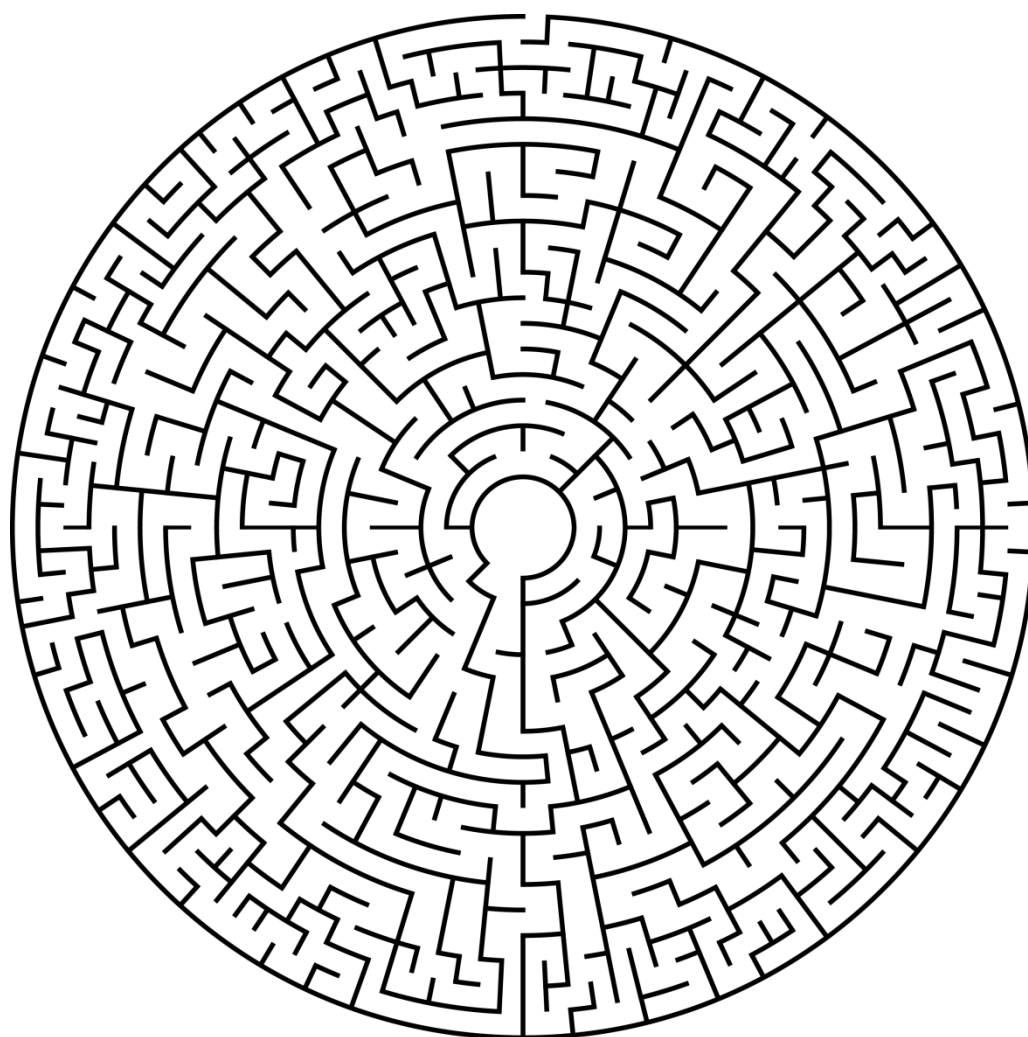


А.С. Васильев, О.Ю. Лашманов

**ОСНОВЫ ПРОГРАММИРОВАНИЯ В СРЕДЕ
LabVIEW**



Санкт-Петербург

2015

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

УНИВЕРСИТЕТ ИТМО

А.С. Васильев, О.Ю. Лашманов
ОСНОВЫ ПРОГРАММИРОВАНИЯ
В СРЕДЕ LabVIEW

Учебное пособие

 **УНИВЕРСИТЕТ ИТМО**

Санкт-Петербург

2015

Васильев А.С., Лашманов О.Ю. Основы программирования в среде LabVIEW. – СПб: Университет ИТМО, 2015. – 82 с.

В учебном пособии рассмотрены основные понятия и концепции графического программирования в среде LabVIEW 2010. Первый раздел пособия содержит теоретические сведения основ программирования в среде LabVIEW, второй - варианты задач по основным разделам информатики для выполнения студентами лабораторных и домашних работ. Задачи рассчитаны на освоение и понимания концепции и базового функционала графического программирования в среде LabVIEW.

Учебное пособие предназначено для студентов по направлению подготовки бакалавров и магистров 200400 – «Оптехника» и по специальности 200401 – «Электронные и оптико-электронные приборы и системы специального назначения».

Рекомендовано Учебно-методическим объединением вузов Российской Федерации по образованию в области приборостроения и оптехники для студентов высших учебных заведений, обучающихся по направлению подготовки бакалавриата 12.03.02 (200400) – Оптехника и специалитета 12.05.01 (200401) – Электронные и оптико-электронные приборы и системы специального назначения протокол №1 от 24.02.2015.



Университет ИТМО – ведущий вуз России в области информационных и фотонных технологий, один из немногих российских вузов, получивших в 2009 году статус национального исследовательского университета. С 2013 года Университет ИТМО – участник программы повышения конкурентоспособности российских университетов среди ведущих мировых научно-образовательных центров, известной как проект «5 в 100». Цель Университета ИТМО – становление исследовательского университета мирового уровня, предпринимательского по типу, ориентированного на интернационализацию всех направлений деятельности.

© Университет ИТМО, 2015

©Васильев А.С., Лашманов О.Ю., 2015

Введение

LabVIEW или Laboratory Virtual Instrument Engineering Workbench (Среда разработки лабораторных виртуальных приборов) представляет собой мощную и гибкую среду графического программирования, которая широко используется в промышленности, образовании и научно-исследовательских лабораториях в качестве стандартного инструмента для сбора данных и управления приборами. LabVIEW - многоплатформенная среда, которая может быть установлена на компьютерах с операционными системами семейства Windows, MacOS или Linux.

Концепция LabVIEW сильно отличается от последовательной природы традиционных языков программирования. С помощью графического языка программирования LabVIEW, именуемого «G» (от англ. Graphical - графический), задача программирования сводится к написанию графической блок-диаграммы, которая компилирует алгоритм в машинный код. Являясь превосходной программной средой для бесчисленных применений в области науки и техники, LabVIEW помогает решать задачи различного типа, затрачивая значительно меньше времени и усилий по сравнению с написанием традиционного программного кода.

LabVIEW находит применение в самых разнообразных сферах человеческой деятельности. В соответствии со своим названием он первоначально использовался в исследовательских лабораториях, да и в настоящее время является наиболее популярным программным пакетом как в лабораториях фундаментальной науки так и в отраслевых промышленных лабораториях. Все более широкое применение LabVIEW находит в образовании - в университетских лабораторных практикумах - особенно по предметам электротехники, механики и физики. Важным достоинством LabVIEW является возможность управления процессом измерения в автоматическом или интерактивном режиме. Для обработки и анализа данных используется обширный набор функциональных библиотек (общего назначения и специализированных). Взаимодействие с исследователем или оператором осуществляется с помощью продуманного и простого в программировании графического интерфейса.

1 Запуск LabVIEW, окна, палитры

Программы LabVIEW называются виртуальными приборами (ВП, VI – virtual instruments), так как они функционально и внешне подобны реальным (традиционным) приборам. Однако они столь же подобны программам и функциям на популярных языках программирования, таких как C или Basic. Здесь и далее мы будем называть программы LabVIEW виртуальными приборами или ВП, причем вне зависимости от того, соотносится их вид и поведение с реальными приборами или нет.

После запуска LabVIEW появляется стартовое диалоговое окно (рис.1), предлагающее выбрать дальнейшие действия. Окно разделено на две части, которые предлагают две группы вариантов действий пользователя. Левая часть окна позволяет создать новый ВП (Blank VI), пустой проект (Empty project) или создать ВП по шаблону (VI from Template). Также можно открыть недавние проекты и файлы (ссылки под надписью Open).

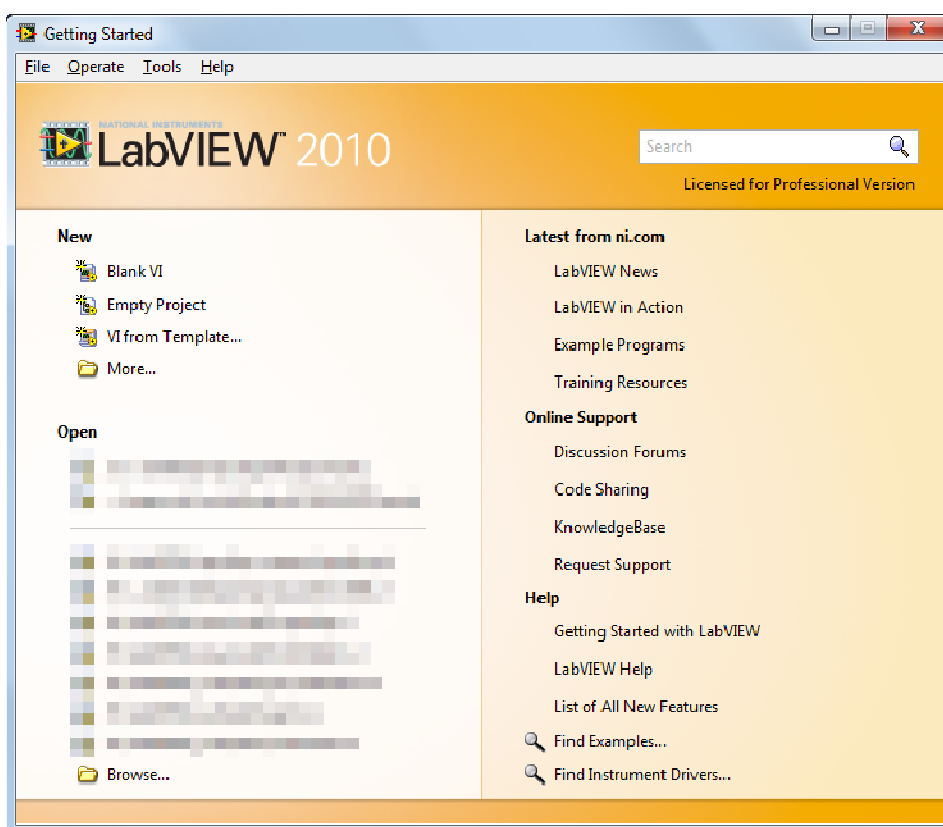


Рисунок 1 – Стартовое диалоговое окно LabVIEW

Правая часть диалогового окна позволяет получить доступ к справочным материалам (Help), материалам на сайте ni.com (Online Support), найти и запустить готовые примеры ВП (Find Example).

При создании пустого ВП (Blank VI) открываются два окна, содержащие лицевую панель (Front Panel) и блок-диаграмму (Block Diagram) (рис.2) виртуального прибора.

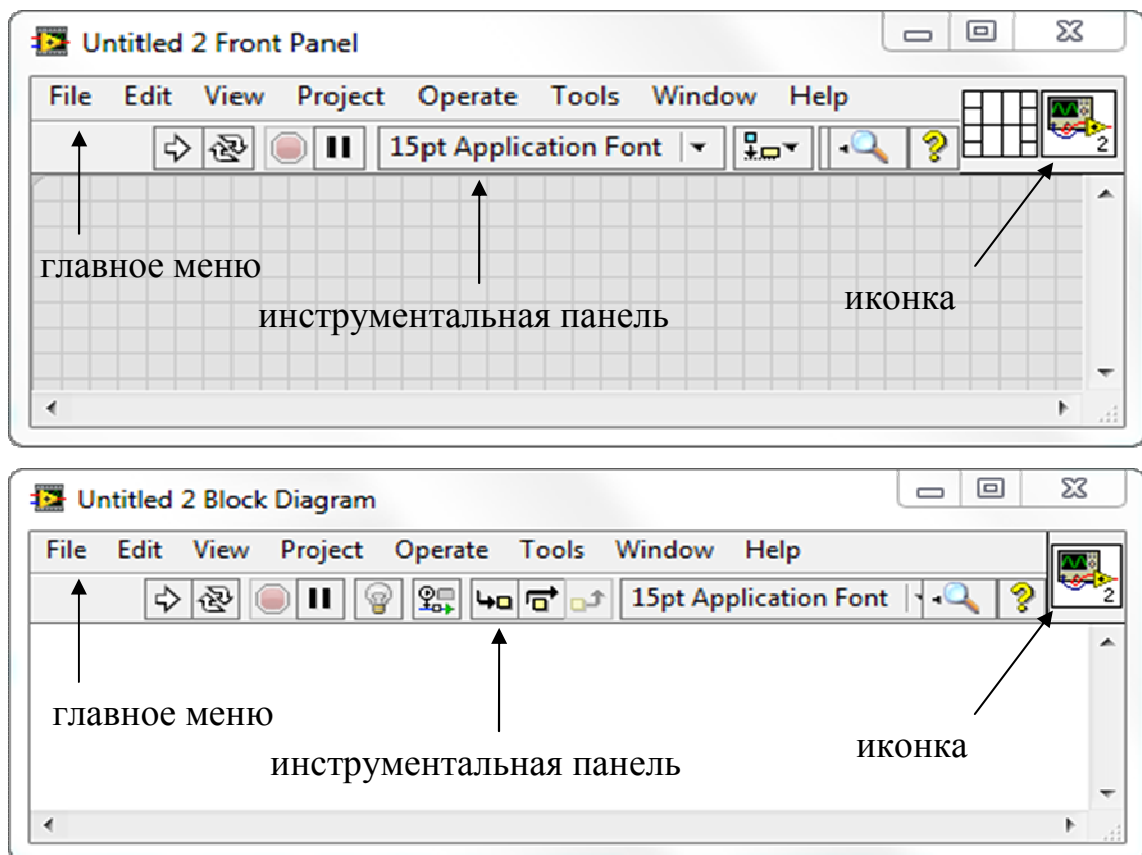


Рисунок 2 – Лицевая панель и блок-диаграмма ВП

На лицевой панели разрабатывается внешний вид будущего виртуального прибора. На ней создаются элементы управления и отображения, которые являются интерактивными средствами ввода и вывода данных этого виртуального прибора.

На блок-диаграмме разрабатывается исходный код будущего виртуального прибора. В отличие от классических языков программирования, исходный код LabVIEW представляет собой блок-диаграмму, где все команды, операторы циклов и сравнения изображаются графическими обозначениями. Блок-диаграмма состоит из узлов, терминалов и проводников данных.









В верхней части каждого окна располагается главное меню с пунктами File, Edit, View, Project, Operate, Tools, Windows и Help. Под главным меню расположена инструментальная панель, служащая для запуска и редактирования ВП. Инструментальная панель окна блок-диаграммы отличается дополнительными инструментами для отладки

ВП. В правом верхнем углу каждого окна находится иконка, наложенная на соединительную панель ВП (последняя показана на лицевой панели).

Окна блок-диаграммы и лицевой панели содержат инструментальную панель (рис. 2).

В таблице 1 описаны функции кнопок инструментальных панелей, которые служат для отладки и управления ходом исполнения ВП.

Таблица 1 – Описание функций кнопок инструментальных панелей

Кнопка	Описание
	Кнопка Запуск (Run) работоспособного ВП
	Вид кнопки Запуск (Run) при наличии ошибок в блок-диаграмме ВП
	Вид кнопки Запуск (Run) ВП в процессе выполнения
	Вид кнопки Запуск (Run) в процессе выполнения подприбора
	Кнопка Непрерывный запуск (Run Continuously) вызывает непрерывный запуск ВП до момента нажатия кнопки Стоп (Stop) или Прервать (Abort)
	Кнопка Прервать выполнение (Abort Execution) вызывает остановку выполняющегося ВП
	Кнопка Пауза (Pause) временно останавливает выполнение ВП
	Кнопка Подсветка выполнения (Highlight Execution) вызывает режим анимационного показа процесса передачи данных по блок-диаграмме и отображения значений данных на выходе узлов и терминалов
	Кнопка Сохранять (Не сохранять) значения провода (Retain (Do Not Retain) Wire Values) позволяет сохранить последнее значение, переданное по проводу, и просмотреть его после окончания выполнения при установке Пробника данных (Probe Data)
	Кнопка Начало пошагового выполнения (Start Single Stepping)
	Кнопка Шаг через (Step Over)
	Кнопка Выход из пошагового выполнения (Step Out) завершает пошаговое выполнение ВП

Свободное пространство каждого окна образует рабочую область, снабженную горизонтальной и вертикальной полосами прокрутки. При построении ВП в рабочей области лицевой панели визуально размещаются элементы управления и индикации, на панели блок-диаграммы составляется блок-диаграмма. Для одновременного отображения блок-диаграммы и лицевой панели на экране можно использовать меню Окно ⇒ Панели слева и справа (Windows ⇒ Tile Left and Right) или комбинацию клавиш <Ctrl+T>.

Построение ВП осуществляется с помощью трех вспомогательных палитр: палитры Элементов управления и отображения (Controls Palette),

палитры Функции (Functions Palette) и палитры Инструменты (Tools Palette). Все перечисленные палитры можно вывести для постоянного или временного отображения и разместить в любом месте экрана. Вывод для постоянного отображения осуществляется с помощью пунктов главного меню Вид (View).

Палитру инструментов можно вывести с помощью щелчка правой кнопки мыши при нажатой клавише <Shift>. Термин инструмент подразумевает специальный операционный режим курсора мыши. Все операции по созданию, редактированию и отладке ВП выполняются с помощью палитры Инструменты (Tools Palette) (рис.3).



Рисунок 3 – Палитра инструментов

При выборе определенного инструмента из палитры инструментов значок курсора мыши приобретает форму этого инструмента. При включенном автоматическом выборе инструмента наведение курсора на объект лицевой панели или блок-диаграммы LabVIEW приводит к автоматическому выбору соответствующего инструмента из палитры инструментов. Автоматический выбор инструментов включается нажатием кнопки Автоматический выбор инструмента (Automatic Tool Selection)



палитры инструментов или нажатием клавиш <Shift+Tab>. Выбор любого другого инструмента приводит к отключению автоматического выбора инструмента. При этом можно циклически менять инструменты с помощью клавиши <Tab>. Для переключения между инструментами Перемещение и Соединение на блок-диаграмме или между инструментами Перемещение и Управление на лицевой панели достаточно нажать пробел.

Для размещения элементов управления и отображения данных на лицевую панель используется палитра элементов управления и отображения (рис. 4). Палитра элементов управления и отображения доступна только с лицевой панели. Для вывода палитры на экран следует щелкнуть правой кнопкой мыши в рабочем пространстве лицевой панели.

Все элементы управления и отображения на палитре сгруппированы по разделам. Каждый раздел может делиться, в свою очередь, на подразделы (числовые, строковые индикаторы, кнопки и т.д.).

Как было сказано ранее, на блок-диаграмме разрабатывается исходный текст программы. Для этого используется палитра функций

(Functions palette). Для вывода палитры на экран следует щелкнуть правой кнопкой мыши в рабочем пространстве блок-диаграммы. Все элементы на палитре сгруппированы по разделам (рис. 5).

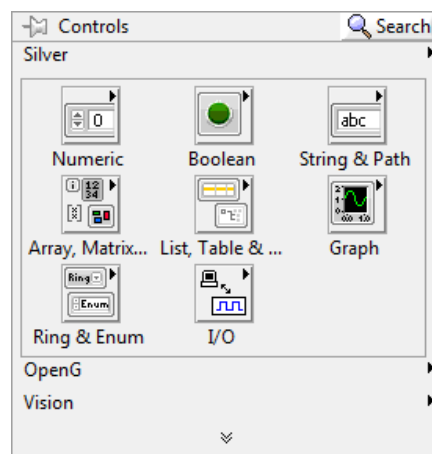


Рисунок 4 – Палитра элементов управления и отображения

Для проектирования ВП в среде LabVIEW необходимо сформировать его лицевую панель и разработать блок-диаграмму. При формировании лицевой панели производятся выбор и установка на ней элементов управления и индикаторов из палитры элементов данной панели. Аналогично при разработке блок-диаграммы производятся выбор и установка на ней функциональных элементов и подприборов из палитры функций данной панели.

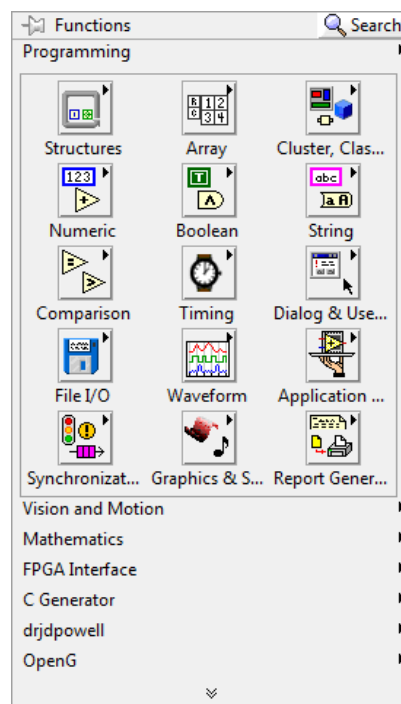


Рисунок 5 – Палитра функций

Установка каждого элемента на лицевой панели сопровождается появлением соответствующего терминала данных (terminal) на панели блок-диаграммы. Терминалы элементов управления представляют порты ввода информации в блок-диаграмму, а терминалы индикаторов – порты вывода информации из блок-диаграммы на лицевую панель. Для обработки введенной информации и программного управления параметрами и режимами работы элементов лицевой панели на панели блок-диаграммы размещаются необходимые константы, функции (Functions), подприборы (SubVI) и структуры (Structures), которые также имеют терминалы для ввода и вывода информации. Все перечисленные элементы представляют узлы (nodes) блок-диаграммы, которые соединяются с терминалами элементов управления и индикации и между собой линиями, называемыми проводниками (wires). В такой схеме через узлы в процессе обработки проходят данные, идущие по проводникам от входных терминалов к выходным. Узлы – это объекты на блок-диаграмме, которые имеют одно или более полей ввода/вывода данных и выполняют алгоритмические операции ВП. Они аналогичны операторам, функциям и подпрограммам текстовых языков программирования.

2 Принцип потока данных

Принцип потока данных (Dataflow) – глобальный концепт программирования, согласно которому LabVIEW выполняет код, написанный на языке «G». Принцип потока данных подразумевает архитектуру параллельных вычислений, в которых выполнение каждой функции выполняется лишь тогда, когда они получают на вход все необходимые данные, при этом последовательность выполнения команд заранее не задаётся. Впервые графическую модель вычислений, управляемых потоком данных, предложил в 1968г. в своей докторской диссертации сотрудник Стэнфордского университета Дуайн Эдэмс.

Основная идея заключается в том, что прохождение данных через узлы в рамках программы определяет порядок исполнения функций программы. В LabVIEW узлы (функции, структуры и виртуальные подприборы) анализируют свои входы, обрабатывают данные и генерируют выходные сигналы. Всякий раз, когда на все используемые входы узла поступают данные, соответствующие инструкции будут добавлены в очередь исполнения процессора. И как только процессор освободится, будут произведены вычисления.

Как только обработка данных внутри узла завершится, он сгенерирует данные на своих выходах, и поток данных продолжит течь к следующему узлу. Соединяя соответствующие входы и выходы узлов, программист задает порядок их выполнения. Такой подход к программированию

позволяет производить обработку нескольких потоков независимо и параллельно друг другу.

На рисунке 6 приведен пример программы, выполняющей умножение y на число 3 и суммирование результата с переменной x , аналогичные операции выполняются для переменных $x2$ и $y2$.

