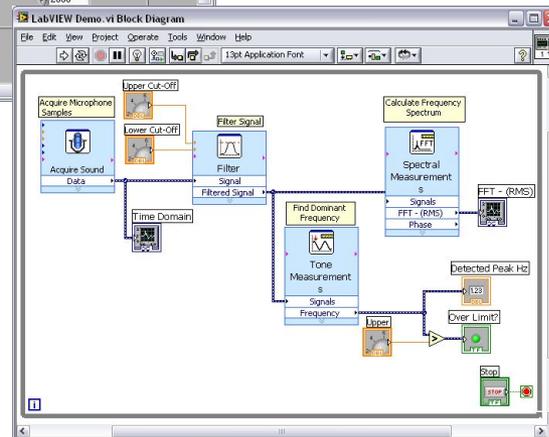
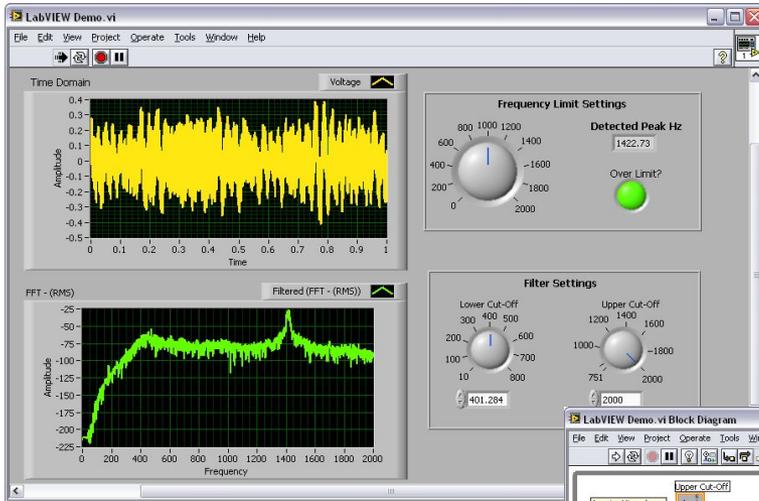


# Введение в LabVIEW

Графическая среда программирования  
для инженеров и ученых



# LabVIEW™



**LabVIEW** (англ. **L**aboratory **V**irtual **I**nstrumentation **E**ngineering **W**orkbench) — это среда разработки и платформа для выполнения программ, созданных на графическом языке программирования «G» фирмы National Instruments (США)

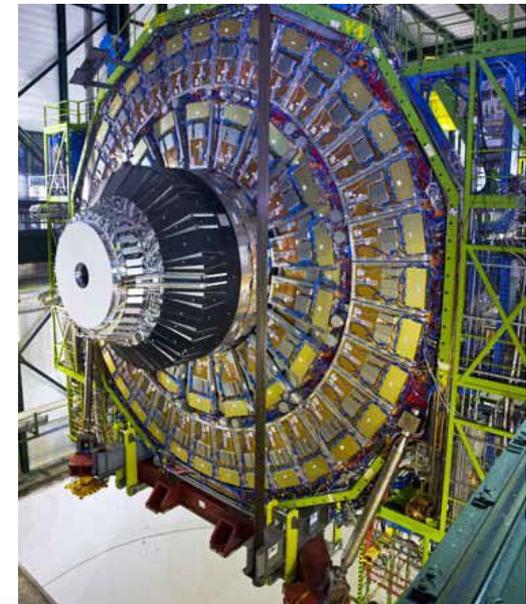
Первая версия LabVIEW была написана в 1986 году, продукцию которой покупают более 25 тысяч организации, расположенных в 90 странах мира. Последние десять лет National Instruments входит в число 100 лучших работодателей США.

## Те же самые технологии National Instruments используются для программирования робота в школе и контроля потока элементарных частиц в большом адронном коллайдере

Школьники и подростки от 6 до 14 лет используют специализированную версию LabVIEW для создания собственных роботов на базе конструктора Lego серий Wedo и Mindstorms.



127 модульных систем National Instruments, установленных в 108 коллиматорах большого адронного коллайдера позволили контролировать поток элементарных частиц с точностью до 20 мкм.



## Имитатор ГЛОНАСС

*Новосибирский государственный технический университет (НГТУ)*

Разработанный в НГТУ имитатор сигналов ГЛОНАСС на основе программно-аппаратных средств National Instruments был представлен в виде конечного продукта. Компания National Instruments выступила площадкой для продвижения настоящего продукта на рынок. С 2009 года НГТУ успешно продает имитатор ГЛОНАСС при информационной поддержке National Instruments.

### **Учебно-производственный инновационный центр по микроэлектронике**

*Санкт-Петербургский технический колледж управления и коммерции*

Центр сквозного проектирования и прототипирования позволяет студентам проходить полный цикл обучения от идеи до готового продукта. Оснащение Центра производственным оборудованием позволяет наладить мелкосерийное производство. Обучение разбито на 3 этапа:

- Теоретическая подготовка студентов
- Практические навыки в ходе выполнения реальных работ
- Курсовые проекты, целью которых является проектирование, разработка приборов или микроэлектронных устройств с последующим их тестированием

## Промышленные решения

### Примеры применения технологий National Instruments в мире



#### **Первый в мире 3D оптический томограф реального времени**

*Университет Касато, Япония*

Новейшая методика оптической когерентной томографии (ОКТ) и 320-канальная система сбора и обработки данных на основе технологий NI PXI и NI FlexRIO позволила ученым из Японии первыми в мире реализовать идею 3D ОКТ имиджинга с беспрецедентной скоростью 12 объемных образов в секунду. Благодаря высокому пространственному разрешению и глубине томографического сечения данные технологии значительно упрощают диагностику онкологических заболеваний трахеи и органов пищеварительной системы.



#### **Испытательный стенд для тестирования атомного реактора нового поколения**

*Университет штата Орегон, США*

Тесная взаимосвязь исследований, проведенных университетом и технологий National Instruments, позволили разработать надежную и точную систему сбора, обработки, хранения и чтения данных. Кроме того работа, проделанная исследовательским центром помогла корпорации Westinghouse в кратчайшие сроки добиться максимальной безопасности проекта.

## Промышленные решения

### Примеры применения технологий National Instruments в мире



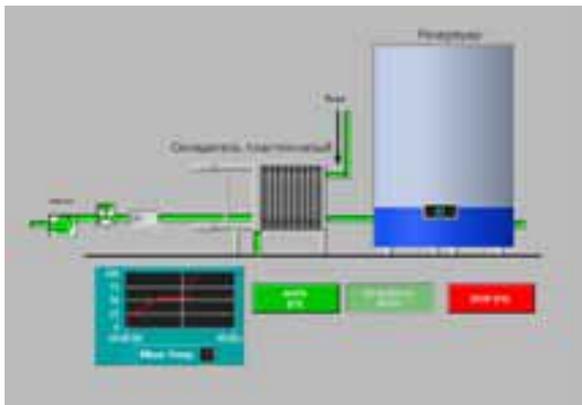
#### **Самолет – лаборатория AN 30 «ОПТИК Э»**

*Институт оптики атмосферы СО РАН*

Основной целью разработки было изготовление комплекса аппаратуры, предназначенного для измерения количества загрязняющих веществ в воздухе и на подстилающей поверхности. Система регистрации представляет собой блок подготовки аналоговых сигналов, систему сбора данных с беспроводной передачей данных, а также программного обеспечения разработанного на LabVIEW. Раз в секунду система регистрирует и сохраняет данные в текстовый файл.

#### **Разработка АСУ линией поточного производства**

*Орловский государственный технический университет*



Целью является разработка системы управления и контроля типовой технологической поточной линии производства кефира резервуарным способом с охлаждением продукта в резервуарной емкости для сквашивания молока и фасовкой в полиэтиленовые пакеты. Данная автоматизированная система управления поточной линией производства молочной продукции предназначена для внедрения в Орловском филиале предприятия «Юнимилк».

# Программа в LabVIEW называется **Виртуальным прибором (ВП)**

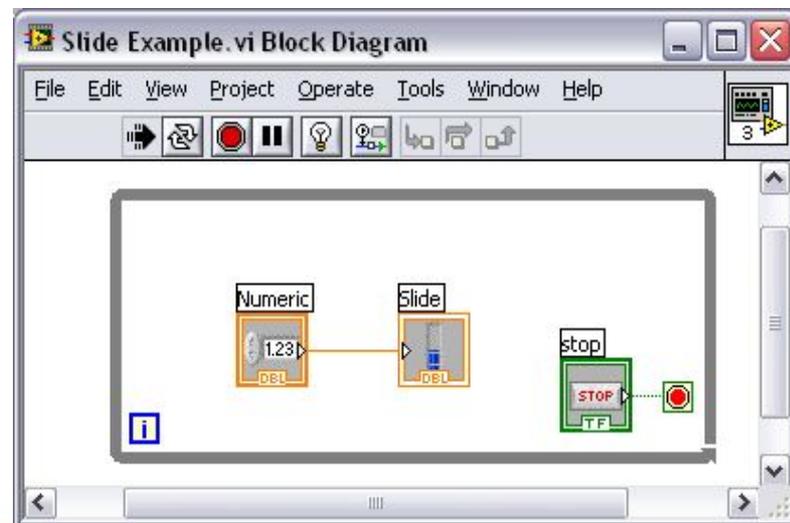
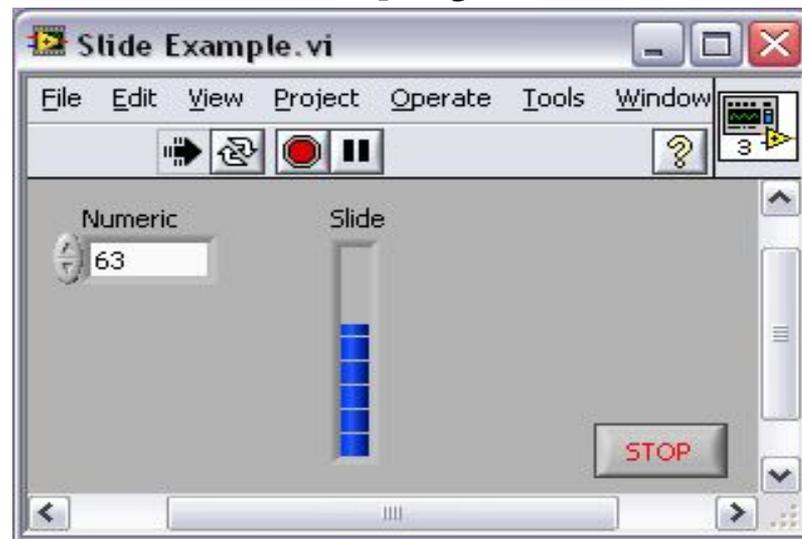
Каждый ВП имеет 2 окна

## Лицевая панель

- Интерфейс пользователя
  - Элементы управления = Входы
  - Индикаторы = Выходы

## Блок диаграмма

- Графический код
  - Данные передаются по проводникам от элементов управления, через функции, к индикаторам



ВП симулируют реальные физические приборы, например, осциллограф или мультиметр.

LabVIEW содержит полный набор инструментов для сбора, анализа, представления и хранения данных.

LabVIEW имеет встроенные возможности для работы в компьютерных сетях Интернет, используя LabVIEW Web Server и программные стандарты TCP/IP и Active X.

LabVIEW – это 32-х разрядный компилятор, который создает как автономные модули (.EXE), так и совместно используемые динамические библиотеки (.DLL).

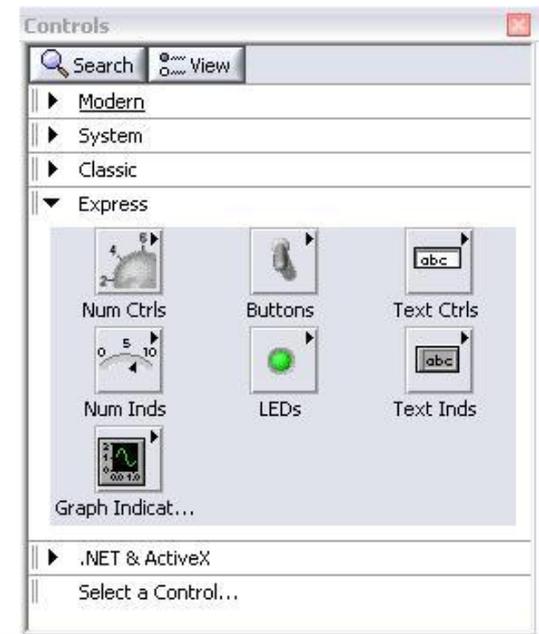
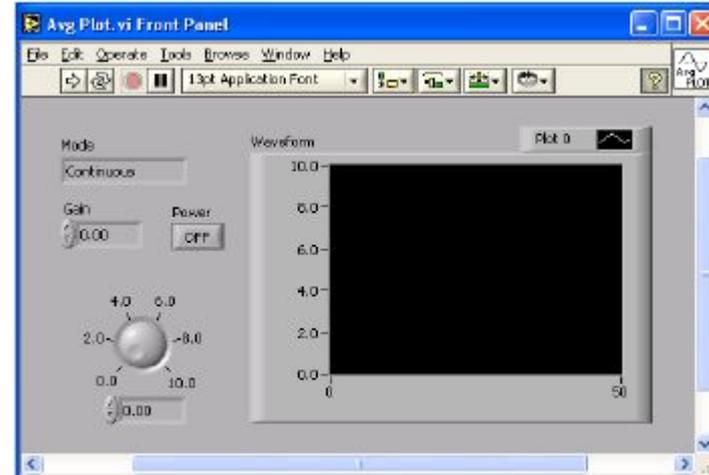
Лицевая панель – это интерфейс пользователя ВП.  
Лицевая панель создается с использованием палитры Элементов (Controls).

Эти элементы делятся:

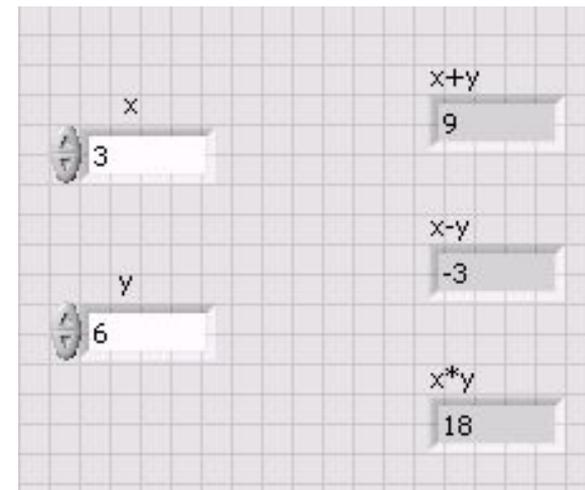
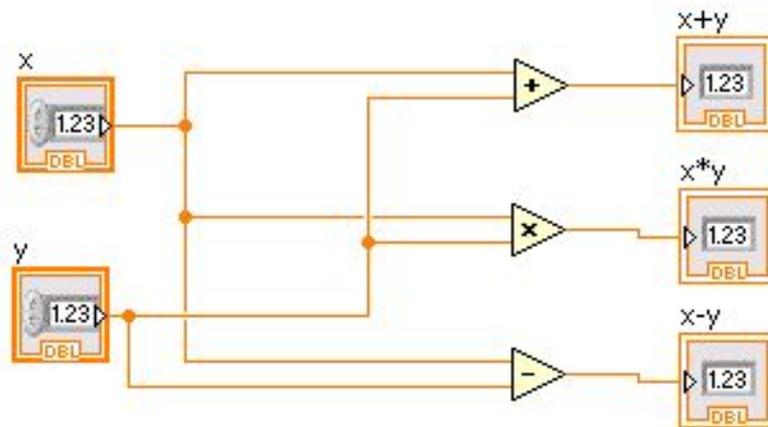
- **Управления** - средства ввода данных (кнопки, переключатели, ползунки и др.);
- **Отображения** - средствами отображения данных (графики, цифровые табло, светодиоды и т.д.)

Данные, вводимые на лицевой панели ВП, поступают на блок-диаграмму, где ВП производит с ними необходимые операции.

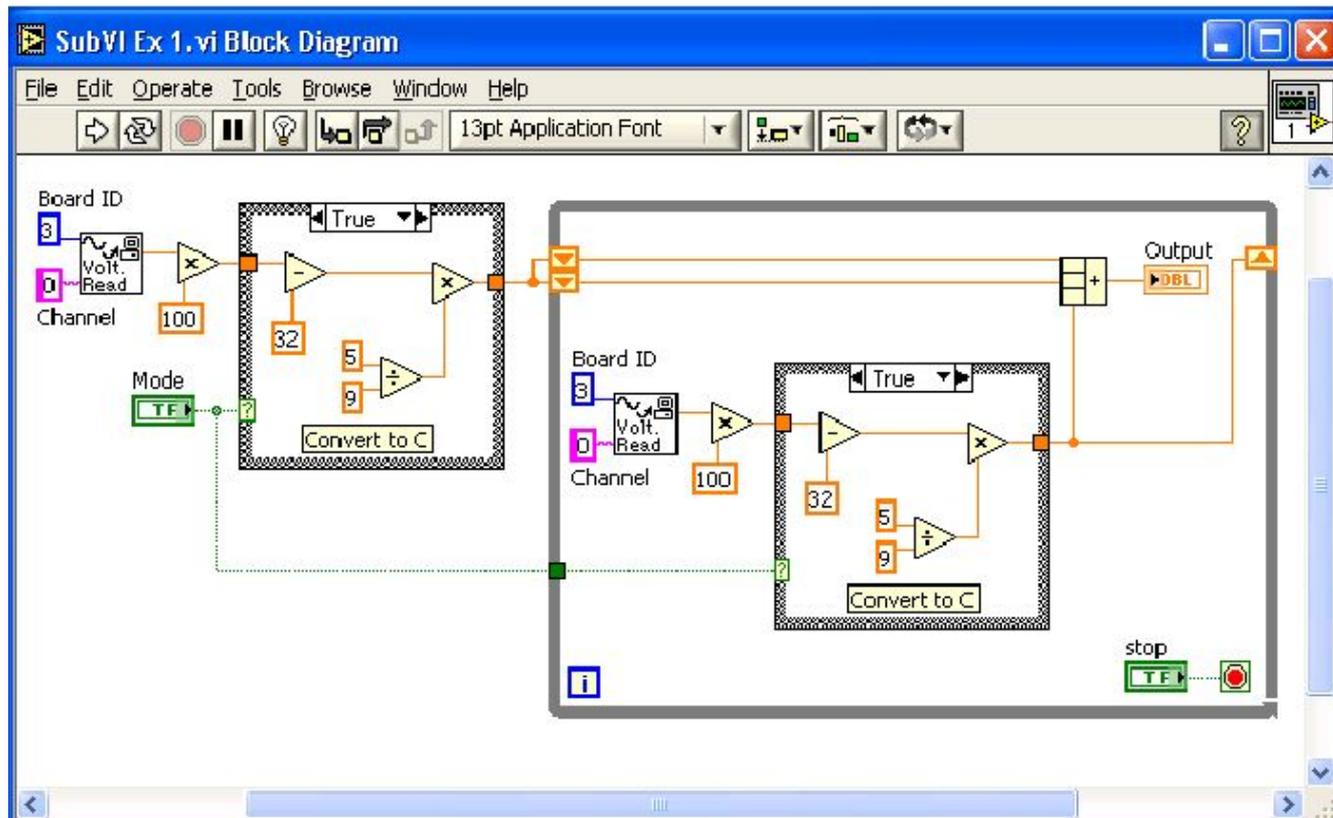
Результат вычислений передается на элементы отображения информации на лицевой панели ВП.



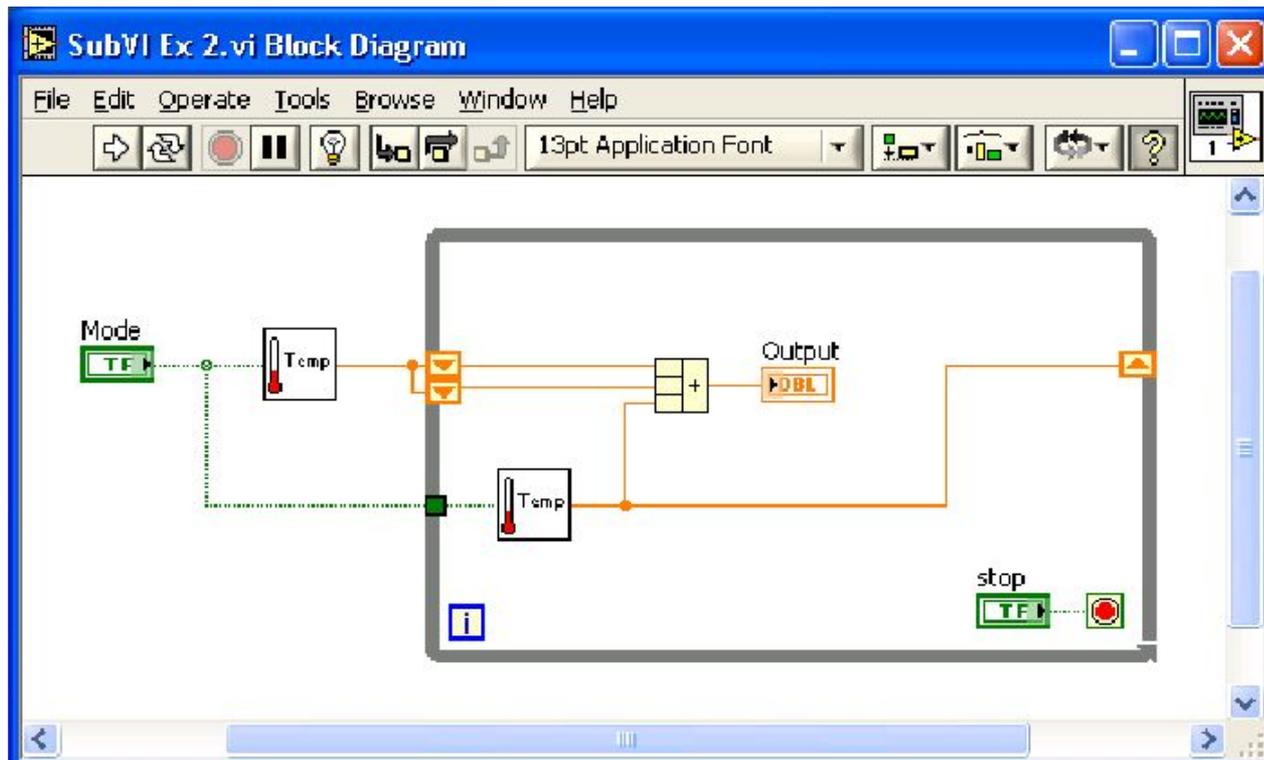
Следующий пример показывает блок-диаграмму и соответствующую ей лицевую панель:



Преимущество LabVIEW заключается в иерархической структуре ВП. Количество уровней в иерархии не ограничено. Использование подпрограммы ВП помогает быстро изменять и отлаживать блок-диаграмму. Например, приведенная ниже блок-диаграмма содержит две идентичные операции.



