LabVIEW 8.20: справочник по функциям





Суранов А. Я.



Суранов А. Я.

LabVIEW 8.20 Справочник по функциям

2-е издание, электронное



Москва, 2023

УДК 621.38 ББК 32.973.26-108.2 С90

Суранов, Александр Яковлевич.

С90 LabVIEW 8.20. Справочник по функциям / А. Я. Суранов. — 2-е изд., эл. — 1 файл pdf : 537 с. — Москва : ДМК Пресс, 2023. — Систем. требования: Adobe Reader XI либо Adobe Digital Editions 4.5 ; экран 10". — Текст : электронный.

ISBN 978-5-89818-456-8

В книге приведено описание функциональных элементов среды проектирования виртуальных приборов LabVIEW 8.20. Описание выполнения функций сопровождается примерами их использования. Для большинства Экспресс-ВП приведены окна конфигурирования с переводом их содержимого. В справочнике большое внимание уделено функциям программирования, математики, обработки сигналов, коммуникации, управления приборами и обмена данными по стандартным интерфейсам. Рассмотрены также новые элементы LabVIEW 8.20 — проект, разделяемая переменная, элементы объектно-ориентированного программирования и язык MathScript. Справочник может быть полезен широкому кругу специалистов, решающих задачи измерения, обработки или моделирования сигналов.

УДК 621.38 ББК 32.973.26-108.2

Электронное издание на основе печатного издания: LabVIEW 8.20. Справочник по функциям / А. Я. Суранов. — Москва : ДМК Пресс, 2008. — 536 с. — ISBN 5-94074-347-1. — Текст : непосредственный.

В соответствии со ст. 1299 и 1301 ГК РФ при устранении ограничений, установленных техническими средствами защиты авторских прав, правообладатель вправе требовать от нарушителя возмещения убытков или выплаты компенсации.

© Суранов А. Я. © Оформление, 34 1 ДМК Пресс, 2008



Введение	6
Благодарности	7

▼ 1

Организация среды LabVIEW и технология

программирования	8
1.1. Панели, палитры и окна LabVIEW	8
1.2. Технология проектирования виртуальных приборов	
1.3. Структуры, массивы и графические индикаторы среды	
LabVIEW	

▼2

Функции программирования LabVIEW	57
2.1. Базовые функции LabVIEW	57
2.1.1. Числовые функции и функции манипуляции данными	57
2.1.2. Логические функции	74
2.1.3. Строковые функции	77
2.1.4. Функции сравнения	103
2.1.5. Функции работы с массивами	109
2.1.6. Функции работы с кластерами и данными с типом Вариант	121
2.1.7. Функции установления времени	127
2.1.8. Функции и ВП ввода/вывода файлов	133
2.2. Дополнительные функции LabVIEW	169
2.2.1. Функции диалога и интерфейса пользователя	169
Функции диалога	170
Функции обработки ошибок	173
Экспресс-ВП палитры	177

	170
Функции подпалитры сооытие	178
Функции меню	183
ВП из подпалитры Курсор	188
Функции из подпалитры Помощь	191
2.2.2. Функции управления приложением	192
2.2.3. Функции и ВП синхронизации	200
Операции уведомителя	200
Операции очереди	204
ВП Семафор	209
ВП Встреча	211
Функции случаев	213
2.2.4. Функции преобразования и отображения графических файлов	215
2.2.5. Функции записи и воспроизведения звуковых сигналов	222

▼3

Математические функции LabVIEW	. 231
3.1. Функции линейной алгебры	. 231
3.2. Функции аппроксимации данных	. 245
3.3. Функции статистической обработки данных	. 257
3.4. Функции численных методов	. 268
3.4.1. Функции интерполяции и экстраполяции	. 269
3.4.2. Функции интегрирования и дифференцирования	275
3.4.3. Функции решения дифференциальных уравнений	278
3.4.4. Функции оптимизации	285
3.5. Окно и узел MathScript	292

▼4

Функции генерации и обработки сигналов LabVIEW	302
4.1. Функции генерации сигналов и шумов	302
4.2. Функции операций с сигналами	312
4.3. Функции преобразований сигналов	325
4.4. Функции спектрального анализа	333
4.5. Функции фильтров	341
4.6. Функции обработки весовыми окнами	363



Функции генерации и измерения параметров осциллограмм.	371
5.1. Базовые функции аналоговых и цифровых осциллограмм	371

5.2.	Функции генерации осциллограмм	394
5.3.	Функции измерения параметров осциллограмм	404

▼6

Функции обмена данными	423
6.1. Разделяемые переменные	423
6.2. Технология передачи данных и функции DataSocket	430
6.3. Функции протоколов передачи данных	436
6.3.1. Функции протоколов TCP/IP	
6.3.2. Функции протоколов UDP	
6.3.3. Функции протокола Bluetooth	
6.3.4. Функции электронной почты	450

▼ 7

Функции поддержки взаимодействия приложений	454
7.1. Технология и функции ActiveX	454
7.2. Технология и функции .NET	460
7.3. Разработка библиотек динамической компоновки	464
7.4. ВП доступа к реестру Windows	470
7.5. ВП управления устройствами ввода и портами ввода/вывода	476

▼ 8

Функции поддержки ввода/вывода данных и стандартных		
интерфейсов	479	
8.1. Функции сбора данных DAQmx	481	
8.2. Функции интерфейса канала общего пользования (GPIB)	499	
8.3. Функции последовательной коммуникации	508	

▼ Приложение 1

Синтаксис узла Форму	\a	51!	5
----------------------	----	-----	---

▼ Приложение 2

Перечень «горячих» клавиш (Keyboard Shortcuts))519
--	------

▼ Приложение 3

Алфавитный указатель	функций	
----------------------	---------	--



Появление в течение последних полутора лет новых версий LabVIEW 8.0 и 8.20 стало еще одним свидетельством быстрого развития этой среды графического программирования. Компания National Instruments выдвигает в качестве ключевых особенностей данных версий концепцию распределенного интеллекта, или распределенной логики, которая включает следующие аспекты:

- формирование на базе LabVIEW единой графической платформы для программирования таких устройств, как настольные компьютеры, системы реального времени, ПЛИС, КПК, встроенные микропроцессоры и сигнальные процессоры, представляющие все узлы распределенной системы – как ведущий, так и целевые. Программирование перечисленных устройств обеспечивается с помощью соответствующих обновленных модулей;
- включение в состав LabVIEW нового **Проекта** (Project) как единого инструмента для обзора системы и обеспечения доступа ко всем ее узлам. Проект позволяет с помощью одного окна просматривать, редактировать, запускать и отлаживать код, работающий на любом целевом блоке;
- использование **Разделяемой переменной** (Shared Variable) нового упрощенного программного интерфейса для совместного использования данных. С помощью разделяемой переменной можно передавать данные между системами, в том числе и системами реального времени, без потери скорости. Для осуществления передачи необходимо лишь сконфигурировать эту переменную с помощью несложных диалоговых окон;
- способность синхронизации внутри и между распределенными устройствами и системами. Измерение времени и синхронизация остаются ключевыми проблемами при построении эффективных измерительных и управляющих систем.

