

СОДЕРЖАНИЕ



Введение	11
Предисловие	13
Обращение к читателю	13
Организация	14
Соглашение о приоритете правил	15
Примечания	15
Благодарности	16
Об авторе	18
▼ 1	
Зачем нужен стиль	19
1.1. Зачем нужен стиль?	19
1.1.1. Простота использования	24
1.1.2. Эффективность	25
1.1.3. Читательность	27
1.1.4. Простота поддержки	29
1.1.5. Надежность	30
1.1.6. Простота	34
1.1.7. Производительность	34
1.1.8. Поддержание стиля	36
1.2. Стиль или быстрота?	37
▼ 2	
Подготовка к хорошему стилю	39
2.1. Техническое задание	39

2.1.1. Советы по документации	42
2.1.2. Проектная документация в LabVIEW	44
2.2. Проектирование	47
2.2.1. Поиск полезных источников	48
2.2.2. Разработка пробной версии	49
2.2.3. Вернитесь к ТЗ	50
2.3. Настройка среды LabVIEW	51
2.3.1. Диалоговое окно опций LabVIEW	51
2.3.2. Повторное использование кода	53
2.4. Структура проекта, именование файлов и управление	57
2.4.1. Расположение файлов	58
2.4.2. Проект LabVIEW	62
2.4.3. Именование файлов	64
2.4.4. Управление исходниками	65
Ссылки	66

▼ 3

Стиль лицевой панели	67
3.1. Расположение	68
3.1.1. Общие правила	69
3.1.2. Панель пользовательского ВП	71
3.1.3. Лицевая панель подприбора	79
3.2. Текст	80
3.2.1. Общие правила	80
3.2.2. Метки элементов	85
3.2.3. Текст в подприборах	88
3.2.4. Текст промышленных ВП	88
3.3. Цвет	90
3.4. Навигация по приложению	92
3.4.1. Элементы управления	92
3.4.2. Ваш стиль	95
3.5. Примеры	96
3.5.1. ВПП из блок-диаграммы	96
3.5.2. Служебный диалог	98
3.5.3. Тестирование и сортировка конденсаторов	99
3.5.4. Центрифуга	101
3.5.5. Спектральный анализатор	103
3.5.6. Интерфейс управления парапланом	104
Ссылки	105

▼ 4

Блок-диаграмма	106
4.1. Расположение	107
4.1.1. Основные части	107
4.1.2. Блоки в ВПП	108
4.2. Соединения	112
4.2.1. Секреты аккуратного соединения	112
4.2.2. Использование кластеров	116
4.3. Поток данных	120
4.3.1. Основы потока данных	120
4.3.2. Когда нужны переменные и последовательности	124
4.3.3. Здесь переменные и последовательности не нужны	126
4.3.4. Оптимизируем поток данных	131
4.4. Примеры	136
4.4.1. ВПП из участка кода	136
4.4.2. ВП Excessively Nested	137
4.4.3. ВП Haphazard	139
4.4.4. ВП Right to Left	140
4.4.5. ВП Left to Right	141
4.4.6. ВП Centrifuge DAQ	142
4.4.7. ВП Screw Inspection	143
4.4.8. ВП Optical Filter Test	144
Ссылки	147

▼ 5

Иконка и контакты	148
5.1. Иконка	150
5.1.1. Основные правила	151
5.1.2. Хитрости создания иконок	154
5.1.3. Международные иконки	158
5.2. Соединительная панель	159
5.3. Примеры	165
5.3.1. Доказательства от противного	165
5.3.2. Драйверы приборов	167
5.3.3. Разные примеры	171
5.3.4. Показательные примеры	173
Ссылки	175

▼ 6

Структуры данных	176
6.1. Методология разработки конструкций данных	177
6.1.1. Выбор элементов управления и типов данных	177
6.1.2. Настройка свойств	189
6.1.3. Создание конструкторов данных	190
6.2. Простые типы данных	193
6.2.1. Логические переменные	193
6.2.2. Численные элементы	196
6.2.3. Специальные численные данные	198
6.2.4. Строка, путь и изображение	201
6.3. Конструкты данных	203
6.3.1. Простые массивы и кластеры	203
6.3.2. Специальные конструкты данных	210
6.3.3. Вложенные структуры данных	211
6.4. Примеры	217
6.4.1. ВП Thermometer	217
6.4.2. Вариант OpenG	219
6.4.3. Случайные данные	222
6.4.4. ВП Centrifuge DAQ	222
Ссылки	225