Можно создать подпрограмму ВП, которая выполнит эту операцию, и можно вызвать эту подпрограмму дважды. Возможно многократное использование подпрограммы ВП в другом виртуальном приборе. Следующий пример демонстрирует использование Temperature VI в качестве подпрограммы на блок-диаграмме.

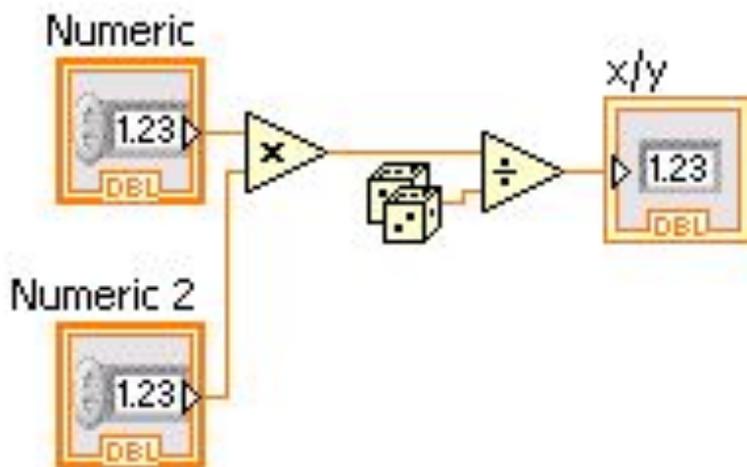


## Последовательность обработки данных

В Visual Basic, C++, Java и большинстве других текстовых языков программирования порядок выполнения всей программы определяется расположением функций программы.

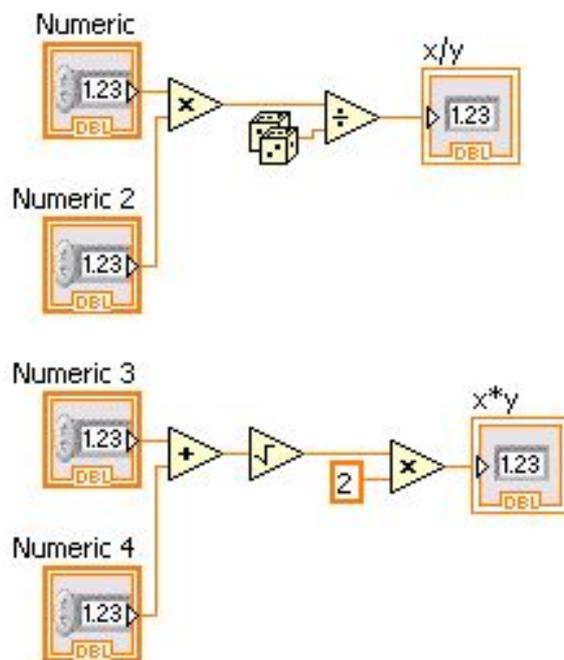
В среде LabVIEW используется потоковая модель обработки данных. Узлы блок-диаграммы выполняют заложенные в них функции, если данные поступили на все необходимые поля ввода/вывода. По окончании выполнения операции одним узлом результаты операции по проводникам данных передаются следующему узлу и т.д. Другими словами, готовность входных данных определяет последовательность выполнения узлов блок-диаграммы.

В качестве примера можно рассмотреть блок-диаграмму, которая умножает два числа и затем делит получившееся произведение на случайное число (0-1). В этом случае блок-диаграмма выполняется слева направо не потому, что объекты помещены в этом порядке, а потому, что одно из полей ввода функции деления не определено, пока не выполнилась функция умножения и не передала данные к функции деления. Узел выполняется только тогда, когда определены его поля ввода данных.



На этом примере последовательность выполнения не определена, т.к. умножение и сложение заданных чисел инициализируется одновременно.

Когда необходимо выполнить одну часть кода блок-диаграммы раньше другой и нет зависимости данных между функциями, для установки порядка выполнения следует использовать методы программирования.

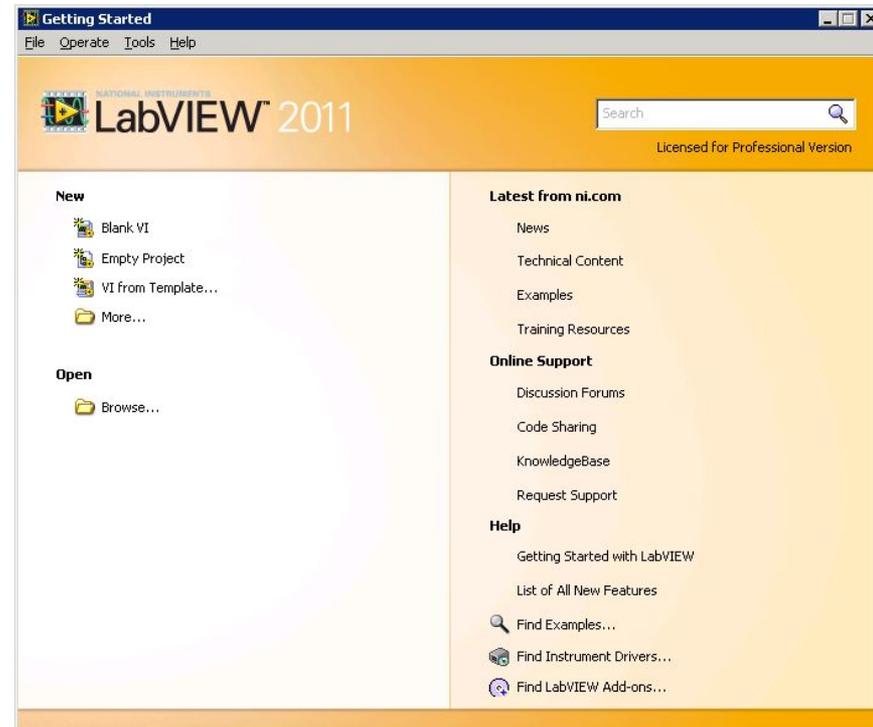


# Организация программной среды LabVIEW

При запуске LabVIEW появляется окно запуска для работы с системой, с помощью которого можно создать новый ВП, проект, открыть уже существующий ВП, найти примеры или получить доступ к подсказке LabVIEW Help. Окно запуска появляется также при закрытии всех лицевых панелей и блок-диаграмм.

Окно запуска содержит следующие компоненты:

- Панель меню со стандартными пунктами, например, File»Exit
- Окно Files, позволяющее открыть или создать ВП. В данном окне находятся вкладки New и Open. Используя вкладку New, Вы можете создать новый ВП, новый проект или загрузить шаблон ВП.Используя вкладку Open, можно открыть созданный ранее ВП.
- Окно Resources позволяет обратиться за помощью или дополнительной информацией к различным текстовым и Internet ресурсам, а также посмотреть встроенные примеры.



# Инструментальная панель лицевой панели

Инструментальная панель используется для запуска и редактирования ВП



Кнопка запуска Run – запускает ВП



Во время работы ВП кнопка Run меняет свой вид, как показано слева, если этот виртуальный прибор высокого уровня.



Если ВП работает в качестве подпрограммы, то кнопка Run выглядит, как показано слева.



Кнопка Run выглядит в виде «сломанной» стрелки, как показано слева, во время создания или редактирования ВП. В таком виде кнопка показывает, что ВП не может быть запущен на выполнение. После нажатия этой кнопки появляется окно Error list, в котором перечислены допущенные ошибки.



Кнопка непрерывного запуска Run Continuously – ВП выполняется до момента принудительной остановки.



Во время выполнения ВП появляется кнопка Abort Execution. Эта кнопка используется для немедленной остановки выполнения ВП.

# Инструментальная панель блок-диаграммы



Кнопка Highlight Execution предназначена для просмотра потока данных через блок-диаграмму (режим отладки). Повторное нажатие кнопки отключает этот режим.



Кнопка Retain Wire Values предназначена для сохранения данных прошедших по проводникам. Включив ее, можно посмотреть значения данных в любом проводнике ВП в любой момент времени.



Кнопка Step Into используется при пошаговом выполнении цикла от узла к узлу, подпрограммы ВП и т.д. При этом узел мигает, обозначая готовность к выполнению.



Кнопка Step Over позволяет пропустить в пошаговом режиме цикл, подпрограмму и т.д.



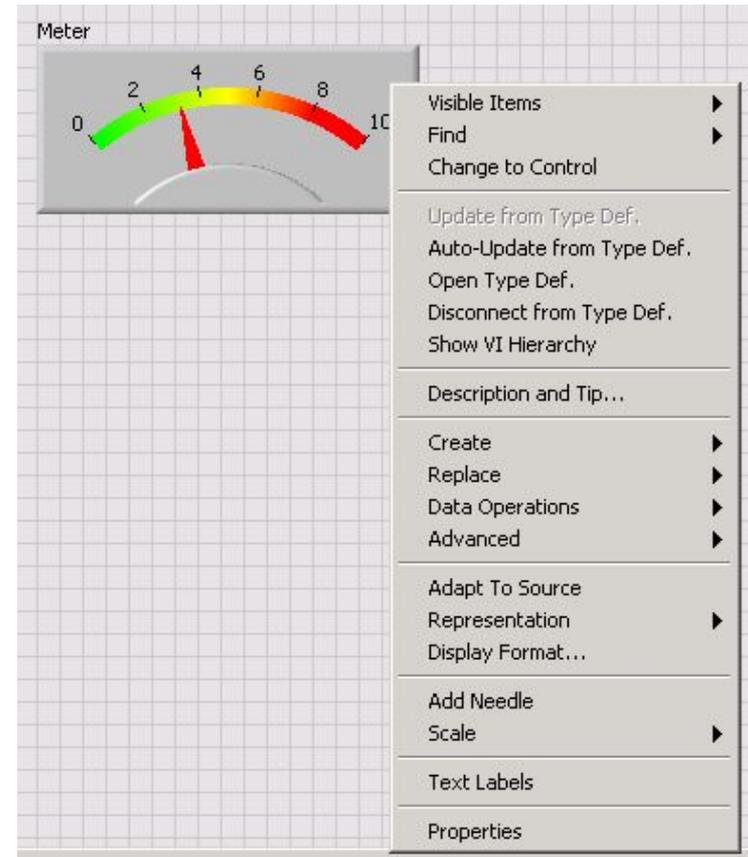
Кнопка Step Out позволяет выйти из цикла, подпрограммы и т.д. Выход из узла предполагает завершение выполнения этого узла в пошаговом режиме и переход в следующий.



Кнопка Warning появляется, когда есть потенциальная проблема с блок-диаграммой, но она не запрещает выполнение ВП. Кнопку Warning можно активизировать, войдя в пункт главного меню Инструменты, далее – Опции, Отладка (Tools»Options»Debugging).

## Контекстное меню

Контекстное меню используется наиболее часто. Все объекты LabVIEW, свободное рабочее пространство лицевой панели и блок-диаграммы имеют свои контекстные меню. Контекстное меню используется для изменения поведения объектов блок-диаграммы и лицевой панели. Контекстное меню вызывается щелчком правой кнопкой мыши на объекте, лицевой панели или блок-диаграмме.



## Палитры

LabVIEW имеет вспомогательные палитры, используемые для создания и выполнения ВП: **Controls Palette** (Палитра Элементов) и **Functions Palette** (Палитра Функций). Эти палитры можно поместить в любом месте экрана.

## Палитра Инструментов

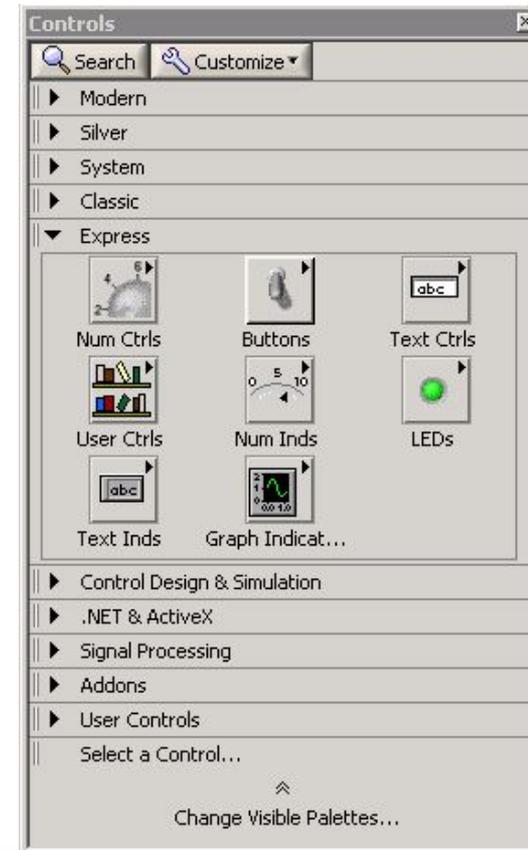
Создавать, редактировать и отлаживать ВП можно с помощью Tools Palette (Палитры Инструментов). Палитра Инструментов доступна как на лицевой панели, так и на блок-диаграмме. Термин инструмент подразумевает специальный операционный режим курсора мыши. При выборе определенного инструмента значок курсора изменяется на значок данного инструмента. Палитра Инструментов доступна через пункт главного меню View»Tools Palette. Палитру Инструментов можно размещать в любой области рабочего пространства блок-диаграммы и лицевой панели.

## Палитры

LabVIEW имеет вспомогательные палитры, используемые для создания и выполнения ВП: **Controls Palette** (Палитра Элементов) и **Functions Palette** (Палитра Функций). Эти палитры можно поместить в любом месте экрана.

### Палитра Элементов (Controls)

Палитра Элементов используется для размещения элементов управления и отображения на лицевой панели. Она доступна только на лицевой панели. Чтобы отобразить палитру Элементов, следует либо выбрать в пункте главного меню View»Controls Palette, либо щелкнуть правой кнопкой мыши в рабочем пространстве лицевой панели.

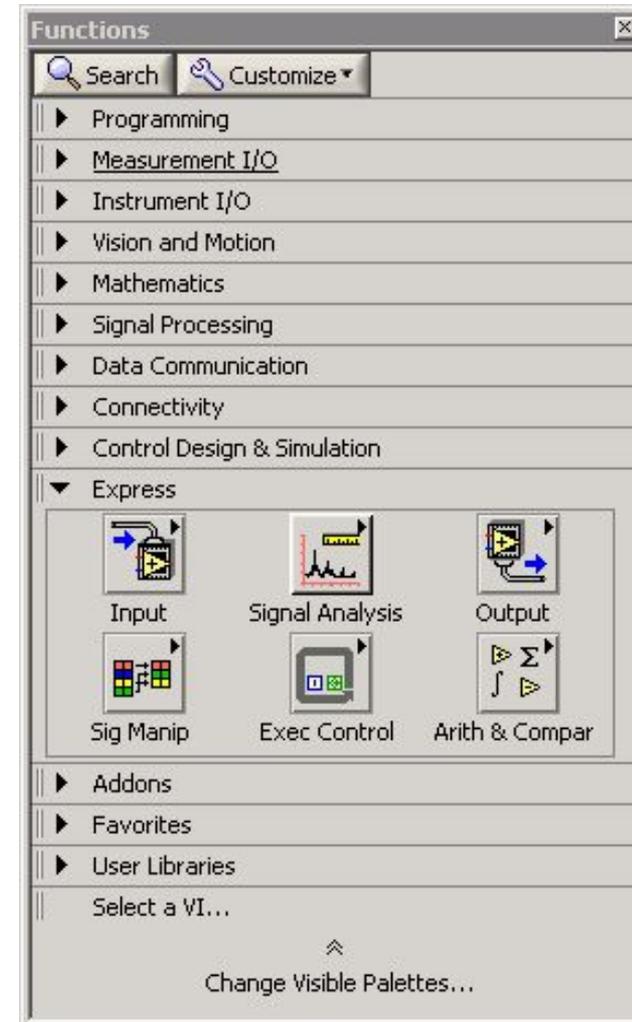


## Палитра Функций (Functions)

Палитра Функций используется для создания блок-диаграммы. Она доступна только на блок-диаграмме. Чтобы отобразить палитру Функций, следует либо выбрать в пункте главного меню View»Functions Palette, либо щелкнуть правой кнопкой мыши в рабочем пространстве блок-диаграммы. Используя кнопку в верхнем левом углу палитры можно зафиксировать ее на экране.

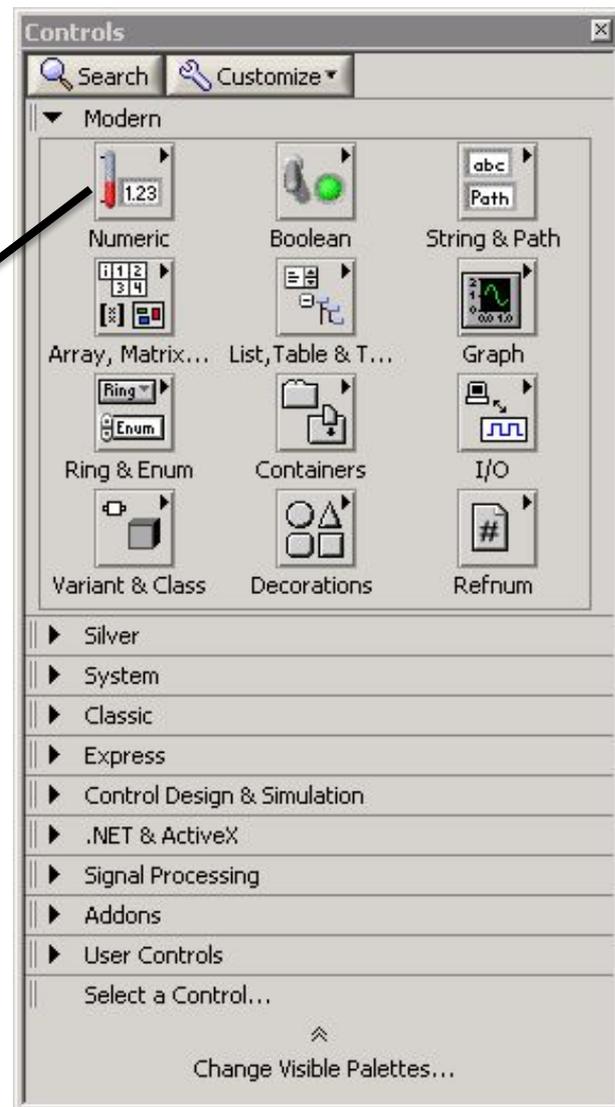
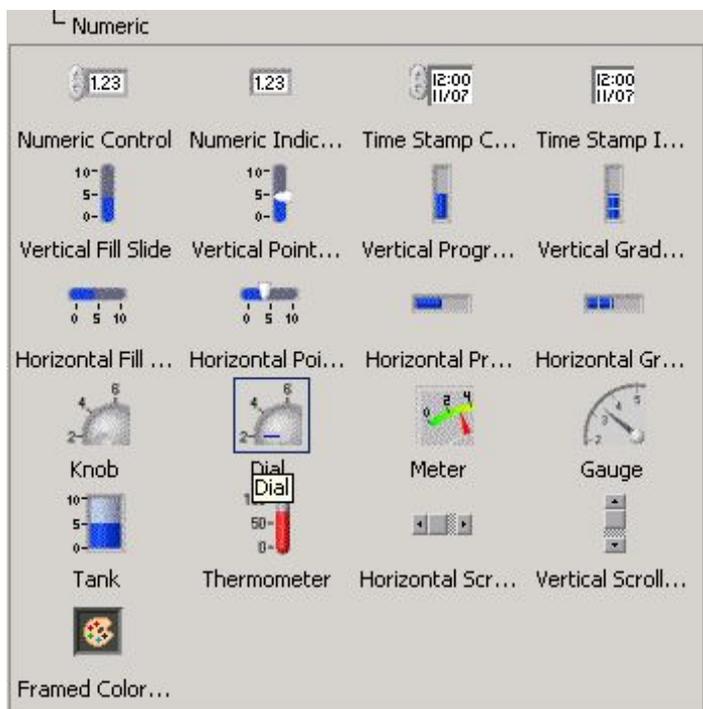
Пункты на палитрах организованы в соответствии с категориями. Можно выбрать способ отображения различных категорий на палитре используя кнопку Customize в верхней части палитры.