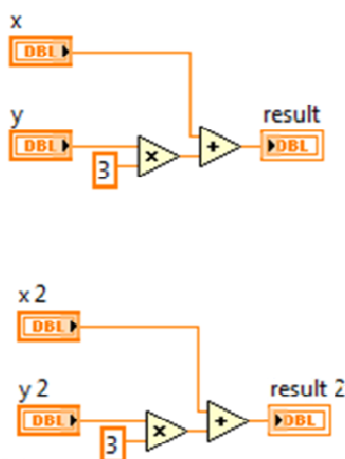


Рисунок 6 – Пример работы потока данных

Приведенный пример показывает работу двух параллельных потоков данных, при этом нельзя наверняка утверждать какая операция выполнится первой, но точно можно сказать, что операция сложения выполнится только после произведения как в одном потоке данных, так и в другом.

3 Типы данных



















Программирование потоков передачи данных в LabVIEW осуществляется при помощи механизма графического связывания терминалов. Терминалы данных имеют прямоугольную форму и содержат буквенно-графическое обозначение, характеризующее тип и форму представления воспринимаемых ими данных. Таким образом, по виду терминала можно определить, является ли он источником или приемником данных, какие типы данных он воспринимает – числовые, логические или строковые, а для числовых – является ли число целым или вещественным. Для определения таких различий используются различия в толщине внешней рамки терминала и направлении треугольной стрелки внутри него, цвет терминала и буквенное или графическое обозначение. Кроме того, вид терминала можно определить и по содержанию контекстного меню.

LabVIEW оперирует различными видами терминалов управляющих элементов, индикаторов, узлов, констант, специализированных структур. Перечень и вид терминалов элементов ввода/вывода данных LabVIEW с

характеристикой типа, цвета и значения по умолчанию приведены в таблице 2.

Таблица 2 – Описание типов данных, используемых в LabVIEW

Элемент управления	Индикатор	Тип данных	Цвет	Значение по умолчанию
		Числовой с плавающей запятой расширенной точности	Оранжевый	0
		Числовой с плавающей запятой двойной точности	Оранжевый	0
		Числовой с плавающей запятой одинарной точности	Оранжевый	0
		Число с фиксированной точкой	Фиолетовый	0
		64-разрядное целое число со знаком	Синий	0
		32-разрядное целое число со знаком	Синий	0
		16-разрядное целое число со знаком	Синий	0
		8-разрядное целое число со знаком	Синий	0
		Беззнаковое 64-битовое целое число	Синий	0
		Беззнаковое 32-битовое целое число	Синий	0
		Беззнаковое 16-битовое целое число	Синий	0
		Беззнаковое 8-битовое целое число	Синий	0

		Комплексный с плавающей запятой одинарной точности	Оранжевый	0+i0
		Комплексный с плавающей запятой двойной точности	Оранжевый	0+i0
		Комплексный с плавающей запятой расширенной точности	Оранжевый	0+i0
		Массив - включает тип данных в квадратные скобки и принимает цвет этого типа данных	Серый	-
		Перечисляемый тип	Синий	-
		Логический	Зеленый	FALSE
		Строковый	Розовый	Пустая строка
		Путь	Морской волны	Не путь (not a path)
		Кластер - включает разные типы данных	Коричневый или розовый	-

Линии связи между терминалами отображают потоки данных в разрабатываемом приложении. Данные могут передаваться только в одном направлении: от источника к приемникам сигнала. Причем цвет и толщина линий связи характеризуют типы передаваемых данных. Так линия одинарной толщины соответствует скалярной величине, утолщенная линия соответствует одномерному массиву, двойная линия – двумерному массиву, утолщенная двойная линия – массиву с размерностью более 2.

Тип представляемых данных может быть задан в меню свойств созданного терминала (Properties - Data Type), либо в контекстном меню Representation (рис. 7)

Важным отличием LabVIEW при работе с типами данных является свойство полиморфизма. Полиморфизмом называется способность

функции обрабатывать данные разных типов. Функция сама «понимает» какие данные подаются на вход и обрабатывает их соответствующим образом.

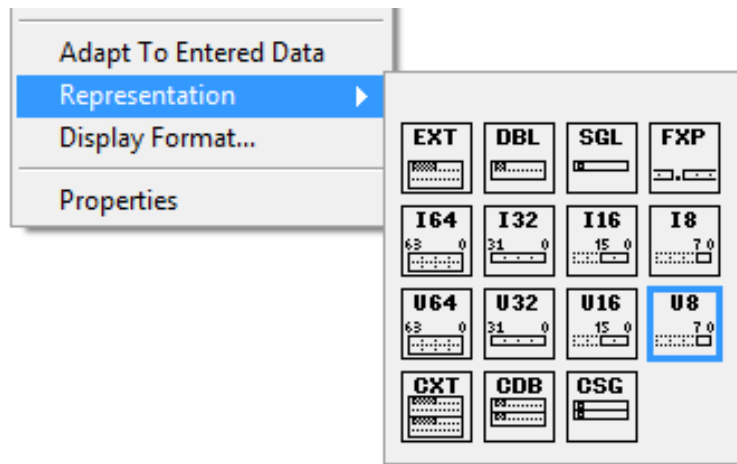


Рисунок 7 – Контекстное меню представления типа данных

Например, функция add при подаче двух чисел с типом данных U8 вернет число U8. При подаче двух массивов произойдет суммирование каждого элемента с каждым. При этом программист никак не задает это поведение при использовании функции, все происходит автоматически.

4 Базовые функции LabVIEW

LabVIEW содержит большое количество встроенных функций, которые сгруппированы в палитре функций по категориям (Programming, Mathematics, Signal Processing, Vision and Motion, Connectivity и т.д.). В связи с большим числом и определенным отличием в уровне сложности функций, входящих в состав подпалитр, их целесообразно разделить на две группы: базовые и дополнительные. К числу базовых функций программирования могут быть отнесены функции, расположенные в следующих подпалитрах: Числовые (Numeric), Логические (Boolean), Строковые (String), Сравнение (Comparison).

4.1 Числовые функции

Для выполнения арифметических, тригонометрических, логарифмических и комплексных операций с числовыми данными и для преобразования типов числовых данных используются математические функции, расположенные в палитре Mathematics (рис. 8).

Палитра математических функций содержит несколько палитр:

- Числовые операции
- Преобразования типов

- Битовые манипуляции
- Комплексные числа
- Масштабирование
- Числа с фиксированной точкой
- Математические константы
- Элементарные функции
 - Тригонометрические функции
 - Логарифмические функции
 - Гиперболические функции
 - Дискретная математика
- Линейная алгебра
- Аппроксимация
- Интерполяция и экстраполяция
- Интегрирование и дифференцирование
- Статистика
- Решение дифференциальных уравнений
- Геометрия
- Полиномы

Используя эти палитры и комбинации функций, можно выполнить большую часть математических операций над числами.

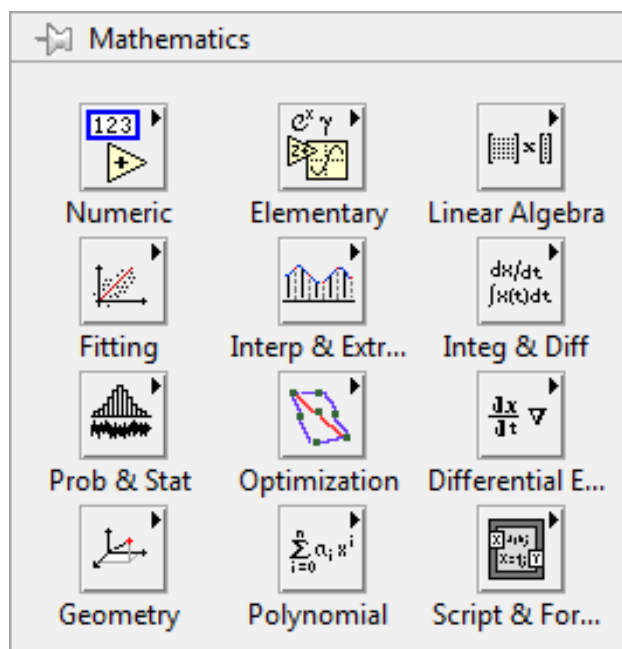


Рисунок 8 – Палитра математических функций

В таблице 3 представлено описание основных математических функций LabVIEW. Все рассмотренные функции являются полиморфными, т.е. на их вход можно подавать скалярные выражения, массивы и их комбинации.

Таблица 3 – Описание математических функций

Название	Пиктограмма	Описание
Add		Возвращает сумму двух операндов
Subtract		Возвращает разность двух операндов
Multiply		Возвращает произведение двух операндов
Divide		Возвращает частное двух операндов
Increment		Возвращает значение входа, увеличенное на 1
Decrement		Возвращает значение входа, уменьшенное на 1
Quotient & Remainder		Возвращает целое частное ($\text{floor}(x/y)$) и остаток ($x-y*\text{floor}(x/y)$) от деления x на y
Absolute Value		Возвращает модуль значения входа
Square		Возвращает квадрат значения входа
Negate		Возвращает значения входа, умноженное на -1
Sine		Возвращает синус значения входа, на входе ожидается значение в радианах
Cosine		Возвращает косинус значения входа, на входе ожидается значение в радианах
Tangent		Возвращает тангенс значения входа, на входе ожидается значение в радианах
Power Of X		Возвращает x в степени y
Power Of 2		Возвращает значение два в степени x
Logarithm Base 10		Возвращает десятичный логарифм от значение входа
Random Number		Возвращает случайное число от 0 до 1

Кроме рассмотренных функций работы с числами, в LabVIEW существует структура Formula Node, в которой программирование осуществляется так же как и в классических текстовых языках. В структуре Formula Node математические выражения и операторы могут быть записаны в текстовом виде. Данную структуру целесообразно использовать при наличии множества переменных или при выполнении сложных расчетов.

Для того чтобы структура Formula Node функционировала, необходимо определить входы и выходы. Для того чтобы добавить вход или выход, нужно нажать правой кнопкой мыши на рамку Formula Node и выбрать add input или add output соответственно. При этом на рамке структуры возникнет вход или выход, в которые вписываются имена

переменных. Стоит помнить, что имена входов и выходов должны соответствовать именам переменных в тексте программы с учетом регистра, написанном в Formula Node.

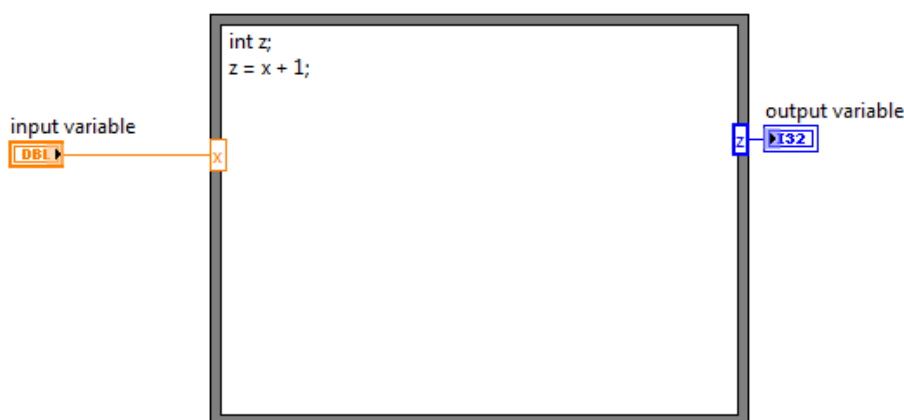


Рисунок 9 – Структура Formula Node

Выражения записываются с использованием функций и операторов, перечисленных в таблице 4. Запись каждой формулы должна заканчиваться символом « ; ».

Таблица 4 – Описание операций структуры Formula Node.

Название	Описание
**	Возведение в степень
+, -, !, ~, ++, --	Унарное суммирование, унарное вычитание, логическое НЕ, битовое дополнение, пред- и постинкремент, пред и постдекремент.
*, /, %	Умножение, деление, модуль (остаток)
>> и <<	Арифметический сдвиг вправо и влево
>, <, >=, <=	Больше, меньше, больше или равно, меньше или равно
!=, ==	Неэквивалентно, эквивалентно
&	Битовое И
^	Битовое исключающее ИЛИ
	Битовое ИЛИ
&&	Логическое И
	Логическое ИЛИ
?:	Условное выражение
abs(x)	Абсолютное значение x
acos(x)	Арккосинус x (рад)
asin(x)	Арсинус x (рад)
atan(x)	Арктангес x (рад)
ceil(x)	Округление до большего числа

$\cos(x)$	Косинус x
$\cosh(x)$	Гиперболический косинус x
$\cot(x)$	Котангес x (рад)
$\operatorname{csx}(x)$	Косеканс x
$\exp(x)$	Экспонента x
$\operatorname{floor}(x)$	Округление до меньшего целого x
$\operatorname{pow}(x,y)$	Возведение x в степень y
$\sec(x)$	Секанс x
$\sin(x)$	Синус x
$\sinh(x)$	Гиперболический синус x
\sqrt{x}	Квадратный корень x
$\tan(x)$	Тангес x
$\tanh(x)$	Гиперболический тангес x

4.2 Логические функции

Логические функции (рис. 10) используются для выполнения логических операций над значениями как простых логических величин, так и массивов этих величин. Функции И (And), ИЛИ (Or), Исключающее ИЛИ (Exclusive Or), НЕ (Not), И-НЕ (Not And), ИЛИ-НЕ (Not Or), Исключающее ИЛИ-НЕ (Not Exclusive Or), Исключение (Implies) являются полиморфными. Оба входа этих функций должны иметь логические или числовые значения, причем эти значения могут быть скалярами, массивами или кластерами. Логические функции расположены в палитре Programming-Boolean (рис. 10).

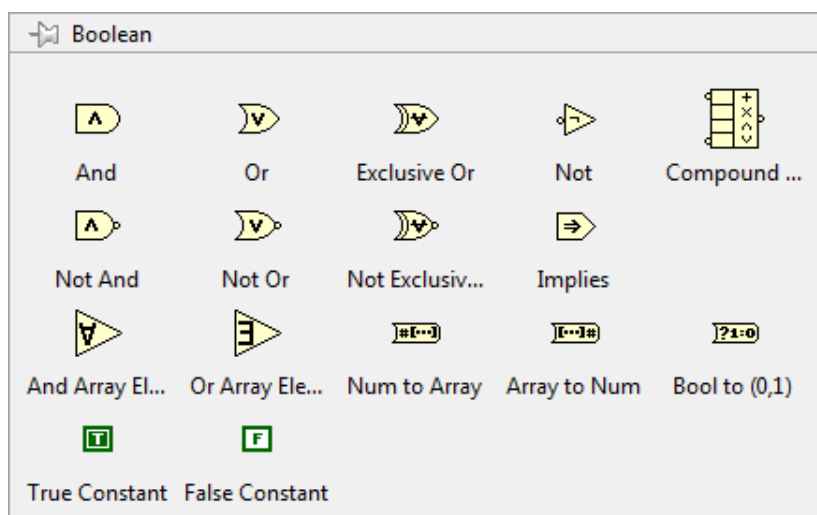









Рисунок 10 – Палитра логических операций

В таблице 5 представлено описание основных логических функций LabVIEW.

Таблица 5 - Описание логических функций

Название	Пиктограмма	Описание
And		Возвращает значение ИСТИНА (1) только при подаче на оба входа значений ИСТИНА (1), иначе возвращает значение ЛОЖЬ (0)
Or		Возвращает значение ЛОЖЬ (0) только при подаче на оба входа значений ЛОЖЬ (0), иначе возвращает значение ИСТИНА (1)
Exclusive Or		Возвращает значение ЛОЖЬ (0) только при подаче на оба входа значений ЛОЖЬ (0) или значений ИСТИНА (1), иначе возвращает значение ИСТИНА (1)
Not		Возвращает значение ЛОЖЬ (0) при подаче на вход значения ИСТИНА (1) и наоборот
Not And		Возвращает значение ЛОЖЬ (0) только при подаче на оба входа значений ИСТИНА (1), иначе возвращает значение ИСТИНА (1)
Not Or		Возвращает значение ИСТИНА (1) только при подаче на оба входа значений ЛОЖЬ (0), иначе возвращает значение ЛОЖЬ (0)
Implies		Инвертирует x и затем выполняет операцию Логическое ИЛИ с входом y Если на входе x значение ИСТИНА и на входе y значение ЛОЖЬ, то функция возвращает ЛОЖЬ Иначе возвращает ИСТИНА

Перечисленные функции также являются полиморфными. При обработке числовых значений они выполняют побитовую обработку чисел.

4.3 Строковые функции

Строки представляют собой последовательности отображаемых и неотображаемых символов в стандарте ASCII. Часто строки используются в качестве простых текстовых сообщений. В процессе управления приборами цифровые данные передаются в виде символьных строк, которые преобразуются затем в цифры. Строковые функции располагаются в палитре Programming – String (рис. 11).

Строковые элементы управления и индикаторы (strings) демонстрируют текстовые данные. Строковые терминалы и проводники, по которым проходят строковые данные, окрашены в розовый цвет.

В таблице 6 представлено описание основных строковых функций LabVIEW.

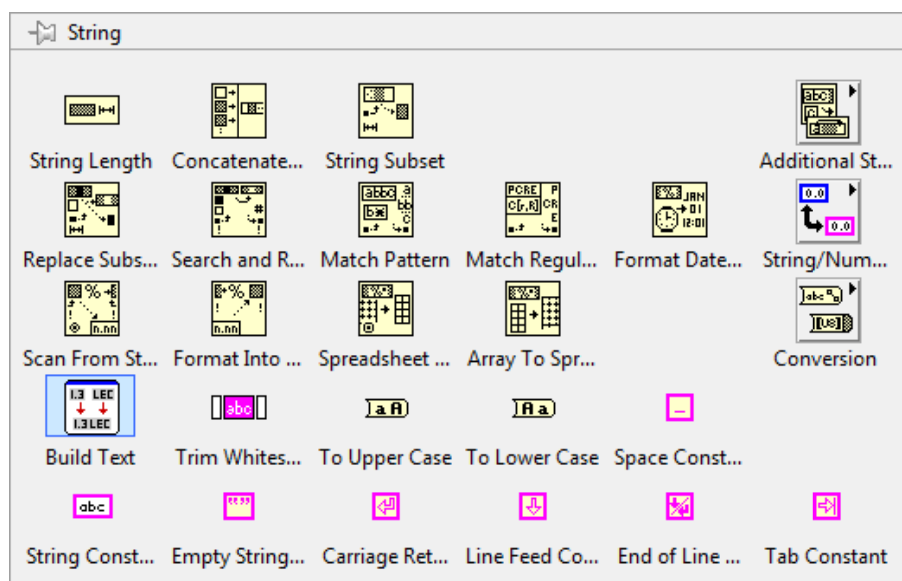










Рисунок 11 – Палитра строковых функций

Таблица 6 - Описание строковых функций

Название	Пиктограмма	Описание
String Length		Функция возвращает число символов (байтов) в строке. На вход строка (string) может быть подана строка, кластер строк или массив кластеров строк
Concatenate Strings		Функция объединяет входные строки и одномерные массивы строк в единственную выходную строку. Для массива строк в объединенную строку входит каждый элемент массива.
String Subset		Функция возвращает часть входной строки (string), начинающуюся со смещения (offset) и содержащую число символов, заданное на входе длина (length).
Search and Replace String		Функция заменяет одну или все образцы подстроки другой подстрокой.

Number To Decimal String		Функция преобразует число (number) в строку десятичных цифр (decimal integer string) с шириной, равной или большей значения, установленного на входе ширина (width)
Decimal String To Number		Функция преобразует цифровые символы строки (string), начиная от смещения (offset), в целое десятичное число и передает его на выход число (number).
String To Byte Array		Функция преобразует строку в массив байтов без знака. Каждый байт в массиве является ASCII кодом соответствующего символа в строке.
Byte Array To String		Функция преобразует массив байтов без знака, представляющих ASCII коды символов, в строку.

4.4 Функции сравнения

Функции сравнения позволяют проверять различные соотношения между скалярными и векторными переменными и константами (равно, не равно, больше, меньше, больше 0, меньше 0 и т. п.), проверять наличие информации, определенного числа или символа, а также выбрать одно из двух значений, выбрать максимальное и минимальное значения, проверить нахождение числа в заданном диапазоне. Функции сравнения располагаются в палитре Programming – Comparison (рис. 12).

Все функции сравнения являются полиморфными и позволяют сравнивать скаляры, массивы и кластеры констант и переменных числового, логического и строкового типа, пути, ссылки, а также структуры на их основе, в частности таблицы, деревья и т. п. Рассмотрим подробнее поведение функции «равно» в зависимости от типов входных данных.

Функция «равно» возвращает значение ИСТИНА, если x равно y , иначе возвращается значение ЛОЖЬ. При сравнении массивов и кластеров предусмотрена возможность режимов сравнение элементов (Compare Elements) (по умолчанию) и сравнение совокупности (Compare Aggregates). Выбор режимов производится с помощью строки Режим сравнения (Comparison Mode) контекстного меню функции. При выборе режима Сравнение элементов выход $x=y?$ представляет массив или кластер

логических скаляров, при выборе режима сравнение совокупности – логический скаляр. Аналогичным образом ведут себя функции «больше», «меньше», «больше либо равно», «меньше либо равно», «не равно».

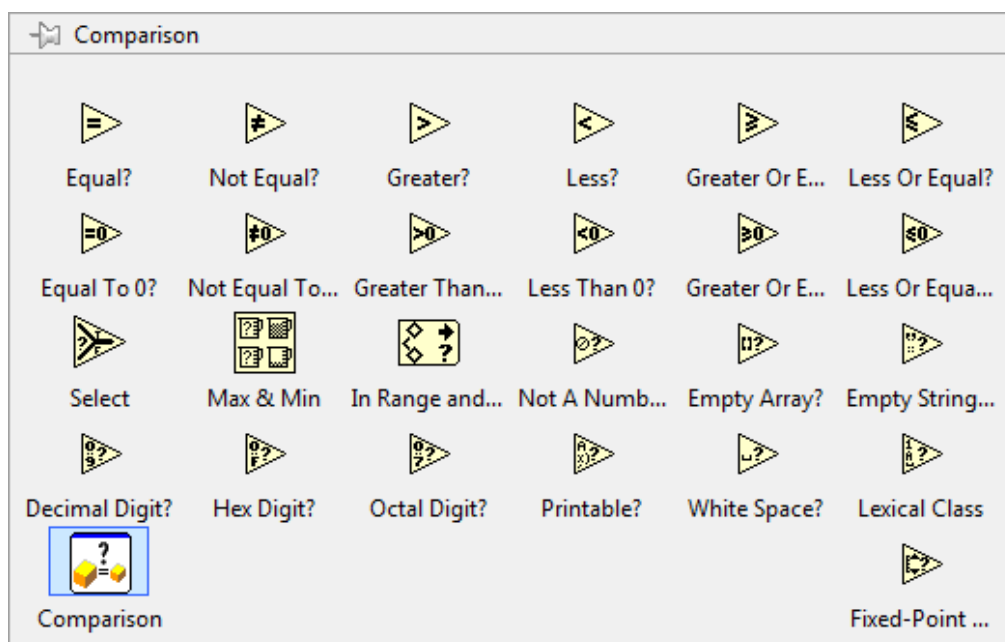




Рисунок 12 – Палитра функций сравнения

В таблице 7 представлено описание основных функций сравнения LabVIEW.

