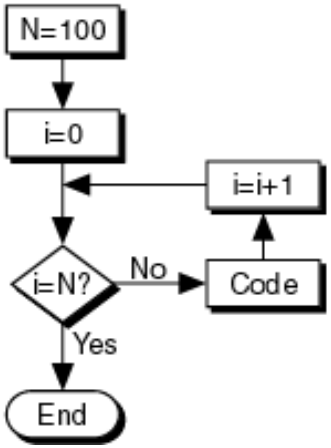
Conditional Terminal

Comparison with While Loops

Numeric Conversion for Count Terminal

For Loops






```
N=100;
i=0;
Until i=N:
    Repeat (code;i=i+1);
End;
```

LabVIEW For Loop

Flowchart

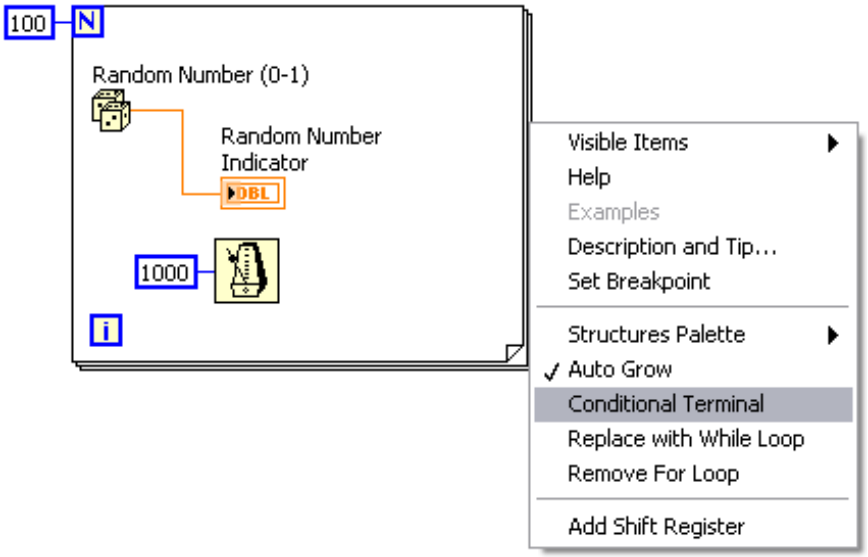
Pseudo Code

For Loops

- Create a For Loop the same way you create a While Loop.
- You can replace a While Loop with a For Loop by right-clicking the border of the While Loop and selecting
  Replace with For Loop from the shortcut menu.
- The value in the count terminal (an input terminal) indicates how many times to repeat the subdiagram in the For Loop.

For Loops – Conditional Terminal

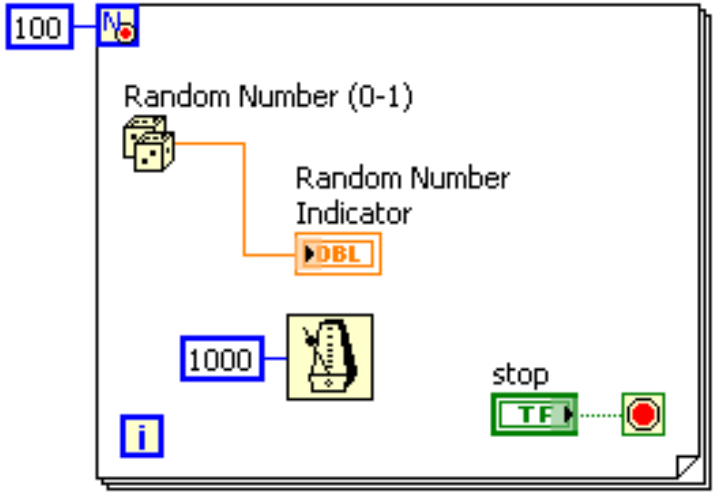
You can add a conditional terminal to configure a For Loop to stop when a Boolean condition is true or an error occurs.