При нажатии на кнопку Search можно перейти в режим поиска какой-либо функции, узла или ВП по названию.



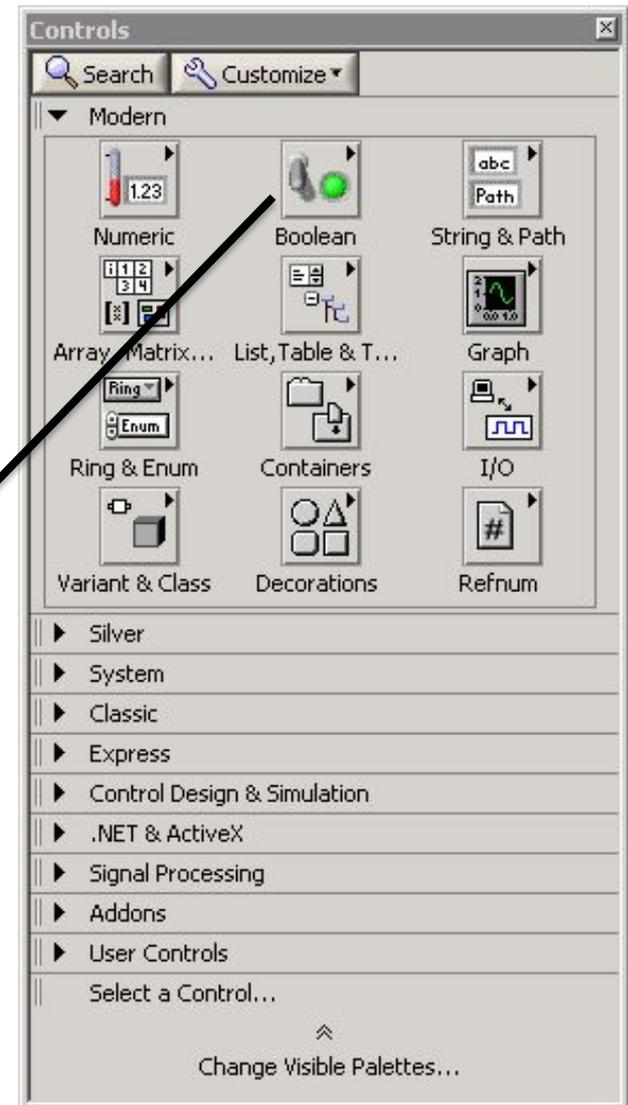
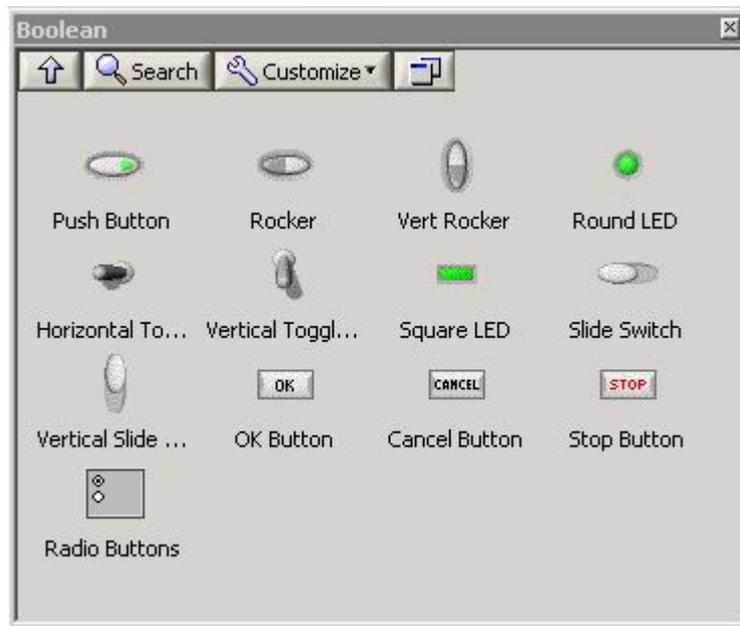
# Палитра Элементов (Controls)

**Подпалитра Числовые элементы (Numeric).**  
Элементы подпалитры используются в качестве источников или приемников числовых данных



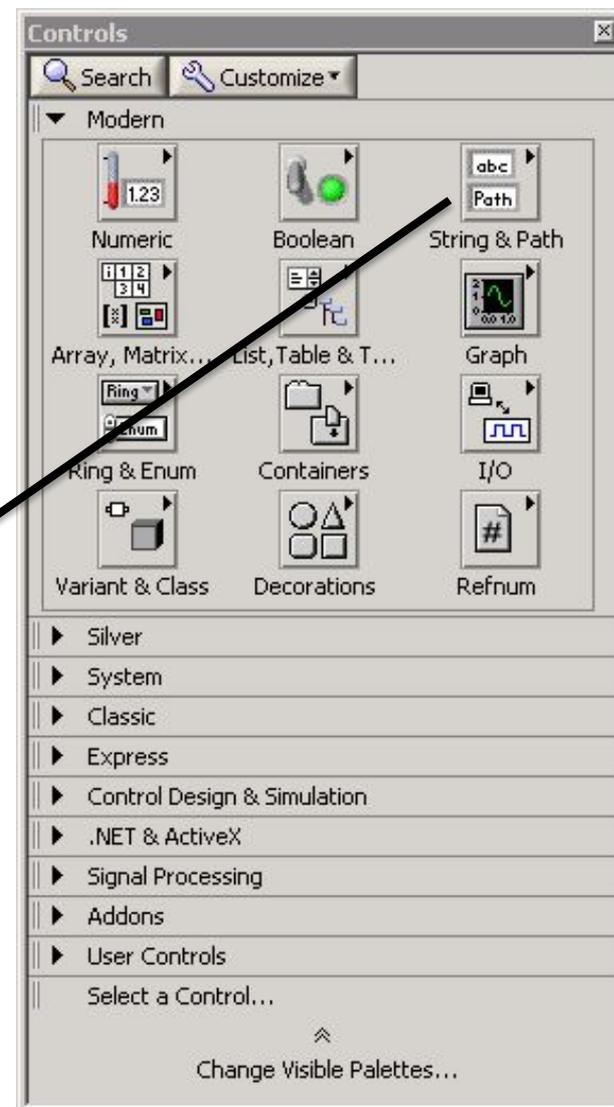
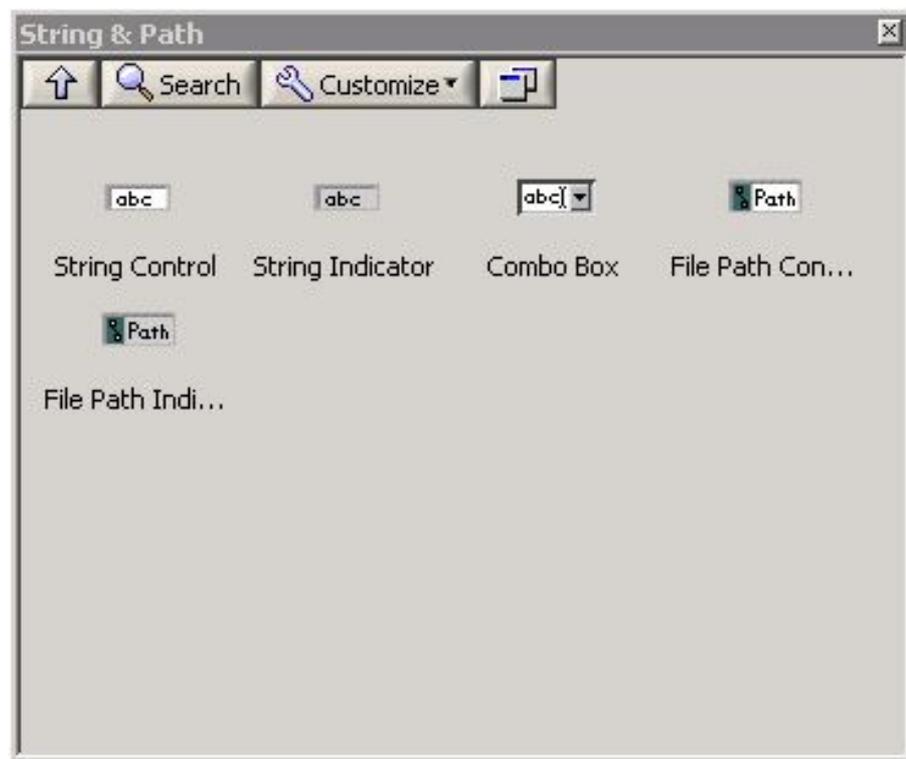
## Подпалитра логических элементов (Boolean).

Подпалитра содержит набор различных переключателей, кнопок и индикаторов, имитирующих действие лампочек и светодиодов. Все элементы могут находиться в двух состояниях, отображающих два состояния логической функции: ИСТИНА (True) и ЛОЖЬ (False)



## Подпалитра Строка и путь (String & Path)

Элементы подпалитры представляют типы данных, которые содержат последовательность букв, СИМВОЛОВ, МАССИВОВ

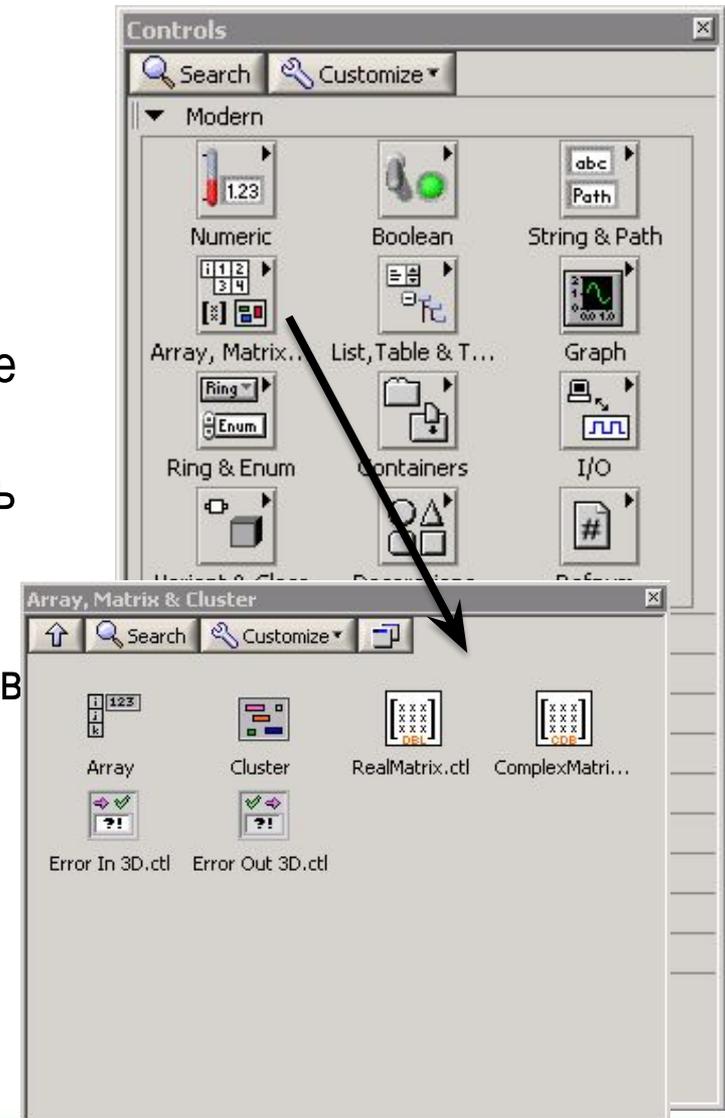


## Подпалитра Массив, матрица и кластер (Array, matrix & cluster)

Подпалитра содержит структуры, которые позволяют создавать массивы или кластеры элементов.

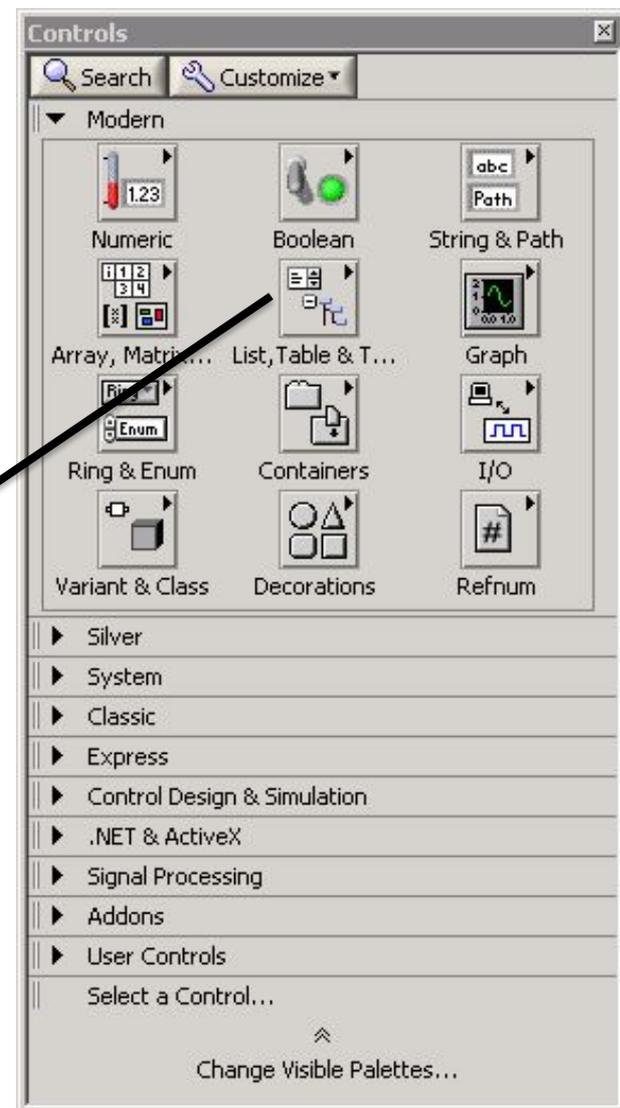
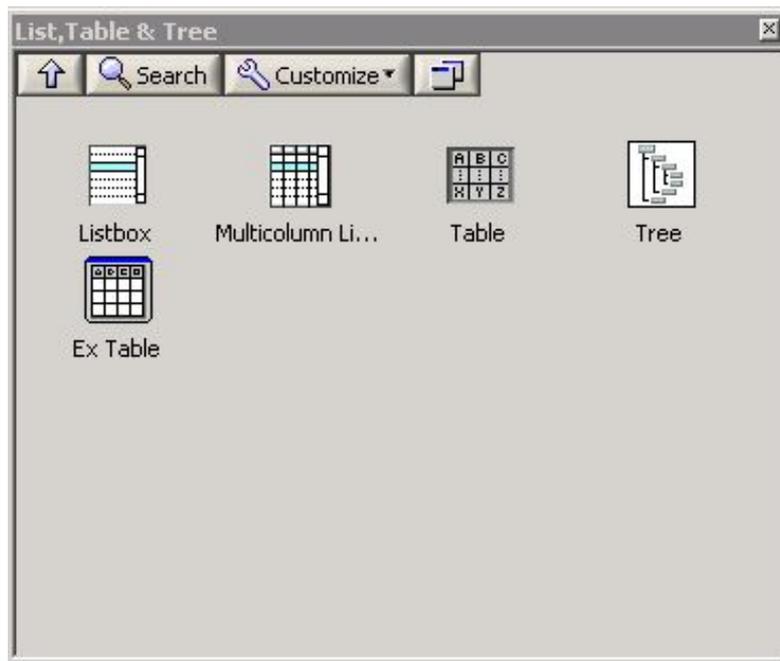
Массивы и кластеры представляют упорядоченное множество элементов соответственно одного или различных типов. Элементами массива могут быть числовые или логические элементы, строки или кластеры. Тип элементов массива определяется типом данных, помещаемых из палитры элементов в шаблон массива.

Матрицы группируют строки и столбцы действительных или комплексных чисел для выполнения операций линейной алгебры.



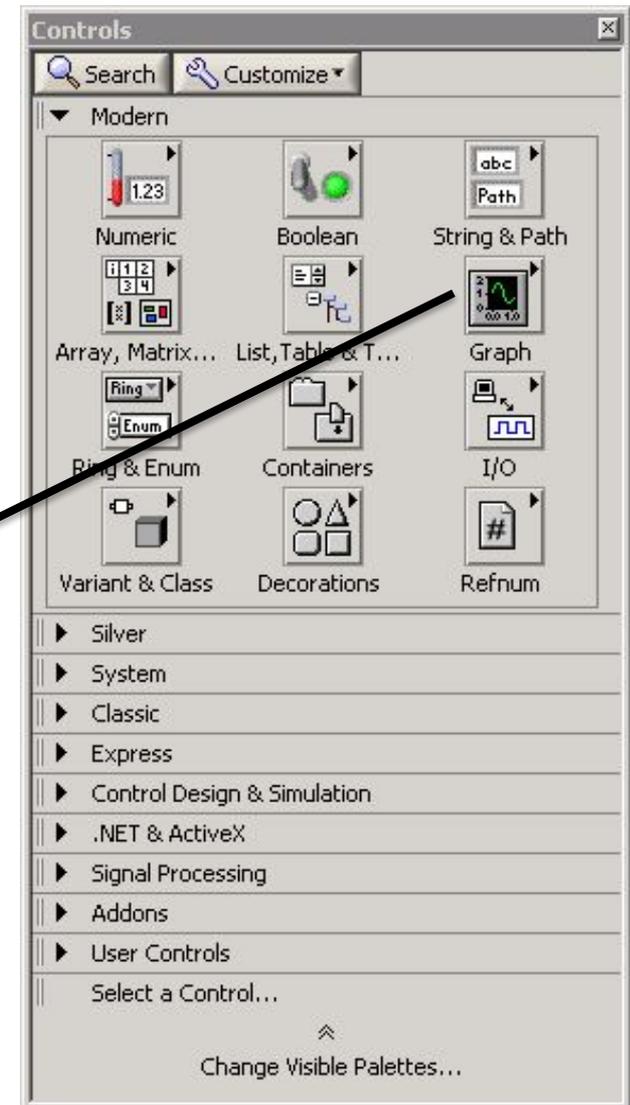
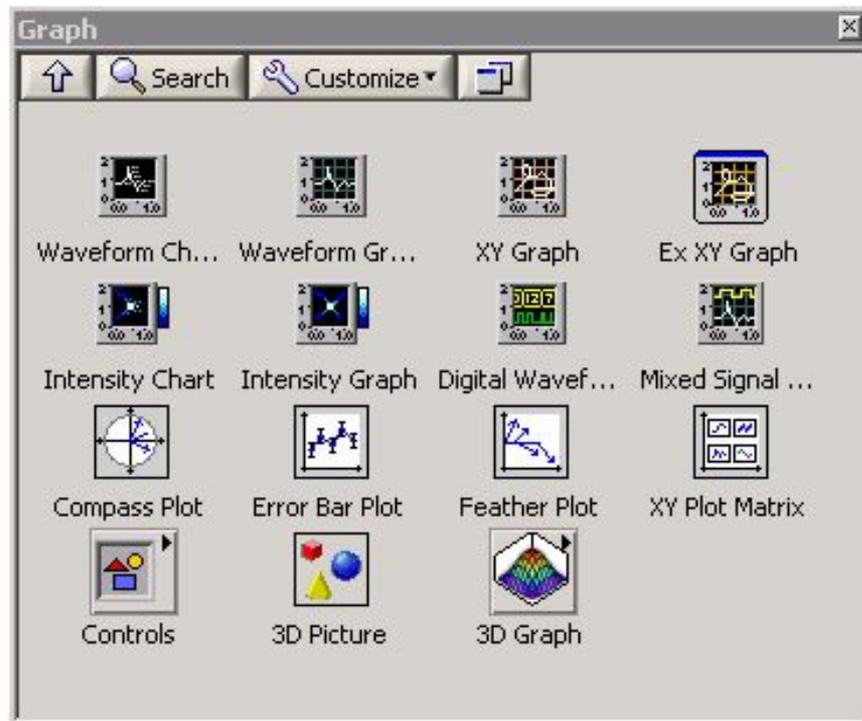
## Подпалитра Список и таблица (List & Table).

Элементы подпалитры представляют собой управляющие или управляемые элементы, позволяющие заносить или отображать буквенную, символьную и цифровую информацию в виде набора строк или ячеек.