Таблица 7 - Описание функций сравнения

Название	Пиктограмма	Описание
Equal?		Функция возвращает значение ИСТИНА, если x равно y , иначе возвращается значение ЛОЖЬ
Not Equal?		Функция возвращает значение ИСТИНА, если x не равно y , иначе возвращается значение ЛОЖЬ
Greater?		Возвращает значение ИСТИНА, если x больше y , иначе возвращается значение ЛОЖЬ
Less?		Возвращает значение ИСТИНА, если x не больше y , иначе возвращается значение ЛОЖЬ
Equal To 0?		Возвращает значение ИСТИНА, если x равно 0, иначе - ЛОЖЬ
Not Equal To 0?		Возвращает значение ИСТИНА, если x не равно 0, иначе - ЛОЖЬ

Select		Функция возвращает значение, подключенное к входам t или f в зависимости от состояния входа s. Если на входе s установлено состояние ИСТИНА, то функция возвращает значение, подключенное к входу t. Если же на входе s установлено состояние ЛОЖЬ, то функция возвращает значение, подключенное ко входу f. Функция является полиморфной
In Range and Coerce		Функция определяет нахождение x в диапазоне, заданном входами верхний предел (upper limit) и нижний предел (lower limit), и дополнительно ограничивает выходное значение указанным диапазоном. Нахождение в диапазоне приводит к появлению на выходе в диапазоне? (In Range?)

5 Структура Варианта

Под ветвлением понимается конструкция языка программирования, обеспечивающая выполнение определённой команды (набора команд) только при условии истинности некоторого логического выражения, либо выполнение одной из нескольких команд (наборов команд) в зависимости от значения некоторого выражения.

В LabVIEW оператор ветвления реализуется на основе структуры Вариант (*Case*). Структура варианта, показанная на рисунке 13, имеет две или более поддиаграмм и аналогична операторам варианта (*switch...case*) или логическим операторам (*if... else*) в текстовых языках программирования. Лишь одна из поддиаграмм выполняется в зависимости от логического, числового или строкового значения, которое подается на терминал селектора. Структура варианта располагается в палитре Programming – Structures.

В теле структуры варианта можно выделить несколько основных элементов: Тело (2) – описывает блок-диаграммы для различных случаев; селектор (1) – отображает значение с которым ассоциирована данная блок-диаграмма. Программист может задавать в поле единичное значение, диапазон или список значений; терминал (3) – определяет, какая из описанных блок-диаграмм будет выполнена.

Если к терминалу структуры вариант подключено логическое значение, то структура будет иметь два варианта: ЛОЖЬ и ИСТИНА. Если к терминалу селектора подключены числовые или строковые данные, то структура может иметь почти неограниченное количество вариантов (ограниченное только типом данных селектора). Первоначально существует

лишь два варианта, которые при помощи контекстного меню можно увеличить до нужного количества. Допустимо использовать несколько значений терминала для одного варианта, отделяя их запятыми. Кроме того, всегда можно выбрать вариант по умолчанию (Default), который будет выполняться, если величина, подаваемая на терминал структуры, не соответствует никакому другому варианту. Стоит помнить, что всегда один (и только один) из вариантов структуры вариант должен быть назначен по умолчанию.

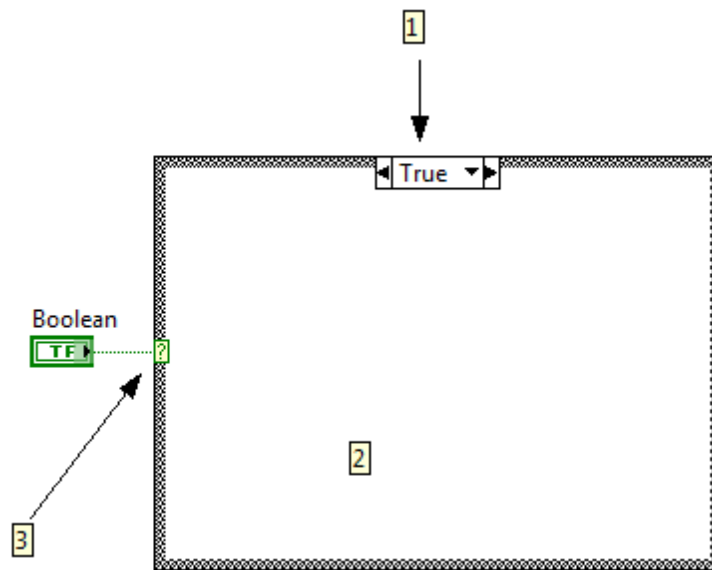


Рисунок 13 – Структура вариант

При размещении структуры варианта на блок-диаграмме в первый раз она будет представлена в логической форме. Для того чтобы использовать в структуре числовые значения, необходимо подать числовой тип данных на терминал селектора. Терминал структуры варианта всегда должен быть подключен.

6 Циклы

Цикл – это разновидность управляющей конструкции в высокоуровневых языках программирования, предназначенная для организации многократного исполнения набора инструкций. Последовательность инструкций, предназначенная для многократного исполнения, называется телом цикла, а единичное выполнение тела цикла называется итерацией. В LabVIEW различают два типа циклов: с фиксированным числом итераций (For Loop) и цикл по условию (While Loop). Структуры циклов расположены в палитре Programming – Structures.

Циклы могут иметь неограниченное количество входных и выходных терминалов для получения и выдачи данных. В терминалы могут подаваться как скалярные, так и векторные значения. Для создания нового терминала достаточно подключить провод к одной из внешних границ рамки цикла. При подключении таким образом массива к структуре For Loop автоматически создается перечисляемый вход. Каждая итерация цикла будет считывать новое значение из массива.

6.1 Цикл с фиксированным числом итераций

Цикл с фиксированным числом итераций (For Loop) осуществляет заданное число итераций выполнения кода внутри данной структуры. На рисунке 14 приведен общий вид структуры цикла с фиксированным числом итераций.

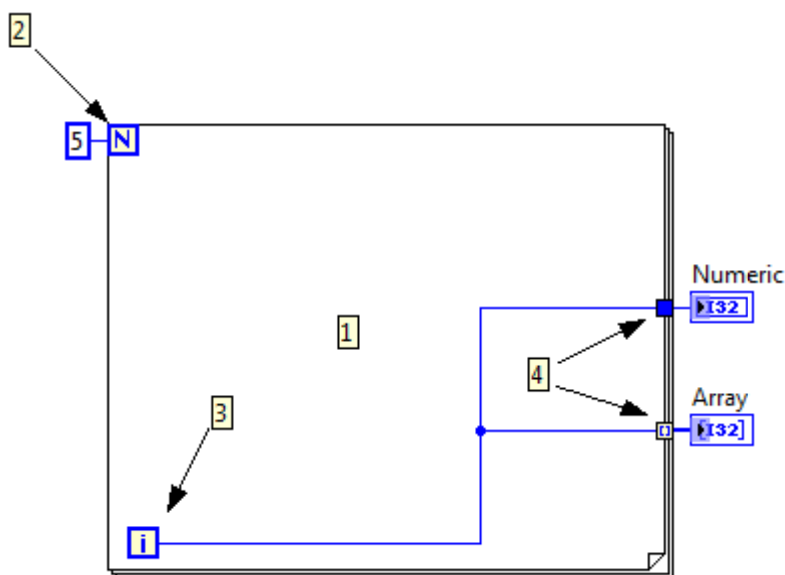


Рисунок 14 – Структура цикла с фиксированным числом итераций

В его составе можно выделить несколько основных компонентов. Тело цикла (1), содержащее инструкции, которые необходимо повторять; терминал количества итераций (2) - с помощью терминала количества итераций программист задает максимальное количество повторений цикла; счетчик итераций (3), отсчитывающий номер текущей итерации; выходные терминалы (4), на которых появляются значения только после завершения работы цикла. По умолчанию на выходе структуры цикла с фиксированным числом итераций создается терминал, содержащий массив значений.

6.2 Цикл по условию

Цикл по условию (While Loop) выполняет многократное повторение операции над потоком данных, пока не выполнится логическое условие выхода. На рисунке 15 приведен общий вид структуры цикла по условию.

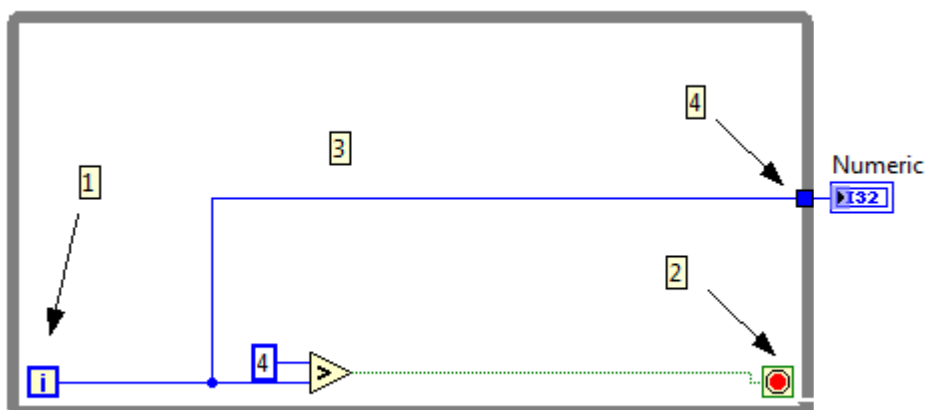


Рисунок 15 – Структура цикла по условию

В его структуре можно выделить несколько основных компонентов. Тело цикла (3), содержащее инструкции, которые необходимо повторять; счетчик итераций (1), отсчитывающий номер текущей итерации; терминал условия завершения (2), имеющий два варианта работы – остановка, если на входе терминала ИСТИНА и остановка, если на входе терминала ЛОЖЬ (переключение между режимами осуществляется про помощи щелчка правой кнопкой мыши (stop if true/false)); выходные терминалы (4), на которых появляются значения только после завершения работы цикла. По умолчанию на выходе структуры цикла по условию создается терминал скалярного вида.

6.3 Сдвиговые регистры

Сдвиговые регистры используются при работе с циклами для передачи значений из текущей итерации в следующую. Установка сдвиговых регистров производится на вертикальных сторонах структуры с помощью опции добавить сдвиговый регистр (Add Shift Register), выбираемой из контекстного меню структуры при щелчке правой кнопкой мыши на одной из этих сторон.

Тип и размерность сдвигового регистра определяются типом и размерностью данных, подключаемых к правому (входному) терминалу сдвигового регистра. Количество разрядов (циклов задержки) левого терминала сдвигового регистра можно изменить с помощью инструмента перемещения или с помощью опций добавить элемент (Add Element), удалить элемент (Remove Element) контекстного меню структуры. По умолчанию при первом запуске виртуального прибора начальные значения левых терминалов сдвигового регистра равны значению по умолчанию для типа данных регистра. Если эти терминалы не были подключены, то при последующих запусках в них будут находиться значения, оставшиеся от предыдущих запусков виртуального прибора. Для инициализации левого

терминала сдвигового регистра необходимо подключить константу, элемент управления или функцию инициализации массивов. Цикл с неинициализированным сдвиговым регистром используется при неоднократном запуске ВП для присвоения выходному значению сдвигового регистра значения, взятого с последнего выполнения ВП.

Предусмотрена возможность создания нескольких сдвиговых регистров в одной структуре цикла. К тому же сдвиговый регистр может иметь несколько левых терминалов сдвигового регистра для возможности работы с несколькими значениями предыдущих итераций.

Единичные сдвиговые регистры могут быть преобразованы в туннели с помощью опции заменить туннелем (Replace with Tunnel). После замены туннель на правой вертикальной границе структуры цикла будет представлять терминал выхода данных из цикла, а туннель на левой границе - терминал входа данных. Операция является обратимой, т.е. входной или выходной терминалы могут быть преобразованы в пару сдвиговых регистров (Replace with shift register).

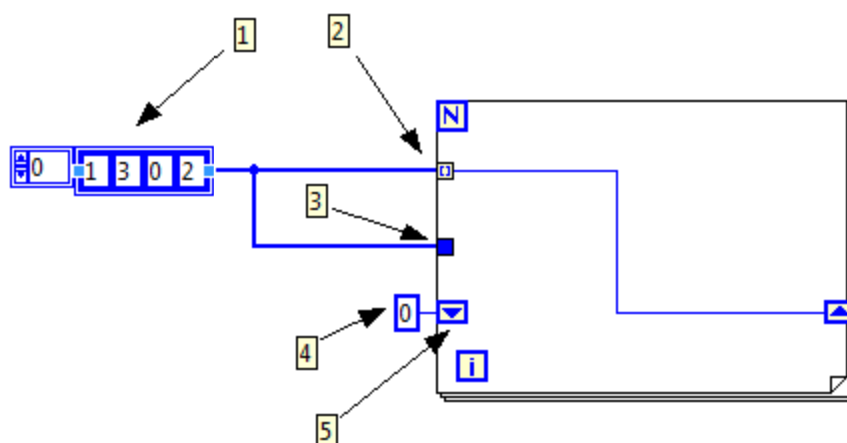


Рисунок 16 – Терминалы структуры цикла

На рисунке 16 представлены все рассмотренные виды терминалов циклов. В данном примере на вход структуры For Loop подается массив (1) при этом автоматически создается автоиндексируемый терминал (2), дополнительно создается неиндексируемый вход (3) и сдвиговый регистр (5). На инициализирующий вход сдвигового регистра подается инициализирующее значение (4) равное нулю. Каждую итерацию с перечисляемого терминала на вход сдвигового регистра передаются значения массива (1, 3, 0, 2). При наличии автоиндексируемого входа структуры For Loop нет необходимости указывать максимальное количество итераций (терминал N), оно будет определено автоматически, исходя из размера массива.

7 Массивы

Массив - набор однотипных данных (элементов), расположенных в памяти непосредственно друг за другом, доступ к которым осуществляется по индексу (индексам). Массив является структурой с произвольным доступом, т.е. программист может осуществлять доступ к любому элементу по его индексу в произвольном порядке.

Количество используемых индексов массива может быть различным. Массивы с одним индексом называют одномерными, с двумя – двумерными и т. д. Одномерный массив нестрого соответствует вектору в математике, двумерный - матрице. Чаще всего применяются массивы с одним или двумя индексами.

Основными характеристиками массива является его размерность, тип входящих в него элементов и его размер (количество элементов по каждой размерности).

Все массивы в LabVIEW являются динамическими. Динамические массивы делают работу с данными более гибкой, так как не требуют предварительного определения хранимых объемов данных. Максимальное количество элементов на каждую размерность составляет 2^{31} . Значения индексов лежат в диапазоне от 0 до N-1, где N – количество элементов массива.

Элементом массива может быть любой тип данных, за исключением массива, таблицы или графика.

В LabVIEW массивы могут быть созданы как вручную на лицевой панели или на блок-диаграмме, так и программно. На лицевой панели могут быть созданы массивы элементов управления или индикаторов, на блок-диаграмме - массивы констант. Программно массивы создаются с помощью структур и соответствующих функций.

Для формирования массивов на лицевой панели необходимо разместить на ней шаблон массива (Array) из палитры Массив, матрица и кластер (Array, Matrix & Cluster) палитры элементов управления. В окне отображения элемента может быть помещен объект данных - элемент управления или индикатор в соответствии с типом формируемого массива. Помещение объекта в шаблоне приводит к присвоению терминалу массива цвета и типа, помещенного объекта. После задания типа массива он может использоваться для ввода или вывода данных (рис. 17).

Аналогичным образом создается массив констант на блок-диаграмме. Для создания массива констант необходимо разместить на диаграмме шаблон массива констант (Array Constant) из палитры Массив (Array) и поместить в него константу необходимого типа.

Для программного формирования и обработки массивов используются структуры Цикл с фиксированным числом итераций (For Loop) и Цикл по условию (While Loop), имеющие в своем составе

индексную переменную i . Если элементы массива формируются в структуре цикла с фиксированным числом итераций, то их преобразование в массив происходит в терминале вывода данных, находящемся по умолчанию в режиме включить индексирование (Enable Indexing). В этом режиме терминал вывода данных представляет пустотелый двойной квадрат. Формирование массива сопровождается изменением толщины провода с данными после его выхода из структуры. Выключение режима индексирования осуществляется с помощью строки отключить индексирование (Disable Indexing) контекстного меню терминала. При этом из структуры будет выводиться только последний элемент.

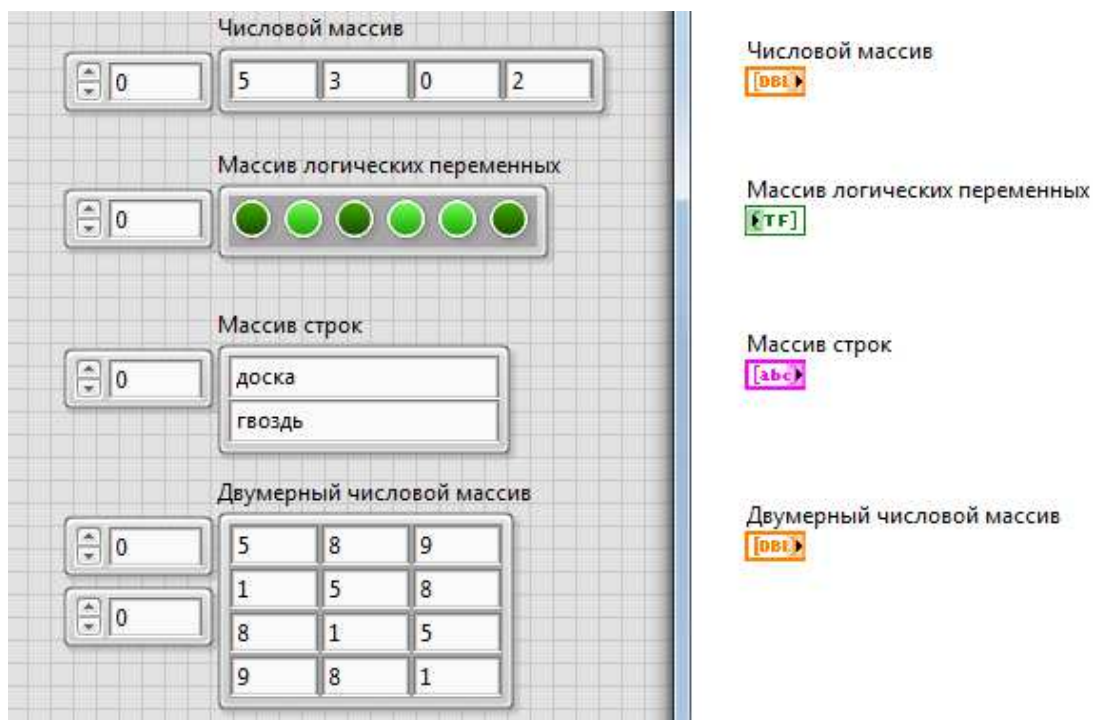


Рисунок 17 – Виды массивов

Описанный режим автоматического индексирования и аккумуляции массивов на границе цикла путем добавления одного нового элемента в каждом повторении цикла называется автоиндексированием (auto-indexing).

При вводе массива в структуру цикла терминал ввода данных по умолчанию также находится в состоянии включить индексирование (рис.18). В данном режиме терминал передает в цикл по одному элементу в каждую итерацию. В этом случае структура автоматически определяет размерность массива и нет необходимости задавать значение терминала числа итераций. При подаче на вход нескольких массивов разной длины структура настраивается на самый короткий массив. То же самое происходит и при одновременно подключенном терминале числа итераций.

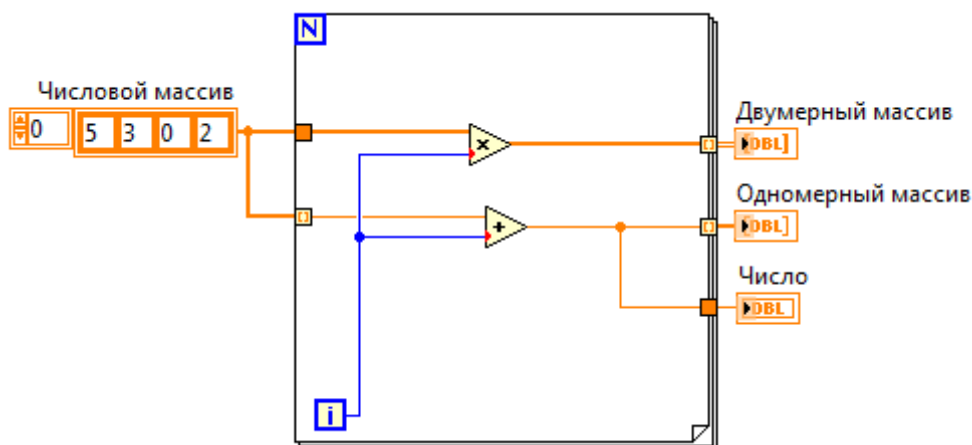


Рисунок 18 – Варианты ввода/вывода данных в/из цикла

Если терминал входа перевести в состояние отключить индексирование, массив будет вводиться в цикл целиком (рис. 18). При этом, естественно, способность автоиндексирования теряется и значение числа циклов должно быть задано

На рис. 18 видно, что при выводе одномерного массива из цикла в режиме автоиндексирования он преобразуется в двумерный массив. Массив такой же размерности может быть создан на основе скалярных переменных при использовании структуры из внутреннего и внешнего циклов. Увеличение размерности массива, созданного оператором на лицевой панели или в блок-диаграмме, может быть выполнено с помощью строки добавить размерность (Add Dimension) контекстного меню элемента управления / отображения индекса массива.

В таблице 8 представлено описание основных функций LabVIEW для работы с массивами.

Таблица 8 - Описание основных функций работы с массивами

Название	Пиктограмма	Описание
Array Size		Возвращает количество элементов в массиве.
Index Array		Возвращает элемент массива по его индексу.
Build Array		Возвращает массив, построенный из элементов на входе. Элементы на входе должны быть одного типа.

Insert Into Array		Возвращает массив со вставленным на указанную позицию элементом.
Delete From Array		Возвращает массив с удаленным элементом. Удаляется как одиночный элемент, так и часть массива.
Array Max & Min		Возвращает максимальное и минимальное значения в массива, а так же их координаты.
Transpose 2D Array		Выполняет операцию транспонирования над исходным массивом.

8 Ввод/вывод данных в/из файла

Операции ввода и вывода (I/O) запрашивают информацию из файла и сохраняют информацию в файле на диске. LabVIEW имеет ряд гибких функций ввода и вывода наряду с простыми функциями, которые обеспечивают выполнение практически всех операций ввода и вывода. Функции ввода/вывода находятся в палитре Programming-File I/O (рис. 19).

Для функций, работающих с файлами, необходимо ввести путь к размещению файла, который выглядит как строка. Путь является особым видом данных для работы с файловой системой, указывая местоположение файла. Если переменная пути к файлу не была подключена, то функция вызовет диалоговое окно и попросит выбрать файл или ввести его имя. Будучи вызванными, функции откроют или создадут файл, прочитают или запишут данные, а затем закроют файл.

Одним из наиболее распространенных способов хранения данных в файле является форматирование текстового файла. Такой файл может быть открыт в любой программе, работающей с таблицами или текстом (Excel, Notepad и т.д.). В большинстве табличных форматов столбцы отделяются табуляцией, а строки - символами конца строк (EOL). Функции Записать в файл табличного формата (Write To Spreadsheet File) и Считать из файла табличного формата (Read From Spreadsheet File) взаимодействуют с файлами табличного формата. При этом функция имеет Write To Spreadsheet File на входе одномерный или двумерный массив для записи, а

так же флаг продолжения записи в файл. Функция Read From Spreadsheet File является полиморфной, пользователь может выбирать в каком типе данных считывать файл (double, integer, string). Эти функции очень удобны при записи и чтения массивов данных.