Помимо этого, в новых версиях LabVIEW произошли определенные изменения как в структуре, так и в содержании палитр элементов лицевой панели и в особенности палитр функций блок-диаграммы. Появились элементы, поддерживающие объектно-ориентированное программирование, а также текстовый язык математических расчетов MathScript.

Перечисленные и ряд других нововведений определили необходимость соответствующей модернизации справочника по функциям среды LabVIEW [1]. Модернизация коснулась прежде всего структуры и содержания глав с описанием функций, которые приведены в соответствие с новой организацией палитр блок-диаграммы, а именно в соответствие с категориями функций. Так, в частности, глава 2 содержит теперь функции из категории **программирования** (Programming), глава 3 – **математические функции** (Mathematics), а глава 4 – функции из категории **обработка сигналов** (Signal Processing). При этом в связи с большим объемом функций программирования глава 2 разделена на две подглавы, содержащие базовые и дополнительные функции. Помимо этого, многочисленные функции работы с осциллограммами, доступные как в палитре программирования, так и в палитре обработки сигналов, рассмотрены в отдельной главе 5.

Последующие три главы (с 6 по 8) посвящены функциям, входящим в состав категорий обмена данными (Data Communication), поддержки взаимодействия приложений (Connectivity), контроля ввода/вывода (Measurement I/O) и связи с приборами (Instrument I/O) и отражают содержимое соответствующих палитр.

В справочник добавлен указатель функций, в котором функции сгруппированы по категориям, а в каждой категории оригинальные названия функций упорядочены по алфавиту. Отсутствие номера страницы после названия функции означает, что эта функция в справочнике не рассмотрена.

В приложение вынесена информация о синтаксисе узла Формула и о «горячих» клавишах.

Разделы, посвященные определенной группе функций, начинаются с кратких пояснений, после чего приводится описание функций, оформленное в виде набора таблиц. Порядок следования функций обычно соответствует порядку их просмотра в палитре – слева направо и сверху вниз. В конце раздела могут быть приведены примеры применения описанных функций. В качестве примеров в большинстве случаев использовались модернизированные в той или иной степени ВП из набора примеров NI Example Finder LabVIEW.

В каждой таблице, посвященной определенной функции, приводятся, как правило, два изображения функции с подключенными элементами управления и индикаторами. При этом первое (левое) изображение имеет ярлыки (labels) элементов управления и индикаторов на английском языке, а второе – на русском. В нижней части таблицы даются пояснения по назначению и параметрам входов и выходов функции. При этом обязательные входы функции на изображении выделяются полужирным шрифтом, а текст пояснений к рекомендуемым и необязательным входам имеет уменьшенный шрифт. В большинстве случаев для экономии места входы и выходы ошибок на изображении функций ввиду их однотипности не подключались.

Благодарности

Автор выражает благодарность руководителю инновационных программ NI в Российской Федерации **П. Р. Сепояну** за всемерную поддержку работы по подготовке этого справочника.



1.1. Панели, палитры и окна LabVIEW

Запуск LabVIEW 8.20 приводит к выводу окна **первоначального запуска** (Getting Started) (рис. 1.1), которое предлагает две группы вариантов дальнейших действий пользователя: **Файлы** (Files) и **Ресурсы** (Resources). Первая группа содержит два раздела: **Новый** (New) и **Открыть** (Open). С помощью строк меню раздела **Новый** можно открыть **чистый виртуальный прибор (BII)** (Blank VI), **пустой проект** (Empty project), **ВП из шаблона** (VI from Template) или перейти к более подробному варианту диалогового окна **Новый** (строка меню More...). В свою очередь, строки меню раздела **Открыть** позволяют открыть существующие файлы.

Строки меню раздела **Ресурсы** позволяют вызвать справочную информацию по различным аспектам работы в LabVIEW, ознакомиться с новыми элементами LabVIEW 8.20, обратиться к Web-ресурсам и к примерам разработанных ВП.

При выборе в разделе **Новый** строки **чистый ВП** (Blank VI) открываются два окна, содержащие лицевую панель (рис. 1.2) и панель блок-диаграммы (рис. 1.3) виртуального прибора (ВП).

В верхней части каждого окна размещена традиционная для приложений Windows полоса главного меню с одинаковыми для обоих окон пунктами File, Edit, View, Project, Operate, Tools, Windows и Help. Ниже полосы меню расположена полоса инструментальной панели, служащая для запуска и редактирования ВП. Полоса инструментальной панели окна блок-диаграммы отличается дополнительными кнопками для отладки ВП. В правом верхнем углу каждой панели находится иконка, наложенная на соединительную панель ВП (последняя показана на лицевой панели).

Свободное пространство каждой панелей образует рабочую область, снабженную горизонтальной и вертикальной полосами прокрутки. При построении ВП в рабочей области лицевой панели визуально размещаются элементы управления

🖾 Getting Started	X
LabVIEW 8.2	
Files	Resources
New	New To LabVIEW?
🕍 Blank VI	Getting Started with LabVIEW
Empty Project	LabVIEW Fundamentals
VI from Template	Guide to LabVIEW Documentation
🗁 More	LabVIEW Help
2	Upgrading Lab¥IEW?
Open	MathScript
Drowse	3D Picture Control
	LabVIEW Object-Oriented Programming
	List of All New Features
	Web Resources
	Discussion Forums .
	Training Courses
	LabVIEW Zone
	Examples
	Find Examples

Рис. 1.1. Вид окна первоначального запуска LabVIEW 8.2

💌 VI 1.vi Front Panel *	-ox
Eile Edit View Project Operate Tools Window Help	
✓ (現) (⇒ (型) ● ■ 13pt Application Font マ いいい * ● ● ■ 13pt Application Font マ いいいい * ● ● ● ■ 13pt Application Font マ ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	; ; ; ;
	1=0
My Project.lvproj/My Computer <	►:

Рис. 1.2. Вид лицевой панели ВП

💌 VI 1.vi Block Diagram *	-OX
<u>File E</u> dit <u>V</u> iew <u>P</u> roject <u>O</u> perate <u>T</u> ools <u>W</u> indow <u>H</u> elp	
✓ - (U) 今 🕑 💷 😵 💷 🖬 🗗 🗗 14pt Ari	al Cyr 🔽 🚛 🙃 🐡 🔊 🚟
My Project.lvproj/My Computer	

Рис. 1.3. Вид панели блок-диаграммы ВП

и индикации, формирующие интерфейс пользователя, а на панели блок-диаграммы составляется блок-диаграмма – графический исходный код ВП. Для одновременного отображения данных панелей в левой и правой половинах экрана целесообразно использовать меню **Окно ⇒ Панели слева и справа** (Windows **⇒** Tile Left and Right) или нажать «горячую» клавишу «Т». Клавиша становится «горячей» при нажатии одновременно с ней одной или более служебных клавиш. В данном случае должна быть нажата клавиша «Ctrl», далее такое сочетание обозначается <Ctrl+T>. Перечень «горячих» клавиш приведен в приложении 2.