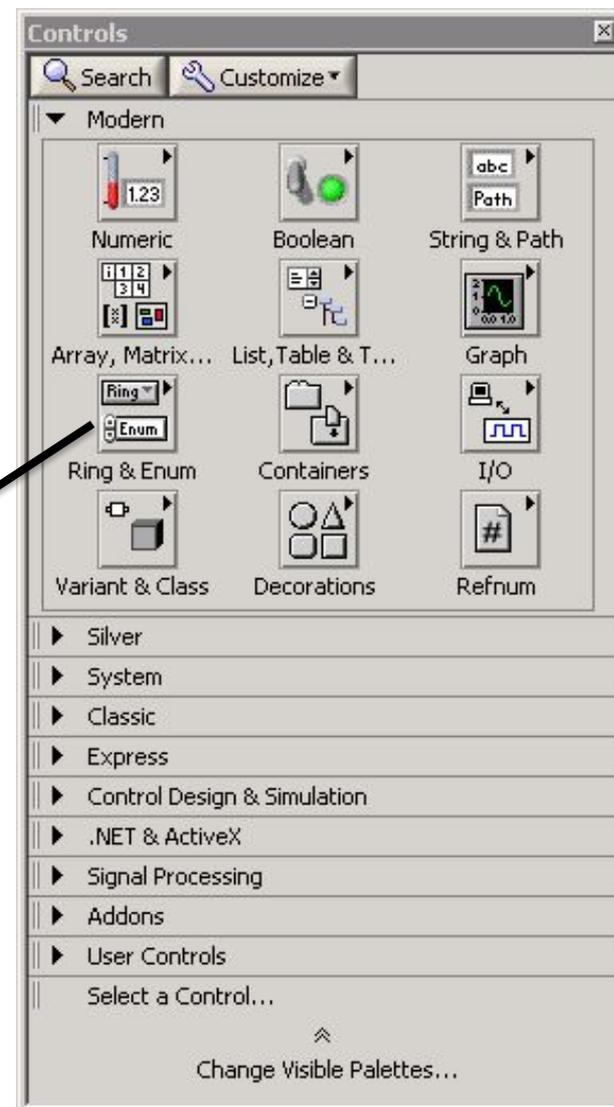
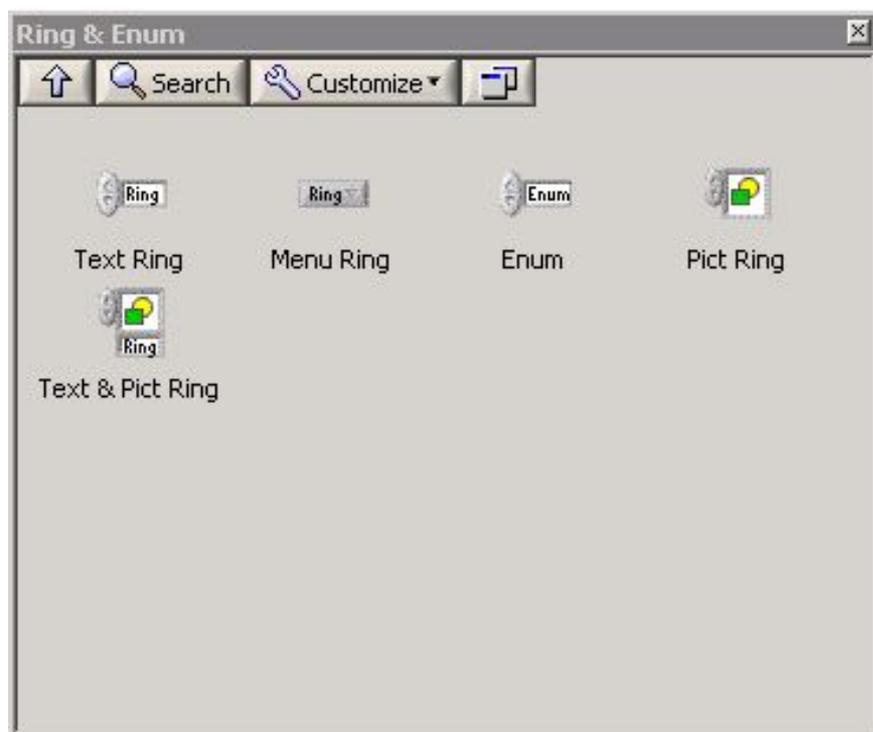
## Подпалитра График (Graph).

Подпалитра содержит набор объектов, которые применяются для отображения временных или функциональных зависимостей реальных или расчетных сигналов.



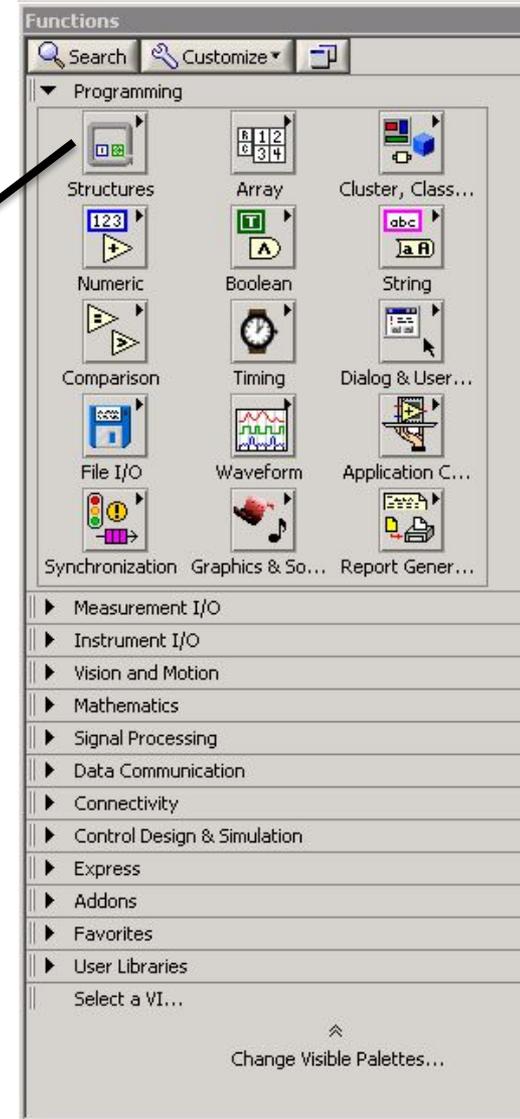
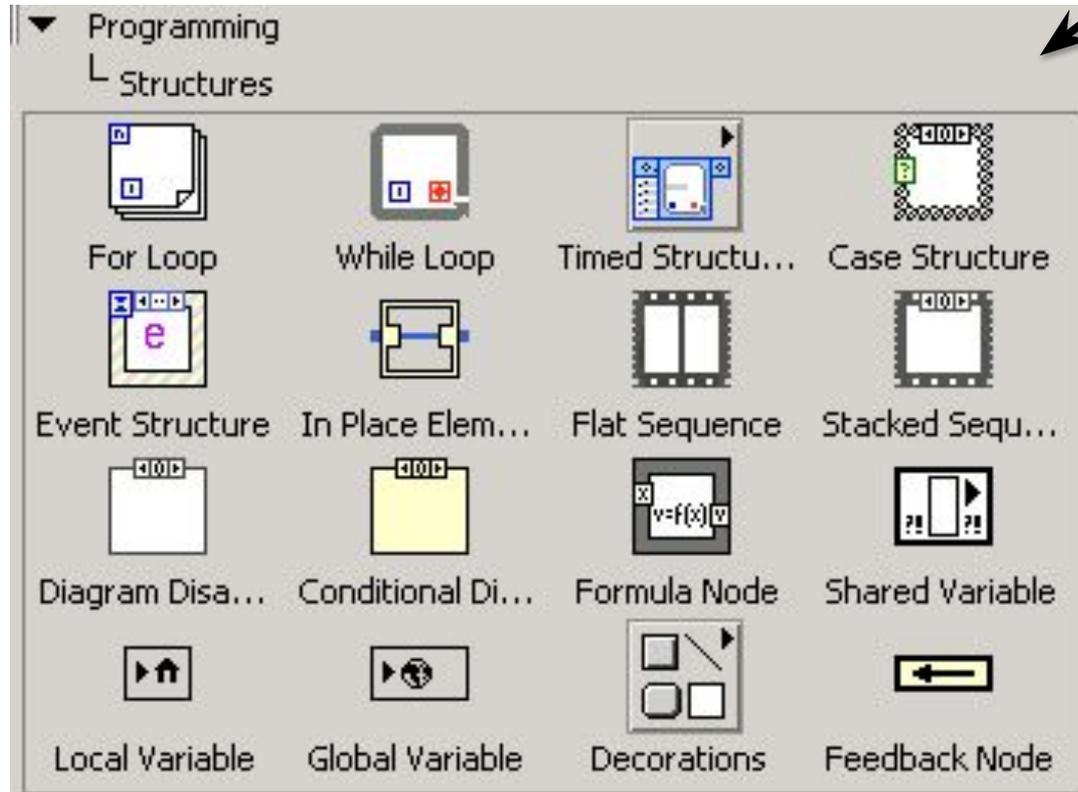
## Подпалитра Выпадающий список и перечень (Ring & Enum).

Элементы подпалитры используются для создания выпадающего меню или списка из которых выбирается нужный элемент при работе программы.

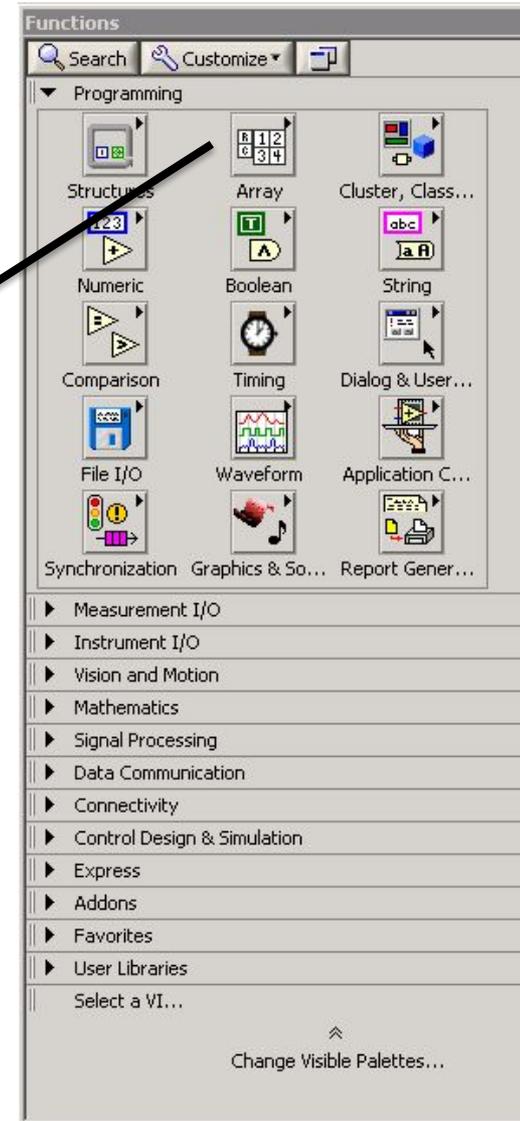
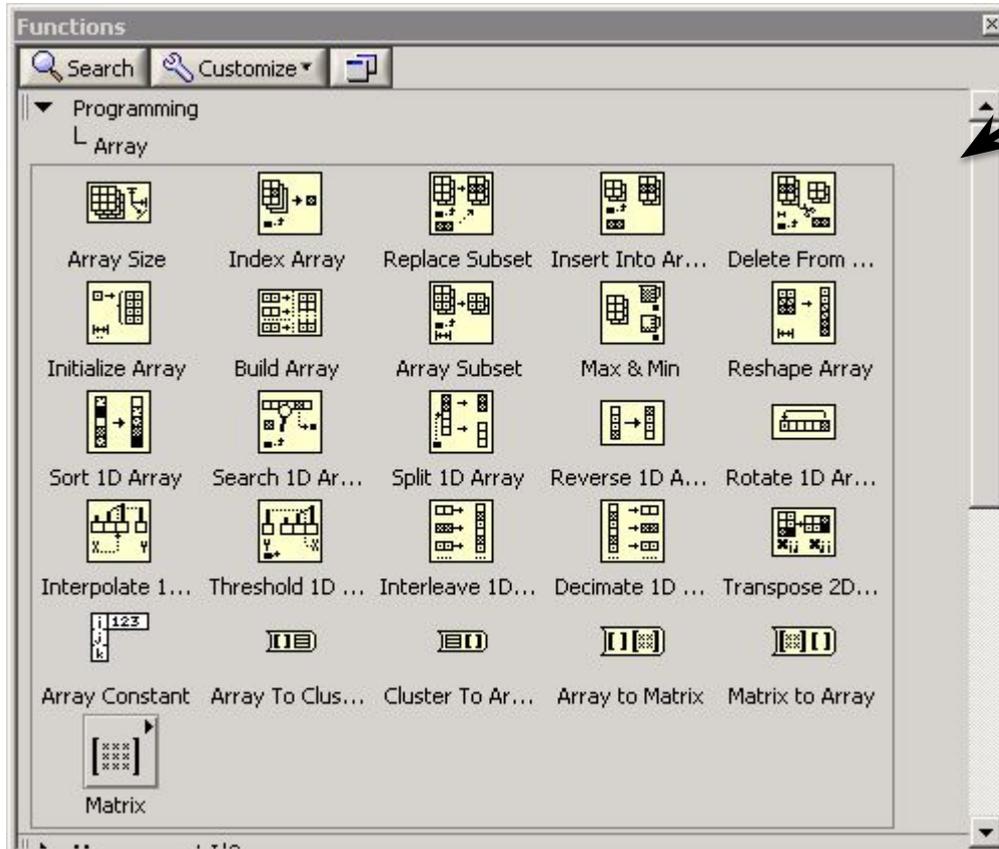


# Палитра функций (Functions)

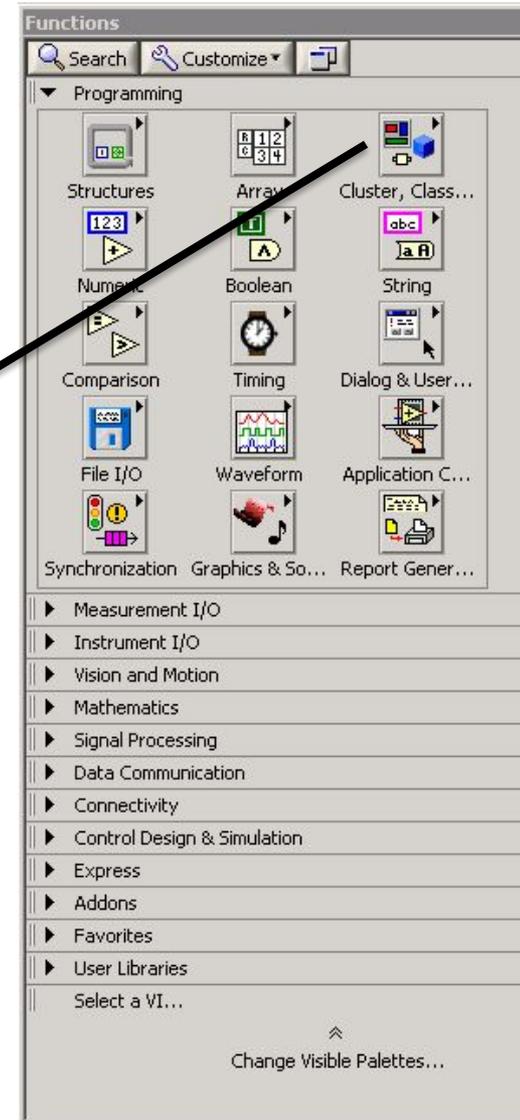
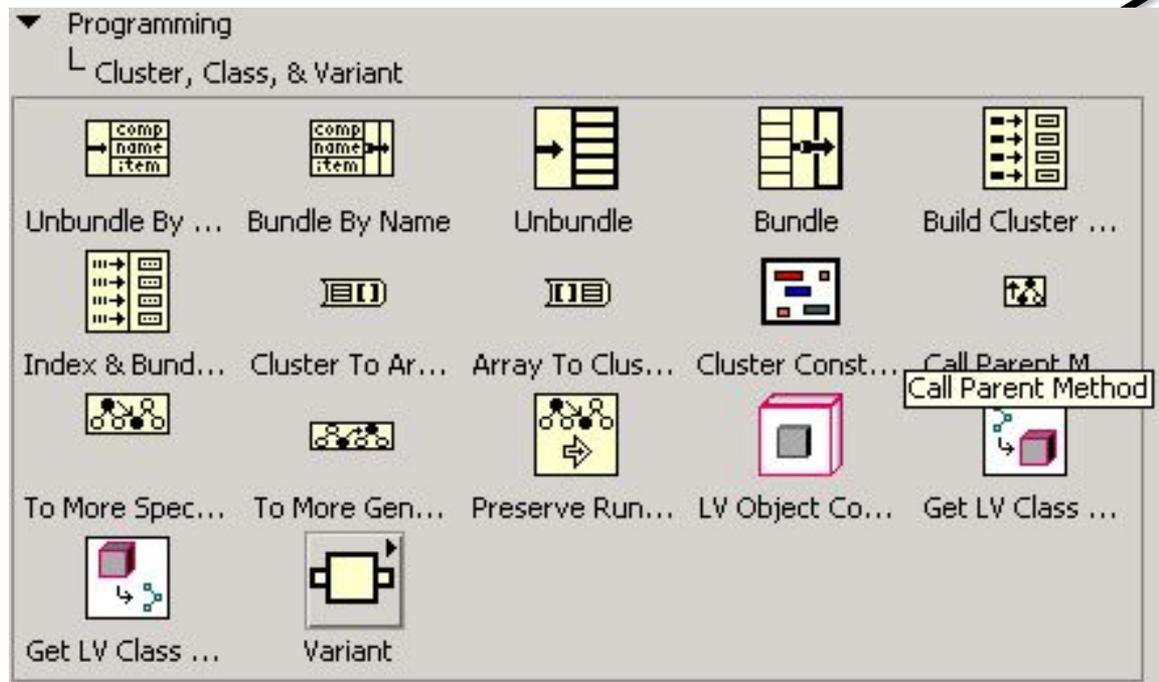
Подпалитра Структуры (Structures) содержит набор структур, управляющих выполнением ВП.



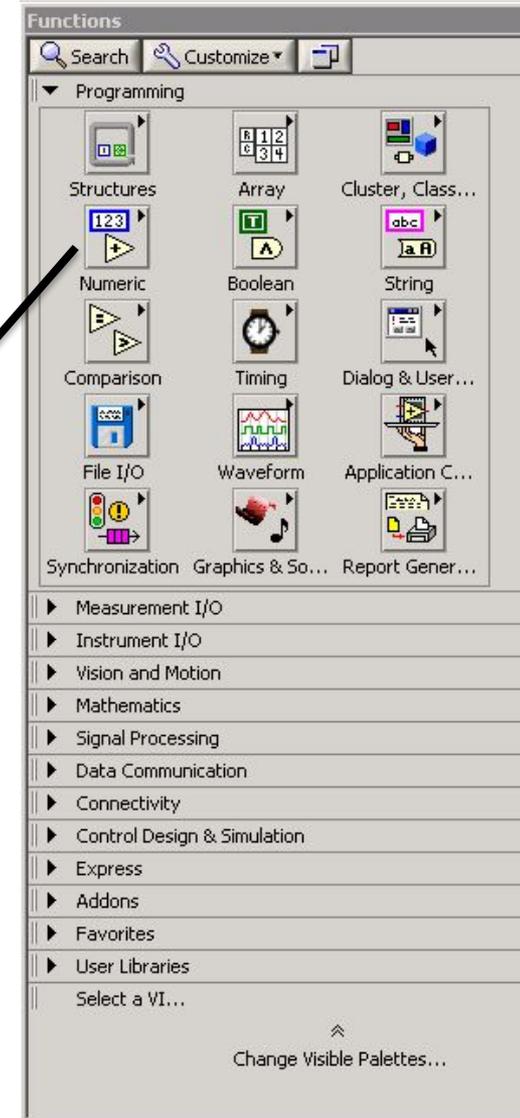
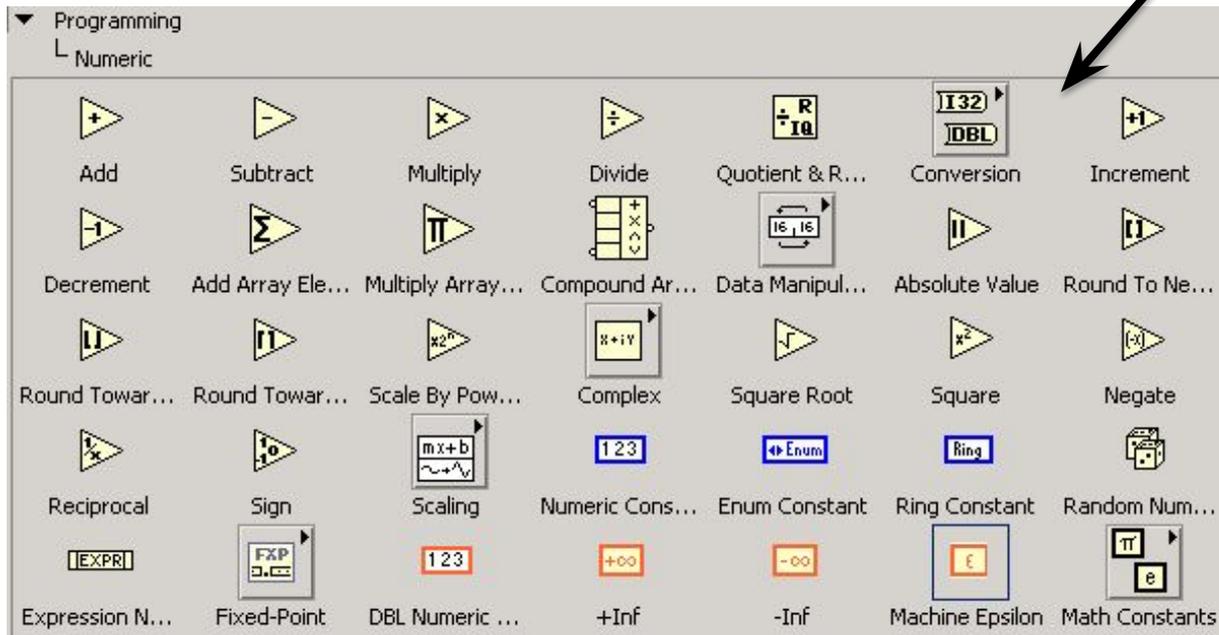
**Подпалитра Массив (Array)** включает функции выделения или замены элементов массивов, выделения подмассивов, транспонирования, сдвига, изменения размерности и т. п.