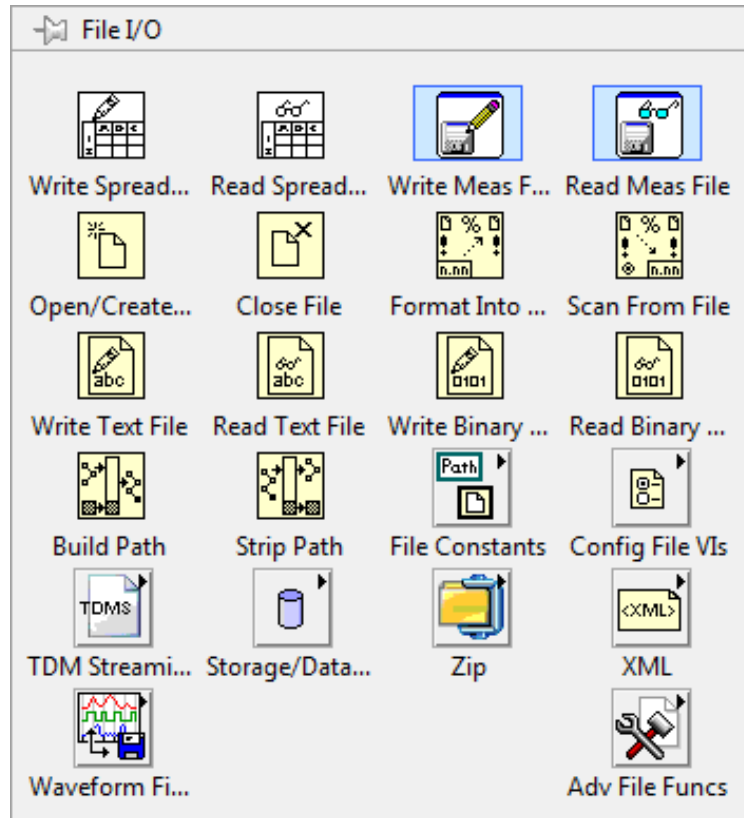








Рисунок 19 – Палитра ввода/вывода в файл

Если необходимо сформировать простой текстовый файл, то используется функция Write to Text File и Read from Text File. Эти функции позволяют записывать любые строки в указанный текстовый файл. Однако при использовании этих функций необходимо соблюдать определенный алгоритм работы и использовать дополнительные модули. Так для записи или чтения необходимо открыть файл при помощи функции Open/Create/Replace File, при этом необходимо обязательно указывать действия при операции с файлом (открыть, создать, заменить, открыть или создать, заменить или создать, заменить или создать с подтверждением). После того как с файлом будут произведены заданные действия, файл необходимо закрыть. Эта операция нужна для того, чтобы операционная система знала, что доступ к файлу больше не заблокирован и другая программа может взаимодействовать с ним.

В таблице 9 представлено описание основных функций LabVIEW для работы с файлами.

Таблица 9 - Описание основных функций работы с файлами

Название	Пиктограмма	Описание
Read from Text File		Возвращает содержимое текстового файла в виде строки
Write to Text File		Осуществляет запись текстовой строки в файл
Open/Create/Replace File		Возвращает ссылку на открытый файл и начинает работу с ним
Close File		Завершает работу с файлом по ссылке.
Write To Spreadsheet File		Функция преобразует массив чисел или строк в текстовую строку и записывает эту строку в виде нового текстового файла или добавляет строку к существующему файлу. Эта функция используется для создания текстового файла, воспринимаемого большинством табличных приложений. По умолчанию разделение между значениями производится при помощи знака табуляции.
Read From Spreadsheet File	 Double ▾	Функция считывает текстовый файл в таблицу значений, поддерживаются различные форматы входных данных (строки, целые, числа с плавающей точкой).

9 Создание подприбора

В LabVIEW очень важным аспектом является повторное использование кода. Для этих целей служат виртуальные подприборы.

Подприбор - это отдельный ВП, применяемый другим ВП. Новый ВП можно использовать в качестве подприбора на блок-диаграмме высокоуровневого ВП, снабдив иконкой и определив его соединительную панель. Виртуальный подприбор LabVIEW является аналогом подпрограммы в языке С или другом подобном языке программирования. Поскольку не имеется никаких ограничений по количеству подпрограмм в языке С, также не существует и каких-либо ограничений по количеству подприборов, применяемых в ВП LabVIEW (если позволяет память).

Для того чтобы превратить любой ВП в подприбор, необходимо определить какие данные в него поступают и какие данные он возвращает. Это производится при помощи назначения соответствующих элементов управления и индикаторов терминалам ВП. Палитра коннекторов ВП находится в верхнем правом углу ВП (рис.2), для ее отображения необходимо щелкнуть на иконку ВП правой кнопкой мыши и нажать show connectors. Существует большое количество конфигураций (паттернов) палитры коннекторов (рис. 20). В одном ВП рекомендуется применять одинаковые паттерны (самый распространенный 4-2-4-2).

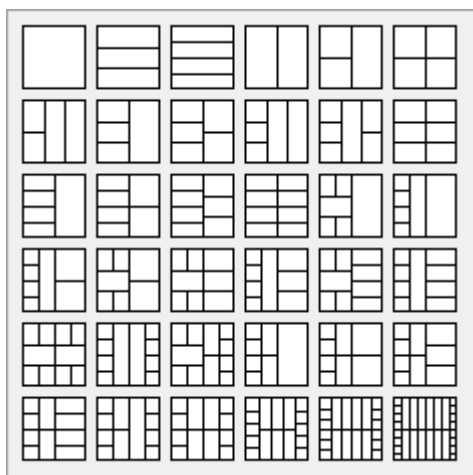


Рисунок 20 – Палитра коннекторов подприбора

Чтобы назначить соответствие элементов подпрограммы выводам коннекторов, необходимо подвести указатель мыши к контакту и нажать левую кнопку мыши. Активная клетка коннектора подсветится. Далее следует подвести указатель к требуемому элементу подпрограммы и нажать левую кнопку мыши. Цвет коннектора изменится в зависимости от типа данных элемента (рис. 21). Так, если элемент принимает значения действительного типа, то коннектор будет оранжевым, если это булевый тип - то зеленый и т.д.

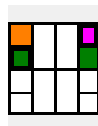


Рисунок 21 – Иконка коннекторов подприбора с подключенными элементами

Каждая подпрограмма может иметь свою уникальную иконку. Иконка – это изображение размера 32×32 пикселя с произвольной картинкой и/или текстом. Для того чтобы задать иконку ВП, необходимо

дважды кликнуть на его иконку в правом верхнем углу окна и переопределить ее при помощи встроенного редактора.

Иконки очень маленькие: всего 32×32 пикселя, поэтому они должны быть краткими, но понятными. Рекомендуется использовать символы и шрифты, которые будут легко читаться при таких размерах. Глиф – это графический, легко узнаваемый символ. К ним относятся, например, знаки дорожного движения. Это отличный пример совмещения графики и текста на ограниченном пространстве для сообщения важной информации. Библиотеки глифов, значков и других символов обычно есть в любом приложении и на многих сетевых ресурсах. Например, в зоне разработчиков (NI Developer Zone) такая библиотека есть по адресу www.ni.com/devzone/idnet/library/icon_art_glossary.htm.

Лучшие иконки состоят из глифа, лаконичного текста и окрашены в 2 или 3 цвета. Глиф занимает половину или 2/3 места, текст – от одного до трех слов или понятных сокращений.

10 Стиль программирования

Блок-диаграмма LabVIEW представляет собой уникальное средство отображения кода, отличное от большинства других языков программирования. Принцип потока данных реализует программу на таких элементах, как проводники, терминалы, структуры, узлы. У каждого элемента есть свое назначение и уникальные возможности. Они почти не ограничивают возможности и воображения программиста. В результате появилось множество стилей написания программ в LabVIEW.

Одну и ту же задачу можно решить множеством способом, но функционально программы будут эквивалентными. На первый взгляд кажется, что предпочтения, стиль программирования и оформление – личное дело каждого, но это только кажется. На самом деле все это напрямую влияет на качество программы.

Стиль программирования в LabVIEW прямо связан с простотой использования, эффективностью, надежностью программы и читабельностью, простотой изменения и структуры кода всего приложения.

При написании программ в LabVIEW необходимо придерживаться основных правил хорошего стиля:

- Располагать объекты плотно;
- Ограничивать размер блок-диаграммы одного подприбора одним экраном по ширине или высоте;
- Иконка и понятное описание должны быть у каждого подприбора;
- Избегать петель и изгибов проводников;

- Параллельные проводники должны проходить на одинаковом расстоянии;
- Избегать наложения объектов на блок-диаграмме;
- Ограничивать длину проводника: его начало и конец должны быть видны на экране;
- Добавлять метки для длинных проводников;
- Объединять связанные данные в кластеры;
- Всегда направлять поток данных слева направо.

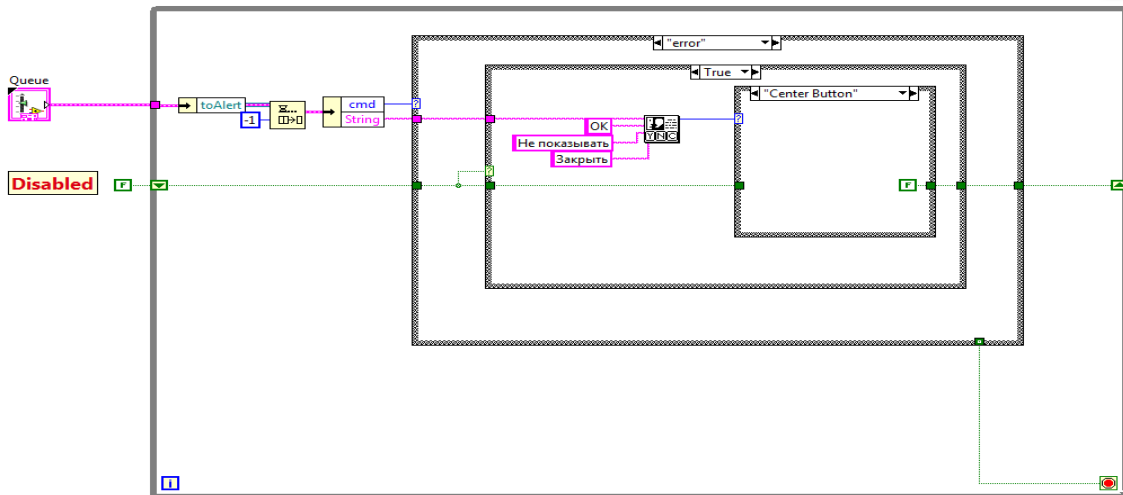


Рисунок 22 – Пример хорошего стиля программирования

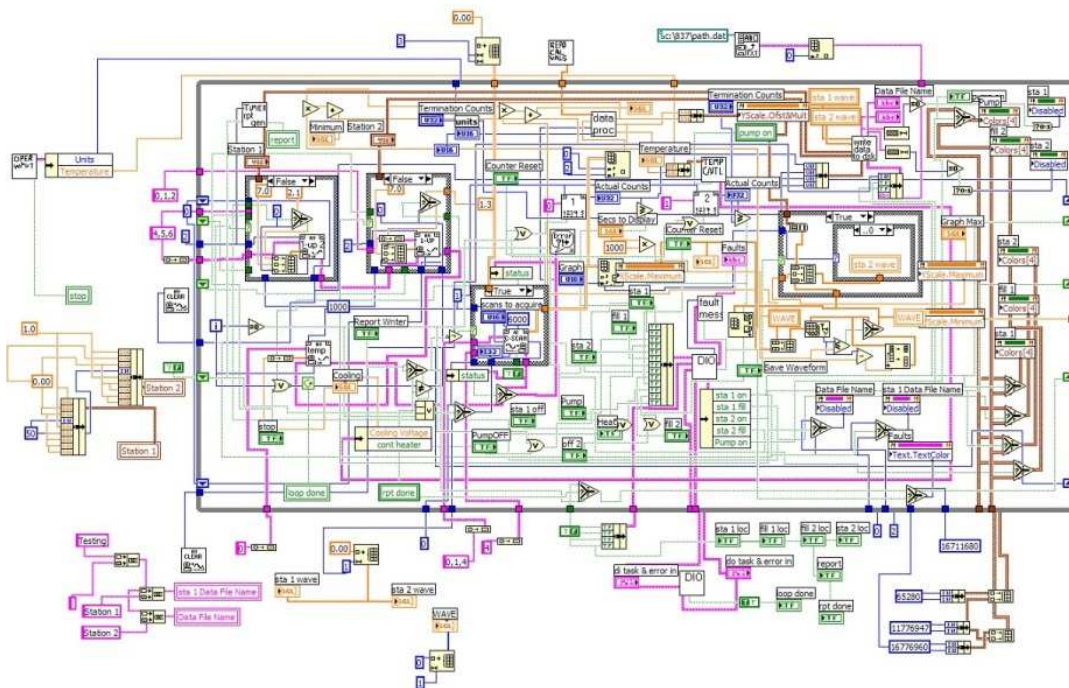


Рисунок 23 – Пример плохого стиля программирования

Соблюдение этих правил поможет повысить читаемость программ и сэкономить время на их поддержку в будущем. На рисунках 22 и 23 представлен пример хорошего и плохого стиля программирования соответственно. Как нетрудно догадаться, работу, выполненную в первом варианте, будет легче защитить и исправить.

Лабораторная работа №1: Вычисление по формулам

Задание

1. Разработать виртуальный прибор, который вычисляет значения двух эквивалентных числовых формул $y_1 \sim y_2$, $z_1 \sim z_2$ с указанными в варианте индивидуального задания значениями исходных данных;
2. Вычисление числовых формул y_1 , z_1 выполнить с помощью структуры Formula Node;
3. Вычисление числовых формул y_2 , z_2 выполнить с помощью базовых математических функций LabVIEW;

Примечание

Две числовых формулы эквивалентны, если для всех возможных значений переменных их математические значения равны;

Будем считать, что значения эквивалентны, если они отличаются не более чем на 10^{-5}

Содержание отчета

1. Название работы и номер варианта индивидуального задания;
2. Фамилия, имя, отчество, номер группы студента;
3. Текст задания, формулы в том виде, как они приведены в варианте индивидуального задания;
4. Области допустимых значений для переменных y_1 и y_2 ;
5. Распечатка блок-диаграммы виртуального прибора

Варианты индивидуального задания

№	Исходные данные	Формулы
1.	$m=2.3$ $\alpha=0.23$ $\beta=1.2$	$y_1 = \frac{m^2 - m - 6 - (m + 3)\sqrt{m^2 - 4}}{m^2 + m - 6 - (m - 3)\sqrt{m^2 - 4}}; \quad y_2 = \frac{-\sqrt{m + 2}}{\sqrt{m - 2}};$ $z_1 = (\cos \alpha - \cos \beta)^2 + (\sin \alpha - \sin \beta)^2; \quad z_2 = 4 \sin^2 \frac{\alpha - \beta}{2};$
2.	$x=5.3$ $\alpha=0.3$ $\beta=0.1$	$y_1 = \frac{x + \sqrt{x^2 - 4x}}{x - \sqrt{x^2 - 4x}} - \frac{x - \sqrt{x^2 - 4x}}{x + \sqrt{x^2 - 4x}}; \quad y_2 = \sqrt{x^2 - 4x};$ $z_1 = \sin(\alpha + \beta) \cdot \sin(\alpha - \beta) \cdot \sec^2 \alpha \cdot \sec^2 \beta; \quad z_2 = \operatorname{tg}^2 \alpha - \operatorname{tg}^2 \beta;$

3.	$m_1 = 0.47$ $m_2 = 2.47$ $\alpha = 0.1$	$y_1 = \frac{\sqrt{(2m+3)^2 - 24m}}{2\sqrt{m} - \frac{3}{\sqrt{m}}};$ <p><i>npu</i> $m \leq 1.5$ $y_1 \sim y_2$; <i>npu</i> $m > 1.5$ $y_1 \sim y_3$;</p> $y_2 = -\sqrt{m}; \quad y_3 = \sqrt{m};$ $z_1 = \frac{\cos \alpha + \sin \alpha}{\cos \alpha - \sin \alpha}; \quad z_2 = \operatorname{tg} 2\alpha + \sec 2\alpha;$
4.	$a = 3.5$ $b = -2.1$ $\alpha = 0.1$	$y_1 = \frac{a^2 - b^2}{a - b} - \frac{a^3 - b^3}{a^2 - b^2}; \quad y_2 = \frac{ab}{a + b};$ $z_1 = \frac{\cos 2\alpha}{\operatorname{ctg}^2 \alpha - \operatorname{tg}^2 \alpha}; \quad z_2 = \frac{1}{4} \sin^2(2\alpha);$
5.	$m_1 = 0.65$ $m_2 = 1.65$ $\alpha = 1.43$	$y_1 = \frac{m^5 + m^4 \cdot \sqrt[3]{2} + \sqrt[3]{4m^9}}{ m^3 - 1 - 1};$ <p><i>npu</i> $m > 1$ $y_1 \sim y_2$; <i>npu</i> $m \leq 1$ $y_1 \sim y_3$;</p> $y_2 = \frac{m^3}{m - \sqrt{2}}; \quad y_3 = -(m^2 + m\sqrt[3]{2} + \sqrt[3]{4});$ $z_1 = \frac{1 - 2\sin^2 \alpha}{1 + \sin 2\alpha}; \quad z_2 = \frac{1 - \operatorname{tg} \alpha}{1 + \operatorname{tg} \alpha};$
6.	$x = 0.3$ $\alpha = 0.77$	$y_1 = (\sqrt{1-x^2} + 1) \cdot \left(\frac{1}{\sqrt{1+x}} + \sqrt{1-x} \right); \quad y_2 = \sqrt{1+x};$ $z_1 = \frac{\cos(3\pi - 2\alpha)}{2\sin^2\left(\frac{5}{4}\pi + \alpha\right)}; \quad z_2 = \operatorname{tg}\left(\alpha - \frac{5}{4}\pi\right);$
7.	$x = 4.3$ $\alpha = 1.23$	$y_1 = \frac{x^4 - x^3 - x + 1}{x^3 - 5x^2 + 7x - 3} x - 3 ; \quad y_2 = x^2 + x + 1;$ $z_1 = \frac{\sin 4\alpha}{1 + \cos 4\alpha} \cdot \frac{\cos 2\alpha}{1 + \cos 2\alpha}; \quad z_2 = \operatorname{ctg}\left(\frac{3}{2}\pi - \alpha\right);$
8.	$a = 12.3$ $\alpha = 0.24$	$y_1 = \left(\frac{\sqrt{a}}{2} - \frac{1}{2\sqrt{a}} \right)^2 \left(\frac{\sqrt{a}-1}{\sqrt{a}+1} - \frac{\sqrt{a}+1}{\sqrt{a}-1} \right); \quad y_2 = \frac{1-a}{\sqrt{a}};$ $z_1 = \frac{\sin 2\alpha - \sin 3\alpha + \sin 4\alpha}{\cos 2\alpha - \cos 3\alpha + \cos 4\alpha}; \quad z_2 = \operatorname{tg} 3\alpha;$

9.	$m=1.8$ $\alpha=0.43$ $\beta=0.58$	$y_1 = \frac{4m(m + \sqrt{m^{2-1}})^2}{(m + \sqrt{m^2 - 1})^4 - 1}; \quad y_2 = \frac{1}{\sqrt{m^2 - 1}};$ $z_1 = \frac{tg 2\alpha + ctg 3\beta}{ctg 2\alpha + tg 3\beta}; \quad z_2 = \frac{tg 2\alpha}{tg 3\beta};$
10.	$a=2.3$ $\alpha=0.75$	$y_1 = \frac{\sqrt{(2a+1)^3} + \sqrt{(2a-1)^3}}{\sqrt{4a+2}\sqrt{4a^2-1}}; \quad y_2 = 4a - \sqrt{4a^2-1};$ $z_1 = \frac{\sin\left(\frac{\pi}{2} + 3\alpha\right)}{1 - \sin(3\alpha - \pi)}; \quad z_2 = ctg\left(\frac{5}{4}\pi + \frac{3}{2}\alpha\right);$
11.	$a=0.7$ $x=0.44$ $y=0.82$	$y_1 = \frac{1}{2(1+\sqrt{a})} + \frac{1}{2(1-\sqrt{a})} - \frac{a^2+2}{1-a^3}; \quad y_2 = \frac{-1}{a^2+a+1};$ $z_1 = \cos^4 x + \sin^2 y + \frac{1}{4}\sin^2 2x - 1; \quad z_2 = \sin(y+x) \cdot \sin(y-x);$
12.	$a=5.1$ $\alpha=0.1$	$y_1 = \frac{\sqrt{a}+1}{a\sqrt{a}+a+\sqrt{a}}; \quad y_2 = a-1;$ $z_1 = \frac{\sin \alpha - \sin 3\alpha + \sin 5\alpha}{\cos \alpha - \cos 3\alpha + \cos 5\alpha}; \quad z_2 = tg 3\alpha;$
13.	$a=5.3$ $b=2.1$ $\alpha=0.75$	$y_1 = \frac{(a^2 - b^2) \cdot (\sqrt[3]{a} - \sqrt[3]{b})}{\sqrt[3]{a^4} + \sqrt[3]{ab^3} - \sqrt[3]{a^3b} - \sqrt[3]{b^4}}; \quad y_2 = a - b;$ $z_1 = \cos 4\alpha - \sin 4\alpha \cdot ctg 2\alpha; \quad z_2 = \cos 2\alpha - 2\cos^2 \alpha;$
14.	$a=1.7$ $b=2.8$ $\alpha=0.22$	$y_1 = \frac{a^{\frac{1}{2}} + ab^{-1}}{a^{\frac{1}{3}} - a^{\frac{1}{6}}b^{\frac{1}{3}} + b^{\frac{2}{3}}} - \frac{a}{\sqrt[3]{b}}; \quad y_2 = a^{\frac{5}{6}};$ $z_1 = \cos 4\alpha \cdot tg 2\alpha - \sin 4\alpha; \quad z_2 = \frac{2tg \alpha}{tg^2 \alpha - 1};$
15.	$x=1.4$ $y=2.8$ $\alpha=0.66$ $\beta=0.82$	$y_1 = \frac{x^{\frac{2}{3}} + 2 \cdot \sqrt[3]{xy} + 4y^{\frac{2}{3}}}{(\sqrt[3]{x^4} - 8y \cdot \sqrt[3]{x}) : \sqrt[3]{xy}}; \quad y_2 = \frac{1}{\sqrt[3]{\frac{x}{y}} - 2};$ $z_1 = (\cos \alpha - \cos \beta)^2 - (\sin \alpha - \sin \beta)^2;$ $z_2 = -4\sin^2 \frac{\alpha - \beta}{2} \cdot \cos(\alpha + \beta);$