Построение ВП осуществляется с помощью трех вспомогательных палитр: палитры **Элементы управления** (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 разделов меню **Вид** (View). Так, в частности, при активном окне лицевой панели с помощью строки **Палитра элементов управления** (Controls Palette) меню **Вид** на эту панель можно вывести палитру элементов, а при активном окне панели блок-диаграммы на нее можно вывести палитру функций, пользуясь строкой **Палитра функций** (Functions Palette) этого же меню. Для вывода палитры инструментов необходимо использовать строку **Палитра инструментов** (Tools Palette) меню **Вид**.

Однако может оказаться, что пользователю будет более удобен временный вывод первых двух палитр, который реализуется как вызов контекстного меню каждой панели с помощью щелчка на ее рабочем пространстве правой кнопкой мыши (ПКМ). Выбор конкретного объекта из палитры элементов или палитры функций производится путем перемещения курсора мыши по разделам палитр. Выбранный объект берется из палитры с помощью щелчка левой кнопкой мыши (ЛКМ) и переносится в заданную область соответствующей панели, после чего фиксируется в этой области повторным щелчком ЛКМ (технология **Перенес и бросил** (Drag and Drop)). Эту же операцию можно выполнить с помощью щелчка ЛКМ на выбранном объекте, последующего удержания клавиши во время переноса объекта и отпускания клавиши в момент его фиксации. Такие объекты палитры функций, как **Структуры** (Structures), или строковые константы, перед фиксацией могут быть увеличены до необходимых размеров путем рисования модифицированным курсором мыши прямоугольного контура объекта при постоянно нажатой ЛКМ.

Временную версию палитры инструментов можно вывести с помощью щелчка ПКМ при нажатой клавише <Shift>.

Рассмотрим более подробно назначение пунктов главного меню, кнопок инструментальных панелей, палитр инструментов, элементов и функций.

Выше были перечислены пункты главного меню, среди которых можно выделить пункты, встречающиеся в большинстве приложений Windows, такие как File, Edit, View, Windows, Help, и пункты, являющиеся характерными для LabVIEW, такие как Project, Operate и Tools. Ниже в таблице приведено краткое описание функций пунктов главного меню.

Файл	Используется для открытия новых или существующих ВП и проектов,
(File)	закрытия, сохранения и вывода на печать ВП и проектов, а также для доступа к свойствам ВП
Правка	Применяется для редактирования панелей ВП, поиска объектов
(Edit)	и удаления неисправных проводников с блок-диаграммы, создания
	подприборов и установления значений элементов по умолчанию
Вид	Используется для вывода доступных палитр, списка ошибок, иерархии
(View)	ВП и взаимосвязей между ВП и подприборами, работы с проводником классов
Проект	Позволяет работать с проектом: создавать новый, открывать или
(Project)	сохранять существующий, добавлять элементы, получать информацию
	о файлах или устанавливать свойства проекта
Управление	Реализует запуск и прерывание выполнения ВП, соединение с удален-
(Operate)	ной панелью и отладку приложения или общей библиотеки
Инструменты	Используется для запуска программы анализа измерений и автомати-
(Tools)	зации (МАХ) и поиска драйверов приборов, поиска ВП на диске,
	открытия окна MathScript, управления библиотеками ВП, управления
	соединением с удаленными ВП и публикацией панелей ВП в Web,
	конфигурирования ВП и для выполнения ряда прикладных функций
Окно	Используется для отображения окон LabVIEW и списка открытых ВП
(Window)	и проектов
Справка	Служит для получения информации об элементах и о функциях
(Help)	LabVIEW

На рис. 1.4 показан вид полосы инструментальной панели на лицевой панели, а на рис. 1.5 – вид аналогичной панели на панели блок-диаграммы.



Рис. 1.4. Вид инструментальной панели на лицевой панели



Рис. 1.5. Вид инструментальной панели на панели блок-диаграммы

Далее в таблице кратко описаны функции кнопок инструментальных панелей.

⇔	Кнопка Запуск (Run) работоспособного ВП
ŝ	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 Синхронизировать с другими экземплярами приложения (Synchronize with Other Application Instances) вносит изменения ВП во все экземпляры приложения. После нажатия этой кнопки отмена изменений невозможна. Эта кнопка появляется только при редактировании ВП, который открыт в нескольких экземплярах приложения

Следующие пять кнопок инструментальной панели блок-диаграммы используются при отладке программы, в том числе и при пошаговой отладке.



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)

Кнопки **Начало пошагового выполнения** (Start Single Stepping) или **Шаг через** (Step Over) вызывают пошаговое выполнение ВП

Кнопка **Выход из пошагового выполнения** (Step Out) завершает пошаговое выполнение ВП

Кнопки, рассмотренные ниже, позволяют форматировать текстовые объекты панелей, изменять размеры и расположение объектов панелей.

между ними

Кнопка Установки текста (Text Settings) позволяет выбирать и устанавливать шрифт, размер, стиль и цвет текста LabVIEW Пть объекты (Align Objects) позволяет выровнять объекты



•Oo`

по центру Кнопка Распределить объекты (Distribute Objects) позволяет распределить объекты панелей равномерно относительно их центов или краев, установить равномерные промежутки (Gaps) между объектами или удалить промежутки



Кнопка **Изменить размеры объекта** (Resize Objects) позволяет изменить размеры объектов на лицевой панели



Кнопка **Изменить порядок** (Reorder) позволяет изменить порядок расположения объектов на панели при их перекрытии или зафиксировать положение объектов на панели Кнопка Ввести текст (Enter Text) служит для завершения ввода текста

Кнопка Показать окно контекстной справки (Show Context Help Window) позволяет открыть окно контекстной справки

Все операции по созданию, редактированию и отладке ВП выполняются с помощью палитры **Инструменты** (Tools Palette) (рис. 1.6).

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) тием клавиш <Shift+Tab>. Выбор любого другого инструмента приводит к отключению автоматического выбора инстру-

Рис. 1.6

мента. 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.

Ниже в таблице приведены краткие пояснения по инструментам, входящим в палитру.