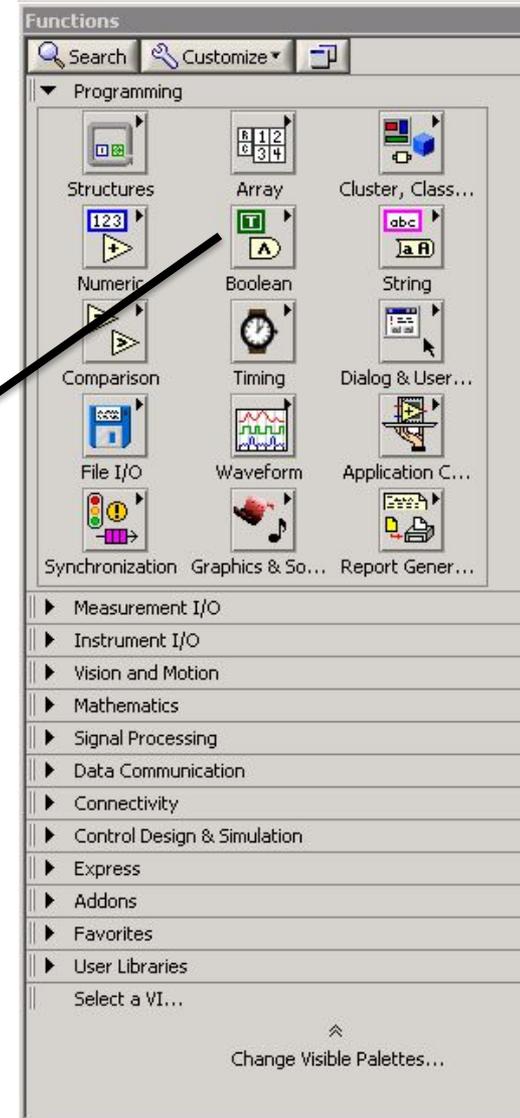
**Подпалитра Кластер Класс и Вариант (Cluster Class & Variant)** включает функции формирования кластеров и массивов кластеров, выделения элементов из кластеров, а также кластерную константу.



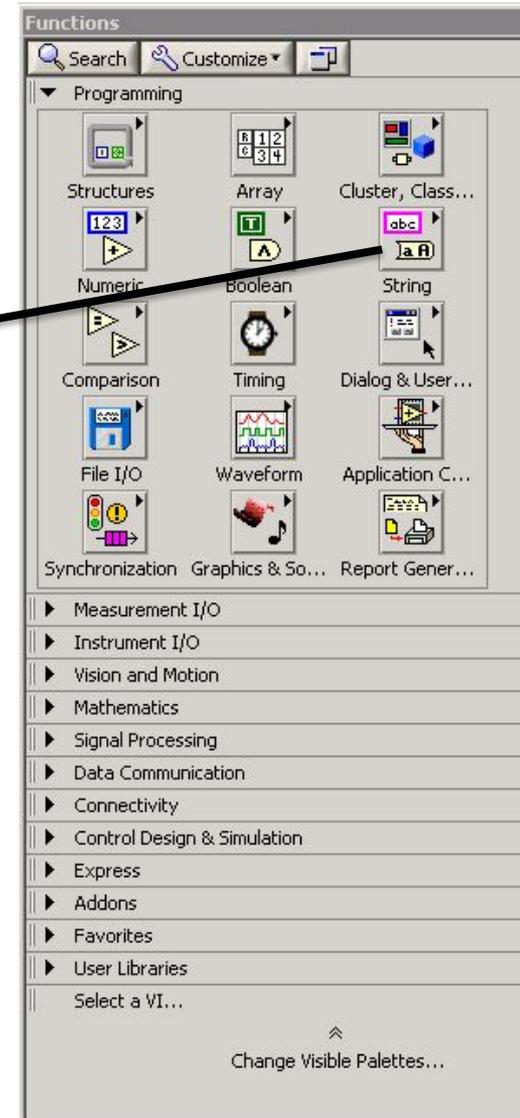
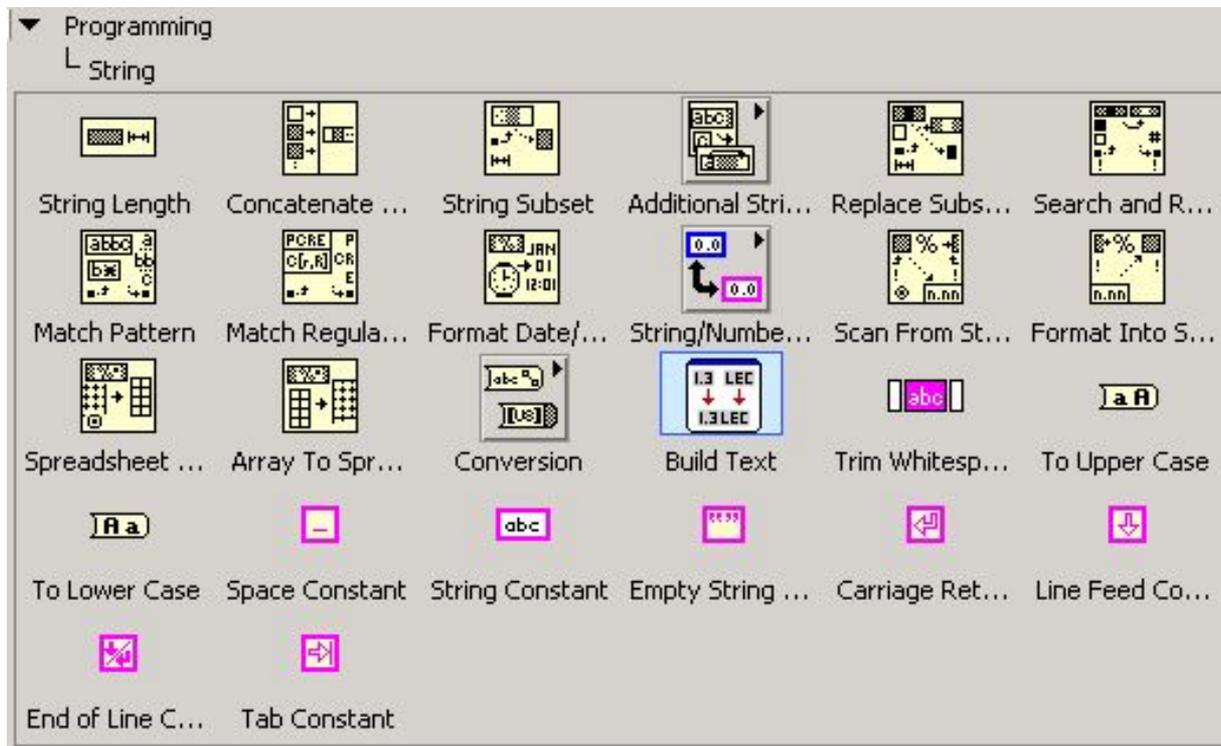
Подпалитра Числа (Numeric) содержит полный набор математических функций, функций преобразования форматов чисел и набор констант.



Подпалитра Логика (Boolean) включает набор функций И (AND), ИЛИ (OR), Исключающее ИЛИ (Exclusive Or), НЕ (Not) и логические константы.

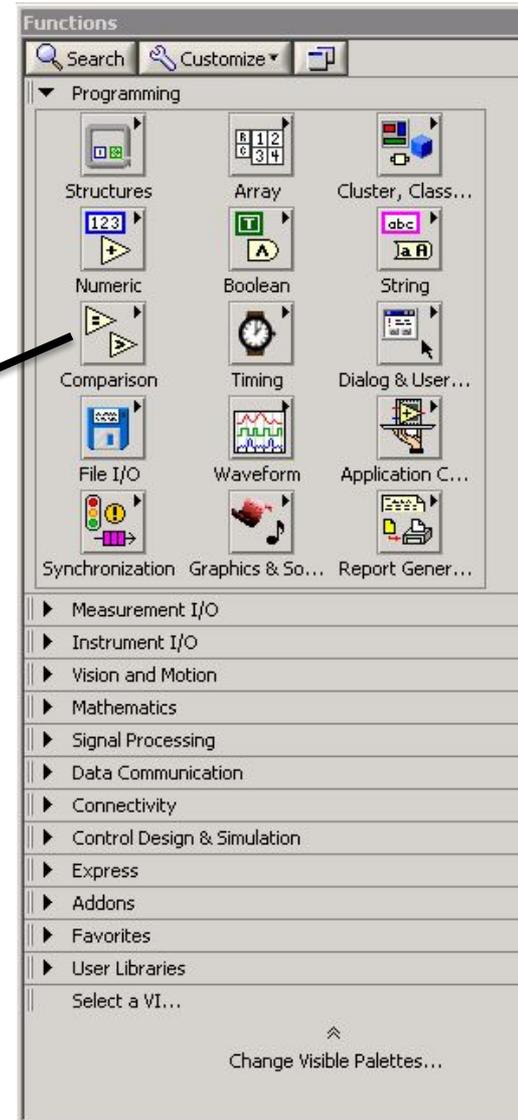


**Подпалитра Строка (String)** содержит ряд функций обработки строковых переменных, функции взаимного преобразования чисел и строк, а также строковые КОНСТАНТЫ.

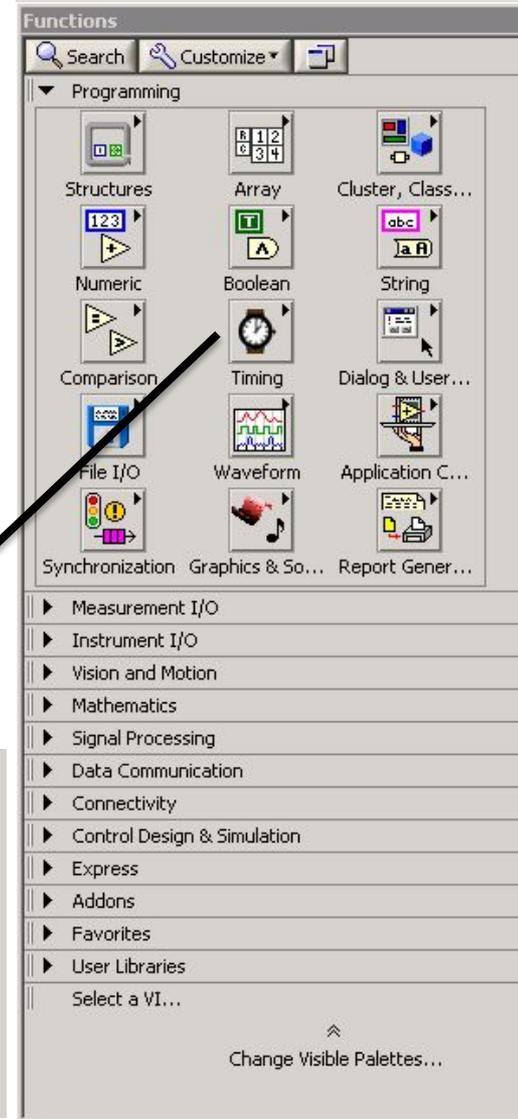


## Функции подпалитры Сравнение (Comparison)

формируют логическую переменную в зависимости от результата сравнения входных переменных или позволяют определять соотношение чисел, нахождение числа в заданном диапазоне, тип числа или тип символа.



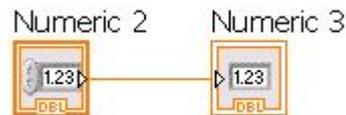
Функции подпалитры **Время (Timing)** позволяют задавать или определять временные интервалы или текущее время.



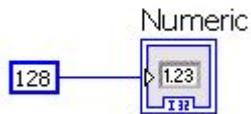
# Основные типы данных

## *Numeric* (численный тип)

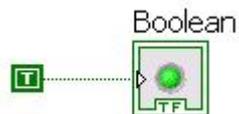
> Floating point - число с плавающей запятой, отображается в виде оранжевых терминалов. Может быть представлено в виде single (32-bit), double (64-bit) или extended (128-bit) precision (с одиночной, двойной или расширенной точностью). Число с плавающей запятой может быть комплексным;



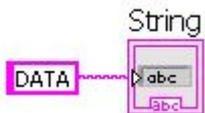
> Integer - целочисленный тип, отображается в виде синих терминалов. Возможны три представления целых чисел: 8, 16 и 32 бита. Один бит может использоваться для знака числа, если это число является знаковым целым.



**Boolean** - логический тип, отображается в виде зеленых терминалов. Логический тип может принимать только два значения: 0 (FALSE) или 1 (TRUE).



**String** - строковый тип, отображается в виде розовых терминалов. Строковый тип данных содержит текст в ASCII формате.



# Подпрограммы в LabVIEW (Виртуальные подприборы)

Большие возможности и удобство использования LabVIEW обусловлены модульным принципом его программ. Создавая и применяя подприборы (ВПП), можно строить ВП модуль за модулем.

Подприбор представляет собой ВП, используемый в другом ВП (или вызванный другим ВП).

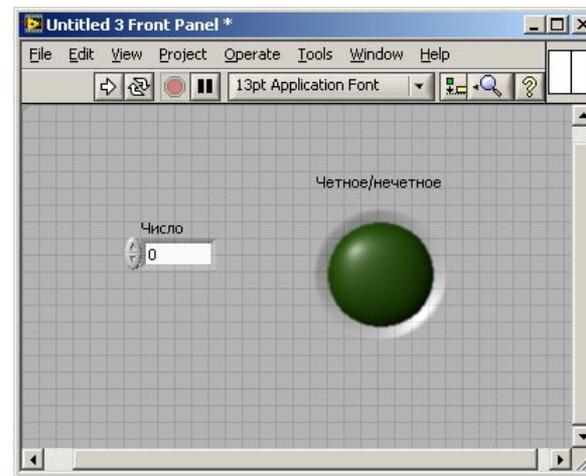
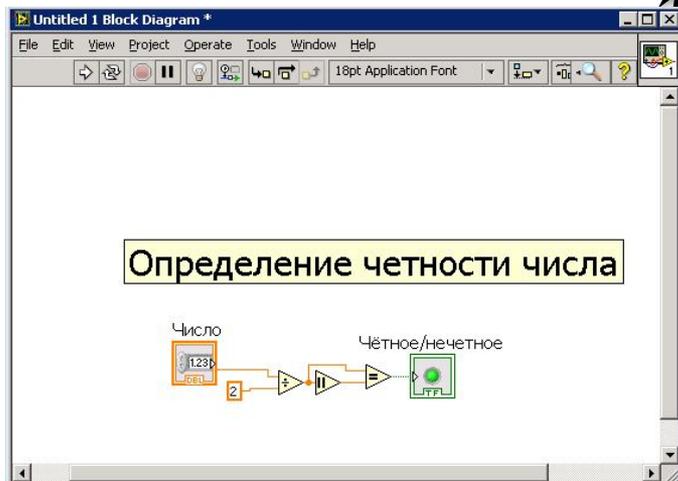
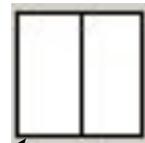
Узел данных ВПП состоит из:

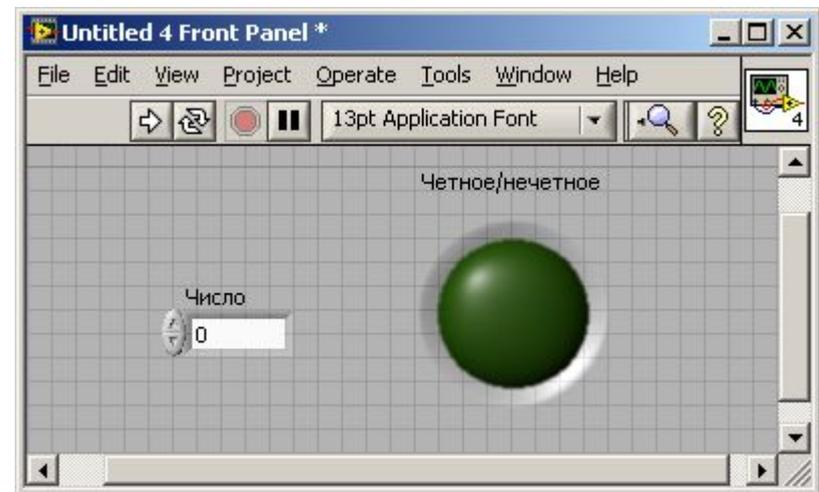
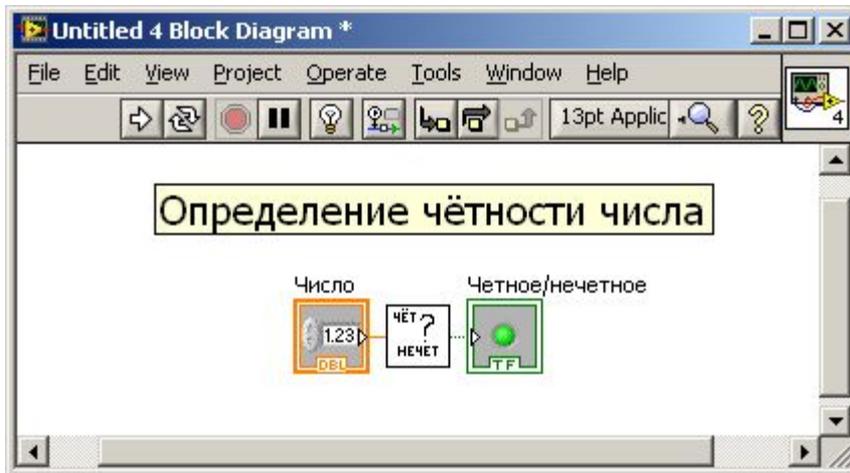
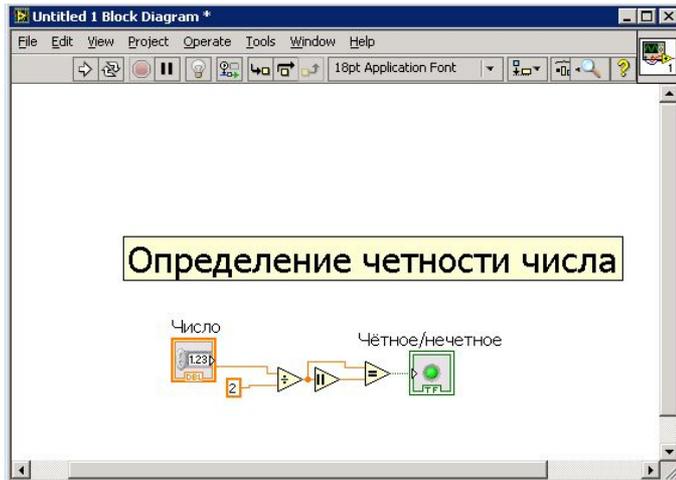
- Иконки;
- Соединительной панели.

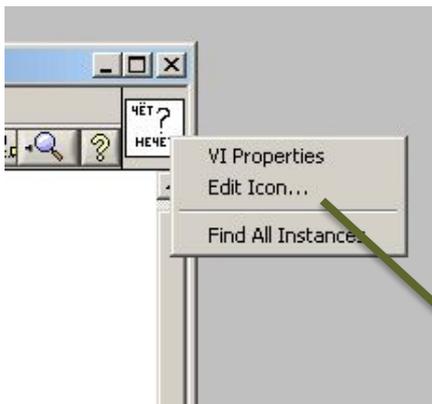


Иконка

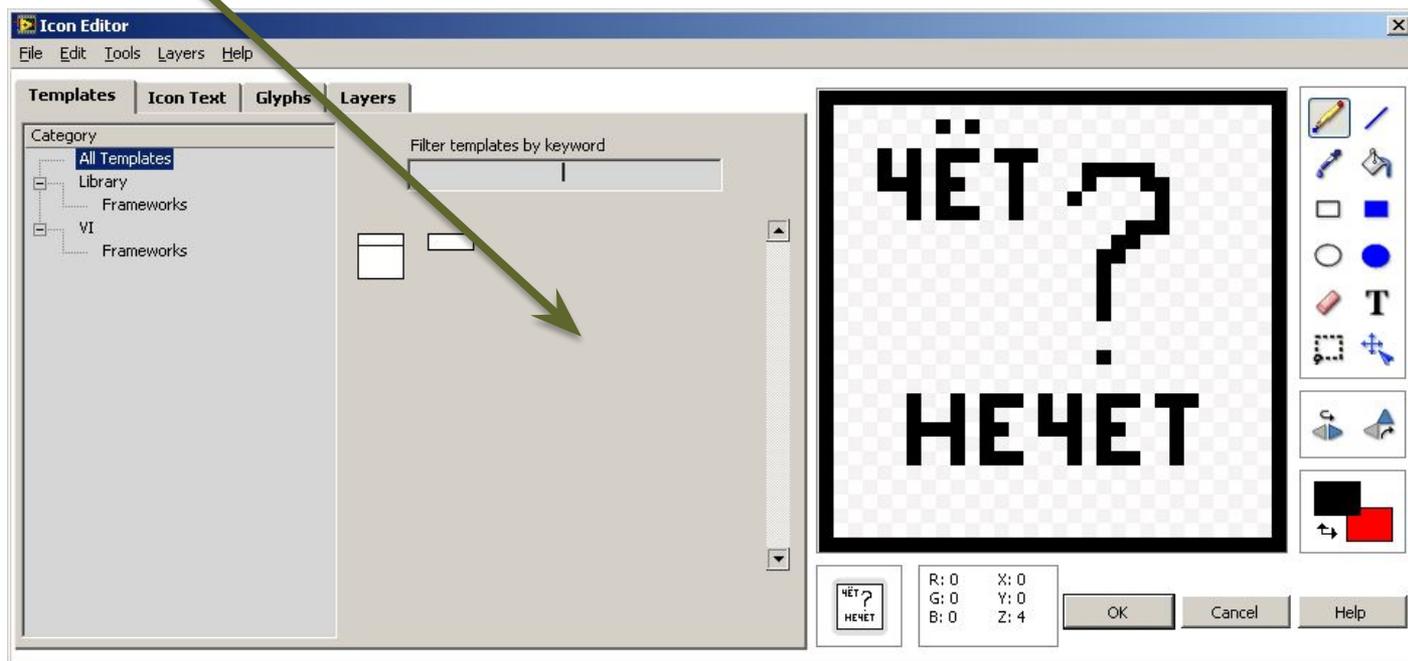
Соединительная панель

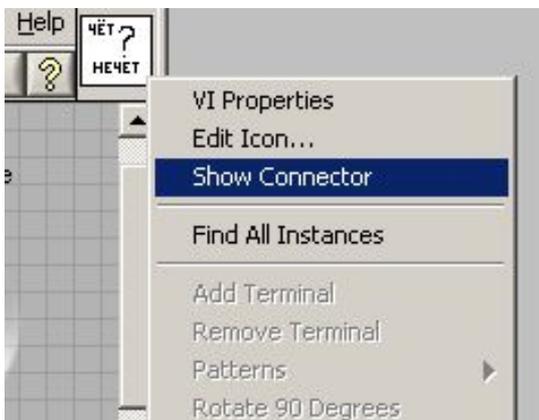






Для изменения иконки ВПП, нужно выбрать пункт Редактировать иконку (Edit Icon...) из контекстного меню иконки в верхнем правом углу окна блок-диаграммы ВП.