16.	$a=4.3$ $\alpha=0.43$	$y_1 = \frac{a^3 - 3a^2 + 4 + (a^2 - 4)\sqrt{a^2 - 1}}{a^3 + 3a^2 - 4 + (a^2 - 4)\sqrt{a^2 - 1}}; \quad y_2 = \frac{(a-2)\sqrt{a+1}}{(a+2)\sqrt{a-1}};$ $z_1 = \cos \alpha + \cos 2\alpha + \cos 6\alpha + \cos 7\alpha;$ $z_2 = 4 \cos \frac{\alpha}{2} \cdot \cos \frac{5}{2} \alpha \cdot \cos 4\alpha;$
17.	$b=4.8$ $\alpha=0.23$	$y_1 = \frac{b^2 - 3b - (b-1)\sqrt{b^2 - 4} + 2}{b^2 + 3b - (b+1)\sqrt{b^2 - 4} + 2} \cdot \sqrt{\frac{b+2}{b-2}}; \quad y_2 = \frac{1-b}{1+b};$ $z_1 = \frac{\cos 4\alpha + 1}{\operatorname{ctg} \alpha - \operatorname{tg} \alpha}; \quad z_2 = \frac{1}{2} \sin 4\alpha;$
18.	$x_1=2.8$ $x_2=4.8$ $\alpha=0.97$	$y_1 = \sqrt{\frac{4}{x} + \frac{1}{4x^{-1}} - 2} + \sqrt{\frac{1}{4x^{-1}} + \frac{1}{4x} + \frac{1}{2}};$ $\text{npu } x \leq 4 \quad y_1 \sim y_2; \quad \text{npu } x > 4 \quad y_1 \sim y_3;$ $y_2 = \frac{5}{2\sqrt{x}}; \quad y_3 = \frac{2x-3}{2\sqrt{x}};$ $z_1 = \sin^2\left(\frac{7}{8}\pi - 2\alpha\right) - \sin^2\left(\frac{9}{8}\pi - 2\alpha\right); \quad z_2 = \frac{\sin 4\alpha}{\sqrt{2}};$
19.	$x=1.4$ $y=2.8$ $\alpha=0.5$ $\beta=0.34$	$y_1 = \frac{\sqrt{x^3} + \sqrt{xy^2} - \sqrt{x^2y} - \sqrt{y^3}}{\sqrt[4]{y^5} + \sqrt[4]{x^4y} - \sqrt[4]{xy^4} - \sqrt[4]{x^5}}; \quad y_2 = -(\sqrt[4]{x} + \sqrt[4]{y});$ $z_1 = \operatorname{ctg}^2 \alpha - \operatorname{ctg}^2 \beta; \quad z_2 = \frac{\cos^2 \alpha - \cos^2 \beta}{\sin^2 \alpha \cdot \sin^2 \beta};$
20.	$a=5.1$ $\alpha=0.3$	$y_1 = \left(\frac{1+\sqrt{a}}{\sqrt{1+a}} - \frac{\sqrt{1+a}}{1+\sqrt{a}}\right)^2 - \left(\frac{1-\sqrt{a}}{\sqrt{1+a}} - \frac{\sqrt{1+a}}{1-\sqrt{a}}\right)^2;$ $y_2 = \frac{16a\sqrt{a}}{(1-a^2) \cdot (a-1)};$ $z_1 = (1 + \sec 2\alpha + \operatorname{tg} 2\alpha) \cdot (1 - \sec 2\alpha + \operatorname{tg} 2\alpha); \quad z_2 = 2\operatorname{tg} 2\alpha;$
21.	$a=15.1$ $\alpha=1.23$	$y_1 = \left(\frac{1+a+a^2}{2a+a^2} + 2 - \frac{1-a+a^2}{2a-a^2}\right)^{-1} (5-2a^2); \quad y_2 = \frac{4-a^2}{2};$ $z_1 = \cos \alpha + \sin \alpha + \cos 3\alpha + \sin 3\alpha;$ $z_2 = 2\sqrt{2} \cos \alpha \cdot \sin\left(\frac{\pi}{4} + 2\alpha\right);$

22.	$x=3.1$ $y=0.8$ $\alpha=0.81$	$y_1 = \frac{x}{x^2 + y^2} - \frac{y(x-y)^2}{x^4 - y^4}; \quad y_2 = \frac{1}{x+y};$ $z_1 = \cos^2\left(\frac{3}{8}\pi - \frac{\alpha}{4}\right) - \cos^2\left(\frac{11}{8}\pi + \frac{\alpha}{4}\right); \quad z_2 = \frac{\sqrt{2}}{2} \sin \frac{\alpha}{2};$
23.	$a=2.3$ $b=1.89$ $\alpha=0.23$	$y_1 = \frac{(2a-b)^2 + 2b^2 - 3ab}{2a^{-1} + b^2}; \quad y_2 = \frac{4a^2 - 3ab}{2 + ab^2}; \quad y_2 = a - b;$ $z_1 = \sin^6\left(\frac{\alpha}{2}\right) - \cos^6\left(\frac{\alpha}{2}\right); \quad z_2 = \frac{1}{4}(\sin^2 \alpha - 4)\cos \alpha;$
24.	$b=3.8$ $\alpha=0.28$	$y_1 = \frac{\sqrt{2b+2\sqrt{b^2-4}}}{\sqrt{b^2-4}+b+2}; \quad y_2 = \frac{1}{\sqrt{b+2}};$ $z_1 = \sin^2\left(\frac{15}{8}\pi - 2\alpha\right) - \cos^2\left(\frac{17}{8}\pi - 2\alpha\right); \quad z_2 = \frac{-\cos 4\alpha}{\sqrt{2}};$
25.	$p=0.7$ $\alpha=0.54$	$y_1 = \left(\left(1-p^2\right)^{\frac{1}{2}} - \left(1+p^2\right)^{\frac{1}{2}}\right)^2 + 2\left(1-p^4\right)^{\frac{1}{2}}; \quad y_2 = \frac{2}{(1-p^4)};$ $z_1 = \operatorname{tg} \alpha + \operatorname{ctg} \alpha + \operatorname{tg} 3\alpha + \operatorname{ctg} 3\alpha; \quad z_2 = \frac{8\cos^2 2\alpha}{\sin 6\alpha};$
26.	$a=3.5$ $b=0.72$ $\alpha=0.62$	$y_1 = \frac{\left(\sqrt{a^2 + a\sqrt{a^2 - b^2}} - \sqrt{a^2 - a\sqrt{a^2 - b^2}}\right)^2}{2\sqrt{a^2 b} \left(\sqrt{\frac{a}{b}} + \sqrt{\frac{b}{a}} - 2\right)};$ $y_2 = \frac{(\sqrt{a} + \sqrt{b})^2}{a - b};$ $z_1 = 1 - \frac{1}{4}\sin^2 2\alpha + \cos 2\alpha; \quad z_2 = \cos^2 \alpha + \cos^4 \alpha;$
27.	$a=12.3$ $\alpha=0.43$	$y_1 = \left(\frac{a+2}{\sqrt{2a}} - \frac{a}{\sqrt{2a+2}} + \frac{2}{a-\sqrt{2a}}\right) \cdot \frac{\sqrt{a}-\sqrt{2}}{a+2}; \quad y_2 = \frac{1}{\sqrt{a+\sqrt{2}}};$ $z_1 = \frac{\sin 2\alpha + \sin 5\alpha - \sin 3\alpha}{\cos \alpha + 1 - 2\sin^2 2\alpha}; \quad z_2 = 2\sin \alpha;$
28.	$a=8.6$ $b=1.3$ $c=3.3$ $\alpha=0.75$	$y_1 = (a^2 - b^2 - c^2 + 2bc) \cdot \frac{a+b-c}{a+b+c}; \quad y_1 = (a+c)^2 - b^2$ $z_1 = 2\sin^2(3\pi - 2\alpha) \cdot \cos^2(5\pi + 2\alpha);$ $z_2 = \frac{1}{4} - \frac{1}{4}\sin\left(\frac{5}{2}\pi - 8\alpha\right);$

29.	$a = 6.3$ $\alpha = 0.1$ $\beta = 0.7$	$y_1 = \frac{1 + 2a^{\frac{1}{4}} - a^{\frac{1}{2}}}{1 - a + 4a^{\frac{3}{4}} - 4a^{\frac{1}{2}}} + \frac{a^{\frac{1}{4}} - 2}{\left(a^{\frac{1}{4}} - 1\right)^2}; \quad y_2 = \frac{1}{\sqrt[4]{a} - 1};$ $z_1 = \frac{\sin \alpha + \cos(2\beta - \alpha)}{\cos \alpha - \sin(2\beta - \alpha)}; \quad z_2 = \frac{1 + \sin 2\beta}{\cos 2\beta};$
30.	$m = 0.4$ $n = 2.1$ $\alpha = 0.43$	$y_1 = \frac{(m-1)\sqrt{m} - (n-1)\sqrt{n}}{\sqrt{m^3n + mn + m^2} - m}; \quad y_2 = \frac{\sqrt{m} - \sqrt{n}}{m};$ $z_1 = \frac{\operatorname{tg} \alpha - \operatorname{sec} \alpha}{\cos \alpha - \operatorname{ctg} \alpha}; \quad z_2 = \operatorname{tg} \alpha \cdot \operatorname{sec} \alpha;$

Лабораторная работа №2: Попадание точки в заданную область

Задание

- Область заштрихована и ограничена линиями контура. Описать математически систему неравенств, обеспечивающую попадание произвольной точки в заштрихованную область;
- Программу оформить в виде подприбора (структура подприбора показана на рисунке 24):
 - Входы - координата X, координата Y;
 - Выходы - результат логического выражения и строковый ответ «точка попадает в область» / «точка не попадает в область»;

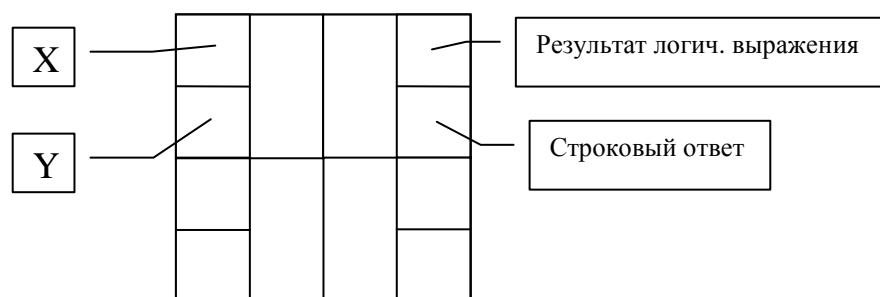
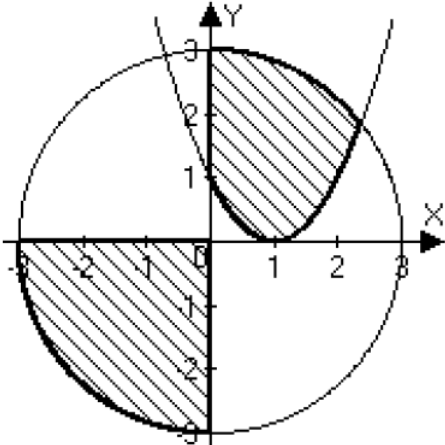
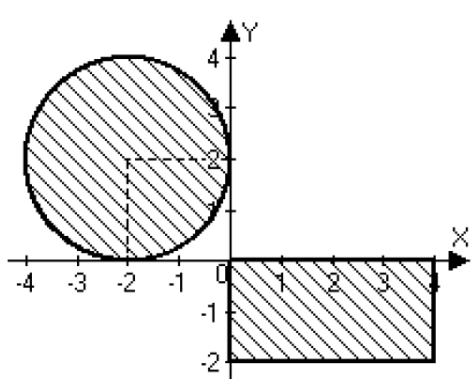
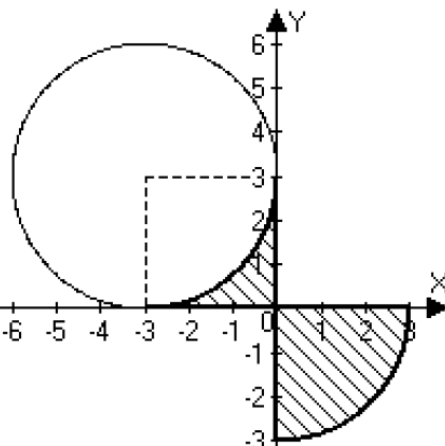
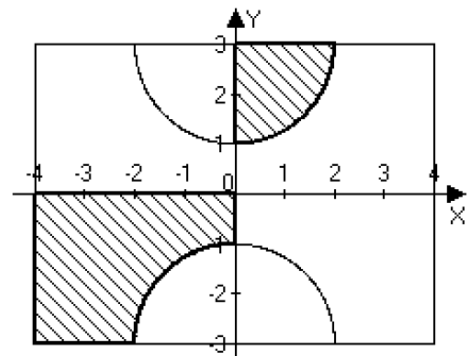
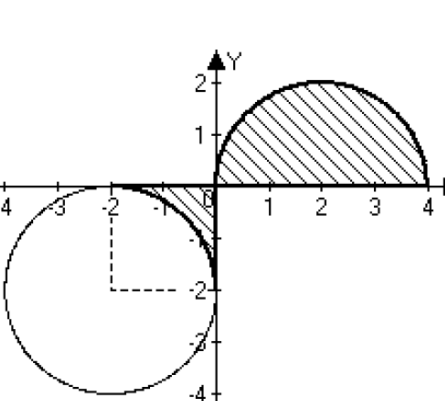
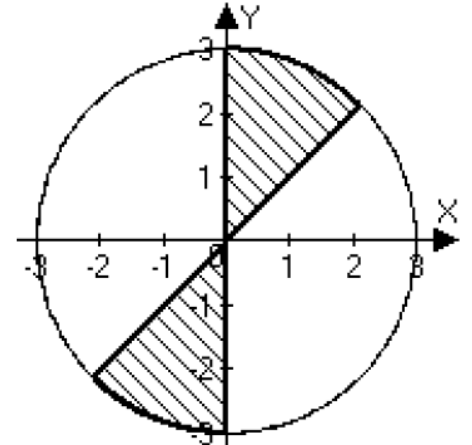


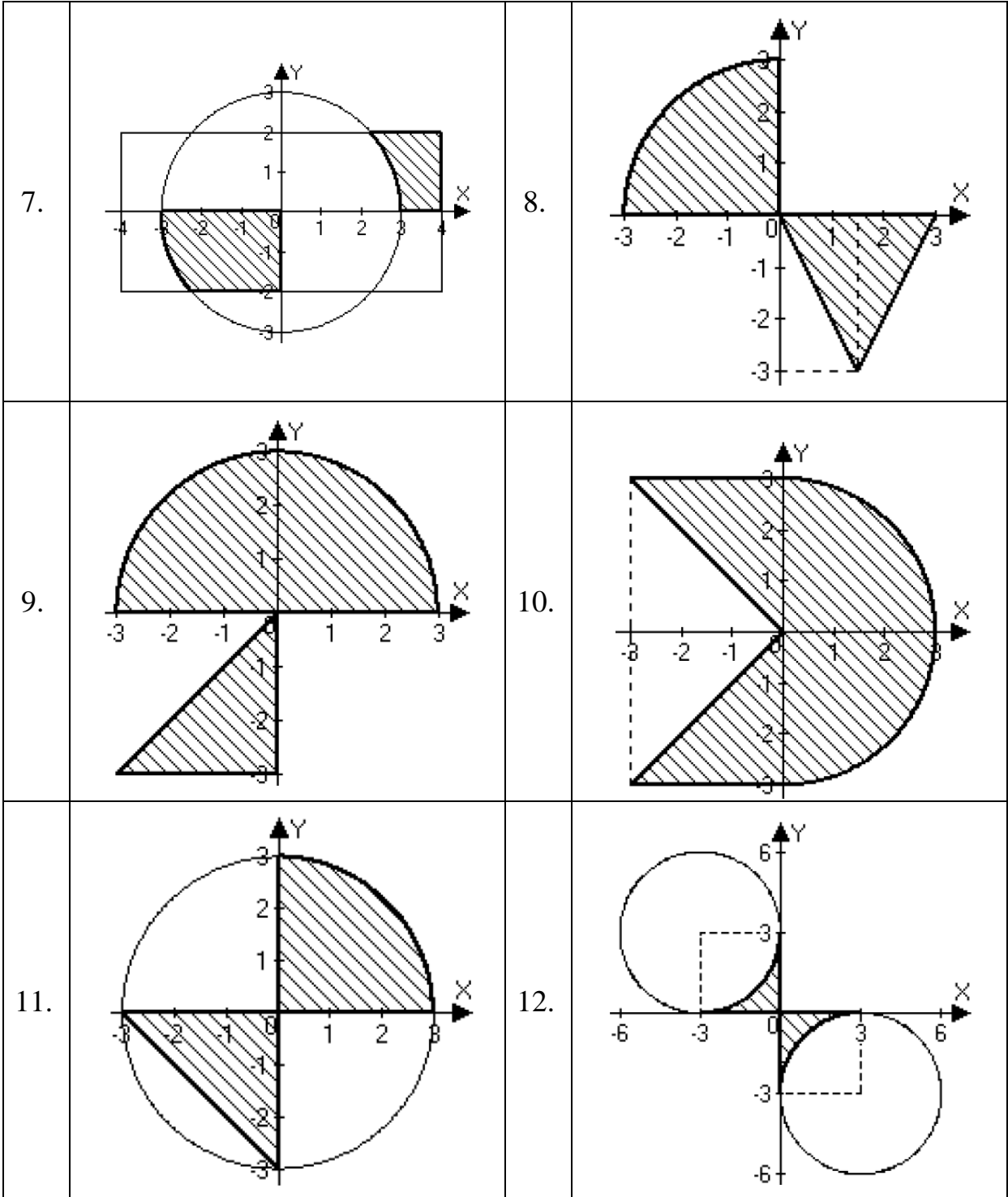
Рисунок 24 – Структура подприбора

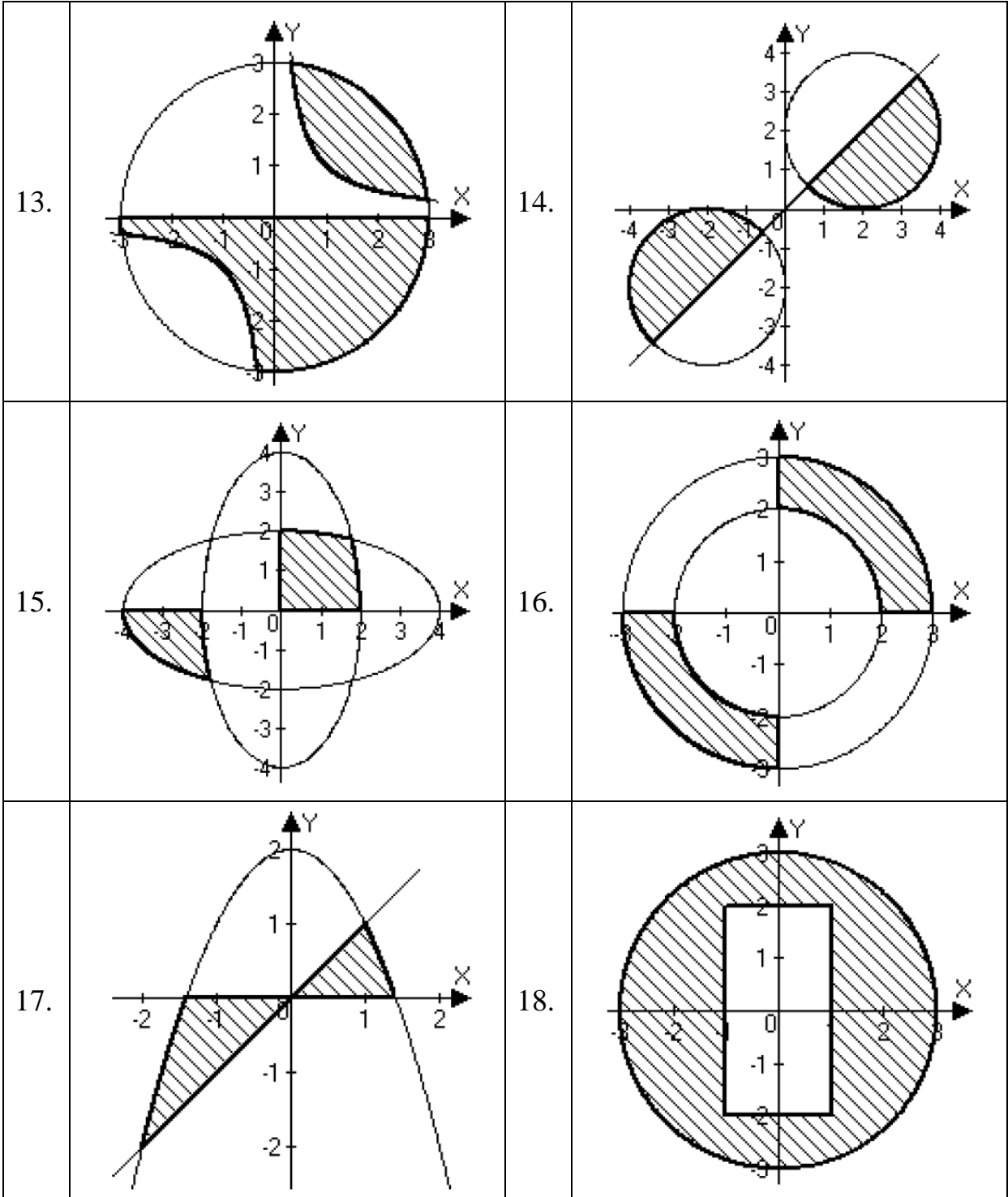
Содержание отчета

- Название работы, номер варианта индивидуального задания.
- Фамилия, имя, отчество и номер группы студента.
- Рисунок в том виде, как он приведен в варианте задания.
- Уравнения всех линий, ограничивающих заштрихованную область.
- Система неравенств координат всех точек заштрихованной области.
- Блок – диаграмма программы.

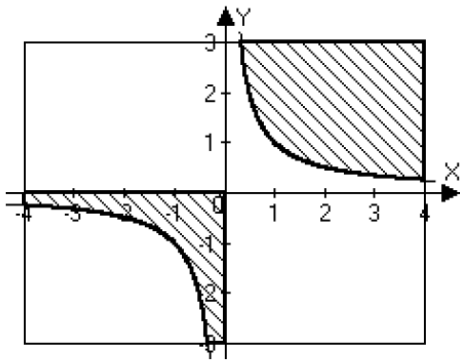
Варианты индивидуального задания

№	Задание	№	Задание
1.		2.	
3.		4.	
5.		6.	

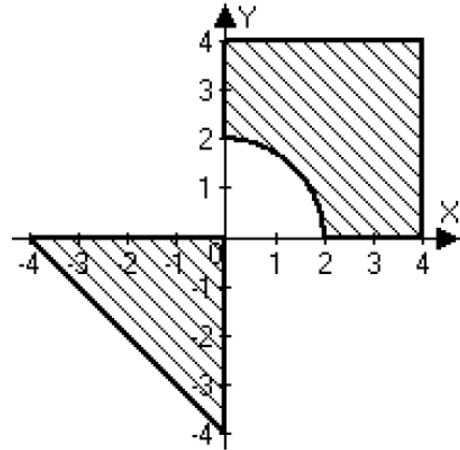




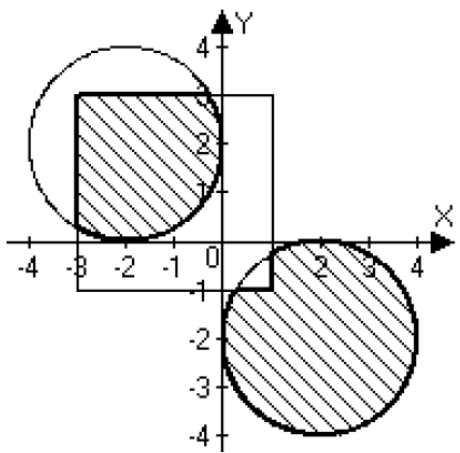
19.



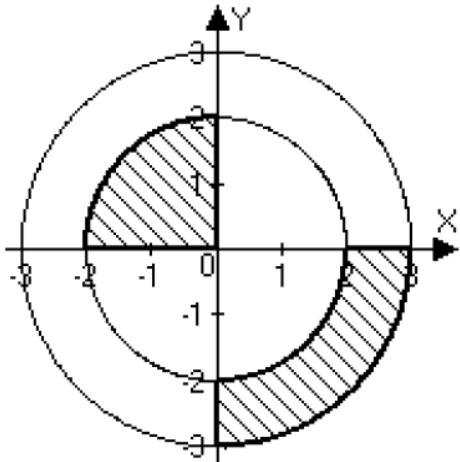
20.