▼ 7

Обработка ошибок	226
7.1. Основы обработки ошибок	227
7.1.1. Отслеживание ошибок	228
7.1.2. Отчеты об ошибках	234
7.1.3. Коды ошибок	239
7.2. Обработка ошибок в ВПП	241
7.3. Определение приоритетов ошибок	246
7.4. Советы по обработке ошибок	252
7.4.1. Соединение структур	252
7.4.2. Слияние ошибок	252
7.4.3. Очищение ошибок	254
7.4.4. Автоматическая обработка ошибок	256
7.5. Примеры	257
7.5.1. Постоянное получение данных и запись в файл	257
7.5.2. Suss Interface Toolkit	258

7.5.3. Слияние параллельных ошибок	259
7.5.4. ВП Screw Inspection	260
7.5.5. ВП Test Executive	262
Ссылки	265

▼ 8

Шаблоны	266
8.1. Простые шаблоны	268
8.1.1. Шаблон ВПП Immediate	268
8.1.2. Шаблон Functional Global	271
8.1.3. Шаблон Continuous Loop	272
8.1.4. Цикл с обработкой событий	278
8.2. Конечные автоматы	283
8.2.1. Классический конечный автомат	287
8.2.2. Конечный автомат с очередью	288
8.2.3. Событийно управляемый конечный автомат	292
8.2.4. Автомат событий	295
8.3. Составные шаблоны	298
8.3.1. Параллельные циклы	300
8.4. Объектные структуры сложных приложений	304
8.4.1. Динамическая объектная структура	304
8.4.2. Объектная структура приложения со многими циклами	311
8.4.3. Модульная объектная структура приложения со многими циклами	316
8.5. Примеры	320
8.5.1. ВП Elapsed Time	320
8.5.2. ВП Poll Instrument Response	321
8.5.3. Нетрадиционный конечный автомат	323
8.5.4. ВП Centrifuge DAQ	326
8.5.5. Утилита управления датчиком	327
8.5.6. Распределенная управляющая система	328
Ссылки	330

▼ 9

Документация	331
9.1. Документация лицевой панели	333
9.2. Блок-диаграмма	337
9.3. Описание иконки и ВП	345

9.4. Online-документация	346
9.5. Примеры	350
9.5.1. ВПП из участка блок-диаграммы	350
9.5.2. ВП Filter Test	350
9.5.3. Тщательное описание элементов управления	351
9.5.4. Профиль температуры	352
Ссылки	353

▼ 10

Экспертная оценка программы	354
10.1. Самостоятельная экспертная оценка программы	355
10.1.1. ВП Analyzer Toolkit	355
10.1.2. Контрольный список ручной проверки	366
10.2. Экспертные проверки	370
Ссылки	375

▼ Приложение А

Глоссарий	376
------------------------	------------

▼ Приложение Б

Сводка основных правил стиля	388
---	------------

Предметный указатель	397
-----------------------------------	------------

Введение



В течение длительного времени я работал над созданием ВП в R&D-отделе LabVIEW в National Instruments. Во время первой недели моего обучения в LabVIEW я решил написать LabVIEW-версию моей любимой карточной игры «Set». Мне потребовалось немало времени, но в конце концов я закончил ее, и это был, вероятно, самый ужасный код LabVIEW, который вы когда-либо видели. Как сказал бы Питер Блум (Peter Blume), мой код был образцом «Спагетти». Хуже того, на лицевой панели было множество ярких, кричащих элементов неоновых цветов. Мой код функционировал, но его было невозможно использовать и уж тем более поддерживать. Спустя несколько лет, когда я стал куда лучше разбираться в LabVIEW, я попытался добавить к старому коду несколько новых функций, но быстро сдался, потому что у меня не было ни малейшего понятия, как работает мой код.

Много лет спустя я могу с уверенностью сказать, что мой стиль программирования улучшался не по дням, а по часам. Однако я также без сомнения могу сказать, что все мои проблемы со стилем программирования на LabVIEW в то время исчезли бы, если бы тогда существовала книга «Стиль LabVIEW». Эта книга полностью самосогласованный источник информации, затрагивающий каждый аспект стиля программирования ВП от верхних уровней (прогнозирование и планирование) до мелких деталей (проводники данных со слишком большим количеством изгибов). Если бы я прочел эту книгу во время изучения LabVIEW, мой код было бы во много раз проще использовать и поддерживать с самого начала.