For Loops – Conditional Terminal

For Loops configured with a conditional terminal have:

- A red glyph next to the count terminal.
- A conditional terminal in the lower right corner



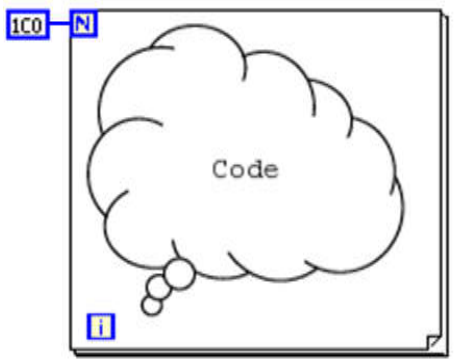
For Loop/While Loop Comparison

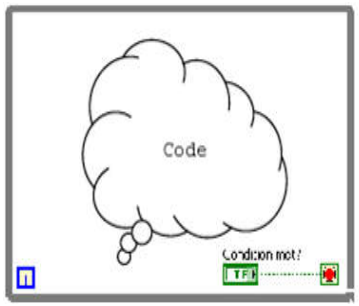
For Loop

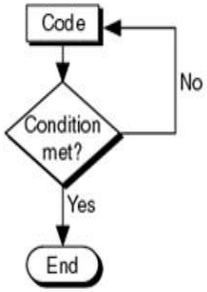
- Executes a set number of times unless a conditional terminal is added.
- Can execute zero times.
- Tunnels automatically output an array of data.

While Loop

- Stops executing only if the value at the conditional terminal meets the condition.
- Must execute at least once.
- Tunnels automatically output the last value.

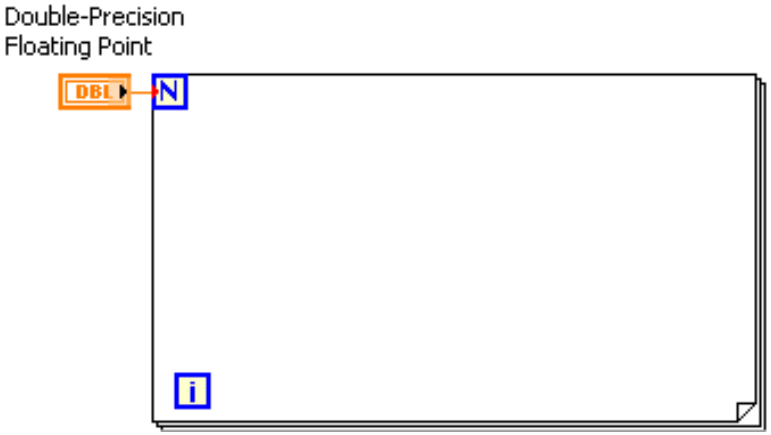


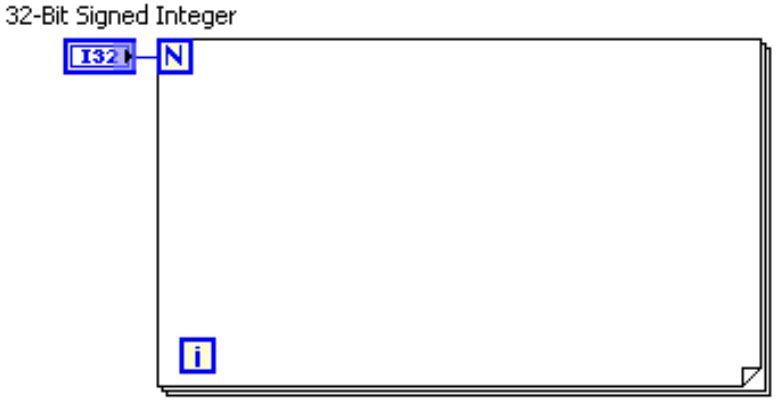




For Loops – Numeric Conversion

- The number of iterations a For Loop executes must be specified in non-negative integers.
- If you wire a double-precision, floating-point numeric value to the count terminal, LabVIEW converts the numeric value to a 32-bit signed integer.