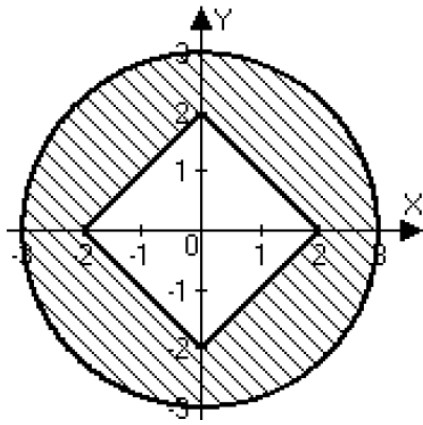
21.



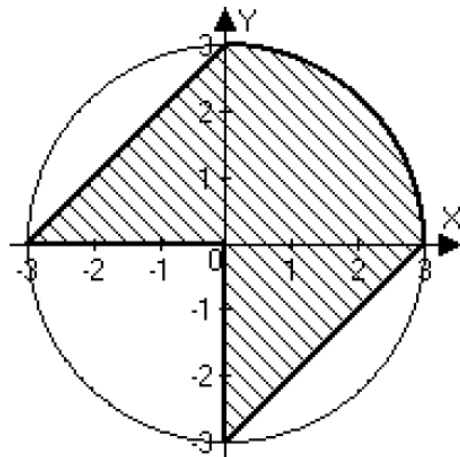
22.



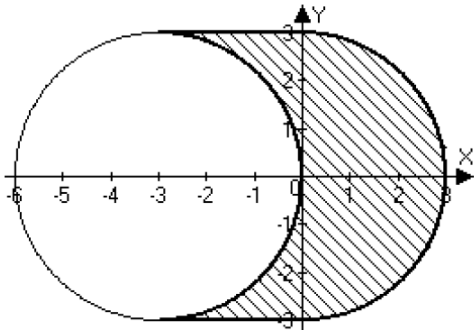
23.



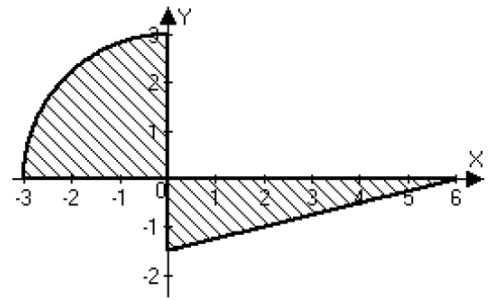
24.



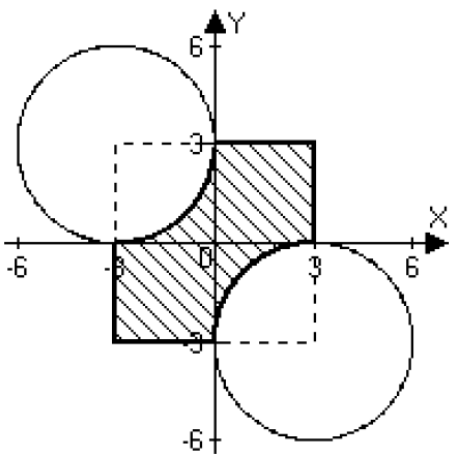
25.



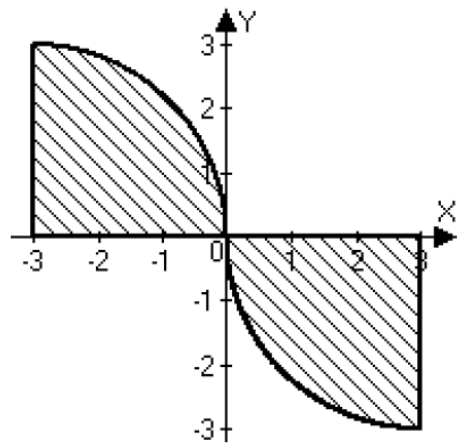
26.



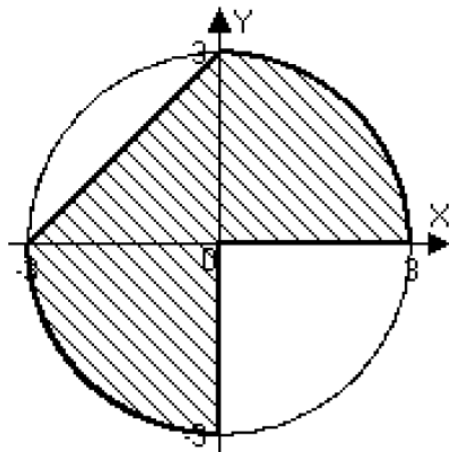
27.



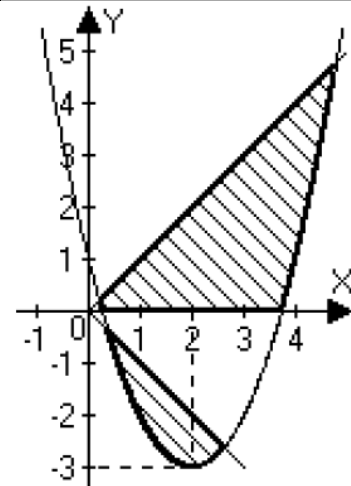
28.



29.



30.



Лабораторная работа №3: Условные операторы

Задание

Разработать виртуальный прибор для решения задачи, согласно заданному варианту индивидуального задания.

Содержание отчета

1. Название работы, номер варианта индивидуального задания.
2. Фамилия, имя, отчество и номер группы студента.
3. Текст задания индивидуального варианта.
4. Блок – диаграмма программы.
5. Блок – схема программы.

Варианты индивидуальных заданий

№	Задание
1.	Дано трехзначное число. Определить: а) верно ли, что все его цифры одинаковые; б) есть ли среди его цифр одинаковые.
2.	Дано двухзначное число. Определить: а) входит ли в него цифра 5; б) входит ли в него цифра a .
3.	Ввести с клавиатуры координаты точек $A_1(x_1, y_1)$, $A_2(x_2, y_2)$, $A_3(x_3, y_3)$. Определить, можно ли через точку A_3 провести прямую, перпендикулярную прямой, проходящую через точки A_1 и A_2 . Ответ вывести в виде сообщения.
4.	Написать программу, которая при вводе числа в диапазоне от 1 до 99 добавляет к нему слово «копейка» в правильной форме. Например: «1 копейка», «5 копеек», «42 копейки».
5.	Ввести с клавиатуры координаты точки $A(x, y)$. Определить, попадает ли точка A в область, ограниченной окружностью $1 = y^2 + (x - 1)^2$. Ответ вывести в виде сообщения.
6.	Определить, является ли шестизначное число «счастливым» (сумма первых трех цифр равна сумме последних трех цифр).
7.	Дано целое число от 1 до 365. Определить, какой день недели выпадает на это число, если 1 января – понедельник.
8.	Траектория снаряда, вылетающего из орудия под углом α с начальной скоростью v_0 , определяется уравнениями $x = v_0 t \cos \alpha$, $y = v_0 t \sin \alpha - 98t^2 / 2$. Определить попадет ли снаряд в цель размером h , находящуюся в плоскости его полета на расстоянии R и высоте H .

9.	Дано трехзначное число. Определить: а) является ли сумма его цифр двухзначным числом; б) является ли произведение его цифр трехзначным числом.
10.	Дано трехзначное число. Определить: а) является ли произведение его цифр больше числа b ; б) кратна ли сумма его цифр трем.
11.	Даны вещественные числа a , b , c , $a \neq 0$. Решить уравнение $ax^2 + bx + c = 0$. Учесть возможность равенства корней.
12.	Ввести с клавиатуры координаты точки $A(x, y)$. Определить, лежит ли точка A в области, ограниченной параболой $y = 2 - x^2$ и прямой $y = -2$. Ответ вывести в виде сообщения.
13.	Дано двухзначное число. Определить: а) входят ли в него цифры 3 и 7; б) входят ли в него цифры (4 и 8) или цифра 9.
14.	Рейтинг бакалавра заочного отделения при поступлении в магистратуру определяется средним баллом по диплому, умноженным на коэффициент стажа работы по специальности, который равен: нет стажа – 1, меньше 2 лет – 13, от 2 до 5 лет – 16. Составить программу расчета рейтинга при заданном среднем балле диплома (от 3 до 5) и вывести сообщение о приеме в магистратуру при проходном балле 45.
15.	Даны целочисленные координаты трех вершин прямоугольника, стороны которого параллельны координатным осям. Определить координаты четвертой вершины.
16.	Дано четырехзначное число. Определить: а) равна ли сумма двух первых его цифр сумме двух последних; б) кратна ли 7 сумма его цифр.
17.	Дано четырехзначное число. Определить: а) кратно ли произведение его цифр трем; б) кратно ли числу a произведение его цифр.
18.	Даны вещественные положительные числа a , b , c , d . Выяснить, можно ли прямоугольник со сторонами a , b поместить в прямоугольник со сторонами c , d так, чтобы стороны одного прямоугольника были параллельны или перпендикулярны сторонам другого прямоугольника.
19.	Даны вещественные положительные числа a , b , c , x , y . Выяснить, пройдет ли кирпич размером $a \times b \times c$ в отверстие размером $x \times y$ при параллельном или перпендикулярном расположении ребер кирпича сторонам отверстия.
20.	Написать программу, которая при вводе числа в диапазоне от 1 до 20 добавляет к нему слово «гривна» в правильной форме. Например, «1 гривна», «2 гривны», «5 гривен».

21.	Ввести число от 1 до 31. Определить ближайшую (следующую) дату полнолуния или новолуния (лунных месяц содержит 28 дней), если последнее полнолуние было 27 августа.
22.	Определить проводник с максимальным сопротивлением, если даны три проводника с удельным сопротивлением r_1, r_2, r_3 и площадью сечения s_1, s_2, s_3 .
23.	Дано натуральное четырехзначное число. Выяснить, является ли оно палиндромом (читается одинаково слева направо и справа налево).
24.	Определить максимальный предельный заряд q_{\max} и минимальный потенциал φ_{\min} для трех шаров с диаметром r_1, r_2, r_3 . Напряженность поля, при которой начинается ударная ионизация, равна $E = 3 \text{ МВ/м}$ $(q_{\max} = \frac{E}{4\pi\epsilon_0 r^2}, \epsilon_0 = 8.85 \cdot 10^{-12} \text{ Ф/м}; \epsilon = 1; \varphi_{\min} = Er)$.
25.	Написать программу для сравнения радиусов окружностей, которые описывают протон и альфа-частица, если влетают в однородное магнитное поле перпендикулярно линиям индукции: а) при одинаковой скорости; б) при одинаковой энергии. Вывести сообщение. Заряд альфа-частицы в 2 раза больше заряда протона, а масса в 4 раза больше.
26.	Локатор может быть ориентирован на одну из сторон света («С» – север, «Ю» – юг, «З» – запад, «В» – восток) и может принимать команды: «1» – поворот направо (90°), «-1» – поворот налево (-90°), «2» – поворот в обратную сторону (180°). Исходная ориентация локатора «С». Определить ориентацию локатора после выполнения последовательности вводимых команд $N1, N2$.
27.	Ввести длину волны и определить попадает ли она в рабочий диапазон длин волн приемника, если емкость конденсатора в его колебательном контуре плавно изменяется от 50 до 500 пФ, а индуктивность катушки постоянна и равна 20 мкГн.
28.	Дано трехзначное число a . Определить: а) является ли произведение его цифр меньше числа a ; б) кратна ли 5 сумма его цифр.
29.	Написать программу, определяющую условие, когда сдвиг фаз между током и напряжением при последовательном соединении сопротивления, емкости и конденсатора будет положительным, отрицательным и равным нулю.
30.	Ввести с клавиатуры координаты точки $A(x_1, y_1)$. Определить, попадает ли точка A в область, ограниченную линиями $y = x$, $y = -x$ и $y = 1$. Ответ вывести в виде сообщения.

Лабораторная работа №4: Решение задач с одномерными массивами

Задание

1. Разработать виртуальный прибор, который включает в себя два подприбора:
 - Первый подприбор генерирует одномерный массив из 20 элементов случайных *целых* чисел в диапазоне значений [-100: 100], либо выдает заранее введенный одномерный массив.
 - Поведение первого подприбора определяется входом булева типа, где значение ИСТИНА – генерирование массива случайных элементов, ЛОЖЬ – выдача введенного массива.
 - Второй подприбор осуществляет операции над одномерным массивом согласно варианту индивидуального задания.
2. На лицевой панели виртуального прибора разместить элемент управления выбора массива, поле ввода одномерного массива, вывод исходного массива, результаты промежуточных вычислений и вычисления по выражению в соответствии с вариантом индивидуального задания. Структура виртуального прибора показана на рисунке 25

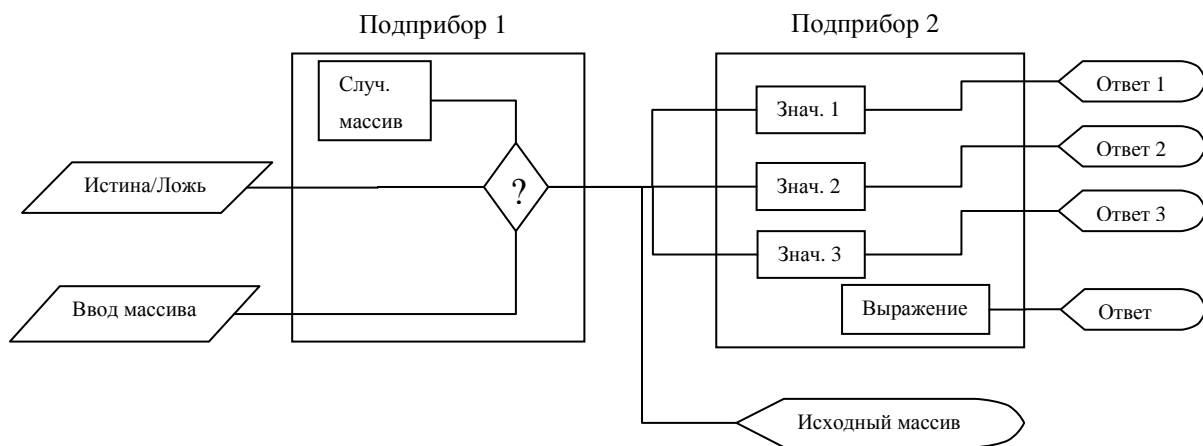


Рисунок 25 – Структура виртуального прибора

3. Все вычисления проводить, используя циклы (без применения встроенных функций LabVIEW для работы с массивами кроме Array Size и Index Array)
4. Дополнительные параметры, если они есть в варианте задания, вводить через элементы управления на лицевой панели.

Содержание отчета

1. Название работы, номер варианта индивидуального задания.
2. Фамилия, имя, отчество и номер группы студента.
3. Текст варианта индивидуального задания.
4. Блок – диаграмма программы.

Варианты индивидуального задания

№	Выражение	Задание
1.	$B + \frac{A}{C+1}$	А – сумма отрицательных элементов с четными номерами; В – максимальный элемент среди N первых элементов; С – среднее геометрическое положительных элементов.
2.	$\frac{R+Q+S}{R \cdot Q \cdot S + 2}$	R – произведение положительных элементов с нечетными номерами; Q – последний положительный элемент с четным номером; S – среднее гармоническое положительных элементов
3.	$\left(H + \frac{E}{H+1} \right) G$	H – количество нулей среди N последних элементов; E – номер минимального элемента; G – среднее квадратичное элементов с четными номерами.
4.	$U + \frac{1}{ S+T +1}$	S – сумма положительных элементов, меньших D; T – минимальный по модулю ненулевой элемент (со знаком); U – среднее арифметическое N первых элементов.
5.	$Z + \frac{X}{10+Y}$	X – количество элементов со значениями из интервала [A, B]; Y – модуль минимального элемента; Z – среднее арифметическое элементов с нечетными номерами.
6.	$\frac{U}{R+1} + S$	U – произведение элементов с четными номерами; R – номер максимального по модулю элемента; S – среднее арифметическое N последних элементов.
7.	$\frac{A \cdot B}{C+2}$	A – произведение ненулевых элементов; B – последний отрицательный элемент с нечетным номером; C – среднее арифметическое положительных элементов.

8.	$\frac{R}{Q+1} + S$	R – сумма всех элементов; Q – номер первого нулевого элемента; S – среднее геометрическое положительных среди N последних элементов.
9.	$\frac{H}{(E+1)(H+1)} + G$	H – сумма положительных среди N первых элементов; E – номер первого отрицательного элемента; G – среднее арифметическое отрицательных элементов.
10.	$Y + \frac{X}{10+Z}$	X – количество элементов, меньших D, с нечетными номерами; Y – максимальный по модулю элемент (со знаком); Z – среднее квадратичное положительных элементов.
11.	$C + \frac{A+B}{A \cdot B + 1}$	A – произведение положительных элементов; B – последний положительный элемент с четным номером; C – среднее квадратичное N первых элементов.
12.	$\frac{T \cdot V + U}{U + 1}$	V – сумма отрицательных элементов, больших D, (D<0); T – номер максимального элемента; U – среднее геометрическое положительных среди N первых элементов.
13.	$(U+T)(S+2)$	S – сумма модулей отрицательных элементов; T – номер минимального по модулю ненулевого элемента; U – среднее квадратичное N последних элементов.
14.	$\frac{H}{(E+1)(H+1)} + G$	H – произведение модулей отрицательных элементов; E – номер последнего положительного элемента; G – среднее арифметическое элементов с четными номерами.
15.	$\frac{X+Z}{Y+2}$	X – сумма элементов, больших D, с четными номерами; Y – номер максимального отрицательного элемента; Z – среднее квадратичное всех элементов.
16.	$B + \frac{A}{C+1}$	A – среднее гармоническое отрицательных элементов; B – номер второго нулевого элемента; C – количество отрицательных элементов с четными номерами.

17.	$C + \frac{A}{10} + \frac{B}{10+A}$	A – сумма положительных элементов; B – номер минимального положительного элемента; C – среднее гармоническое ненулевых элементов.
18.	$\frac{X+Z}{Y+2}$	X – количество положительных элементов с четными номерами; Y – номер последнего нулевого элемента; Z – среднее гармоническое ненулевых элементов, больших D.
19.	$\frac{A \cdot B}{C+2}$	A – произведение отрицательных элементов; B – максимальный по модулю элемент (со знаком); C – среднее арифметическое модулей отрицательных элементов.
20.	$\left(H + \frac{E}{H+1}\right)G$	H – количество отрицательных элементов; E – минимальный элемент среди N последних элементов; G – среднее квадратичное элементов с четными номерами.
21.	$U + \frac{1}{S+T+1}$	S – количество положительных элементов; T – минимальный положительный элемент; U – среднее арифметическое элементов.
22.	$C + \frac{A}{10} + \frac{B}{10+A}$	A – первый положительный элемент с четным номером; B – сумма элементов с нечетными номерами; C – среднее гармоническое ненулевых среди N первых элементов.
23.	$\frac{U}{R+1} + S$	U – произведение ненулевых среди N первых элементов; R – номер максимального по модулю элемента; S – среднее геометрическое положительных элементов.
24.	$\frac{R}{Q+1} + S$	R – максимальный элемент с четным номером; Q – количество ненулевых элементов; S – среднее гармоническое положительных элементов.
25.	$(U+T)(S+2)$	S – сумма N последних элементов; T – максимальный отрицательный элемент; U – среднее квадратичное элементов с четными номерами.
26.	$C + \frac{A+B}{A \cdot B+1}$	A – количество элементов, больших D; B – первый положительный элемент с четным номером; C – среднее арифметическое N первых элементов.

27.	$\frac{H+1}{(C+2)E}$	Н – сумма элементов со значениями из интервала [А, В]; Е – последний положительный элемент с четным номером; С – среднее арифметическое положительных элементов.
28.	$\frac{(H+E) \cdot C}{(E+C+4)}$	Н – произведение элементов со значениями из интервала [А, В]; Е – номер минимального по модулю ненулевого элемента; С – среднее геометрическое положительных среди N последних элементов.
29.	$\frac{R+Q+S}{R \cdot Q \cdot S + 2}$	R – количество элементов, меньших D; Q – модуль минимального элемента; S – среднее квадратичное положительных элементов.
30.	$\frac{R}{R+1} + U + T$	U – минимальный элемент с номером из интервала [K, L]; T – среднее арифметическое элементов с нечетными номерами. R – количество нулей среди N последних элементов.

Лабораторная работа №5: Операторы цикла с пред и постусловием

Задание

Разработать виртуальный прибор для решения задачи, согласно заданному варианту индивидуального задания.

Содержание отчета

1. Название работы, номер варианта индивидуального задания.
2. Фамилия, имя, отчество и номер группы студента.
3. Текст задания индивидуального варианта.
4. Блок – диаграмма программы.
5. Блок – схема программы.

Варианты индивидуальных заданий

№	Задание
1.	Два натуральных числа называют дружественными, если каждое из них равно сумме всех делителей другого, кроме самого этого числа. Найти все пары дружественных чисел, лежащих в диапазоне от 200 до 300.

2.	Дано натуральное число n . Среди чисел $1, 2, \dots, n$ найти все такие, запись которых совпадает с последними цифрами записи их квадрата.
3.	Назовём натуральное число палиндромом, если его запись читается одинаково как с начала так и с конца (пример: 4884, 393, 1, 22). Найти все меньшие 100 натуральных числа, которые являются палиндромами.
4.	Написать программу поиска суммы последовательности положительных чисел, вводимых с клавиатуры, предшествующих первому введенному нулю. Контрольный пример: 1,2,3,-4,5,-2,0.
5.	Натуральное число называется совершенным, если оно равно сумме всех своих делителей за исключением самого себя. Например, $6 = 1 + 2 + 3$. Дано натуральное число n . Получить все совершенные числа, меньшие n .
6.	Написать программу поиска чисел, лежащих в интервале от -5 до 5, в последовательности чисел, вводимых с клавиатуры, предшествующих первому введенному нулю. Контрольный пример: 1,10,-4,5,-16,-5,0.
7.	Вычислить сумму чисел от 1 до N , возведенных в степень M . Возведение в степень оформить как многократное умножение.
8.	Определить 20-е число Фибоначчи. Где последовательность чисел Фибоначчи $\{F_n\}$ задается линейным рекуррентным соотношением: $F_0 = 0, F_1 = 1, F_{n+1} = F_n + F_{n-1}, n \in N$.
9.	Написать программу поиска произведения последовательности отрицательных чисел, вводимых с клавиатуры, предшествующих первому введенному нулю. Контрольный пример: 1,2,3,-4,5,-2,0.
10.	Ежемесячная стипендия студента составляет X руб., а расходы на проживание превышают ее и составляют Y руб. в месяц. Рост цен ежемесячно увеличивает расходы на 3%. Определить, какую нужно иметь сумму денег, чтобы прожить учебный год (10 месяцев), используя только эти деньги и стипендию.
11.	Дано натуральное число n ($n < 9999$) и число m . Найти сумму m -последних цифр числа n .
12.	Найти все четырехзначные числа, у которых все цифры различны.
13.	Назовём натуральное число палиндромом, если его запись читается одинаково как с начала так и с конца (пример: 4884, 393, 1, 22). Найти все меньшие 100 натуральных числа, которые при возведении в квадрат дают палиндром.
14.	Остров Манхэттен был приобретен поселенцами за \$24 в 1626 г. Каково было бы в настоящее время состояние их счета, если бы эти 24 доллара были помещены тогда в банк под 6% годового дохода?
15.	Натуральное число из n цифр является число Армстронга, если сумма его цифр возведенных в n -ую степень равна самому числу. Получите все эти числа состоящие из трех и четырех цифр (пример $1^3 + 5^3 + 3^3 = 153$).