Книгу «Стиль LabVIEW» необходимо прочитать каждому разработчику программного обеспечения на LabVIEW. Не только потому, что она содержит основные правила стиля для новичков, в ней также находятся необходимые обновления для ветеранов программирования. В частности, в главах 6 «Структуры данных», 7 «Обработка ошибок», 8 «Шаблоны» обсуждаются ключевые моменты и техники программирования, недоступные в других источниках. В качестве одного из адвокатов стиля ВП в LabVIEW R&D я настоятельно рекомендую всем новым разработчикам в моей команде прочитать эту книгу, и я уверен, что буду не раз к ней обращаться в ходе дискуссий с моими более опытными коллегами.

Другой поистине уникальной особенностью этой книги является впечатляющее количество примеров, которые Питер использует для иллюстрации хорошего

(а иногда и плохого) стиля программирования. За свой пятнадцатилетний опыт программирования на LabVIEW Питер накопил огромную библиотеку ВП, которые написали он, его работники и клиенты. И он использует эту библиотеку, чтобы проиллюстрировать подходящие моменты в каждой главе. Более того, иногда он рассматривает один пример на протяжении нескольких глав и постепенно применяет к нему все новые и новые правила стиля. Таким образом, мы можем «в реальном времени» видеть, как хороший стиль положительно влияет на разработку ВП.

В то время как LabVIEW отмечает 20-летие, вдохновляя инженеров, ученых и даже детей по всему миру, я абсолютно уверен, что читатели этой книги по стилю LabVIEW оценят то невероятное количество времени и усилий, которое они сэкономят, разрабатывая ВП по правилам хорошего стиля. Так что, если вы новичок в LabVIEW, приготовьтесь к тому, что ваши ВП станут – как сказал бы Питер – «внушающими благоговейный ужас», а если вы эксперт в LabVIEW, то вы тоже узнаете много нового. Как и я!

Дарен Наттингер

Штатный разработчик программного обеспечения, LabVIEW R&D
National Instruments Corporation.

Дарен Наттингер (Darren Nattinger) проработал в National Instruments восемь лет. Сейчас он является ведущим разработчиком VI Analyzer Toolkit и Report Generation Toolkit for Microsoft Office. Он также был рецензентом книги «Стиль LabVIEW».

Предисловие



Книга «Стиль LabVIEW» содержит правила стиля, направленные на улучшение производительности, читабельности, эффективности; упрощение технической поддержки приложений в LabVIEW. В книге приведены объяснения всех правил, с примерами и иллюстрациями. В книге используются работы пионеров LabVIEW-сообщества, эволюционировавшие за годы использования в Bloomy Controls, которые были отрецензированы уважаемыми членами LabVIEW-сообщества. Я призываю вас учиться на моем опыте и опыте сотрудников Bloomy Controls, читая книгу «Стиль LabVIEW». Я надеюсь, что вы получите такое же удовольствие от чтения этой книги, какое получил я, когда писал ее!

Обращение к читателю

Предполагаемые читатели этой книги – это, прежде всего, разработчики, менеджеры и организации, занимающиеся разработкой приложений на LabVIEW. Вы должны обладать практическим знанием фундаментальных принципов LabVIEW и терминологии на уровне курсов LabVIEW Basics I и II, кроме того, необходим опыт разработки и внедрения приложений. Опытные новички могут использовать эту книгу, чтобы сформировать хорошие навыки программирования на LabVIEW в самом начале своей карьеры. Более опытные разработчики, которые овладели основами и готовы перейти на новый уровень, получают наибольшую пользу от прочтения этой книги. Без сомнения, вы уже poznali мощь и гибкость LabVIEW и готовы сконцентрироваться на стиле. Продвинутые разработчики подкрепят свои знания и опыт, получат полезную тему для обсуждения с коллегами. Вы можете использовать книгу «Стиль LabVIEW», чтобы уменьшить затраты сил на обучение и поддержку внутри организации. Менеджеры и организации, которые нанимают множество разработчиков и пользователей, могут получить максимум прибыли, приняв данные правила стиля как стандарт для всей организации. Этот подход гарантирует качество и согласованность во всей организации и поможет вам удовлетворить всем стандартам качества.

