



Самоучитель

Никита Культин

Основы программирования

Delphi 7

2-е издание



Среда разработки

Назначение базовых компонентов

Программирование графики,
мультимедиа и баз данных

Создание справочной системы
и установочного CD

Никита Культин

Основы программирования
в Delphi 7
2-е издание

Санкт-Петербург

«БХВ-Петербург»

2009

УДК 681.3.068+800.92Delphi 7

ББК 32.973.26-018.1

K90

Культин Н. Б.

K90 Основы программирования в Delphi. — 2-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2009. — 640 с.: ил. + CD-ROM — (Самоучитель)

ISBN 978-5-9775-0310-5

Книга является руководством по программированию в среде Delphi 7. Описывается весь процесс разработки программы: от создания диалогового окна до организации справочной системы и создания установочного компакт-диска. Материал включает ряд тем, которые, как правило, остаются за рамками книг, адресованных начинающим — обработка символьной информации, использование динамических структур, работа с файлами. Рассматриваются вопросы программирования графики, мультимедиа и работа с базами данных. Приведено описание процесса создания справочной системы при помощи программы Microsoft HTML Help Workshop, установочного компакт-диска в InstallShield Express. Книга отличается доступностью изложения, большим количеством наглядных примеров. Во втором издании обновлены примеры и описана работа с базами данных Access на основе ADO. Прилагаемый компакт-диск содержит проекты, приведенные в книге.

Для начинающих программистов

УДК 681.3.068+800.92Delphi 7

ББК 32.973.26-018.1

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Игорь Шишигин</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Анна Кузьмина</i>
Компьютерная верстка	<i>Натальи Смирновой</i>
Корректор	<i>Зинаида Дмитриева</i>
Дизайн серии	<i>Инны Тачиной</i>
Оформление обложки	<i>Елены Беляевой</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 28.08.08.

Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 51,6.

Тираж 3000 экз. Заказ №

"БХВ-Петербург", 194354, Санкт-Петербург, ул. Есенина, 5Б.

Отпечатано с готовых диапозитивов

в ГУП "Типография "Наука"

199034, Санкт-Петербург, 9 линия, 12

ISBN 978-5-9775-0310-5

© Культин Н. Б., 2008

© Оформление, издательство "БХВ-Петербург", 2008

Оглавление

ПРЕДИСЛОВИЕ	1
Delphi — что это?	1
Об этой книге	2
ГЛАВА 1. СРЕДА ПРОГРАММИРОВАНИЯ DELPHI.....	5
Установка	5
Начало работы.....	8
Первый проект	12
Форма	12
Компоненты.....	17
Событие и процедура обработки события.....	26
Редактор кода	31
Справочная система.....	36
Структура проекта	36
Сохранение проекта.....	41
Компиляция	42
Запуск программы.....	46
Ошибки времени выполнения	47
Внесение изменений	48
Окончательная настройка приложения	51
Установка приложения на другой компьютер	54
ГЛАВА 2. ОСНОВЫ ПРОГРАММИРОВАНИЯ	57
Программа	57
Этапы разработки программы	57
Спецификация	58
Разработка алгоритма	58
Кодирование	58
Отладка	58
Тестирование	59
Создание справочной системы	59
Создание установочного CD.....	59
Алгоритм и программа.....	60
Компиляция	64

Язык программирования Delphi	65
Тип данных	65
Переменная	67
Константы	69
Инструкция присваивания	71
Стандартные функции	75
Ввод данных	78
Вывод результатов	80
Процедуры и функции	84
Запись инструкций программы	86
Комментарии	88
Стиль программирования	88
ГЛАВА 3. УПРАВЛЯЮЩИЕ СТРУКТУРЫ ЯЗЫКА DELPHI.....	91
Условие	92
Выбор	95
Инструкция <i>if</i>	95
Инструкция <i>case</i>	103
Циклы	116
Инструкция <i>for</i>	116
Инструкция <i>while</i>	121
Инструкция <i>repeat</i>	124
Инструкция <i>goto</i>	127
ГЛАВА 4. СИМВОЛЫ И СТРОКИ	129
Символы	129
Строки	133
Операции со строками	134
ГЛАВА 5. КОНСОЛЬНОЕ ПРИЛОЖЕНИЕ.....	137
Инструкции <i>write</i> и <i>writeln</i>	137
Инструкции <i>read</i> и <i>readln</i>	139
Создание консольного приложения	141
ГЛАВА 6. МАССИВЫ	145
Объявление массива	145
Операции с массивами	147
Вывод массива	147
Ввод массива	149
Поиск минимального (максимального) элемента массива	161