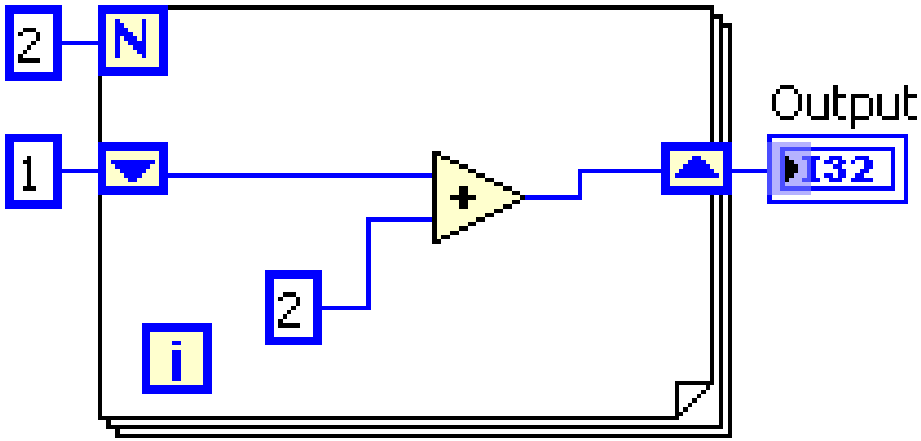




Trimiterea datelor de la o iterație la alta

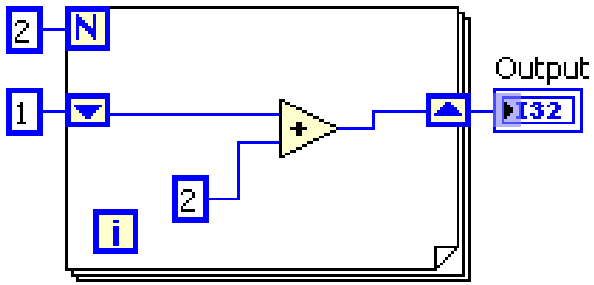
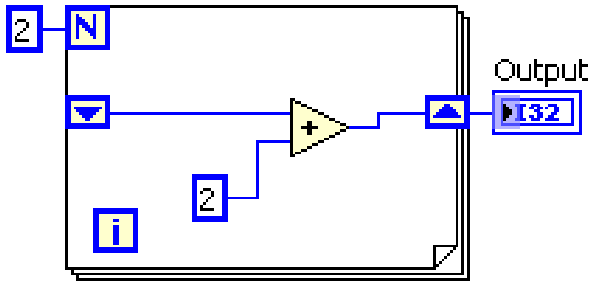
Data Feedback in Loops

- When programming with loops, you often need to know the values of data from previous iterations of the loop.
- Shift registers transfer values from one loop iteration to the next.



Initializing Shift Registers

Run once → VI finishes → Run again

Block Diagram	1st run	2nd run
Initialized Shift Register 	Output = 5	Output = 5
Not Initialized Shift Register 	Output = 4	Output = 8

Use Default if Unwired

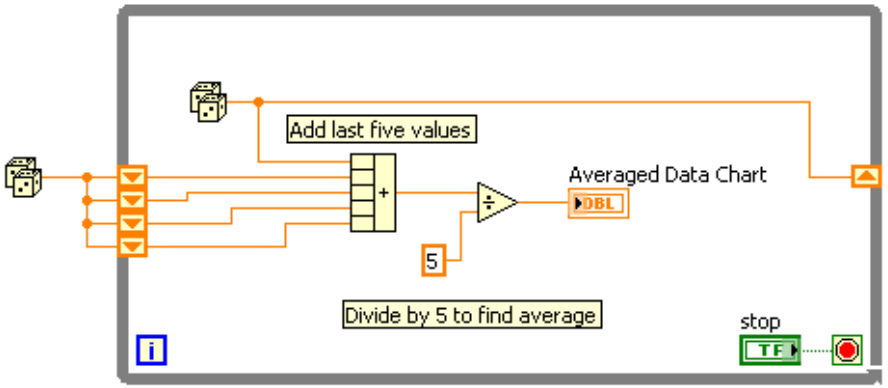
Default values vary by data type:

Data Type	Default Value
Numeric	0
Boolean	FALSE
String	Empty

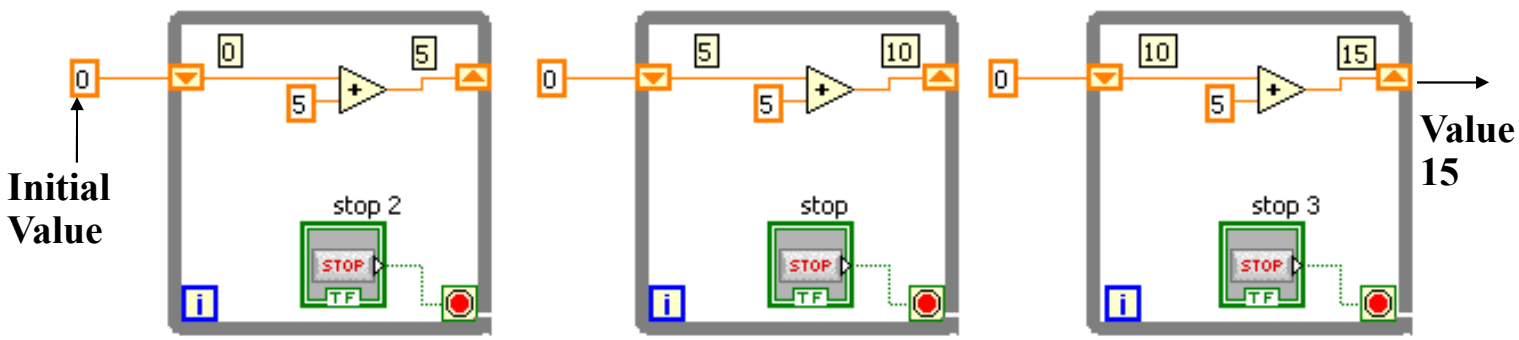
Uninitialized shift registers use default values for first run.

Multiple Previous Iterations

- Stacked shift registers remember values from multiple previous iterations and carry those values to the next iterations.
- Right-click the left shift register and select **Add Element** from the shortcut menu to stack a shift register.