প চন্দ্	Инструмент Управление (Operate Value, «палец») используется для изменения
	значений элементов управления или ввода текста. При работе со стр <u>оков</u> ыми
	элементами управления вид инструмента изменяется на следующий: 🔲
	Инструмент Перемещение (Position/Size/Select, «стрелка») служит для выбора,
	перемещения или изменения размеров объектов. Для изменения размеров
	в LabVIEW 8.20 используются подвижные прямоугольные элементы, появляющи-
	еся в зависимости от допустимого направления изменения в центре сторон
	или на углах контура объекта при установке инструмента Перемещение внутри
	этого контура
Aĭ	Инструмент Редактирование текста (Edit Text, «буква») используется для ввода
Γ <u>i</u>	и редактирования текста и создания свободных меток. При создании текстовых
	элементов вид инструмента изменяется:
	Инструмент Соединение (Connect Wire, «катушка») применяется для соедине-
V	ния объектов на блок-диаграмме. Он также используется для условного
	(невидимого) подключения элементов управления и индикаторов лицевой
	панели к терминалам соединительной панели ВП
h_	Инструмент Контекстное меню объекта (Object Shortcut Menu) вызывает



Инструмент Быстрая прокрутка окна (Scroll Window) используется для просмотра окна без обращения к полосам прокрутки

Инструмент Контрольная точка (Set/Clear Breakpoint) позволяет размещать и удалять контрольные точки на ВП, функциях, узлах, проводниках данных, структурах и приостанавливать в них выполнение программы





۰Đ

Инструмент **Получить цвет** (Get Color, «пипетка») служит для копирования цвета с последующей вставкой с помощью инструмента **Установить цвет** Инструмент **Установить цвет** (Set Color) предназначен для изменения цвета объекта. Он также отображает текущие цвета переднего и заднего плана



Рис. 1.7. Вид палитры элементов лицевой панели в стандартном режиме

Палитра элементов лицевой панели может отображать входящие в ее состав подпалитры элементов по категориям без надписей под иконками (стандартный вид) или с надписями (Icons & Text), в виде папок, организованных в структуре Дерево, или в виде набора вложенных текстовых папок. На рис. 1.7 показан вид палитры элементов лицевой панели, отображаемой в стандартном режиме по категориям. При этом автоматически разворачивается содержимое категории, находящейся на верхнем уровне. В данном случае это категория элементов, имеющих стиль **Современные** (Modern).

Текущее изменение вида отображения палитры производится с помощью нажатия кнопки **Вид** (View) прикрепленной палитры (рис. 1.7) и последовательного выбора строки **Видеть эту палитру как** (View This Palette As) и желаемого вида палитры. Для долговременного изменения вида палитры необходимо с помощью меню **Инструменты** \Rightarrow **Опции** (Tools \Rightarrow Options) вызвать диалоговое окно **Опции**, выбрать в списке меню категорию **Па**литры элементов управления/функций (Controls/Functions Palettes) и в поле со списком **Формат** (Format) выбрать вид палитры.

Набор отображаемых категорий может быть изменен с помощью той же кнопки **Вид** и по-

следовательного выбора строки **Всегда видимые категории** (Always Visible Categories) и необходимых категорий в открывающемся списке. Положение категории в палитре может быть изменено с помощью ее контекстного меню, имеющего пункты **Переместить эту категорию вверх** (Move this Category Up), **Переместить эту категорию вниз** (Move this Category Down), **Переместить в верхнюю часть (Рас**- **ширяемая по умолчанию)** (Move to Top (Expandable by Default)), или простым захватом в области левого края строки и ее переносом в нужное место с помощью мыши.

Изменение содержания подпалитр производится путем их перевода в режим редактирования с помощью меню **Инструменты** ⇒ **Дополнительные** ⇒ **Редактирование набора палитры**... (Tools ⇒ Advanced ⇒ Edit Palette Set...) и вызова контекстного меню подпалитры с набором операций редактирования.

Раскрытая на рис. 1.7 палитра элементов содержит следующие подпалитры.



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



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



Строка и путь (String & Path). Элементы подпалитры представляют типы данных, которые содержат последовательность литер, символов, массивов



Массив, матрица и кластер (Array, Matrix & Cluster). Подпалитра содержит структуры, которые позволяют создавать массивы или кластеры элементов. Массивы и кластеры представляют упорядоченное множество элементов соответственно одного или различных типов. Элементами массива могут быть числовые или логические элементы, строки или кластеры. Тип элементов массива определяется типом данных, помещаемых из палитры элементов в шаблон массива.

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



Список и таблица (List & Table). Элементы подпалитры представляют собой управляющие или управляемые элементы, позволяющие заносить или отображать буквенную, символьную и цифровую информацию в виде набора строк или ячеек



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



Кольцевой список и перечень (Ring & Enum). Элементы подпалитры представляют собой специальные числовые объекты, которые ставят в соответствие 16-битовым целым числам без знака строки, рисунки или то и другое



Контейнеры (Containers). Элементы подпалитры представляют объекты, внутри которых могут размещаться элементы управления и индикации, лицевые панели подприборов и элементы ActiveX



Ввод/вывод (I/O). Подпалитра содержит элементы управления и индикаторы, используемые для передачи установленных пользователем имен каналов DAQ, имен средств VISA и логических имен IVI к ВП ввода/вывода, обеспечивающих связь с приборами или устройствами сбора данных



Ссылка (Refnum). Подпалитра содержит идентификаторы, которые связаны с открытым приложением или файлом

Вариант и класс (Variant & Class)

Оформление (Decorations). Подпалитра служит для размещения на лицевой панели разнообразных графических элементов: линий, стрелок, рамок различной формы, надписей и т. п.



Рис. 1.8. Вид палитры функций панели блок-диаграммы

Палитра функций панели блок-диаграммы имеет такую же организацию и методику настройки, что и палитра элементов (рис. 1.8). В ее состав входят следующие категории функций: **Программирование** (Programming), **Математика** (Mathematics), **Обработка сигнала** (Signal Processing), **Обмен данными** (Data Communication), **Средства взаимодействия** (Connectivity), **Связь с приборами** (Instrument I/O) и **Экспресс** (Express). Порядок перечисления категорий соответствует порядку следования глав справочника, в которых эти функции рассматриваются.

Из перечисленных категория функций **Программирование** (раскрыта на рис. 1.8) является наиболее содержательной и играет ключевую роль при разработке широкого круга ВП. Она включает следующие подпалитры (перечисляются в порядке расположения слева направо и сверху вниз).