Для изменения соединительной панели ВПП, нужно выбрать пункт Показать соединение (Show Connector) из контекстного меню иконки в верхнем правом углу окна лицевой панели ВП.

Далее из контекстного меню выбирается нужная схема соединения (Patterns)



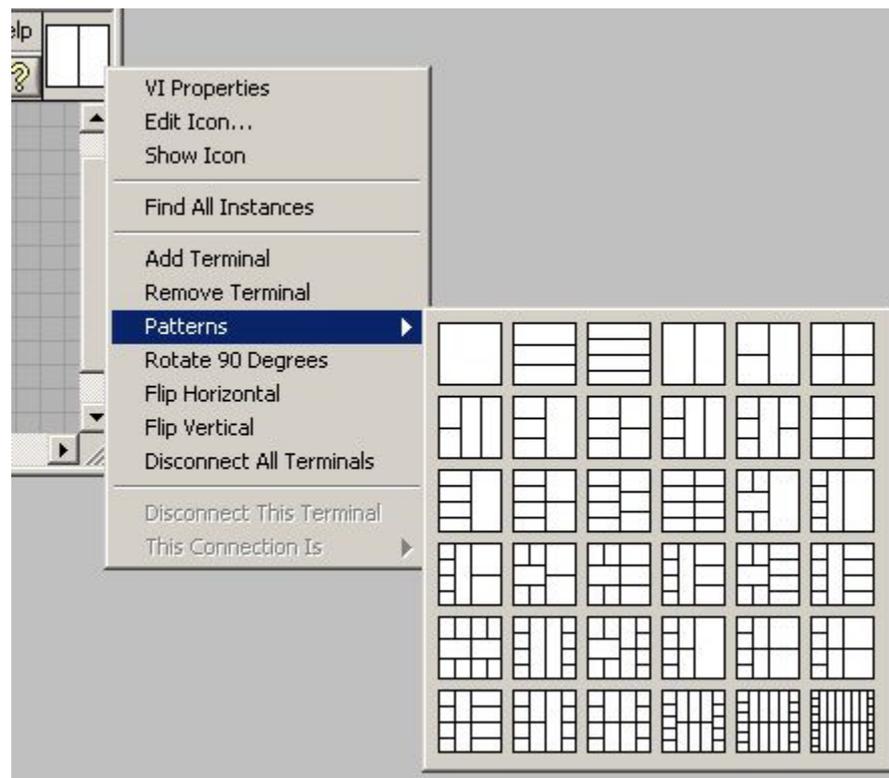
Один вход - один выход



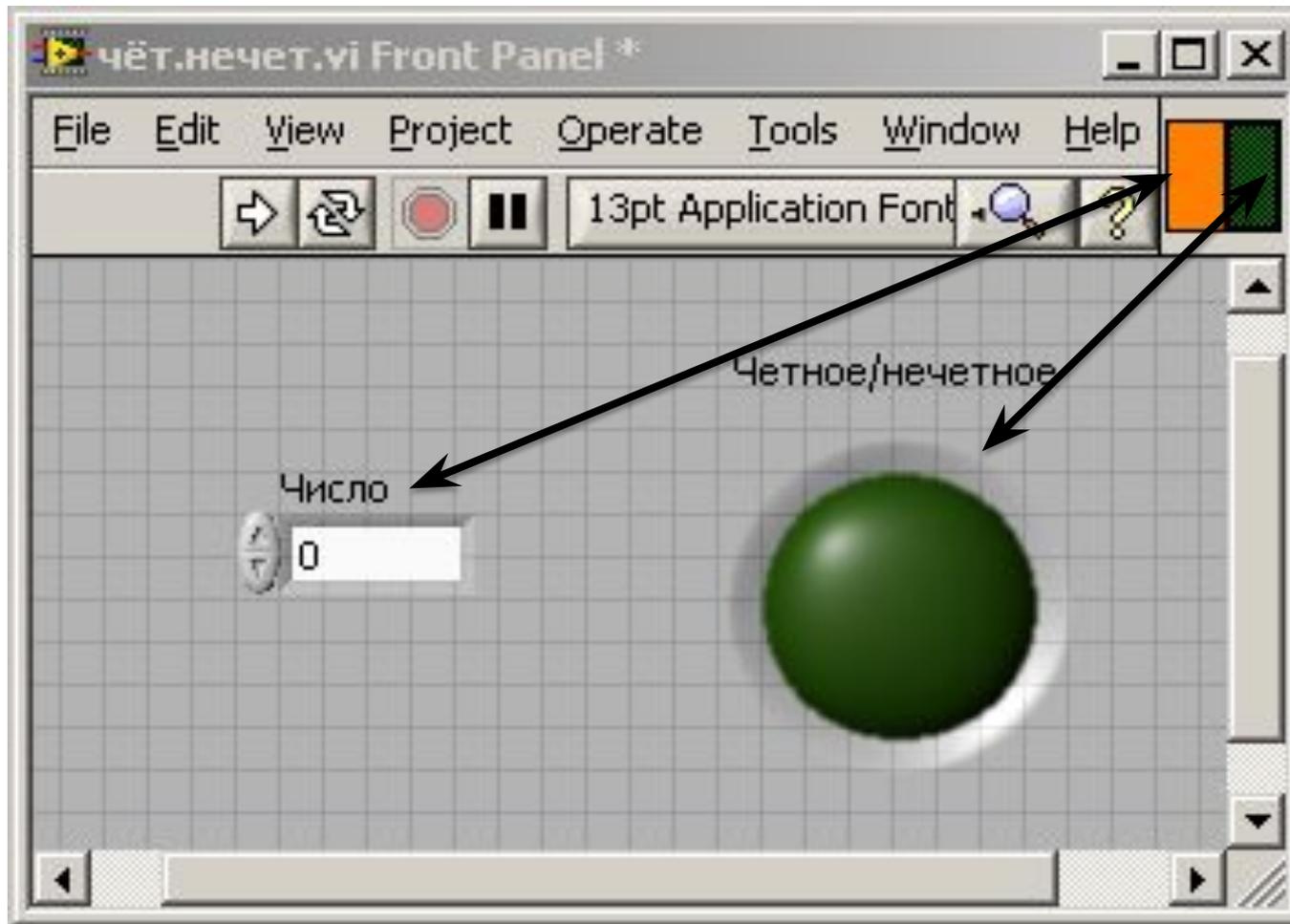
Два входа – один выход



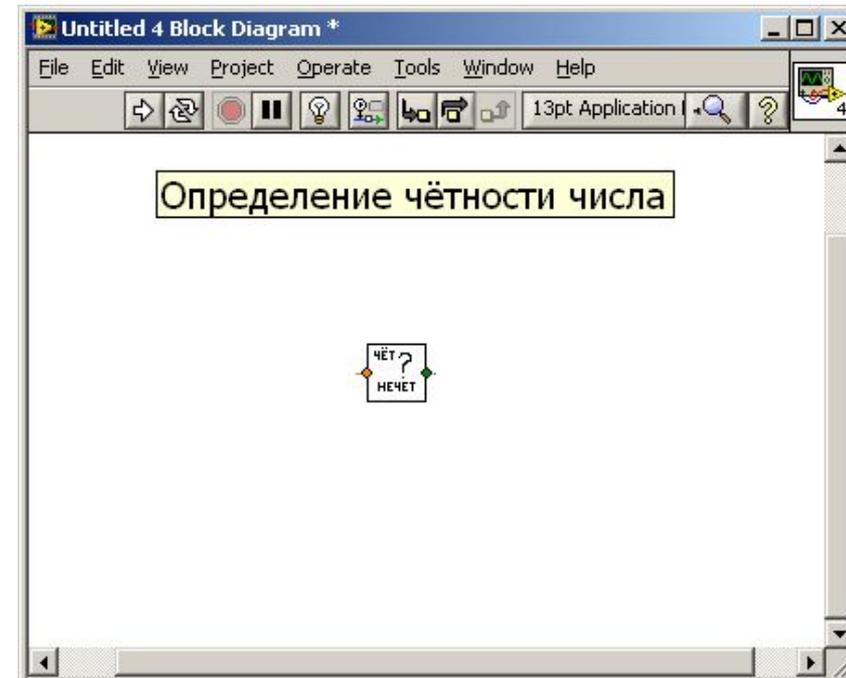
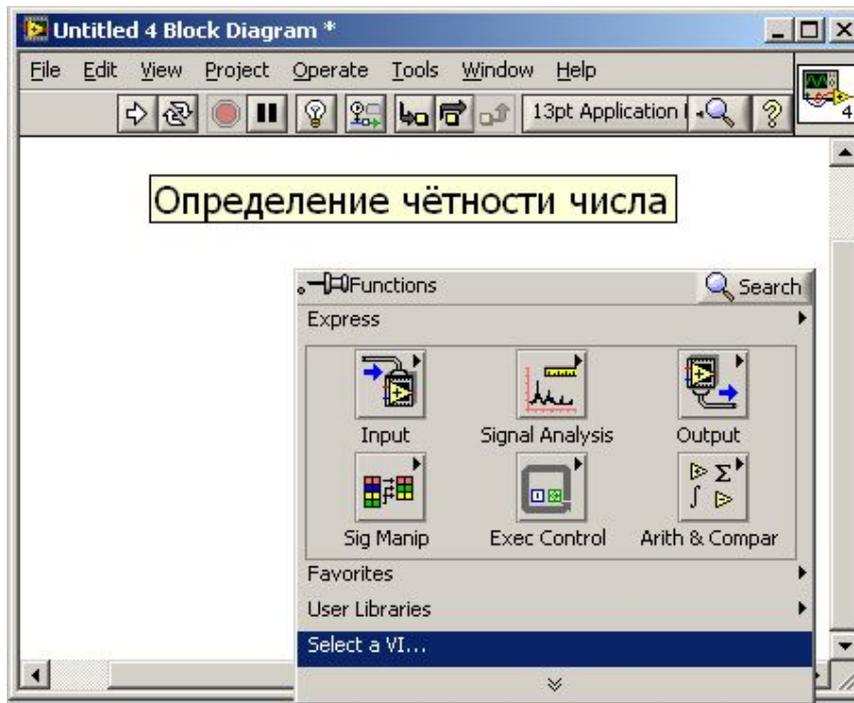
Два входа – два выхода и т.д.



Далее последовательно определяются все входы-выходы будущего ВПП и подприбор сохраняется через меню File - Save As...



Для вызова сохранённого ВПП, нужно выбрать из контекстного меню окна блок-диаграмм пункт Выбор ВП (Select a VI) и через открывшееся окно выбрать файл с сохранённым ВПП. Далее ВПП устанавливается в нужное место панели блок-диаграмм и остаётся добавить к нему элементы контроля и индикации.



# Структуры LabVIEW

Структуры – важный тип узла данных – управляют выполнением ВП.

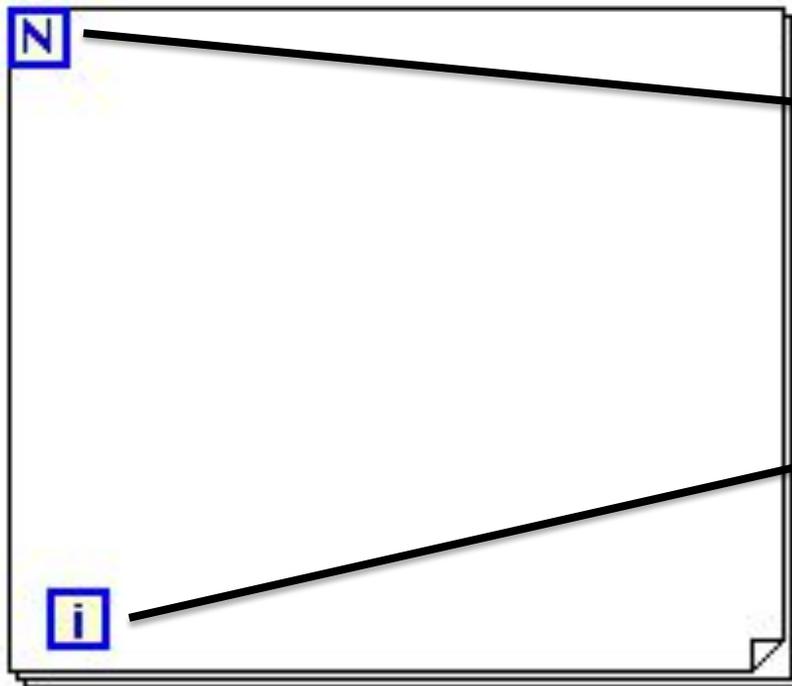
Основные структуры:

- Циклы по условию (For While);
- Циклы с фиксированным числом итераций (For Loop);
- Структуры варианта;
- Структуры последовательности;
- Узел Формула.

## Цикл с фиксированным числом итераций (For Loop)

Цикл выполняет код внутри своих границ (поддиаграмму) некоторое количество итераций (count).

Цикл For находится в палитре Functions>>All Functions>>Structures.



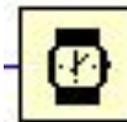
Count(счёт), определяет, сколько раз будет выполняться цикл (нужно подключить, например, константу к этому терминалу).

Счетчик итераций, начальное значение всегда 0 (на первом шаге цикла), в дальнейшем при каждом проходе увеличивается на 1)

Пример программы, выводящей на лицевую панель ВП случайные числа от 0 до 1 с частотой в 1 секунду. Количество цифр для вывода – 10.



Генератор случайных чисел от 0 до 1  
(Function palette>> Programming>> Numeric>>  
Random Number (0-1))

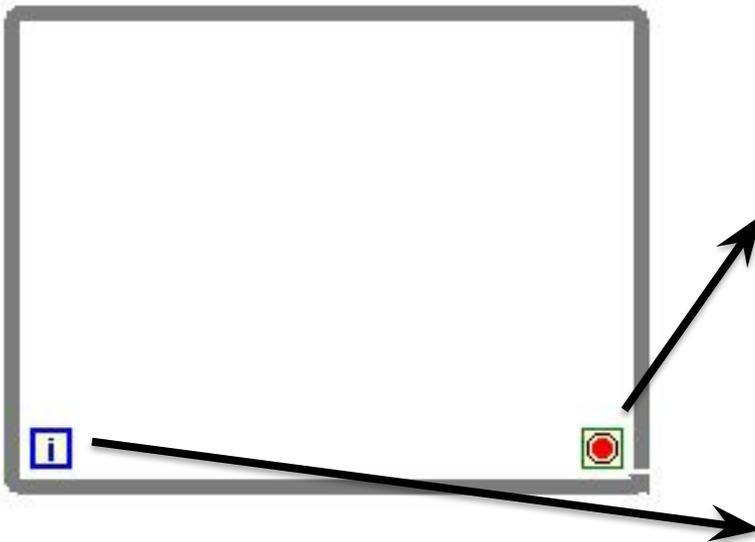


Функция ожидания, ожидает заданное  
количество миллисекунд (Function palette>>  
Programming>> Timing >> Wait(ms))

## Цикл по условию (While Loop)

Цикл выполняет код внутри своих границ (поддиаграмму) до тех пор, пока логическое значение, подключенное к *терминалу условия выхода из цикла (Condition terminal)*, не перейдет в состояние ИСТИНА (ДА – остановить цикл).

Цикл While находится в палитре Functions»Programming»Structures.

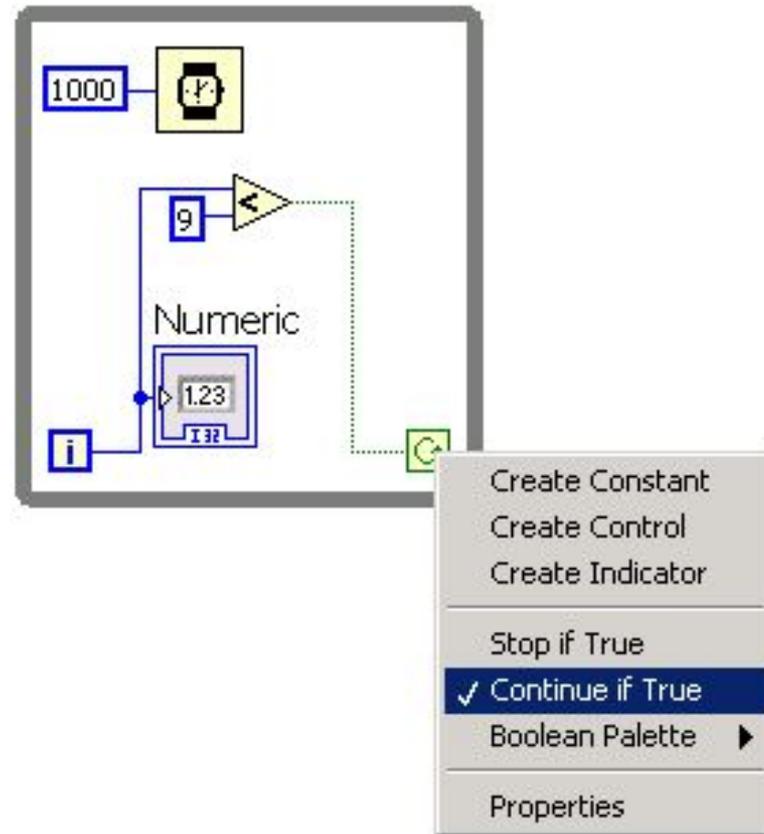
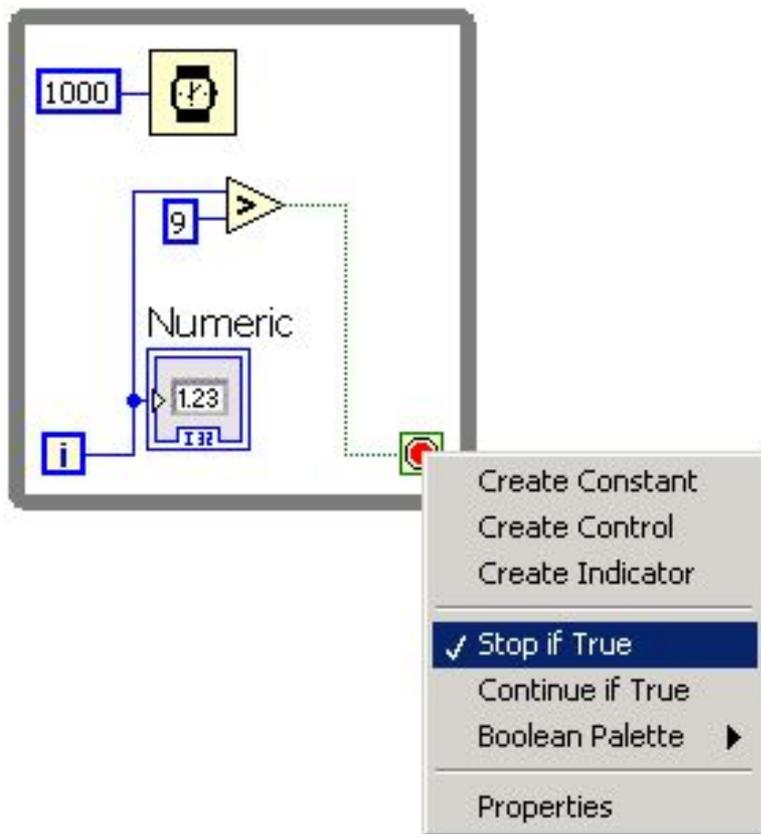


Терминал завершения цикла. Цикл будет выполняться до тех пор, пока этот терминал не получит определенное булевское значение (в данном случае - пока на терминал не поступит true)

Счетчик итераций, начальное значение всегда 0 (на первом шаге цикла), в дальнейшем при каждом проходе увеличивается на 1 (аналогично с циклом For Loop)

Два условия завершения цикла:

- Stop If True (стоп если ИСТИНА)
- Continue if True (продолжить если ИСТИНА)



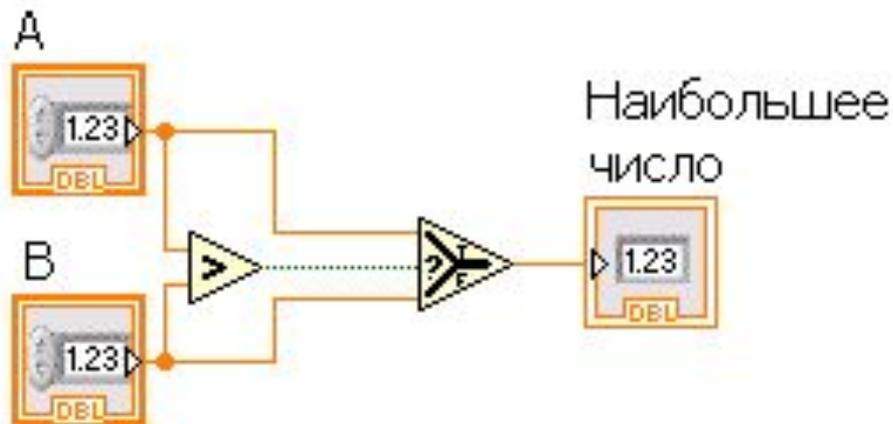
## Структуры варианта (ветвление программы)

При программировании часто ход дальнейшего выполнения программы зависит от какого-либо условия. Например, если случилось А, то делай Б, а если случилось В, то делай Г.