Shift Registers in action



Before Loop Begins

First Iteration

Second Iteration

Last Iteration

Structuri conditionale

I. Case Structures

Parts of a Case Structure

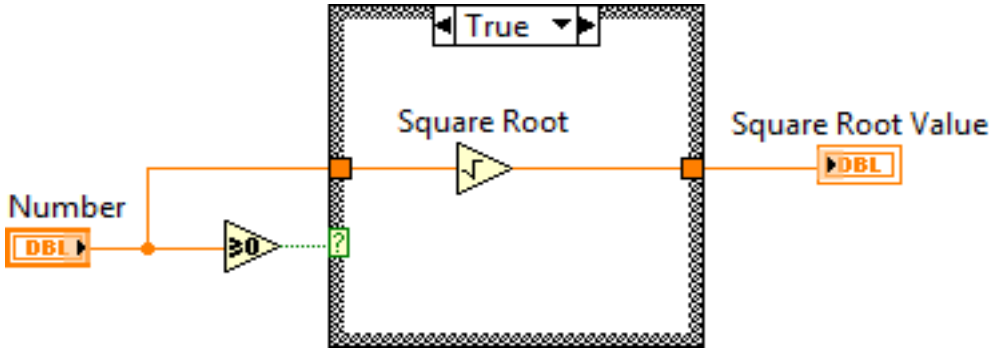
Enum Case Structures

Error Case Structures

Input and Output Tunnels

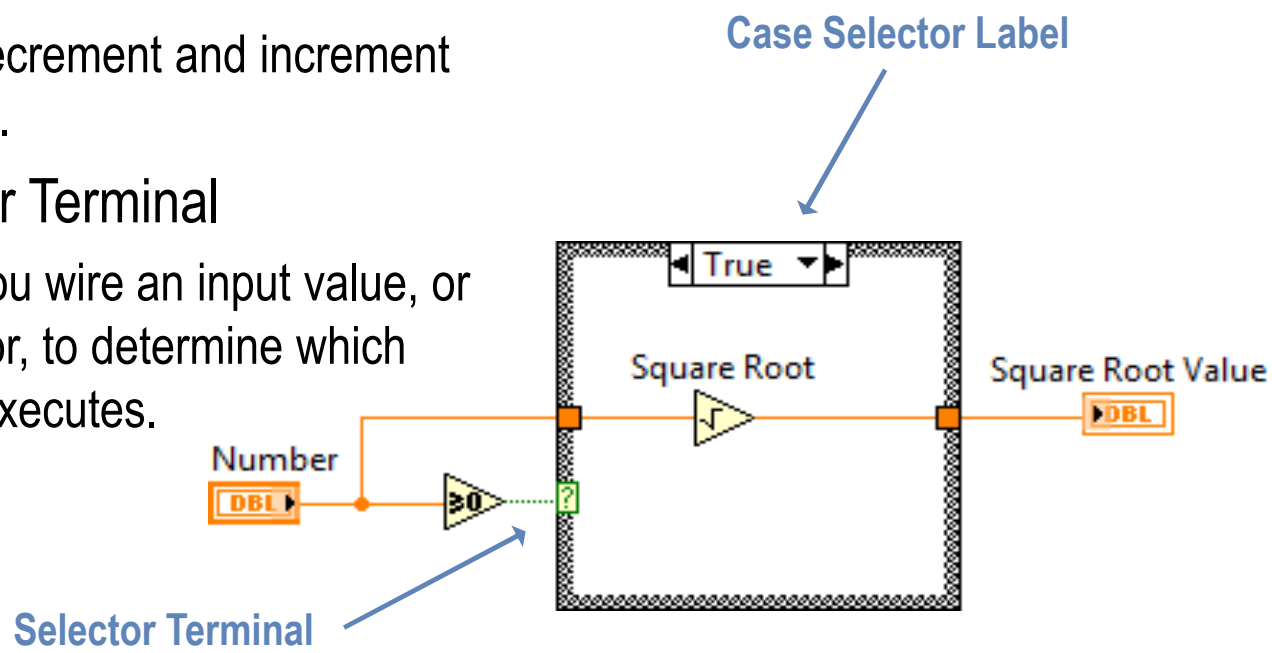
Case Structures

- Have two or more subdiagrams or cases.
- Use an input value to determine which case to execute.
- Execute and display only one case at a time.
- Are similar to **case** statements or **if...then...else** statements in text-based programming languages.



Case Structures

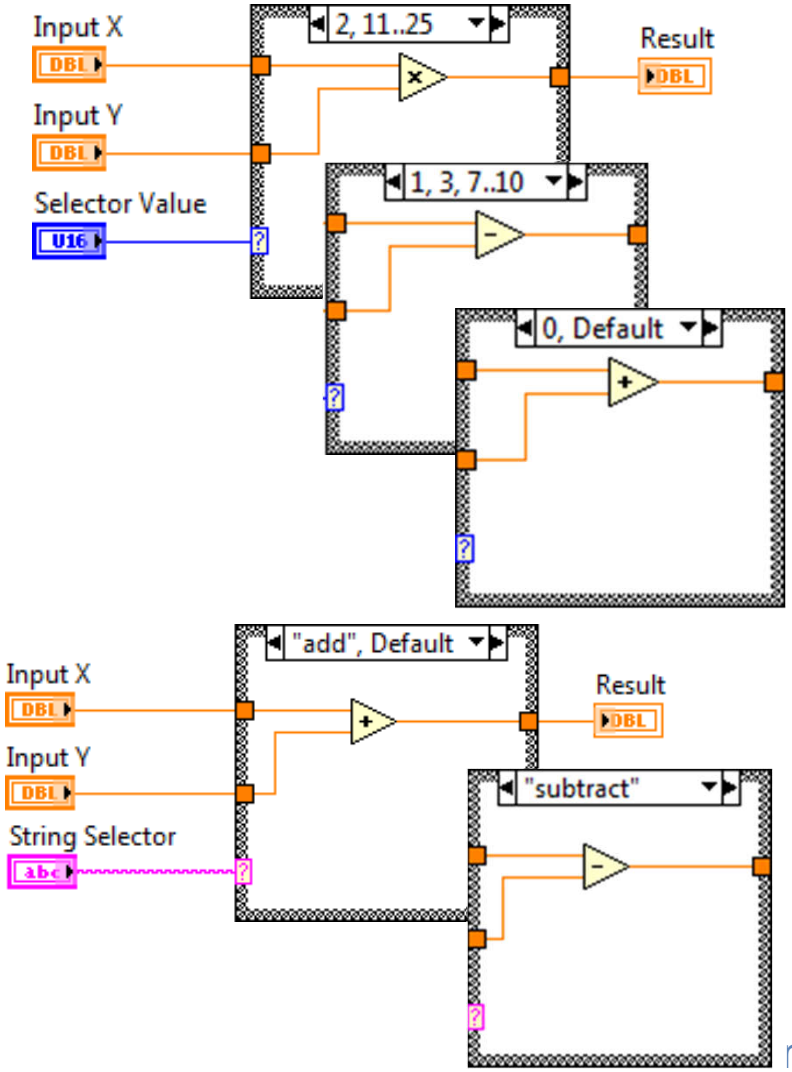
- Case Selector Label
  - Contains the name of the current case.
  - Has decrement and increment arrows.
- Selector Terminal
  - Lets you wire an input value, or selector, to determine which case executes.