из кластеров, а также кластерную константу. Более подробно функции работы с кластерами описаны в разделе 2.1.6



Подпалитра **Числовые** (Numeric) содержит полный набор математических функций, функций преобразования форматов чисел и набор констант. Функции подпалитры рассмотрены в разделе 2.1.1



Подпалитра Логические (Boolean) включает набор функций И (AND), ИЛИ (OR), Исключающее ИЛИ (Exclusive Or), НЕ (Not) и логические константы. Более подробно логические функции рассмотрены в разделе 2.1.2



Подпалитра **Строковые** (String) содержит ряд функций обработки строковых переменных, функции взаимного преобразования чисел и строк, а также строковые константы. Перечень строковых функций и пояснения к ним приведены в разделе 2.1.3



Функции подпалитры **Сравнение** (Comparison) формируют логическую переменную в зависимости от результата сравнения входных переменных или позволяют определять соотношение чисел, нахождение числа в заданном диапазоне, тип числа или тип символа. Данным функциям посвящен раздел 2.1.4



Функции подпалитры **Установления времени** (Timing) позволяют задавать или определять временные интервалы или текущее время. Указанные функции рассмотрены в разделе 2.1.7



Функции подпалитры **Диалога и интерфейса пользователя** (Dialog & User Interface) используются для создания диалоговых окон, содержащих сообщения для пользователя. Эти функции рассмотрены в разделе 2.2.1



Функции подпалитры **Файловый ввод/вывод** (File I/O) выполняют файловые операции записи и считывания данных. Перечень функций файлового ввода/ вывода и пояснения к ним приведены в разделе 2.1.8



Подпалитра Осциллограмма (Waveform) включает разделы Аналоговая осциллограмма (Analog Waveform), Цифровая осциллограмма (Digital Waveform), Файловый ввод/вывод осциллограмм (Waveform File I/O), Измерения осциллограмм (Waveform Measurements) и Генерация осциллограмм (Waveform Generation). Функции подпалитры Осциллограмма рассмотрены в главе 5

Функции подпалитры **Управление приложением** (Application Control) позволяют программно управлять приложением (LabVIEW) или ВП на локальном или





удаленном компьютере. Функции подпалитры рассмотрены в разделе 2.2.2 Функции подпалитры **Синхронизация** (Synchronization) используются для синхронизации параллельно выполняющихся задач и для передачи данных между такими задачами. Функции подпалитры рассмотрены в разделе 2.2.3 Функции подпалитры **Графики и звук** (Graphics & Sound) позволяют строить трехмерные графики, производить запись/чтение графических файлов, формировать различные объекты на рисунке и осуществлять запись/чтение звуковых файлов. Функции преобразования графических файлов и формирования изображений рассмотрены в разделе 2.2.4. Функции работы со звуковой картой и звуковыми файлами рассмотрены в разделе 2.2.5



Функции подпалитры Создание отчета (Report Generation) позволяют формировать отчеты в виде HTML-страниц

В состав подпалитры Структуры (Structures) входят следующие элементы.



Цикл с фиксированным числом итераций (For Loop) осуществляет заданное число итераций выполнения кода внутри данной структуры

Цикл по условию (While Loop) осуществляет итерационное выполнение кода внутри данной структуры до выполнения заданного условия



Структура Вариант (Case Structure) управляет выполнением одного из двух или более фрагментов кода и при выборе по условию аналогична оператору if-then-else текстовых языков, а при выборе по значению числовой или строковой переменной аналогична оператору case Структура Событие (Event Structure) ожидает наступления заданных событий на лицевой панели и производит их обработку



MathScript Node





🖣 Disabled 🔻 🕨







Numeric

Узел MathScript исполняет функции и записи, написанные на языке LabVIEW MathScript, синтаксис которого аналогичен языку MATLAB

Структура Стековая последовательность (Stacked Sequence Structure) позволяет управлять последовательностью выполнения отдельных фрагментов кода путем их размещения в кадрах данной структуры

Структура Открытая последовательность (Flat Sequence Structure) отличается от предыдущей возможностью передачи данных между кадрами без вспомогательных переменных и возможностью вывода данных из любого кадра структуры

Узел **Формула** (Formula Node) позволяет включить фрагмент кода в текстовом представлении. Текстовый язык похож на C, но не идентичен ему

Структура отключения диаграммы (Diagram Disable Structure) имеет одну или несколько поддиаграмм или вариантов, из которых выполняется только одна поддиаграмма под заголовком **Разрешенная** (Enabled). Структура используется для отключения фрагмента блок-диаграммы

Структура отключения по условию (Conditional Disable Structure) имеет одну или несколько поддиаграмм или вариантов, из которых в зависимости от конфигурации LabVIEW использует только одну для продолжения исполнения

Узел **Обратная связь** (Feedback Node) используется для передачи значений между итерациями структур циклов

Разделяемая переменная (Shared Variable) позволяет передавать текущие данные между различными ВП проекта или по сети, при этом источниками или приемниками данных могут быть элементы лицевой панели или блок-диаграммы

Глобальная переменная (Global) используется для передачи данных между ВП на одном компьютере

Локальная переменная (Local) используется для передачи данных между элементами управления или индикаторами без применения проводов

Более подробно особенности построения и функционирования структур рассмотрены в разделе 1.3.

Помимо разработки лицевой панели и блок-диаграммы самого ВП, важное значение имеют конфигурирование его входов-выходов и формирование графического представления ВП для последующего использования в других ВП в качестве подприбора (подпрограммы). Перечисленные функции выполняются с помощью иконки и соединительной панели, размещенных в правом верхнем углу панелей. Изображение, помещенное на иконке, придает разработанному ВП индивидуальность и в большинстве случаев несет информацию о его функциональном назначении. Соединительная панель определяет картину расположения входных и выходных терминалов, посредством которых производятся ввод и вывод данных при использовании ВП в качестве подпрограммы. Для того чтобы элементы лицевой панели могли обмениваться данными с ВП верхнего уровня, они должны быть подключены к терминалам соединительной панели.

В качестве примера на рис. 1.9 показаны иконка и соединительная панель ВП **Записать в файл табличного формата** (Write To Spreadsheet File), находящегося в подпалитре **Файловый ввод/вывод**.

Вызов диалогового окна **Редактор иконки** (Icon Editor) для редактирования изображения иконки осуществляется с помощью строки **Редактировать иконку** (Edit



Рис. 1.9

Icon) контекстного меню иконки на лицевой панели. Вызов функций редактирования соединительной панели производится с помощью строки **Показать соединительную панель** (Show Connector) того же меню.

Ниже в таблице приведено краткое описание инструментов для создания иконки в диалоговом окне **Редактор иконки**.