Организация

Главы в книге «Стиль LabVIEW» содержат правила стиля и примеры, разбитые по темам. В главе 1 «Зачем нужен стиль» обсуждается связь между стилем и простотой использования, эффективностью, читабельностью, простотой поддержки, устойчивостью к ошибкам и производительностью. В главе 2 «Приготовка к хорошему стилю» представлены некоторые соображения, влияющие на стиль еще до того, как вы начнете программировать, к ним относятся спецификация, настройка среды LabVIEW, организация проектов и файлов. Кроме того, в ней представлен специализированный стандарт проектной спецификации в LabVIEW. В главах 3 «Стиль лицевой панели», 4 «Блок-диаграмма» и 5 «Иконка и контакты» рассмотрены основы разработки ВП. В главе 3 находятся правила, касающиеся компоновки, текста, цвета и навигации. В ней подчеркивается различие между правилами для лицевых панелей ВП с графическим интерфейсом и для ВПП. Правила в главе 4 относятся к расположению, соединению элементов и организации потока данных наряду с методами оптимизации потока данных. В главе 5 обсуждается опыт создания хороших иконок и настройки ярлычков, в том числе затрагиваются вопросы шаблонов соединительной панели, назначений и общепринятых норм.

В главе 6 «Структуры данных» представлены правила выбора типа данных и работы с массивами и кластерами. В этой главе также находится несколько таблиц для упрощения выбора типа данных. В этой главе также представлены правила и примеры оптимизации ВП с использованием сложных структур данных. Главы 7 «Обработка ошибок», 8 «Шаблоны», 9 «Документация» выходят за рамки базовых понятий. В главе 7 представлены согласованные правила для последовательной обработки ошибок в приложении, наряду с некоторыми специфичными моментами обработки ошибок внутри ВПП. В главе 8 обсуждается обычная архитектура ВП, обеспечивающая хороший стиль, начиная с простых шаблонов ВПП и кончая шаблонами со многими циклами. В этой же главе обсуждается несколько вариантов конечного автомата в LabVIEW. Кроме того, в главе 8 представлены 3 сложные структуры приложений, включая динамическую структуру, использующую плагины; структуру со многими циклами и модульную структуру со многими циклами, использующую циклические ВПП. В главе 9 представлены правила по документированию вашего исходного кода, включая лицевую панель, блок-диаграмму, иконку и описание ВП. Помимо этого, обсуждается создание и размещение документов on-line. В главе 10 «Экспертная оценка программы», предложено несколько методов проверки исходного кода и усиления правил стиля, включая самопроверки по собственному контрольному списку, автоматические проверки с использованием LabVIEW VI Analyzer Toolkit и проверки другими специалистами. Рассматривается эволюция приложения на каждой стадии.

Приложение включает глоссарий и суммарный свод правил. В Приложении А «Глоссарий» находится список терминов и определений, многие термины в LabVIEW эволюционируют и зависят от контекста. Каждому появляющемуся в тексте специфическому термину или термину, который можно воспринять

неоднозначно, дается определение, и в последующих главах он используется уже именно в этом значении. Для удобства все определения собраны в глоссарии. В Приложении Б «Сводка основных правил стиля» представлены все правила стиля из каждой главы книги.

Соглашение о приоритете правил

Во всей книге «Стиль LabVIEW» используется два уровня приоритета правил, которые различаются шрифтом (полужирный курсив или просто курсив).

Правила с высшим приоритетом – это законы, которые должны соблюдаться всегда, за редким исключением. Они выделены жирным курсивом:

Правило 2.2 Составьте список требований, запишите техническое задание

Правила с обычным приоритетом носят рекомендательный характер, следовать которым считается хорошим тоном, но они не так критичны, как правила с высоким приоритетом, и допускают больше исключений. Они выделены простым курсивом:

Правило 2.3 Придерживайтесь хорошего стиля программирования в LabVIEW

Примечания

1. В разделе «Благодарности» приведен список рецензентов и людей, внесших свой вклад в развитие стиля LabVIEW.
2. Bloomy Controls является партнером National Instruments с офисами в Виндзоре, Коннектикуте, Милфорде, Массачусетсе; и Форте Ли, Нью Джерси. Более подробная информация доступна на сайте www.bloomy.com.
3. LabVIEW Basic I и II – это недельные курсы, предлагаемые сертифицированными центрами обучения NI. Более подробная информация доступна на сайте www.ni.com/training и LabVIEW.ilc.edu.ru.