Case Structures

Selector terminal data types:

- Boolean
  - True case and False Case
- Error Cluster
  - Error Case and No Error Case
- Integer, string, or enum
  - Structure can have any number of cases.
  - Include a Default diagram to avoid listing every possible input value.



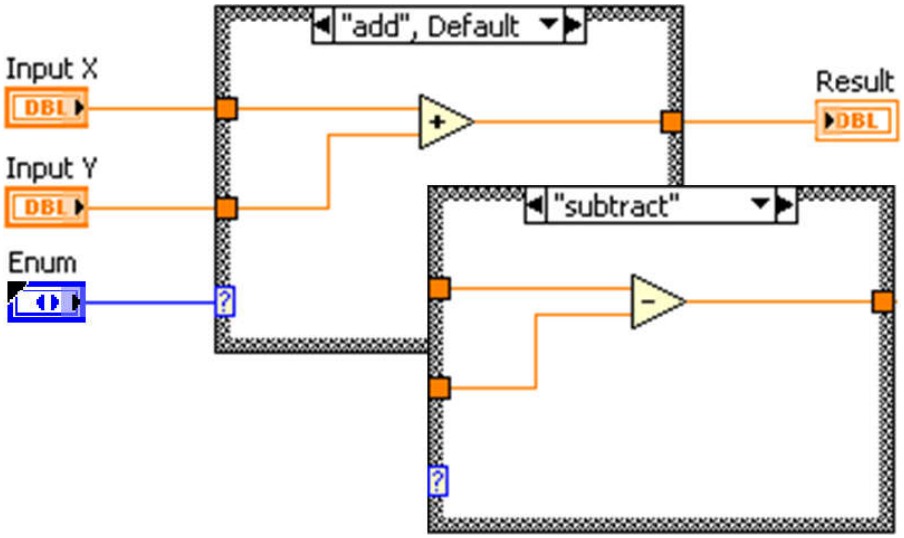
Abstractizarea cazului

Pentru abstractizarea datelor poate fi folosit tipul de date ENUM

Enum data type defineste in realitate un tip de date caracterizat printr-o enumerație de nume(text). Fiecarui nume ii corespunde o valoare. Astfel, in loc sa folosim un integer pentru a selecta cazul, putem folosi un Enum