Инструмент	Название	Функция
Ø	Карандаш	Рисует и стирает элементы изображения
	Линия	Рисует прямые линии
I	Пипетка	Копирует цвет переднего плана с элемента в иконке
٨	Наполненное ведро	Заполняет выделенную область цветом переднего плана
	Прямоугольник	Рисует прямоугольник в цвете заднего плана
	Заполненный прямоугольник	Рисует прямоугольник, окаймленный цветом переднего плана и заполненный цветом заднего плана
	Выбор	Выбирает область иконки для перемещения, копирования или других изменений

Инструмент	Название	Функция
A	Текст	Вводит текст в иконку. Изменение атрибутов шрифта производится с помощью двойного щелчка на этом инструменте
	Передний план/ задний план	Показывает текущий цвет переднего и заднего планов. Выбор цвета производится с помощью щелчка мышью на соответствующем прямоугольнике

Новым элементом рабочей среды LabVIEW 8.0 и 8.20 является оболочка управления проектами – LabVIEW Project, призванная обеспечить разработку распределенных приложений. Проект также поддерживает коллективную разра-

ботку больших приложений за счет включения интегрированных средств управления исходными текстами (Visual SourceSafe, Perforce, Rational ClearCase, PVCS, MKS и CVS) и библиотек **проектов** (Project Libraries), содержащих исходные коды в виде модульных, унифицированных функций, которые можно многократно вызывать из различных подсистем. Проект позволяет создавать загружаемые модули программ в виде **автономного приложения** (Stand-Alone Application) или динамически подключаемой библиотеки (DLL), а также zip-файлы или файлы распространения исходной программы (Source Distribution). Таким образом, простая и дружественная пользователю оболочка проекта LabVIEW позволяет наблюдать, редактировать, загружать, выполнять и отлаживать программный код, работающий на любом узле системы.

Для создания и редактирования проектов LabVIEW служит окно **Проводника проекта** (Project Explorer) (рис. 1.10). Окно отображается при выборе меню **Файл** \Rightarrow **Новый проект** (File \Rightarrow New Project) или **Проект** \Rightarrow **Новый проект** (Project \Rightarrow New Project).



Рис. 1.10. Вид окна проводника проекта

По умолчанию окно Проводника проекта включает следующие узлы:

- Корень проекта (Project root) содержит все прочие узлы окна Проводника проекта. Надпись в этом узле включает имя файла проекта;
- **Мой компьютер** (My Computer) представляет локальный компьютер в качестве платформы проекта;
- **Зависимости** (Dependencies) включает элементы, которые необходимы для ВП, находящихся под узлом платформы;
- Спецификации создания (Build Specifications) включает конфигурации формирования файлов распространения исходных программ и других видов приложений на основе инструментальных средств и модулей LabVIEW. При наличии установленных LabVIEW Professional Development System или Application Builder можно использовать узел Спецификации создания для конфигурации автономных приложений (EXE), библиотек динамической компоновки (DLL), инсталляторов и zip-файлов.

Перед сохранением проекта все его новые несохраненные файлы должны быть сохранены.

К существующему проекту могут быть добавлены новые или существующие папки или файлы, ВП, библиотеки ВП, библиотеки и подбиблиотеки проекта, элементы XControl, разделяемые переменные и объекты классов. Добавление новых элементов может производиться как с помощью строк **Новый** (New), **Добавить файл** (Add File), **Добавить папку** (Add Folder) контекстного меню узла, в который производится добавление, так и с помощью выделения этого узла и вызова меню **Проект Э Добавить к проекту** (Project **Э** Add to Project).

Библиотеки ВП могут быть добавлены к проекту LabVIEW как в виде папки, так и в виде файла. При добавлении библиотеки в виде папки LabVIEW использует имя библиотеки для наименования папки и добавляет ВП библиотеки как элементы в новую папку. Поиск библиотеки производится в диалоговом окне, открывающемся при выборе контекстного меню **Добавить папку** (Add Folder) заданного узла или расположенной под ним папки. Добавление или удаление элементов папки в окне не влияет на файл .llb на диске. С помощью контекстного меню папки **Преобразовать в библиотеку** (Convert To Library) папка может быть преобразована в **библиотеку проекта** (Project Libraries).

При добавлении библиотеки ВП в виде файла входящие в ее состав ВП появляются в окне Проводника проекта. Поиск библиотеки инициализируется с помощью строки **Добавить файл** (Add File) контекстного меню заданного узла. При выборе папки открывается диалоговое окно, в котором можно выделить отдельные файлы ВП или всю библиотеку, установив строку View All, и нажать кнопку **Добавить файл** (Add File).

Библиотеки проекта LabVIEW представляют наборы ВП, определений типа, разделяемых переменных, объектов классов, файлов палитр меню и других файлов, включающих другие библиотеки проектов. При создании и сохранении новой библиотеки проекта LabVIEW создает файл библиотеки проекта (.lvlib), который включает свойства библиотеки проекта и ссылки к файлам, которые входят в ее состав. Файл библиотеки проекта в отличие от библиотеки ВП не содержит самих этих файлов. Структуру библиотеки проекта можно просматривать в окне Проводника проекта или в автономном окне библиотеки проекта.

Библиотеки проекта могут быть полезны для организации файлов в единую иерархию элементов, потенциального дублирования имен ВП, отслеживания версий, ограничения открытого доступа к определенным файлам, ограничения возможности редактирования набора файлов и установки палитры меню для группы ВП.

Библиотеки проекта могут применяться для уточнения имен ВП и других файлов. LabVIEW распознает ВП по имени файл, поэтому LabVIEW может непреднамеренно загрузить и установить ссылку на ВП, имеющий такое же имя, что и другой ВП (проблема, известная как перекрестное связывание). Разные библиотеки проекта могут использовать ВП с одинаковыми именами.

Пользователь может устанавливать номера версий в библиотеке проекта для различения изменений набора файлов с течением времени. Установка номеров версий осуществляется в разделе **номер версии** (Version Number) на странице **общие установки** (General Settings) диалогового окна **свойства библиотеки проекта** (Project Library Properties) (рис. 1.11), вызываемого с помощью строки **Свойства** (Properties) контекстного меню библиотеки.

На этой же странице в разделе **Защита** (Protection) могут быть установлены различные степени ограничения на редактирование библиотеки проекта. В частности, может быть установлено **блокирование** (**без пароля**) (Locked (no pass-

🖻 Library 1. lvlib - Project Library Properties 🛛 🔀				
Category General Settings Documentation Item Settings		General Settings Version Number VI Icon Template Major Minor Fix Build 256 Colors 1 0 0 Image: State		
	~	Default Palette None Library Path С:\Cправочник LV 8\Глава 6\Library 1.lvlib		

Рис. 1.11. Вид диалогового окна Свойства библиотеки проекта

word)) или **защита паролем** (Password-protected). При блокировании библиотеки проекта пользователи не могут добавлять или удалять элементы и не видят элементы, установленные как закрытые.