16.	Написать программу поиска чисел, лежащих в интервале от 3 до 13, в последовательности чисел, вводимых с клавиатуры, предшествующих первому введенному отрицательному числу. Контрольный пример: 1,3,16,7,13,10,2,-1 .
17.	Найти в промежутке от 1 до 1000 числа, у которых пять делителей.
18.	Написать программу поиска суммы последовательности отрицательных чисел, вводимых с клавиатуры, предшествующих первому введенному нулю. Контрольный пример: 1,2,3,-4,5,-2,0.
19.	У студента имеются накопления S руб. Ежемесячная стипендия составляет X руб., а расходы на проживание превышают ее и составляют Y руб. в месяц. Рост цен ежемесячно увеличивает расходы на 3%. Определить, сколько месяцев сможет прожить студент, используя только накопления и стипендию.
20.	Найти все числа кратные семи, и сумма цифр которых также кратна семи.
21.	Дано натуральное число. Найти сколько раз в нем встречается каждая цифра.
22.	Перевести целое число из десятичной системы счисления в восьмеричную, используя алгоритм деления на 8.
23.	Написать программу поиска произведения последовательности чисел, вводимых с клавиатуры, предшествующих первому введенному отрицательному числу. Контрольный пример: 1,2,3,4,5,-2.
24.	Перевести целое число из десятичной системы счисления в двоичную, используя алгоритм деления на 2.
25.	Для чисел от 1 до 1000 . Найти все нечетные сотни в которой есть повторение чисел.
26.	Написать программу поиска произведения последовательности чисел, вводимых с клавиатуры, предшествующих первому введенному нулю. Контрольный пример: 1,2,3,-4,5,-2,0.
27.	Каждая бактерия делится на две в течение одной минуты. В начальный момент имеется одна бактерия. Составьте программу, которая рассчитывает количество бактерий на заданное вами целое значение момента времени (15 минут, 7 минут и т.п.).
28.	Для чисел от 1 до 1000, возвести в куб каждый третий десяток каждой второй сотни.
29.	Перевести целое число из десятичной системы счисления в шестнадцатеричную, используя алгоритм деления на 16.
30.	Билет называют «счастливым», если в его номере сумма первых трех цифр равна сумме последних трех. Подсчитать число тех «счастливых» билетов, у которых сумма трех цифр равна 13. Номер билета может быть от 000000 до 999999 .

Лабораторная работа №6: Решение задач с двумерными массивами

Задание

1. Разработать виртуальный прибор, который:
 - генерирует двумерный массив размера $M \times N$ из случайных чисел в диапазоне $[-100: 100]$ в соответствии с вариантом индивидуального задания. Параметры размера матрицы M и N вводятся с клавиатуры.
 - выводит на лицевую панель исходную матрицу.
 - дополнительные параметры, если они есть в варианте индивидуального задания, вводятся через элементы управления на лицевой панели.
 - выполняет действия в соответствии с индивидуальным заданием с использованием вложенных циклов или встроенных функций LabVIEW для работы с массивами.
 - выводит результаты работы программы на лицевую панель, поясняя каждое значение формулировками из индивидуального задания.
2. Программа должна иметь минимум два подприбора.

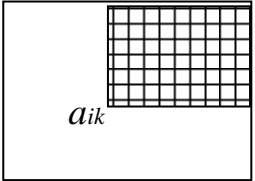
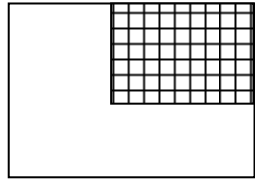
Содержание отчета

1. Название работы, номер варианта индивидуального задания.
2. Фамилия, имя, отчество и номер группы студента.
3. Текст варианта индивидуального задания.
4. Блок – диаграмма программы.

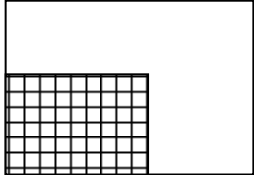
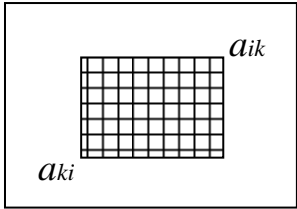
Вариант индивидуального задания

Обе части задания выполняются для одной и той же матрицы.

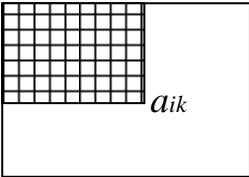
№	Задание
1.	<p>I. Сформировать одномерный массив из элементов, расположенных по «периметру» матрицы (элементы первой и последней строк и первого и последнего столбца). Элементы «периметра» выбирать по часовой стрелке, начиная с верхнего левого элемента (элемента a_{11}). Вывести сформированный одномерный массив.</p> <p>II. Составить и вывести одномерный массив из средних квадратичных значений элементов каждого столбца матрицы.</p>

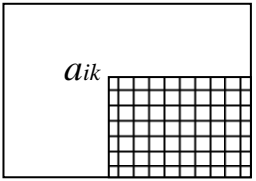
2.	<p>I. Сформировать матрицу B из элементов матрицы A. Каждый элемент b_{ik} должен быть равен сумме элементов матрицы A из области, определяемой индексами i, k в соответствии с рисунком. Вывести результирующую матрицу.</p> <p>II. Составить и вывести одномерный массив из сумм положительных элементов с четными номерами строк в каждом столбце матрицы A.</p> 
3.	<p>I. Отобрать в матрицу B только те строки матрицы A, элементы которых расположены в порядке возрастания. Если в матрице A таких строк нет, вывести сообщение, иначе вывести:</p> <ul style="list-style-type: none"> • номера строк матрицы A, помещенных в матрицу B; • полученную матрицу B. <p>II. Найти и вывести сумму положительных элементов матрицы A с двумя нечетными индексами.</p>
4.	<p>I. Если сумма положительных элементов матрицы больше произведения модулей ее отрицательных элементов, сформировать одномерный массив из положительных элементов матрицы, иначе массив формировать из ее отрицательных элементов. При формировании одномерного массива матрицу просматривать построчно. Вывести:</p> <ul style="list-style-type: none"> • вычисленные значения суммы и произведения; • полученный одномерный массив. <p>II. Составить и вывести одномерный массив из максимальных отрицательных элементов среди второй половины каждой строки матрицы. Если таких элементов в строке нет, соответствующий элемент одномерного массива должен быть равен 0.</p>
5.	<p>I. В каждой строке матрицы каждый отрицательный элемент, расположенный между двумя положительными, заменить их полусуммой. Вывести преобразованную матрицу.</p> <p>II. Найти значение максимального элемента из заштрихованной части преобразованной матрицы и вывести это значение и координаты его в матрице. Границы заштрихованной части выберите сами.</p> 
6.	<p>I. Записать в одномерный массив максимальные отрицательные элементы каждой строки матрицы. Для строк матрицы, не содержащих отрицательных элементов, в результирующий массив записать 0. Вывести полученный одномерный массив.</p> <p>II. Составить и вывести одномерный массив из средних квадратичных значений элементов с четными номерами строк в каждом столбце матрицы.</p>

7.	<p>I. Найти среднее арифметическое значение максимальных по модулю элементов строк матрицы. Вывести:</p> <ul style="list-style-type: none"> • значения максимальных по модулю элементов строк (со знаком); • полученное значение. <p>II. Составить и вывести одномерный массив из количеств элементов со значениями в интервале $[C, D]$ в каждой строке матрицы.</p>
8.	<p>I. В каждом столбце целочисленной матрицы подсчитать сумму положительных элементов с четными значениями и записать ее в одномерный массив. Вывести полученный массив.</p> <p>II. Составить и вывести одномерный массив из минимальных положительных элементов среди первой половины каждого столбца матрицы. Если таких элементов в столбце нет, соответствующий элемент одномерного массива должен быть равен 0.</p>
9.	<p>I. Вычислить среднее арифметическое значение элементов каждого столбца матрицы без учета минимального и максимального элементов этого столбца; записать это значение в одномерный массив. Вывести:</p> <ul style="list-style-type: none"> • значения минимума и максимума столбцов; • полученный одномерный массив. <p>II. Составить и вывести одномерный массив из сумм элементов со значениями из интервала $[C, D]$ в каждом столбце матрицы.</p>
10.	<p>I. Сформировать матрицу B из элементов матрицы A. Каждый элемент b_{ik} должен быть равен сумме элементов матрицы A из области, определяемой индексами i, k в соответствии с рисунком. Вывести результирующую матрицу.</p> <div data-bbox="1125 1137 1385 1317" style="display: inline-block; vertical-align: middle;"> <p>The diagram shows a 6x6 grid of squares. A 3x3 sub-grid is shaded in the bottom-right corner. The top-right element of the shaded area is labeled a_{ik}.</p> </div> <p>II. Найти и вывести среднее геометрическое положительных элементов матрицы A с двумя четными индексами. Если таких элементов в матрице нет, вывести сообщение.</p>
11.	<p>I. Преобразовать <i>целочисленную</i> матрицу, удалив из нее строки, содержащие нулевые элементы, и сдвинув оставшиеся строки. Если нулевых элементов в матрице нет, вывести сообщение. Вывести:</p> <ul style="list-style-type: none"> • номера удаленных строк или сообщение; • преобразованную матрицу. <p>II. Составить и вывести одномерный массив из максимальных отрицательных элементов каждого столбца <i>преобразованной</i> матрицы. Если отрицательных элементов в столбце нет, соответствующий элемент одномерного массива должен быть равен 0.</p>

12.	<p>I. Преобразовать <i>целочисленную</i> матрицу, переставив на первое место столбец с наибольшим количеством нулей (сдвинуть остальные столбцы). При отсутствии в матрице нулевых элементов вывести сообщение, иначе вывести преобразованную матрицу.</p> <p>II. Найти значение минимального ненулевого элемента в заштрихованной части <i>преобразованной</i> матрицы и вывести это значение и его координаты в матрице. Границы заштрихованной части выберите сами. Если таких элементов в матрице нет, вывести сообщение</p>	
13.	<p>I. Сформировать и вывести квадратную матрицу B из элементов квадратной матрицы A. Каждый элемент b_{ik} должен быть равен сумме элементов матрицы A из области, определяемой индексами i, k в соответствии с рисунком.</p> <p>II. Составить и вывести одномерный массив из количеств элементов, меньших D, с четными номерами строк в каждом столбце матрицы A.</p>	
14.	<p>I. Если модуль суммы отрицательных элементов <i>целочисленной</i> матрицы больше суммы ее положительных элементов, сформировать <i>одномерный</i> массив из первых элементов строк матрицы, иначе массив формируется из последних элементов строк матрицы. Если в матрице нет положительных или отрицательных элементов, вывести сообщение, иначе вывести:</p> <ul style="list-style-type: none"> • суммы отрицательных и положительных элементов матрицы; • результирующий массив. <p>II. Найти и вывести координаты и значение максимального элемента матрицы с двумя четными индексами.</p>	
15.	<p>I. Найти максимальный и минимальный по модулю ненулевые элементы матрицы и поменять их местами. Вывести:</p> <ul style="list-style-type: none"> • координаты найденных элементов; • преобразованную матрицу. <p>II. Составить и вывести одномерный массив из произведений ненулевых элементов с нечетными номерами столбцов в каждой строке <i>преобразованной</i> матрицы. Если таких элементов в строке нет, соответствующий элемент одномерного массива должен быть равен 0.</p>	

16.	<p>I. Записать в одномерный массив столбец матрицы с максимальным количеством отрицательных элементов. При отсутствии отрицательных элементов в матрице вывести сообщение. Вывести:</p> <ul style="list-style-type: none"> • количество отрицательных элементов в столбцах или сообщение; • результирующий одномерный массив. <p>II. Составить и вывести одномерный массив из сумм квадратов отрицательных элементов каждой строки матрицы. Если отрицательных элементов в строке нет, соответствующий элемент одномерного массив должен быть равен 0.</p>
17.	<p>I. Если в каждой строке целочисленной матрицы A есть равный нулю элемент, то сформировать матрицу B из строк матрицы A с четными номерами, иначе - из ее строк с нечетными номерами. Вывести:</p> <ul style="list-style-type: none"> • номера строк, в которых нет нулей; • полученную матрицу. <p>II. Составить и вывести одномерный массив из количеств отрицательных элементов с четными номерами столбцов в каждой строке матрицы A.</p>
18.	<p>I. Заменить максимальный элемент матрицы A средним арифметическим его соседей. Соседями элемента a_{ij} в матрице считать элементы $a_{i+1,j}$, $a_{i-1,j}$, $a_{i,j+1}$, $a_{i,j-1}$. Следует учесть, что в зависимости от положения элемента a_{ij} в матрице у него может быть 2, 3 или 4 соседа. Вывести:</p> <ul style="list-style-type: none"> • координаты и значение максимального элемента матрицы; • преобразованную матрицу. <p>II. Найти и вывести количество ненулевых элементов <i>преобразованной</i> матрицы с двумя нечетными индексами.</p>
19.	<p>I. Назовем допустимым преобразованием матрицы перестановку двух строк или двух столбцов. С помощью допустимых преобразований добиться того, чтобы наибольший по модулю элемент матрицы располагался в ее левом верхнем углу. Вывести преобразованную матрицу после каждой перестановки (строки или столбца).</p> <p>II. Составить и вывести одномерный массив из средних арифметических значений элементов с нечетными индексами столбцов в каждой строке <i>преобразованной</i> матрицы.</p>

20.	<p>I. Сформировать матрицу B из элементов матрицы A. Каждый элемент b_{ik} должен быть равен максимальному элементу матрицы A из области, определяемой индексами i, k в соответствии с рисунком. Вывести результирующую матрицу.</p>  <p>II. Составить и вывести одномерный массив из минимальных модулей ненулевых элементов каждой строки матрицы A. Если в строке все элементы равны 0, соответствующий элемент одномерного массив должен быть равен 0.</p>
21.	<p>I. Найти координаты и значение максимума из минимальных элементов каждого столбца матрицы. Вывести:</p> <ul style="list-style-type: none"> • координаты минимальных элементов столбцов; • координаты и значение максимума. <p>II. Составить и вывести одномерный массив из произведений ненулевых элементов каждого столбца матрицы. Если в столбце все элементы равны 0, соответствующий элемент одномерного массива должен быть равен 0.</p>
22.	<p>I. Найти минимальный по модулю ненулевой элемент в каждой строке <i>квадратной</i> матрицы и поменять его местами с элементом этой же строки, находящимся на побочной диагонали матрицы. Вывести:</p> <ul style="list-style-type: none"> • координаты найденных в строках элементов; • преобразованную матрицу. <p>II. Найти и вывести координаты максимального элемента со значением в интервале $[C, D]$ в <i>преобразованной</i> матрице. Если таких элементов в матрице нет, вывести сообщение.</p>
23.	<p>I. Записать в одномерный массив модули разности первого и последнего отрицательных элементов каждого столбца матрицы. Для столбцов, содержащих менее двух отрицательных элементов, в результирующий массив записать (-1) и вывести сообщение. Вывести:</p> <ul style="list-style-type: none"> • для каждого столбца – значения первого и последнего отрицательных элементов или сообщение; • полученный одномерный массив. <p>II. Составить и вывести одномерный массив из максимальных модулей элементов каждой строки матрицы.</p>

24.	<p>I. Сформировать матрицу B из элементов матрицы A. Каждый элемент b_{ik} должен быть равен максимальному элементу матрицы A из области, определяемой индексами i, k в соответствии с рисунком. Вывести результирующую матрицу.</p> <p>II. Составить и вывести одномерный массив из минимальных положительных элементов с четными индексами столбцов в каждой строке матрицы A. Если таких элементов в строке нет, соответствующий элемент одномерного массива должен быть равен 0.</p>	
25.	<p>I. Поменять местами первый и второй положительные элементы в каждом столбце матрицы. Для столбцов матрицы, содержащих менее двух положительных элементов, вывести сообщение. Вывести:</p> <ul style="list-style-type: none"> • координаты элементов, найденных в каждом столбце; • преобразованную матрицу. <p>II. Составить и вывести одномерный массив из сумм отрицательных элементов каждой строки <i>преобразованной</i> матрицы.</p>	
26.	<p>I. Удалить из <i>целочисленной</i> матрицы строки и столбцы, заполненные нулями (оставшиеся строки и столбцы сдвинуть). Вывести результирующую матрицу после каждого удаления (строк или столбцов).</p> <p>II. Найти и вывести произведение ненулевых элементов матрицы со значениями в интервале $[C, D]$. Если таких элементов в матрице нет, вывести сообщение.</p>	
27.	<p>I. Рассматривая построчно <i>целочисленную</i> матрицу A, сформировать три одномерных массива, поместив в массив B - четные элементы, в массив C - элементы, кратные 3, в массив D - элементы, кратные 5. Возможна запись элемента в два или три массива. Вывести полученные одномерные массивы.</p> <p>II. Найти и вывести сумму отрицательных элементов матрицы с двумя четными индексами.</p>	
28.	<p>I. Сформировать квадратную матрицу B из элементов квадратной матрицы A. Каждый элемент b_{ik} должен быть равен максимальному элементу матрицы A из области, определяемой индексами i, k в соответствии с рисунком. Вывести результирующую матрицу.</p> <p>II. Найти и вывести сумму отрицательных элементов матрицы, больших D ($D < 0$). Если таких элементов в матрице нет, вывести сообщение.</p>	

29.	<p>I. В каждом столбце <i>целочисленной</i> матрицы найти количество рядом стоящих пар одинаковых элементов. Результаты записать в одномерный массив. Вывести полученный массив.</p> <p>II. Составить и вывести одномерный массив из сумм последних K элементов каждой строки матрицы. (K меньше числа элементов в строке матрицы).</p>
30.	<p>I. Умножить матрицу A на вектор B. Произведением матрицы $A(M, N)$ и вектора $B(N)$ является вектор $C(M)$, каждый элемент которого вычисляется по формуле $c_j = \sum_{k=1}^n a_{jk} \cdot b_k$. Вывести полученный массив.</p> <p>II. Найти и вывести координаты и значение минимального элемента матрицы. При поиске минимума исключить из рассмотрения первую и последнюю строки и первый и последний столбцы.</p>

Лабораторная работа №7: Решение задач со строками

Задание

Разработать виртуальный прибор для решения задачи, согласно заданному варианту индивидуального задания.

Содержание отчета

1. Название работы, номер варианта индивидуального задания.
2. Фамилия, имя, отчество и номер группы студента.
3. Текст задания индивидуального варианта.
4. Блок – диаграмма программы.

Варианты индивидуального задания

№	Задание
1.	Введите строку, состоящую из 5 символов. Определите, является ли она палиндромом (то есть при чтении наоборот содержание не изменяется, например, слово «БОБ»).
2.	Введите строку из 11 символов. Вычислить произведение входящих в нее целых чисел (без учета их знаков).
3.	Введите строку из 20 символов, состоящей из строчных букв и пробелов. Определить слово, которое начинается и заканчивается на одну и ту же букву.
4.	Введите две строки символов. Определить, являются ли они анаграммами, то есть одна строка получена из другой перестановкой букв. Например, строки "БУК" и "КУБ" или "СОЛЬ" и "ЛОСЬ" являются анаграммами.

5.	Введите строку из 17 символов, содержащую два или более слов, разделенных пробелами. Поменять местами все четные и нечетные слова в строке.
6.	Введите строку из 11 символов (английских символов, знаков препинания и пробелов). Зашифровать его посимвольно операцией «исключающее ИЛИ» с заданным числом из диапазона 0-255. Организовать дешифрование строки.
7.	Строка состоит из слов, в которые входят только русские буквы. Оканчивается строка точкой. Одно слово от другого отделяется одним или несколькими пробелами. Оставить в строке только те слова, в которых чередуются гласные и согласные буквы, и эти слова симметричны относительно своего центра.
8.	Ввести строку, содержащий текст. Определить длину самого короткого слова и самого длинного слова.
9.	Введите строку из 23 символов. Определить, является ли она палиндромом (симметричной с точностью до пробелов) или нет. Например, «А РОЗА УПАЛА НА ЛАПУ АЗОРА» – палиндром. (Предполагается, что все буквы строки – прописные.)
10.	Введите строку из 16 символов. Найти наибольшее целое число (без учета знака числа).
11.	В заданной строке слова зашифрованы — каждое из них записано наоборот. Расшифровать сообщение.
12.	Дана строка, содержащая слова на латинице, разделенные пробелами. Определить, какие буквы в словах совпадают чаще: первые, последние или средние. Позиция средней буквы в слове определяется по формуле: $poz_средн_буквы = длина_слова \hat{:} 2 + 1$, где $\hat{:}$ — операция целочисленного деления.
13.	Дана строка, содержащая текст. Определите, каких букв (строчных или прописных) в нем больше, и преобразуйте следующим образом: если больше прописных букв, чем строчных, то все буквы преобразуются в прописные; если больше строчных, то все буквы преобразуются в строчные; если поровну и тех и других — текст остается без изменения.
14.	Дана строка символов, содержащая слова, разделенные точкой с запятой «;». Набор заканчивается двоеточием «:». Определить, сколько в нем слов, заканчивающихся буквой «а».
15.	Строка не более чем из 100 символов, содержащая произвольный русский текст. Написать, какие буквы и сколько раз встречаются в этом тексте. Ответ должен приводиться в грамматически правильной форме, например «а – 5 раз», «к – 7 раз» и т.д.

16.	Даны две строки символов: <i>A</i> и <i>B</i> . Составьте программу, проверяющую, можно ли из букв, входящих в <i>A</i> , составить <i>B</i> (буквы можно использовать не более одного раза и можно переставлять). Например, <i>A</i> : ИНТЕГРАЛ; <i>B</i> : АГЕНТ – составить можно; <i>B</i> : ГРАФ – составить нельзя.
17.	Введите строку из 7 символов. Определите, является ли она симметричной (Симметричной считается строка, которая одинаково читается слева направо и справа налево).
18.	Дана строка символов, состоящая из предложений, разделенных точками. Напишите программу, производящую следующее форматирование: а) после каждой точки в конце предложения должен стоять хотя бы один пробел; б) первое слово в предложении должно начинаться с прописной буквы. Замечание, текст может быть как на русском, так и на английском языке.
19.	В строке хранятся фамилии и инициалы студентов группы. Требуется напечатать список группы с указанием для каждого студента количества его однофамильцев.
20.	Дана строка, содержащая текст. Отредактировать заданный текст, удаляя из него те слова, которые встречаются в предложении заданное число раз.
21.	Даны две строки, содержащие тексты. Напечатать те слова, которые встречаются в каждой из двух строках.
22.	Дана строка, содержащая текст на английском языке; соседние слова отделены друг от друга запятой, за последним словом – точка. Определить количество слов, которые заканчиваются буквой «и».
23.	Дана строка, содержащая текст; соседние слова отделены друг от друга запятой, за последним словом – точка. Определить количество слов, которые начинаются и оканчиваются одной и той же буквой.
24.	Дана строка, содержащая слова, разделенные пробелами. Найти в ней слова-рифмы для заданного слова (рифма – совпадение трех последних символов).
25.	Даны две строки. Определить, совпадают ли они. Если нет, сообщить номер позиции первого несовпадающего символа.
26.	Дана строка, содержащая текст. Составить программу проверки правильности написания сочетаний «жи»-«ши», «ча»-«ща», «чу»-«щу». Исправить ошибки.
27.	Дана строка, содержащая текст. Текст состоит из цифр, латинских и русских букв. Подсчитать сумму цифр, встречающихся в тексте.
28.	Дана строка, содержащая текст из латинских букв; соседние слова отделены друг от друга запятой, за последним словом – точка. Вывести текст, составленный из первых символов всех слов текста.

29.	Составить программу исправления ошибочного набора текста вида «после символов «.», «!», «?» должен стоять пробел».
30.	Дана строка, содержащая текст. Вывести все слова, предварительно заменив в них первую букву на заглавную.

Лабораторная работа №8: Ввод/вывод данных в файловую систему

Задание

Разработать виртуальный прибор для решения задачи, согласно заданному варианту индивидуального задания.