Enum Case Structure

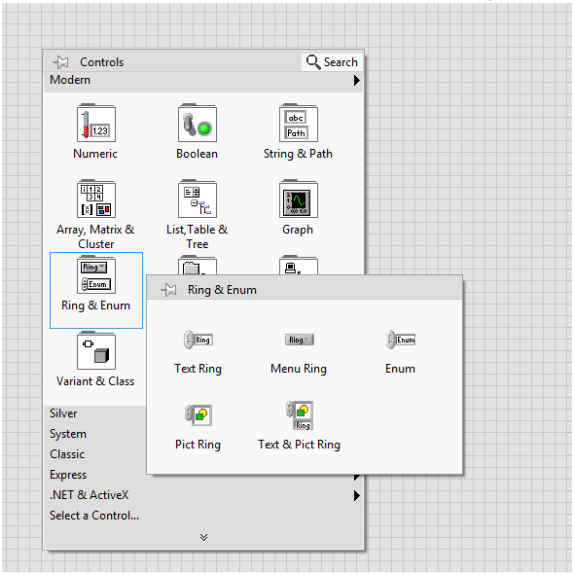
- Gives users a list of items from which to select
- The case selector displays a case for each item in the enumerated type control

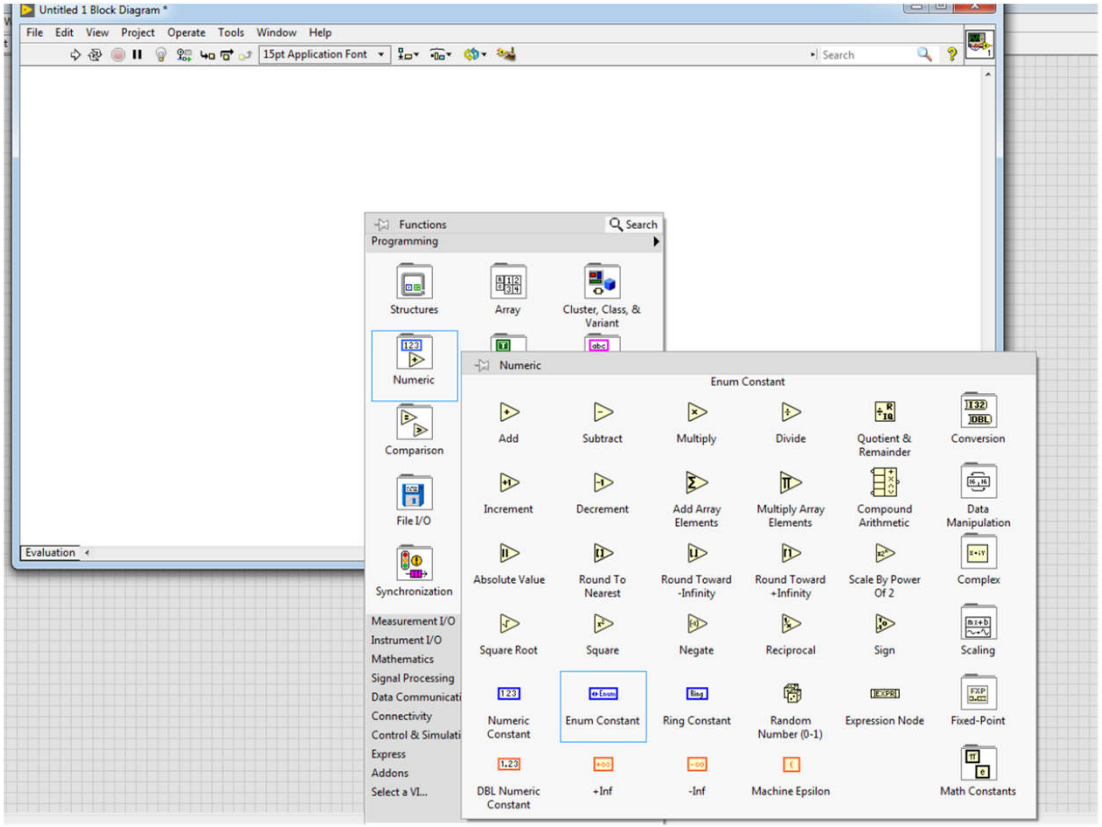


Adaugarea unui Enum

Enum-ul poate fi adaugat ca element tip control, indicator sau constanta(doar in block diagram)

Exemplu utilizare: https://youtu.be/tlmN4C03W1k (pana la case structure)





Enums

- Enums give users a list of items from which to select.
- Each item represents a pair of values.
  - String
  - 16-bit Integer