Библиотека проекта также позволяет ограничить общий доступ к определенным типам файлов. С помощью раздела **область доступа** (Access Scope), расположенного на странице **установки параметров элементов** (Item Settings) диалогового окна **свойства библиотеки проекта**, можно сконфигурировать доступ к элементам и папкам библиотеки проекта как **открытый** (Public) или как **закрытый** (Private). При установке ВП как закрытого для доступа другие ВП, которые не входят в данную библиотеку, не могут вызывать его.

В LabVIEW 8.20 как в проект, так и в другие элементы среды введены средства поддержки объектно-ориентированного программирования (ООП). С ООП связаны такие понятия, как структура класса, инкапсуляция и наследование. Эти понятия используются для создания кода, который можно обновлять и модифицировать без изменения других частей приложения.

В LabVIEW ООП может использоваться для создания типов данных, определяемых пользователем. Такие типы данных создаются с помощью формирования классов LabVIEW. Классы LabVIEW определяют данные, связанные с объектом, а также методы, задающие действия, которые можно выполнять над данными.

Классы LabVIEW могут быть созданы в окне Проводника проекта (рис. 1.12) с помощью контекстных меню таких пунктов, как проект, My Computer, папка и библиотека проекта. В этом окне класс LabVIEW отображается синим кубиком, расположенным слева от названия. После сохранения класса LabVIEW создает файл библиотеки класса с расширением .lvclass, который определяет новый тип данных. В этот файл записываются элемент управления закрытыми данными и информация о любых созданных ВП – членах класса, например такая как перечень и различные свойства ВП. Библиотека класса подобна библиотеке проекта с расширением .lvlib, но отличается тем, что создает новый тип данных. Вид и цвет проводника, по которому передается информация класса, настраивается на странице внешний вид проводника (Wire Appearance) диалогового окна свойства класса.

> Рис. 1.12. Различные варианты создания классов в LabVIEW



При создании класса LabVIEW автоматически создает элемент управления данными с таким же названием и расширением .ctl. В окне Проводника проекта этот элемент отображается в виде синего кубика с зеленым цилиндром, символизирующим сохранение данных. В LabVIEW данные классов являются **закрытыми** (private). Это означает, что доступ к ним могут получить только ВП, являющиеся **членами** класса. Закрытый характер данных обозначается на значке объекта значком ключа красного цвета. Элемент управления данными является единственным файлом библиотеки класса, который определяет кластер закрытых данных для нового типа данных и вид проводника, по которому эти данные передаются. Сохранение элемента управления закрытыми данными в файле библиотеки класса, а не на диске позволяет LabVIEW гарантировать предоставление корректных данных вместе с определением класса.

Cluster of class private data



Рис. 1.13. Вид кластера закрытых данных

Настройка элемента управления закрытыми данными осуществляется в окне **Редактор элемента управления** (Control Editor), которое открывается после двойного щелчка на элементе управления в окне Проводника проекта. Определение закрытого типа данных класса LabVIEW производится путем размещения элементов управления и индикаторов в **Кластере класса закрытых данных** (Cluster of class private data) (рис. 1.13). Значения по умолчанию, которые устанавливаются для элементов кластера, являются значениями по умолчанию для этого класса.

Для доступа к закрытым данным создаются методы, которые реализуются с помощью ВП – членов класса. Эти ВП появляются в окне Проводника про-

екта под элементом управления закрытыми данными класса. Объединение закрытых данных и соответствующих методов и характеризует такой аспект ООП, как инкапсуляция (encapsulation).

ВП – члены класса могут быть созданы из **чистого ВП** (Blank VI), из **шаблона ВП** (VI template), который включает обработку ошибок и объекты класса, а также из **ВП – члена родительского класса** (Ancestor Member VI). Эти варианты создания реализуются с помощью соответствующих строк контекстного меню класса: **новый** ⇒ **ВП** (New ⇒ VI), **новый** ⇒ **динамический ВП** (NewDynamic VI) и **новый** ⇒ **корректировка ВП** (New ⇒ Override VI). При создании ВП – члена класса из ВП – члена родительского класса у них должны совпадать следующие настройки: реентерабельность, предпочтительное исполнение, приоритет и объем доступа, а также терминалы и конфигурация соединительной панели. LabVIEW не подсвечивает опцию **новый** ⇒ **корректировка ВП**, если нет действительного ВП – члена базового класса.

В то время как данные класса всегда закрыты, для ВП – членов класса можно устанавливать различную степень доступа. В разделе область доступа (Access Scope) страницы параметры установки элемента (Item Settings) диалогового

окна **свойства класса** (Class Properties) для методов могут быть установлены следующие варианты доступа:

- Открытая область (Public scope) любой ВП может вызывать ВП члена класса как подприбор (subVI);
- Защищенная область (Protected scope) только ВП, находящиеся внутри того же класса или класса-потомка могут вызывать ВП члена класса. Защищенный ВП член класса отображается в окне Проводника проекта с темно-желтым значком ключа;
- Закрытая область (Private scope) только ВП, находящиеся внутри того же класса, могут вызывать ВП члена класса. Закрытый ВП член класса отображается в окне Проводника проекта с красным значком ключа.

Рассмотренный выше вариант создания ВП – члена класса из ВП – членов родительского класса связан с таким понятием ООП, как **наследование** (inheritace). Наследование позволяет использовать существующий класс в качестве исходной точки для нового класса, в частности использовать открытые и защищенные ВП – члены родительского класса. Варианты наследования устанавливаются на странице **наследование** диалогового окна **свойства** класса.

При упаковке или распаковке класса LabVIEW узлы показывают терминалы только для закрытых данных текущего класса, а не для любых данных, которые класс наследует от родительских классов. Родительские данные являются закрытыми и их можно модифицировать с помощью функций, которые обеспечиваются ВП – членами класса. Эти ВП подобно любому ВП в LabVIEW могут вызывать любой из открытых ВП – членов класса. Но они могут также вызывать и защищенные ВП – члены родительского класса. При создании ВП – члена родительского класса как защищенного член – ВП любого дочернего класса может вызвать метод, но никакие ВП вне иерархии наследования не могут сделать это.

Объектом LabVIEW называется специальный класс, который является первоначальным родителем дерева наследования в ООП LabVIEW. По умолчанию все классы LabVIEW наследуют от объекта LabVIEW. Объект LabVIEW можно использовать для создания ВП, которые могут выполнять общие операции на множестве классов LabVIEW. Для представления объекта LabVIEW на лицевой панели служит одноименный элемент управления, размещенный в подпалитре **Вариант и класс** (Variant & Class).

Большинство методов можно определить с помощью единственного ВП – члена класса в одном классе, но некоторые методы можно определить путем создания нескольких ВП – членов класса с одинаковым именем, находящихся в иерархии класса. В первом случае методы называются **статическими**, во втором – **динамическими**. Статический или динамический тип определяется в соединительной панели ВП – члена класса. Если эта панель включает входной терминал динамической отсылки, то ВП является частью динамического метода. В противном случае ВП – член определяет статический метод. Данные, поступающие на входной терминал динамической отсылки, определяют экземпляр ВП – члена класса, который будет использоваться LabVIEW для реализации метода.