Виртуальный прибор должен формировать исходный файл, выполнять задание и сохранять результат в *новом* файле в соответствии с вариантом индивидуального задания.

Содержание отчета

1. Название работы, номер варианта индивидуального задания.
2. Фамилия, имя, отчество и номер группы студента.
3. Текст задания индивидуального варианта.
4. Блок – диаграмма программы.

Варианты индивидуального задания

№	Задание
1.	Создать файл, куда записать n целых чисел. Из исходного файла сформировать массивы четных и нечетных чисел. Определить наибольший четный компонент файла и наименьший нечетный.
2.	Создать файл, куда записать n целых чисел. На основе исходного файла создать массив удвоенных нечетных чисел. Упорядочить его по возрастанию элементов.
3.	Создать файл, куда записать n целых чисел. Сформировать массив положительных чисел, делящихся на пять без остатка, используя элементы исходного файла. Упорядочить массив по убыванию элементов..
4.	Создать файл, куда записать n вещественных чисел. Из компонентов исходного файла сформировать массивы положительных и отрицательных простых чисел. Вычислить количество нулевых компонентов файла.
5.	Создать файл, куда записать n целых чисел. Из файла создать массив, элементы которого не являются простыми числами и расположены до минимального элемента.

6.	Создать файл, куда записать n целых чисел. Из файла целых чисел сформировать массив, записав в него только ненулевые компоненты, находящиеся после максимального элемента.
7.	Создать файл, куда записать n вещественных чисел. Сделать массив из элементов исходного файла, внося в него числа, превосходящие среднее значение среди положительных значений файла.
8.	Создать файл, куда записать n целых чисел. Из исходного файла сформировать массив, записав в него числа, расположенные в файле до минимального элемента и после максимального.
9.	Создать файл, куда записать n целых чисел. Массив создать из исходного файла. Внести в него числа, расположенные в файле между минимальным и максимальным элементами.
10.	Создать файл, куда записать n целых чисел. Из исходного файла сформировать массив, в котором вначале расположить четные, а затем нечетные числа. Определить номера наибольшего нечетного и наименьшего четного компонентов.
11.	Создать файл, куда записать n целых чисел. В файле поменять местами максимальное среди отрицательных и наибольшее среди простых.
12.	Создать файл, куда записать n целых чисел. Из файла переписать все простые большие среднего арифметические в новый файл.
13.	Создать файл, куда записать n целых чисел. Найти среднее арифметическое среди простых чисел, расположенных после минимального значения.
14.	Создать файл, куда записать n целых чисел. Поменять местами первое совершенное и последнее отрицательное число в файле.
15.	Создать файл, куда записать n целых чисел. Все простые числа из исходного файла записать в массив, который упорядочить по убыванию.
16.	Создать файл, куда записать n целых чисел. Последнюю группу совершенных чисел из исходного файла переписать в текстовый файл.
17.	Создать файл, куда записать n целых чисел. Найти в нем группу положительных элементов наибольшей длины.
18.	Создать файл, куда записать n целых чисел. Из исходного файла сформировать массивы простых и совершенных чисел. Определить наибольшее простое число в файле и наименьшее совершенное.
19.	Создать файл, куда записать n целых чисел. Из файла создать массив, элементы которого являются простыми числами и расположены после максимального значения файла.
20.	Создать файл, куда записать n целых чисел. Из файла целых чисел сформировать массив, записав в него только кратные 3 значения, находящиеся до минимального элемента файла.

21.	Создать файл, куда записать n вещественных чисел. Сделать массив из элементов исходного файла, внося в него числа, превосходящие среднее значение среди четных значений файла.
22.	Создать файл, куда записать n вещественных чисел. Поменять местами последнее простое число в файле с наименьшим совершенным.
23.	Создать файл, куда записать n вещественных чисел. Найти сумму последней группы отрицательных элементов.
24.	Создать файл, куда записать n целых чисел. Найти произведение первой группы, состоящей из простых чисел.
25.	Создать файл, куда записать n целых чисел. Найти разность между суммой простых чисел, находящихся в файле и максимальным значением файла.
26.	Создать файл, куда записать n целых чисел. Найти разность между суммой четных чисел и произведением нечетных чисел, находящихся в файле.
27.	Создать файл из действительных чисел. Создать два новых файла - в первом файле записать все чётные числа исходного файла, а все нечётные во второй файл. Порядок следования чисел сохраняется.
28.	Создать файл из вещественных чисел. Из файла удалить все компоненты, находящиеся после первого чётного компонента.
29.	Создать файл, компонент которого состоит из двух переменных: первая переменная – число, вторая – символ. Поместить в качестве первого компонента файла последний компонент, значение первой составляющей которого больше заданного числа.
30.	Создать файл из строковых данных. Из файла удалить первый компонент, содержащий заданную строку.

Список литературы

1. Лупов С.Ю., Муякшин С.И., Шарков В.В. LabVIEW в примерах и задачах. Учебно-методические материалы по программе повышения квалификации «Обучение технологиям National Instruments». Нижний Новгород, 2007, 101 с.
2. Джеффри Тревис: LabVIEW для всех. Пер. с англ. Клушин Н. А. – М.: ДМК Пресс; ПриборКомплект, 2005. – 544 с.: ил.
3. Суранов А. Я. LabVIEW 8.20: Справочник по функциям. – М.: ДМК Пресс, 2007. – 536 с.
4. Блюм П.. LabVIEW: стиль программирования. Пер. с англ. Под ред. Михеева П. – М.: ДМК Пресс, 2007 – 400 с.
5. LabVIEW user manual. National Instruments corp., 2010
6. Звягин В.Ф., Яньшина Н.А., Голыничев В.Н. Практикум по современному Фортрану в курсе Информатики. Учебное пособие – СПб: СПбГУИТМО, 2010. – 134 с.
7. Леонов Ю. Г., Глазунова Л. В., и др.: Сборник задач по программированию. – Одесса: ОНАС им. А.С. Попова, 2011. – 212 с.
8. Станевко Г.И., Колесникова Т.Г., Давыденко В.А. Программирование и основы алгоритмизации: лабораторный практикум /. – Кемерово: КемТИПП, 2010. – 127 с.

Содержание

1 Запуск LabVIEW, окна, палитры	4
2 Принцип потока данных	9
3 Типы данных	10
4 Базовые функции LabVIEW	13
5 Структура Варианта	22
6 Циклы	23
7 Массивы	27
8 Ввод/вывод данных в/из файла	30
9 Создание подприбора	32
10 Стил ь программирования	34
Лабораторная работа №1: Вычисление по формулам	37
Лабораторная работа №2: Попадание точки в заданную область	42
Лабораторная работа №3: Условные операторы	48
Лабораторная работа №4: Решение задач с одномерными массивами	51
Лабораторная работа №5: Операторы цикла с пред и постусловием	55
Лабораторная работа №6: Решение задач с двумерными массивами	58
Лабораторная работа №7: Решение задач со строками	65
Лабораторная работа №8: Ввод/вывод данных в файловую систему	68
Список литературы	71

Миссия университета – генерация передовых знаний, внедрение инновационных разработок и подготовка элитных кадров, способных действовать в условиях быстро меняющегося мира и обеспечивать опережающее развитие науки, технологий и других областей для содействия решению актуальных задач.

КАФЕДРА ОПТИКО-ЭЛЕКТРОННЫХ ПРИБОРОВ И СИСТЕМ

Кафедра создавалась в 1937-38 годах и существовала под следующими названиями:

- с 1938 по 1958 год - кафедра военных оптических приборов;
- с 1958 по 1967 год - кафедра специальных оптических приборов;
- с 1967 по 1992 год - кафедра оптико-электронных приборов;
- с 1992 года - кафедра оптико-электронных приборов и систем.

Кафедру возглавляли:

- с 1938 по 1942 год - профессор К.Е. Солодилов;
- с 1942 по 1945 год - профессор А.Н. Захарьевский (по совместительству);
- с 1945 по 1946 год - профессор М.А. Резунов (по совместительству);
- с 1947 по 1972 год - профессор С.Т. Цуккерман;
- с 1972 по 1992 год - заслуженный деятель науки и техники РСФСР, профессор Л.Ф. Порфирьев;
- с 1992 по 2007 год - заслуженный деятель науки РФ, профессор Э.Д. Панков.
- с 2007 года по настоящее время - почетный работник высшего профессионального образования, профессор В.В. Коротаев.

История кафедры началась в 1937-38 годах с организации в Ленинградском институте точной механики и оптики (ЛИТМО) кафедры военных оптических приборов. Первым заведующим кафедрой был К.Е. Солодилов, до этого возглавлявший Центральное конструкторское бюро (ЦКБ) Всесоюзного объединения оптико-механической промышленности (ВООМП). Преподавателями кафедры стали сотрудники этого ЦКБ - М.А. Резунов, М.Я. Кругер, С.Т. Цуккерман, В.А. Егоров, Б.М. Кулежнов.

В годы Великой Отечественной войны кафедра была эвакуирована в Черепаново, где обязанности заведующего кафедрой выполнял профессор А.И. Захарьевский. Преподавателями кафедры по состоянию на 01.04.1945 г были профессор Чулановский, доцент Кругер, ст. преподаватель Гриневич, ассистенты Дедюлин и Погарев. После возвращения в Ленинград кафедрой в 1945-46 годах по совместительству заведовал

начальник конструкторского бюро (КБ) Государственного оптического института им. С.И. Вавилова (ГОИ) М.А. Резунов.

В начале 1947 года кафедру возглавил профессор С.Т. Цуккерман, который руководил ею до 1972 года. В 1958 году кафедра была реорганизована в кафедру специальных оптических приборов, а в 1967 году в кафедру оптико-электронных приборов (ОЭП).

Создание С.Т. Цуккерманом в предвоенные годы книги «Точные механизмы» (М.: Оборонгиз, 1941) является значительным вкладом в развитие отечественного точного приборостроения. С.Т. Цуккерман является автором более 120 научных работ и более 50 изобретений. В предвоенные, военные и послевоенные годы С.Т. Цуккерман работал над созданием прицельных устройств для зенитной и авиационной артиллерии. Он был одним из создателей серийного авиационного гироскопического прицела АСП с автоматической выработкой поправки на упреждение, который устанавливался на истребителях МиГ, а также механического ракурсного прицела для мелкокалиберной зенитной артиллерии, широко применяемого во время войны во Вьетнаме.

В 1958 г. при кафедре была организована отраслевая лаборатория «Специальные оптические приборы» с достаточно сильной группой конструкторов-разработчиков.

С.Т. Цуккерман и старший научный сотрудник А.С. Гридин руководили разработкой приборов управления по лучу (ПУЛ), предназначенных для управления движением различных подвижных объектов по прямой линии или по программе.

В начале 60-х годов старший научный сотрудник Г.Г. Ишанин занимался разработкой фотометрической аппаратуры, предназначенной для паспортизации оптико-электронных приборов и систем различного назначения.

Значительное влияние на содержание подготовки специалистов и научных исследований оказало привлечение к работе на кафедре выдающегося специалиста в области оптико-электронного приборостроения, члена-корреспондента Российской академии наук (РАН), Героя Социалистического Труда, лауреата Ленинской премии профессора М.М. Мирошникова, который, работая на кафедре ОЭП с 1969 года по 1976 год в должности профессора по совместительству, поставил и читал курс «Теория оптико-электронных приборов».

С 1972 года по 1992 год кафедрой ОЭП заведовал заслуженный деятель науки и техники РСФСР, профессор Л.Ф. Порфирьев, известный специалист в области автоматических ОЭПиС в комплексах навигации и управления авиационной и космической техникой. Соответственно тематика выполнения научно-исследовательских работ на кафедре приобрела новые направления, существенно увеличилось число тем,

носящих поисковый фундаментальный характер. Были разработаны новый учебный план и программы учебных дисциплин.

Л.Ф. Порфирьев является автором 19 учебников, учебных пособий и монографий, среди которых можно выделить такие как «Теория оптико-электронных приборов и систем» (Л.: Машиностроение, 1980), «Основы теории преобразования сигналов в оптико-электронных системах» (Л.: Машиностроение, 1989). Результаты его работ можно оценить как значительный вклад в разработку общей теории оптико-электронных систем.

Л.Ф. Порфирьев как руководитель проводил достаточно жесткую кадровую политику, при которой на кафедре оставались работать только те сотрудники, которые отличались преданностью делу. При этом он оказывал всемерную поддержку сотрудникам кафедры по разработке ими различных направлений теории и практики оптико-электронного приборостроения. По результатам научно-исследовательских работ в этот период защитили диссертации на соискание ученой степени доктора технических наук Г.Н. Грязин (1983 г.), Е.Г. Лебедько (1985 г.), Э.Д. Панков (1986 г.), Г.Г. Ишанин (1988 г.), защищено много диссертаций на соискание ученой степени кандидата технических наук.

В этот период под руководством Э.Д. Панкова начали проводиться исследования по разработке новых оптико-электронных систем измерения взаимного положения разнесенных в пространстве объектов.

Г.Н. Грязин, перешедший на кафедру с радиотехнического факультета в конце 60-х годов, продолжил свои работы в области прикладного телевидения, в частности, по разработке систем наблюдения за быстродвижущимися объектами и быстропротекающими процессами.

С 1975 года заведующим отраслевой лабораторией стал старший научный сотрудник А.Н. Тимофеев, который продолжил исследования по разработке методов и средств контроля пространственного положения объектов с помощью ОЭП с оптической равносигнальной зоной для машиностроения, энергетики, строительства, судостроения и железнодорожного транспорта.

С 1975 года, после увольнения в запас, из Ленинградской военной инженерной краснознаменной академии (ЛВИКА) им. А.Ф. Можайского на кафедру пришел работать в должности профессора С.П. Авдеев, известный специалист в области ОЭПиС космических аппаратов. Он поставил курсы и читал лекции по учебным дисциплинам «Оптико-электронные приборы», «Оптико-электронные приборы систем управления», «Оптико-электронные приборы для научных исследований».

Существенное влияние на содержание подготовки специалистов и научных исследований оказало привлечение к работе на кафедре лауреата Ленинской и Государственной премий профессора Б.А. Ермакова, известного специалиста в области физической оптики и

оптико-электронного приборостроения. Б.А. Ермаков работал на кафедре ОЭП с 1979 года по 1992 год в должности профессора по совместительству и поставил курс «Оптико-электронные приборы с лазерами».

В 70-80 годах под руководством доцента Е.Г. Лебедево проводились исследования законов отражения лазерного излучения от нестационарных поверхностей и протяженных объектов, исследования в области теории идентификации объектов по их излучению в сложной фоновой ситуации. Создан комплекс для лазерной локации крупногабаритных морских объектов сложной конфигурации и водной поверхности. В этих работах принимали участие доценты О.П. Тимофеев и С.Б. Лукин.

В 70-90 годах под руководством Л.Ф. Порфирьева был разработан ряд астродатчиков, систем астроориентации и космической навигации (В.И. Калинин, А.Л. Андреев, С.Н. Ярышев).

С 1992 г. заведующим кафедрой является заслуженный деятель науки Российской Федерации, профессор Э.Д. Панков. В 1992 году кафедра была переименована в кафедру оптико-электронных приборов и систем (ОЭПИС).

Под руководством Э.Д. Панкова в 70-90-х годах были проведены разработки ряда оптико-электронных приборов и систем специального и гражданского применения, нашедших практическое внедрение и способствующих научно-техническому прогрессу и укреплению обороноспособности нашей страны.

В частности, исследования и разработки в области линейных и угловых измерений позволили приступить к решению общей проблемы согласования отсчетных баз на нестационарно деформируемых объектах с помощью оптико-электронных систем.

В рамках указанной проблемы доцентом И.А. Коняхиним проводились исследования, результаты которых можно классифицировать как разработку теории построения автоколлимационных систем с компонентами нарушенной типовой конфигурации.

В то же время доцентом В.В. Коротаевым разработан ряд поляризационных приборов и измерительных установок. Теоретическим результатом работ явилась разработка методологии анализа поляризационных свойств оптических систем с изменяющейся ориентацией элементов. По результатам указанных работ В.В. Коротаев (в 1997 г.) и И.А. Коняхин (в 1998г.) защитили диссертации на соискание ученой степени доктора технических наук.

Применение многоэлементных приемников в системах пеленгации дало толчок развитию телевизионных систем технического зрения, измерительных телевизионных систем и систем обработки изображений. Результаты этих исследований были использованы доцентом А.Л. Андреевым при постановке учебных курсов «Оптико-электронные системы с ЭВМ», «Специализированные аппаратные и программные

средства ОЭП», «Автоматизированные телевизионные вычислительные комплексы», а также доцентом С.Н. Ярышевым при постановке им в 1993 году учебной дисциплины «Видеотехника».

Указанные курсы обеспечиваются лабораторным практикумом на базе рабочих мест, оснащенных персональными компьютерами, объединенными в локальную сеть. Рабочие места оснащены аппаратными и программными средствами цифровой видеозаписи и обработки изображений. В этот период Г.Н. Грязиным были подготовлены дисциплинам: «Телевизионные системы», «Прикладное телевидение и телевизионно-вычислительные комплексы» (совместно с А.Л. Андреевым).

На основе обобщения методик расчета оптико-электронных систем различного назначения и принципа действия в 1981 году были развернуты работы по созданию элементов систем автоматизированного проектирования ОЭП. За период с 1981 по 1987 год под руководством И.А. Коняхина были разработаны оригинальные пакеты прикладных программ расчета параметров систем измерения пространственного положения объектов.

Развитие компьютерной техники и программного обеспечения общего назначения позволило создать проблемно-ориентированное программное обеспечение поддержки проектирования ОЭП на системотехническом уровне.

По результатам научных работ сотрудниками кафедры ОЭПиС выпущено в свет 15 монографий, 11 учебников и учебных пособий. На кафедре подготовлено 14 докторов наук, а также более 110 кандидатов наук.

На разработки кафедры получены авторские свидетельства СССР и патенты Российской Федерации на более чем 200 изобретений. Наибольший вклад в изобретательскую деятельность внес Э.Д. Панков - автор 123 изобретений, из которых 33 внедрены в промышленности.

При заявлении научно-педагогической школы «Оптико-электронное приборостроение» в 2009 году были сформулированы следующие основные научно-технические результаты, достигнутые в период с 1938 по 2009 годы:

- разработаны принципы построения военных оптико-механических приборов;
- разработаны принципы построения точных механизмов;
- разработаны принципы построения оптико-электронных приборов с оптической равносигнальной зоной;
- систематизированы теоретические основы и принципы построения оптико-электронных приборов;
- разработаны методы описания импульсных сигналов, идентификации и классификации объектов в системах нестационарной лазерной локации;

– разработаны теория, принципы построения и методы расчета импульсных телевизионных систем наблюдения быстро движущихся объектов;

– обнаружен термоупругий эффект в кристаллическом кварце и создан новый тип приемников оптического излучения;

– разработана теория построения автоколлимационных систем с компонентами нарушенной типовой конфигурации;

– разработана методология анализа поляризационных свойств оптических систем с изменяющейся ориентацией элементов;

– систематизированы теоретические основы и принципы построения измерительных систем на основе матричных фотопреобразователей;

– разработаны основы построения ОЭС согласования отсчетных баз на нестационарно деформируемых объектах.

Основоположники научной школы:

- Солодилов Константин Евгеньевич, заведующий кафедрой с 1938 г. по 1942 г., профессор;
- Цуккерман Семен Тобиасович, заведующий кафедрой с 1947 г. по 1972 г., профессор;
- Мирошников Михаил Михайлович, директор ГОИ, д.т.н., профессор, профессор кафедры ОЭП с 1967 г. по 1978 г.; член-корреспондент Российской Академии наук, Герой Социалистического Труда, лауреат Ленинской премии.
- Порфирьев Леонид Федорович, заведующий кафедрой с 1972 г. по 1992 г., д.т.н., профессор, Заслуженный деятель науки и техники РСФСР.
- С 2007 г. заведующим кафедрой является почетный работник высшего профессионального образования Российской Федерации, профессор В.В. Кортаев.

На кафедре была открыта подготовка по новой специализации инженеров «Оптико-электронные приборы и системы обработки видеоинформации» и новая магистерская программа «Оптико-электронные методы и средства обработки видеоинформации».

В 2007 году был создан научно-образовательный центр оптико-электронного приборостроения (НОЦ ОЭП).

Научно-образовательный центр оптико-электронного приборостроения выполняет научно-исследовательские и опытно-конструкторские работы по созданию видеоинформационных и информационно-измерительных приборов различного назначения, высокоточных приборов для измерения линейных, угловых и других физических величин в промышленности, энергетике, на транспорте, а также систем технического зрения и обработки видеоинформации. К выполнению научно-исследовательских и опытно-конструкторских работ широко привлекаются студенты, аспиранты, молодые специалисты,

молодые кандидаты наук. Научно-образовательный центр является активным участником Федеральной целевой программы «Научные и научно-педагогические кадры инновационной России» на 2009-2013 годы.

Направления научных исследований кафедры ОЭПиС в 2007-2012 годах.

Развитие теоретических основ и принципов построения оптико-электронных приборов и систем, в том числе:

- видеоинформационных измерительных систем;
- видеоинформационных систем наблюдения;
- видеоинформационных импульсных систем наблюдения быстро движущихся объектов;
- комплексированных телевизионно-тепловизионных систем наблюдения,
- ОЭПиС обеспечения техносферной безопасности;
- ОЭПиС согласования отсчетных баз на нестационарно деформируемых объектах;
- автоколлимационных систем с компонентами нарушенной типовой конфигурации;
- ОЭПиС цветового и спектрального анализа объектов;
- фотометрических систем аттестации ОЭПиС, источников и приемников оптического излучения;
- систем лазерной локации с нестационарным облучением;
- ОЭС сепарации полезных ископаемых.

По результатам исследований в этот период на кафедре были защищены 14 диссертаций на соискание ученой степени кандидата технических наук.

Идет активное пополнение преподавательского состава молодыми кандидатами наук. В настоящее время на кафедре работает 7 кандидатов наук в возрасте до 35 лет.

Мы занимаемся разработкой оптико-электронных приборов и систем в целом:

- системотехническое проектирование,
- разработка (выбор) оптической системы,
- разработка конструкции,
- разработка (выбор) электроники и средств обработки информации,
- разработка программного обеспечения,
- сборка, юстировка, настройка и испытания.

Мы учим тому, что сами умеем делать!

По итогам конкурсов ведущих научно-педагогических коллективов СПб НИУ ИТМО 2007-2011 годов кафедра занимала призовые места.

С 2011 года подготовка бакалавров, магистров и специалистов на кафедре ОЭПиС осуществляется по Федеральным государственным образовательным стандартам третьего поколения (ФГОС).

Подготовка бакалавров по направлению:

200400 «Оптотехника» (профиль - Оптико-электронные приборы и системы). Срок обучения - 4 года

Подготовка магистров по направлению:

200400 Оптотехника.

Магистерские программы:

– Оптико-электронные методы и средства обработки видеоинформации

– Оптико-электронные приборы и системы безопасности

Срок обучения – 2 года.

Подготовка инженеров по специальности:

200401 -Электронные и оптико-электронные приборы и системы специального назначения.

Специализация:

– Оптико-электронные информационно-измерительные приборы и системы. Срок обучения – 5,5 лет.

Подробная информация о кафедре ОЭПиС имеется на сайте кафедры:
<http://oeps.ifmo.ru/>

Васильев Александр Сергеевич
Лашманов Олег Юрьевич

Основы программирования в среде LabVIEW

Учебное пособие

В авторской редакции

Редакционно-издательский отдел Университета ИТМО

Зав. РИО

Н.Ф. Гусарова

Подписано к печати

Заказ №

Тираж

Отпечатано на ризографе

Редакционно-издательский отдел
Университета ИТМО
197101, Санкт-Петербург, Кронверкский пр., 49