### Functions>>Programming>>Comparison>>Select

Функция выбирает одно из двух значений, в зависимости от того, какое значение (True или False) на третьем входе.



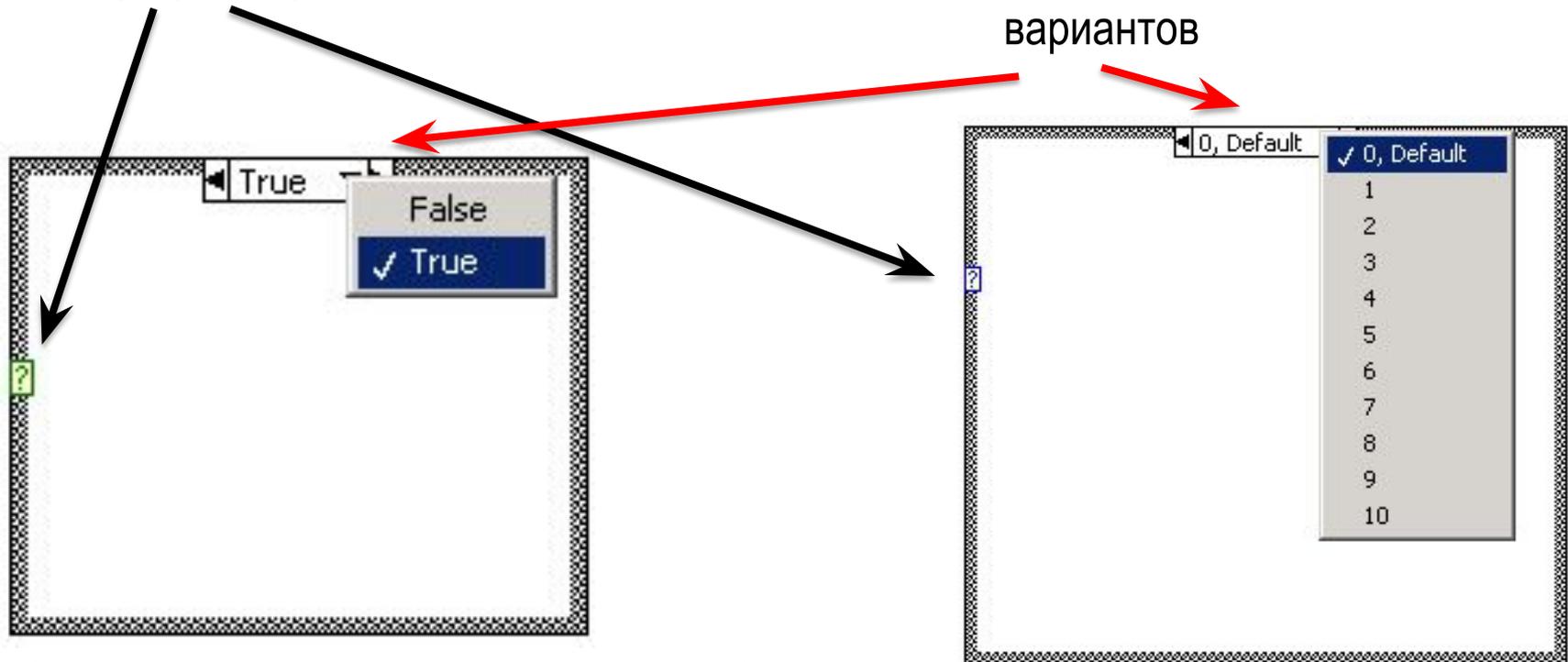
Программа сравнения двух чисел и вывода наибольшего из них. Как видно из примера функция Select имеет два входа для данных и один для условия. После проверки условия на индикатор передается наибольшее число для вывода.

# Case Structures

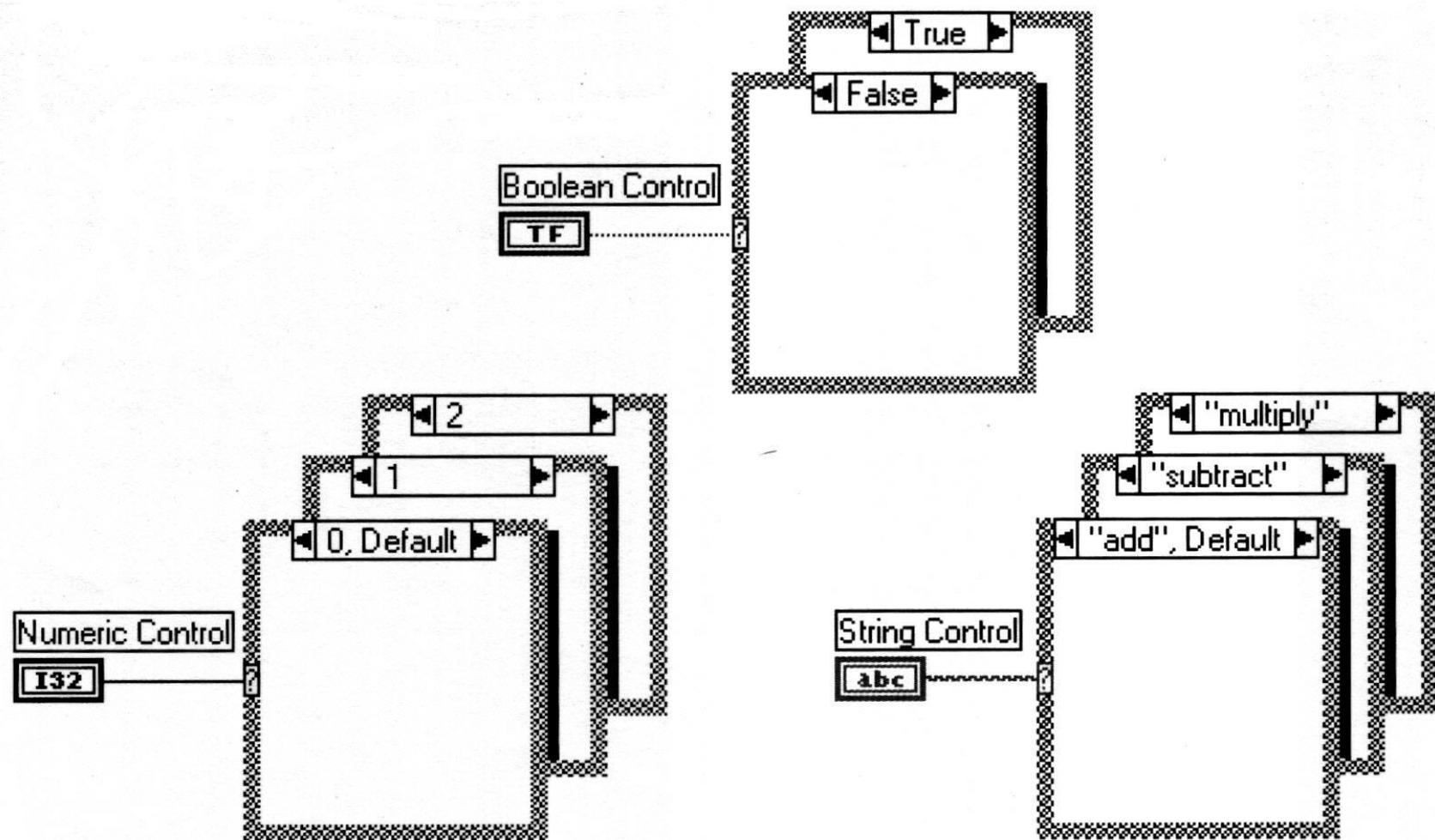
Структура Case позволяет осуществить выбор одного из двух (или более) случаев, и выполнить в зависимости от этого выбора нужную подпрограмму.

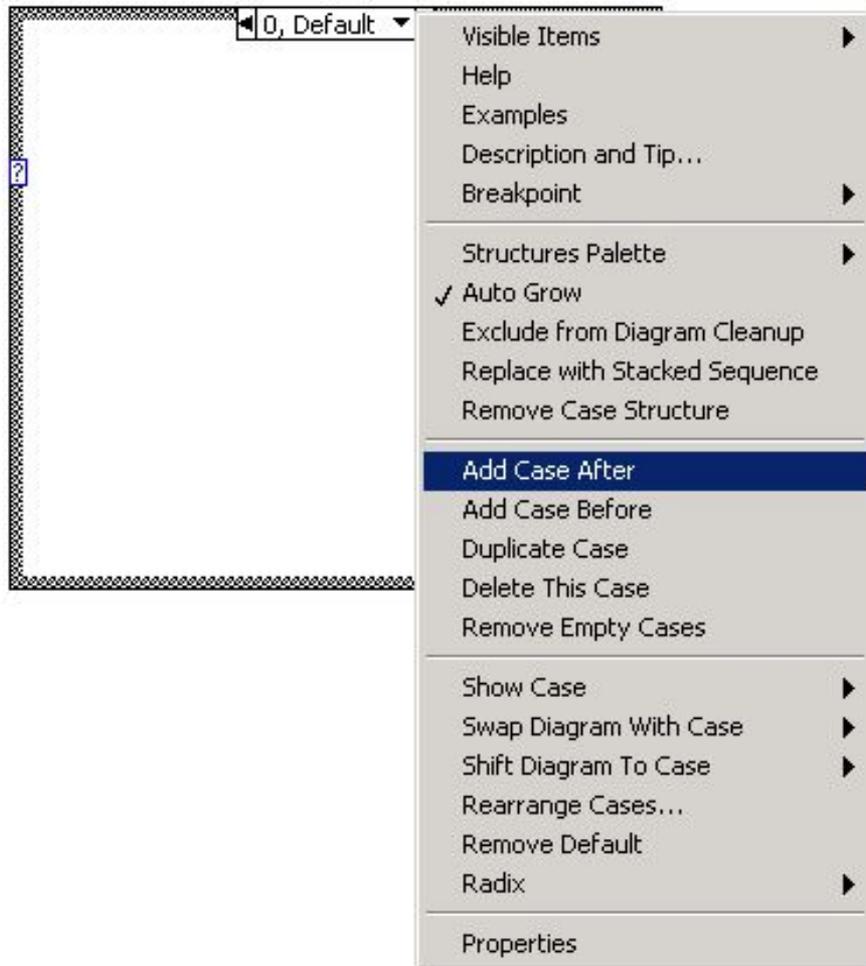
Вход структуры – управление вариантами

Переключатель возможных вариантов



Case structure похожа на конструкцию if...then...else в обычных текстовых языках программирования



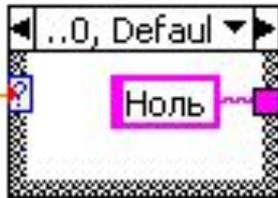


Для добавления дополнительных случаев (вариантов разветвления программы) необходимо выбрать пункт Add Case After или Add Case Before (добавить случай до/после выбранного)

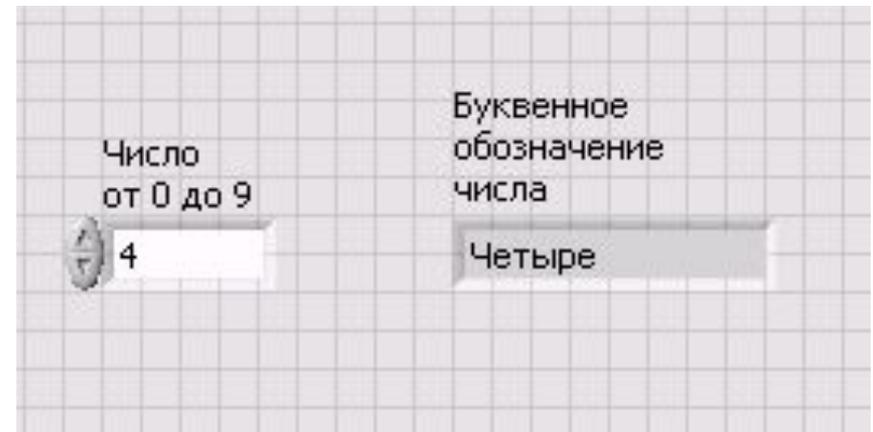
Если при выполнении программы на вход подался сигнал, который не был запрограммирован, то выполняется вариант, выбранный по умолчанию (default)

# Пример перевода чисел в буквенные названия (3 – три, 5 – пять...)

Число  
от 0 до 9

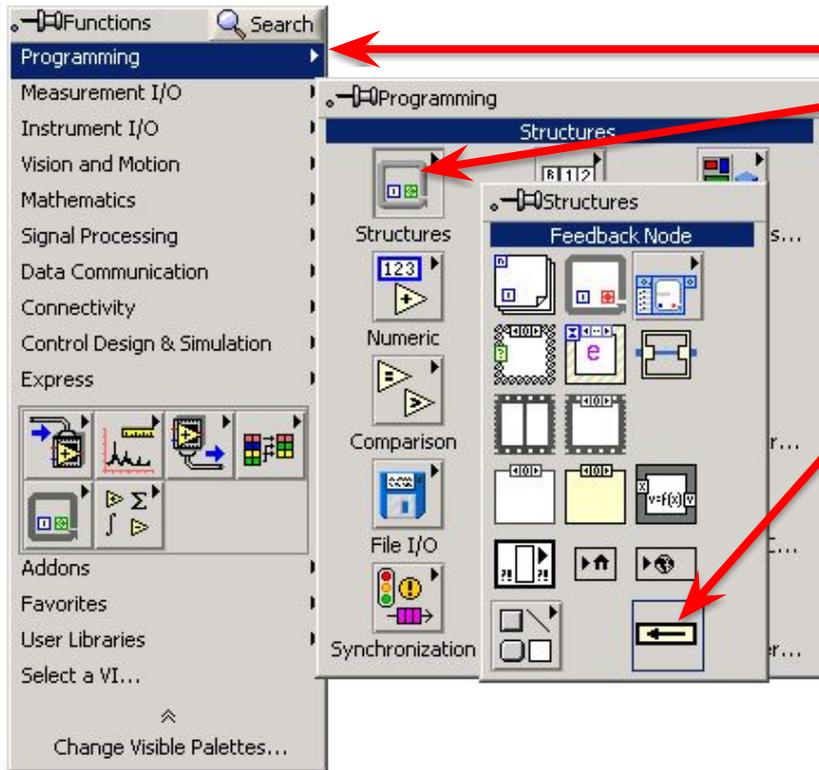


Буквенное  
обозначение  
числа

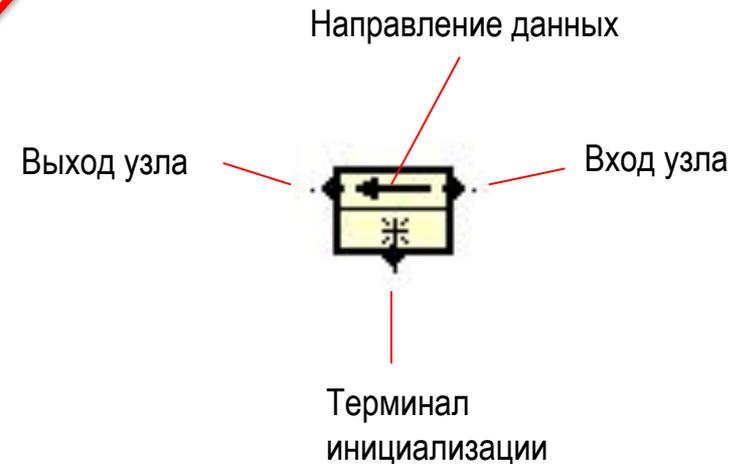


# Узел обратной связи (Feedback Node)

Работает по принципу сдвигового регистра. Автоматически LabVIEW вставляет этот узел, когда выходные терминалы какого-либо элемента соединяются со входными, т.е. происходит зацикливание. Стрелка внутри узла обратной связи показывает направление потока данных.

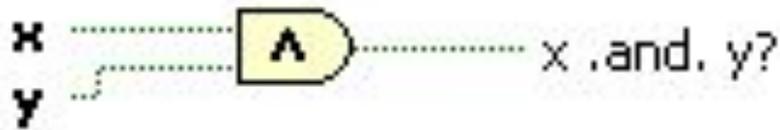


На панели блок-диаграмм из контекстного меню – *Programming – Structures – Feedback Node*



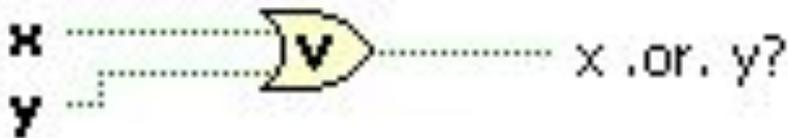
# Логические элементы

## Элемент «И» (AND)



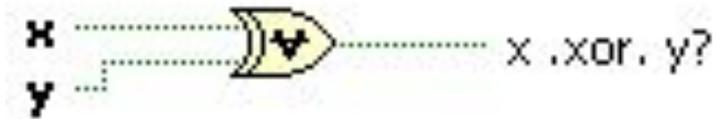
x	y	result
0	0	0
0	1	0
1	1	1

## Элемент «ИЛИ» (OR)



x	y	result
0	0	0
0	1	1
1	1	1

## Элемент «Исключающее ИЛИ» (Exclusive OR)



$x$	$y$	result
0	0	0
0	1	1
1	1	0

## Элемент «И-НЕ» (NOT AND)



$x$	$y$	result
0	0	1
0	1	1
1	1	0

## Элемент «ИЛИ-НЕ» (NOT OR)



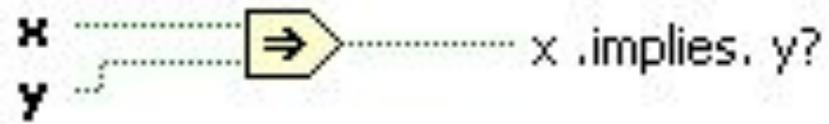
x	y	result
0	0	1
0	1	0
1	1	0

## Элемент «Исключающее ИЛИ-НЕ» (NOT AND)



x	y	result
0	0	1
0	1	0
1	1	1

## Элемент «Исключение» (Implies)



x	y	result
0	0	1
0	1	1
1	1	1
1	0	0

# Составные данные LabVIEW: массивы и кластеры

**Массив (array)** – набор элементов данных одного типа.

Массив:

- Одномерный
- Многомерный

Доступ к массивам осуществляется посредством их индексов. Индекс каждого элемента находится в диапазоне от 0 до N-1, где N – полное количество элементов массива.

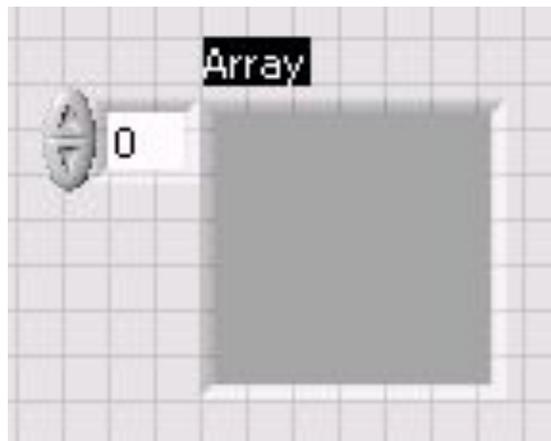
Индекс	0	1	2	3	4	5	6	7	8	9
10-элементный массив	12	32	43	32	46	95	43	5.4	6.1	65.4

Пример одномерного (1D) массива. Первый элемент имеет индекс 0.

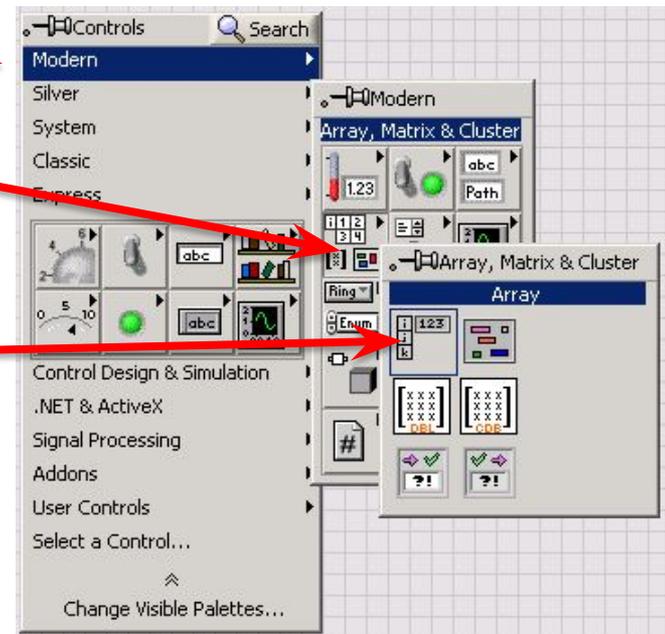
Чтобы создать элементы управления и индикаторы для сложных типов данных, такие как массивы и кластеры, необходимо последовательно выполнить два действия:

1. Определить элементы массива (управления или индикаторов);
2. Объединить *шаблон массива* (array shell) с объектом данных, который может быть числовым, логическим, маршрутным или строковым.