1.2. Технология проектирования виртуальных приборов

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.

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 проходит поток данных (data flow), идущий 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.

Таким образом, описанная технология формирования ВП является основой для **потоковой модели** обработки данных, когда поток данных входит (втекает) в узлы-источники, проходит через узлы обработки данных и выходит (вытекает) через узлы – приемники данных. При этом порядок обработки данных определяется целиком полнотой подхода данных к терминалам узлов. Такая концепция работы программы в LabVIEW существенно облегчает, по сравнению с текстовыми языками, разработку **многозадачных** и **многопоточных программ**.

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.

Еще один способ идентификации терминала связан с отображением иконки соответствующего элемента. Такое отображение включается с помощью опции

Отображать в виде иконки (View as Icon) контекстного меню терминала. В качестве примера можно сравнить вид терминала числового элемента ввода данных в форме с плавающей запятой одинарной точности в его традиционном представ-

лении **DBL** и в представлении в виде иконки **BBL** Однако представление в виде иконки приводит к потере пространства блок-диаграммы, что не всегда удобно.

Перечень и вид терминалов элементов ввода/вывода данных LabVIEW 8.20 с характеристикой типа, цвета и значения по умолчанию приведены в таблице.

Элемент управ- ления	Элемент индика- ции	Тип данных	Цвет	Значение по умол- чанию
SGL	I SGL	Числовой с плавающей запятой одинарной точности	Оранжевый	0,0
DBL	DBL	Числовой с плавающей запятой двойной точности	Оранжевый	0,0
EXT	NEXT	Числовой с плавающей запятой расширенной точности	Оранжевый	0,0
CSG	ECSG.	Комплексный с плавающей запятой одинарной точности	Оранжевый	0,0+i0,0
CDE	CDB	Комплексный с плавающей запятой двойной точности	Оранжевый	0,0+i0,0
CXT	ICXT.	Комплексный с плавающей запятой расширенной точности	Оранжевый	0,0+i0,0
18	18	Числовой байтовый со знаком	Синий	0
116	FI16	Числовой 16-битовый со знаком	Синий	0
132	132	Числовой 32-битовый со знаком	Синий	0
1641	F [64	Числовой 64-битовый со знаком	Синий	0
U8	▶U8	Числовой байтовый без знака	Синий	0
U16	D U16	Числовой 16-битовый без знака	Синий	0
U321	JJ32	Числовой 32-битовый без знака	Синий	0
0641	F U64	Числовой 64-битовый без знака	Синий	0
I) X	<64,64>-битовая отметка времени		12:00:00. 000 AM 1/1/1904
	FO	Нумерованный	Синий	_

28 Организация среды LabVIEW и технология программирования

Элемент управ- ления	Элемент индика- ции	Тип данных	Цвет	Значение по умол- чанию
TF	P TF	Логический	Зеленый	FALSE
abc	Pabe	Строковый	Розовый	Пустая строка
		Массив – заключает тип данных в квад- ратные скобки и принимает цвет этого типа данных	Различный	_
DBL	[DBL]	Матрица из вещественных чисел –	Оранжевый	
		используется для выполнения матрич- ных операций и отличается от массива типом проводника		
[CDB]	[CDB]	Матрица из комплексных чисел	Оранжевый	
		Кластер – заключает разные типы данных. Кластерный тип данных имеет коричневый цвет, если все элементы кластера числовые, и розовый, если элементы разных типов. Кластер ошибкиокрашен в темно-желтый цвет, а кластер объекта класса LabVIEW – в темно-красный	Коричневый, розовый, темно-желты или темно- красный	— Й
P	▶ ⁹ ~~g	Путь – сохраняет положение файла или каталога, используя стандартный синтаксис используемой платформы	Морской волны	<Не путь>
		Динамический (Dynamic) – (Express VIs). Включает данные, связанные с сигналог и атрибутами, которые обеспечивают информацию о сигнале, такую как имя сигнала или дата и время получения данных	Фиолетовый м	
		Осциллограмма – кластер элементов, которые содержат дату, начальное время и интервал выборок сигнала	Коричневый	_
LWI) MU	Цифровая осциллограмма – переносит начальное время, Δ х, цифровые дан- ные, любые атрибуты цифровой осциллограммы	Зеленый	
010	▶ 0101	Цифровые данные – содержат циф ровые данные, отображаемые в бито- вом виде по строкам и столбцам	Зеленый	

Элемент управ- ления	Элемент индика- ции	Тип данных	Цвет	Значение по умол- чанию
	ÞB	Ссылка – действует как уникальный идентификатор для таких объектов, как файл, устройство или сетевое соединение	Морской волны	_
01	ÞŌ	Вариант – включает имя элемента управления или индикатора, информа- цию о типе данных и сами данные	Фиолетовый	_
170) [/0	I/O name – передает имя канала платы сбора данных (DAQ), ресурсное имя VISA или логическое имя IVI с целью их конфигурации	Фиолетовый	_
		Рисунок – отображает рисунок, который содержит линии, окружности, текст и другие графические элементы	Синий	_

Настройка параметров объектов лицевой панели (элементов управления и индикаторов) и терминалов блок-диаграммы производится с помощью контекстного меню, содержащего команды и опции. Контекстное меню открывается с помощью щелчка ПКМ на объекте. Опции, входящие в состав контекстного меню, зависят от типа объекта. Вместе с тем в состав контекстного меню многих объектов входит ряд одинаковых пунктов. Сочетание общего и индивидуального можно показать на примере контекстных меню числового элемента управления (рис. 1.14) и его терминала (рис. 1.15).

Краткие описания функций пунктов контекстных меню приведены в таблице.

Видимые объекты	Позволяет показывать или скрывать определенные элементы
(Visible Items)	оформления, такие как ярлыки, заголовки, полосы прокрутки или
	соединительные терминалы
Найти терминал	LabVIEW находит и выделяет на блок-диаграмме терминал
(Find Terminal)	элемента лицевой панели
Найти элемент	LabVIEW находит и выделяет на лицевой панели элемент управле-
управления	ния, имеющий заданный терминал на блок-диаграмме
(Find Control)	
Заменить	Позволяет заменить элемент управления лицевой панели на
на индикатор	индикатор. Для индикатора аналогичный пункт выполняет обратное
(Change to Indicator)	действие
Сделать элемент	Позволяет сделать элемент управления невидимым. Для невиди-
управления	мого элемента аналогичный пункт меню выполняет обратное
невидимым	действие
(Hide Control)	