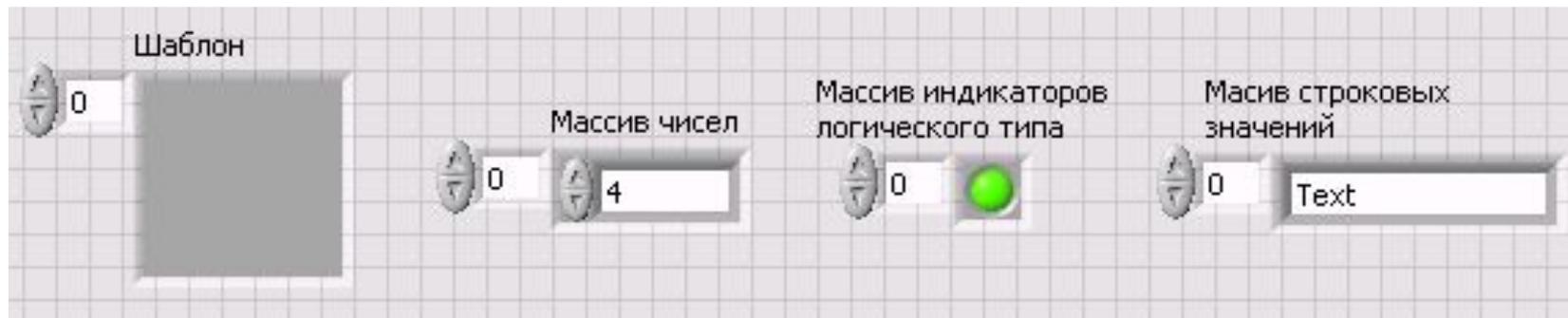
Шаблон массива находится на палитре Элементов управления (Controls на Лицевой панели ВП) – Modern – Array, Matrix & Cluster - Array



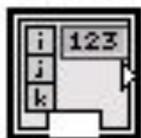
Шаблон массива на Лицевой панели ВП



Для создания массива нужно переместить объект данных в окно отображения элементов на Лицевой панели ВП. После этого шаблон меняет свой вид, показывая, что произошло согласование типа данных. Все элементы массива должны быть либо элементами управления, либо индикаторами, но не их комбинацией.



Шаблон



Массив чисел



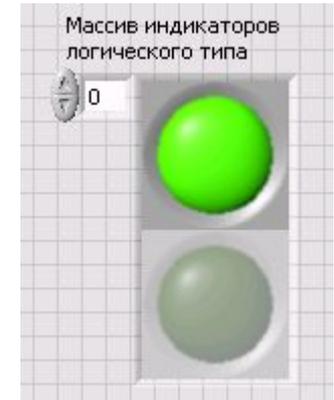
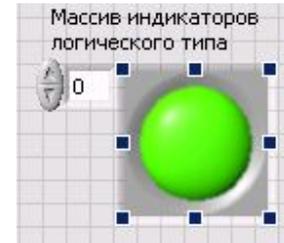
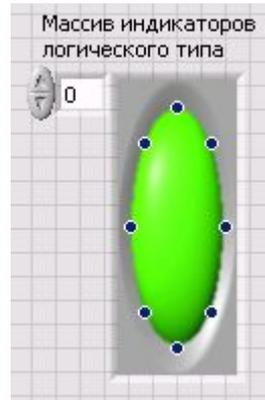
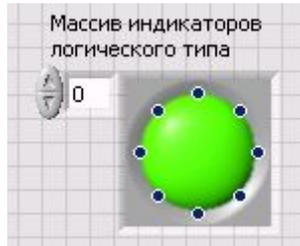
Массив индикаторов логического типа



Массив строковых значений



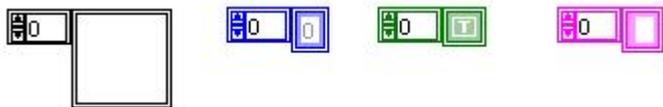
После задания типа данных массива, можно приступить к их вводу.  
Для изменения размеров объекта в массиве, можно воспользоваться инструментом перемещения и растянуть объект за появившиеся маркеры.



При наведении курсора на границу массива тоже появляются маркеры, но они изменяют количество отображаемых элементов массива.

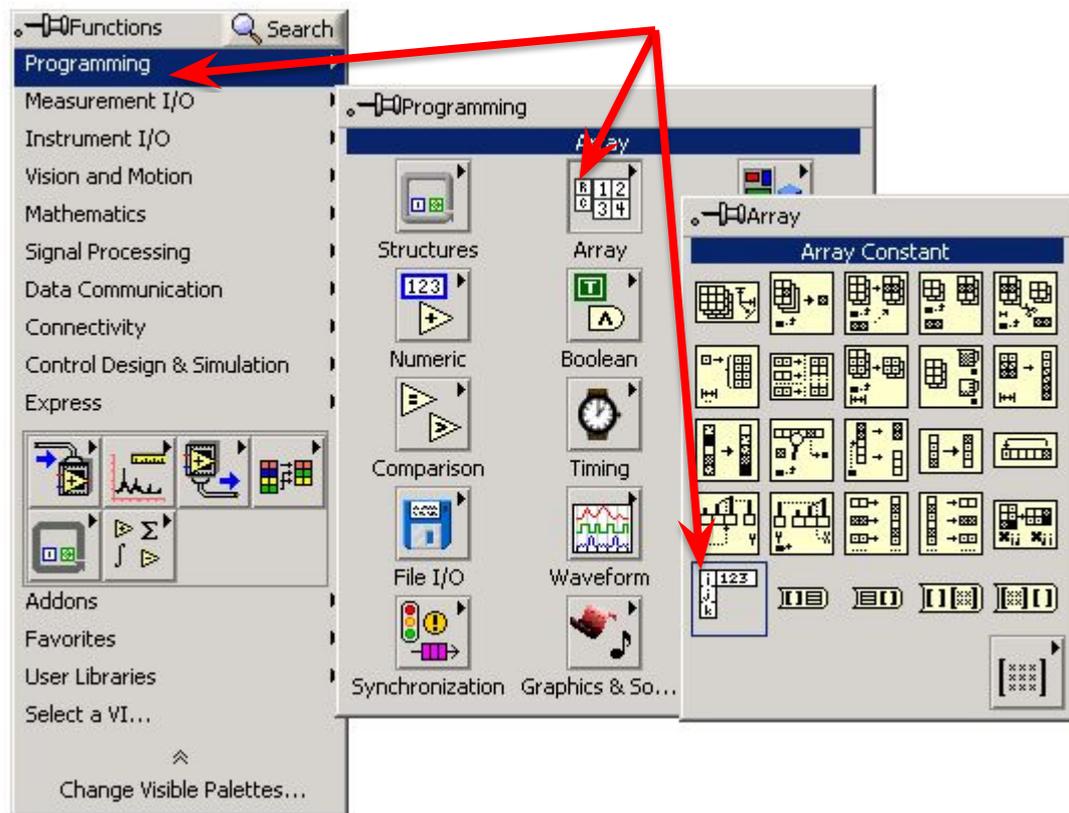
Одномерный массив можно растягивать по горизонтали и по вертикали. На данные в массиве это не повлияет. Формально массив в LabVIEW - всегда строка, вне зависимости от отображения. Также изменение количества отображаемых элементов не означает добавление новых данных в массив.

Создать массив констант можно так же, как и обычные константы (числовые, логические, строковые). Массив констант находится в Палитре Функций (Functions на Панели блок-диаграмм) – Programming - Array – Array Constant



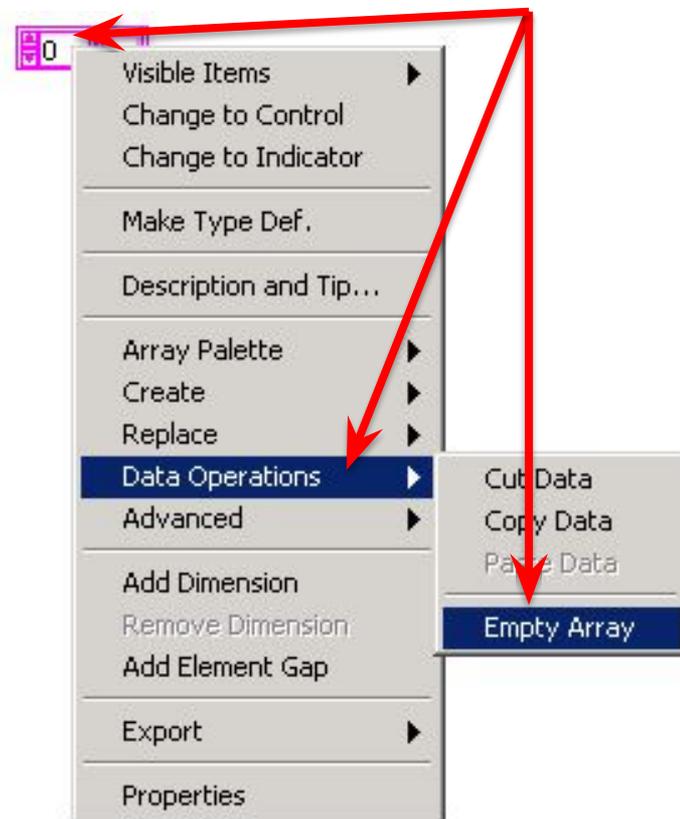
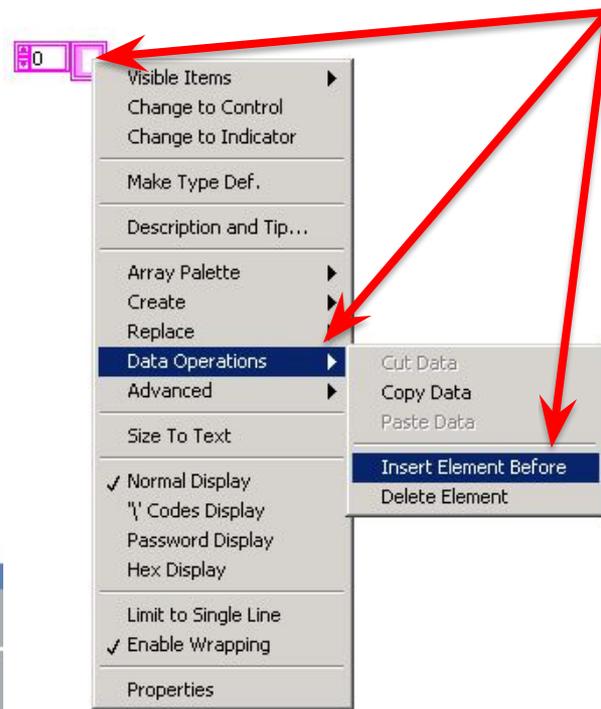
После вызова шаблона массива констант нужно поместить в него необходимый тип данных (в виде констант).

Массив констант используется, например, для инициализации сдвиговых регистров.



Для очистки массива необходимо в контекстном меню индикатора индекса на панели Блок-диаграмм выбрать **Операции с данными - Очистить массив (Data operations - Empty Array)**

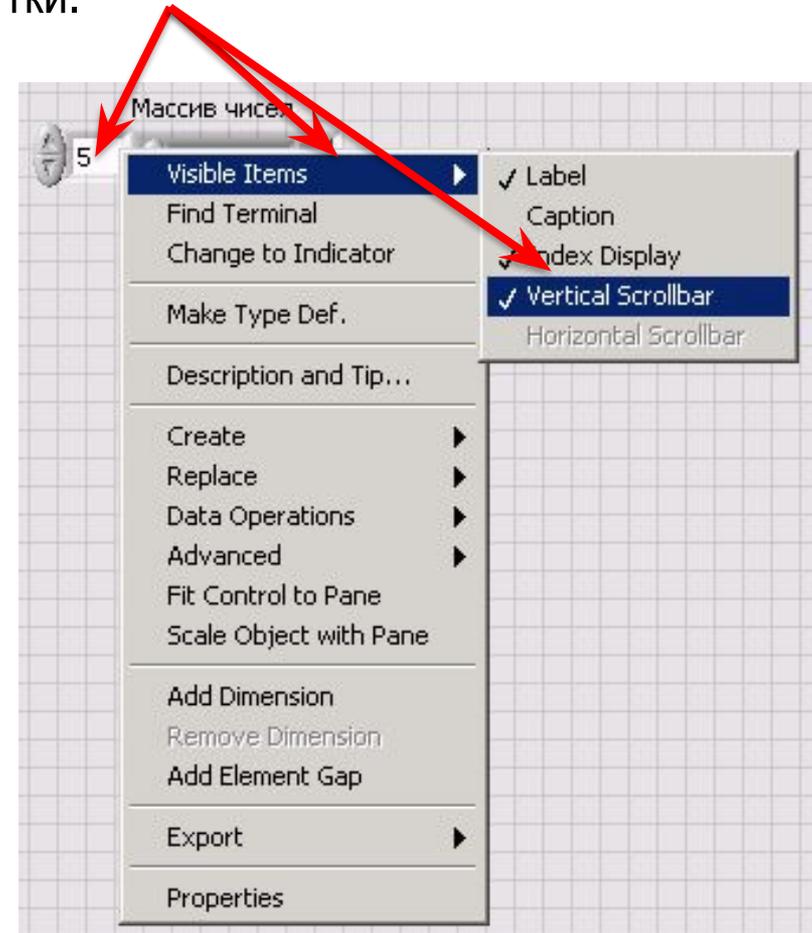
Если необходимо вставить или удалить элемент массива, в контекстном меню элемента массива выбрать **Операции с данными – Вставить элемент выше (Data operations – Insert Element Before)** или **Операции с данными – Удалить элемент (Data operations – Delete Element)**



## Полоса прокрутки

Если массив содержит большое количество элементов, то в нём проще ориентироваться с помощью полосы прокрутки.

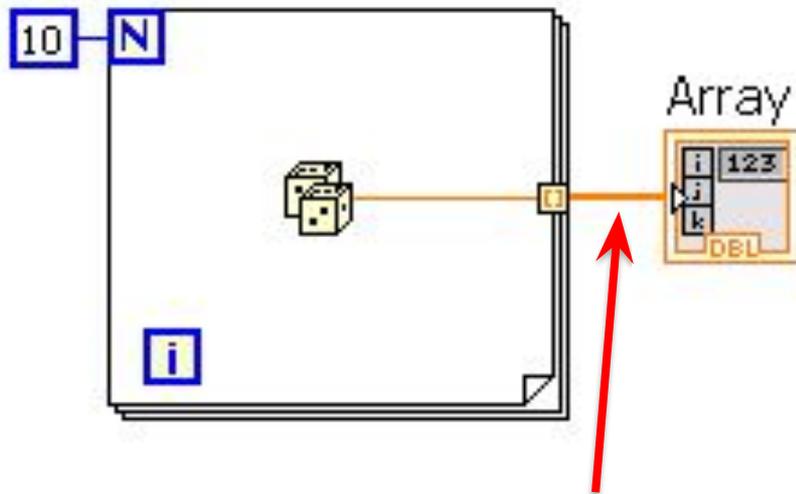
Для добавления полосы прокрутки в массив, нужно в *контекстном меню индикатора индекса* на Лицевой панели выбрать **Видимые элементы – Вертикальная (Горизонтальная) полоса прокрутки** (Visible Items – Vertical (Horizontal) Scrollbar)



## Использование автоматического индексирования в циклах

Цикл с фиксированным числом итераций (For Loop) и цикл по условию (While Loop) могут автоматически поочерёдно обращаться к элементам входящего массива или накапливать новый массив на выходе, добавляя элементы по одному, эта способность называется **автоиндексированием (auto-indexing)**.

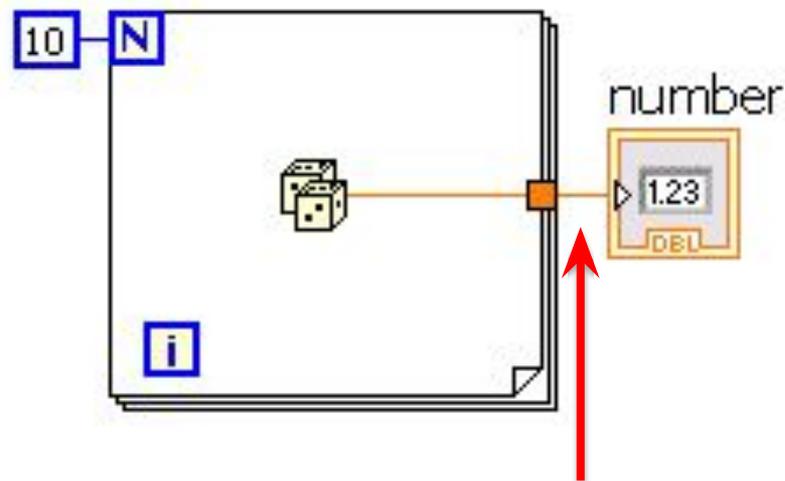
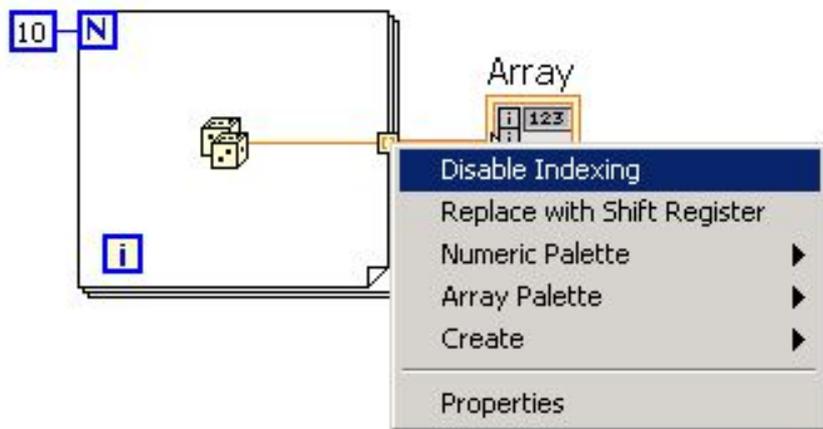
По умолчанию автоиндексирование включено в цикле с фиксированным числом итераций и выключено в цикле по условию.



В цикле с фиксированным числом итераций массив автоматически создаётся на его границе. Каждая итерация создаёт следующий элемент массива. После завершения цикла накопленный массив данных поступает на элемент отображения. До завершения цикла данные не доступны.

Проводник утолщается (на выходе получается одномерный массив)

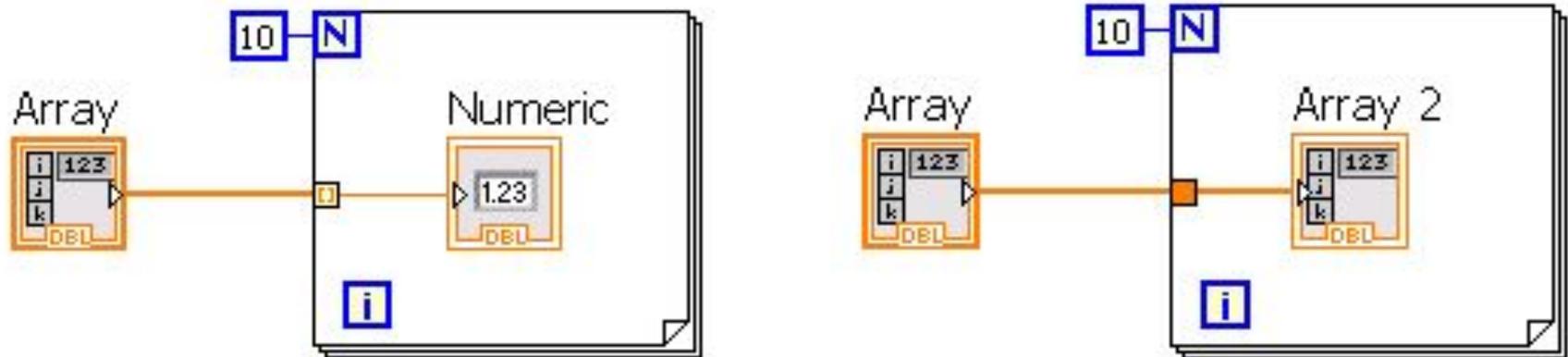
Если необходимо вывести скалярную величину из цикла с фиксированным числом итераций без создания массива, можно отключить автоиндексирование из контекстного меню туннеля – Disable Indexing



Проводник не меняет толщины, на выходе скалярная величина

Если автоиндексация отключена, то из цикла выходит только одно последнее значение.

Автоиндексирование применяется также для ввода массива данных в цикл. Если индексирование включено, то цикл будет переходить к следующему индексу массива при каждой итерации. При этом проводник утончается после входа в цикл.



Если автоиндексирование отключено, то массив целиком передаётся в цикл. При этом проводник не меняет толщины после входа в цикл.

## Использование автоиндексирования для установки количества повторений циклов с фиксированным количеством итераций

При разрешении автоиндексирования для массива, входящего в цикл с фиксированным количеством итераций, LabVIEW автоматически устанавливает число повторений равным размеру массива.

Если задать взаимоисключающие значения, например задав число повторений цикла и введя в действие автоиндексирование или разрешить автоиндексировать два массива разного размера, то LabVIEW установит значение терминала количества итераций в наименьший из вариантов.