Поиск в массиве заданного элемента.....	164
Сортировка массива.....	174
Многомерные массивы.....	180
Ошибки при использовании массивов	186
ГЛАВА 7. ПРОЦЕДУРЫ И ФУНКЦИИ.....	189
Функция	194
Объявление функции	194
Использование функции.....	196
Процедура.....	200
Объявление процедуры	200
Использование процедуры.....	201
Повторное использование функций и процедур.....	204
Создание модуля	204
Использование модуля	206
ГЛАВА 8. ФАЙЛЫ.....	211
Объявление файла	211
Назначение файла	212
Вывод в файл.....	213
Открытие файла для записи.....	213
Запись в файл	213
Ошибки открытия файла.....	216
Закрытие файла	218
Пример программы.....	218
Ввод из файла.....	221
Открытие файла	222
Чтение данных из файла.....	223
Конец файла	225
ГЛАВА 9. ТИПЫ ДАННЫХ, ОПРЕДЕЛЯЕМЫЕ ПРОГРАММИСТОМ	229
Перечисляемый тип	229
Интервальный тип	231
Запись.....	232
Объявление записи	233
Инструкция <i>with</i>	234
Ввод и вывод записей в файл	235
Динамические структуры данных	246
Указатели.....	246
Динамические переменные.....	248
Списки.....	249
Упорядоченный список.....	254

ГЛАВА 10. ВВЕДЕНИЕ В ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ	263
Класс	263
Объект	264
Метод	266
Инкапсуляция и свойства объекта	267
Наследование	269
Директивы <i>Protected</i> и <i>Private</i>	270
Полиморфизм и виртуальные методы	271
Классы и объекты Delphi	276
ГЛАВА 11. ГРАФИКА	279
Холст	279
Карандаш и кисть	280
Карандаш	280
Кисть	282
Вывод текста	286
Методы вычерчивания графических примитивов	288
Линия	288
Ломаная линия	291
Окружность и эллипс	295
Дуга	296
Прямоугольник	297
Многоугольник	298
Сектор	299
Точка	300
Вывод иллюстраций	304
Битовые образы	310
Мультипликация	313
Метод базовой точки	316
Использование битовых образов	320
Баннер	329
ГЛАВА 12. МУЛЬТИМЕДИА	335
Компонент <i>Animate</i>	335
Компонент <i>MediaPlayer</i>	341
Воспроизведение звука	343
Запись звука	350
Просмотр видеороликов и анимации	352
Создание анимации	355

ГЛАВА 13. РЕКУРСИЯ.....	361
Понятие рекурсии	361
Примеры программ.....	365
Поиск файлов	365
Кривая Гильберта.....	370
Поиск пути.....	373
Поиск кратчайшего пути.....	380
ГЛАВА 14. ОТЛАДКА ПРОГРАММЫ.....	383
Классификация ошибок	383
Предотвращение и обработка ошибок.....	385
Отладчик.....	388
Трассировка программы.....	388
Точки останова программы.....	390
Наблюдение значений переменных	393
ГЛАВА 15. СПРАВОЧНАЯ СИСТЕМА	397
Справочная система WinHelp	399
Файл справочной информации.....	400
Создание справочной системы	403
Создание проекта справочной системы.....	403
Добавление в проект файла справочной информации	405
Характеристики окна справочной системы.....	405
Назначение числовых значений идентификаторам разделов справки	408
Компиляция проекта.....	409
Доступ к справочной информации.....	410
HTML Help Workshop.....	412
Подготовка справочной информации	412
Создание файла справки	418
Отображение справочной информации	426
ГЛАВА 16. ПРИМЕРЫ ПРОГРАММ.....	429
<i>Экзаменатор</i>	429
Требования к программе	430
Файл теста.....	430
Форма приложения	433
Отображение иллюстрации.....	435
Доступ к файлу теста	435
Текст программы	437
Запуск программы.....	447

Игра <i>Сапер</i>	448
Правила.....	449
Представление данных.....	450
Форма.....	451
Игровое поле.....	454
Начало игры.....	454
Игра.....	458
Справочная информация.....	462
Информация о программе.....	463
Листинги.....	466
MP3-плеер.....	477
Форма.....	477
Регулятор громкости.....	481
Перемещение окна.....	482
Листинг.....	483

ГЛАВА 17. КОМПОНЕНТ ПРОГРАММИСТА..... 491

Выбор базового класса.....	491
Создание модуля компонента.....	492
Тестирование модуля компонента.....	497
Установка компонента.....	500
Ресурсы компонента.....	500
Установка.....	502
Ошибки при установке компонента.....	505
Тестирование компонента.....	505
Удаление компонента.....	509
Настройка палитры компонентов.....	511

ГЛАВА 18. БАЗЫ ДАННЫХ..... 513

База данных и СУБД.....	513
Локальные и удаленные базы данных.....	513
Структура базы данных.....	514
Механизмы доступа к данным.....	514
Компоненты доступа к данным.....	515
Создание базы данных.....	515
Программа работы с базой данных.....	516
Доступ к данным.....	516
Отображение данных.....	521
Выбор информации из базы данных.....	527
Работа с базой данных в режиме формы.....	535
Установка программы работы с базой данных на другой компьютер.....	543

ГЛАВА 19. СОЗДАНИЕ УСТАНОВОЧНОГО ДИСКА	545
Программа InstallShield Express	545
Образ установочного диска	546
Новый проект	546
Структура.....	548
Выбор устанавливаемых компонентов.....	550
Конфигурирование системы пользователя	552
Настройка диалогов.....	553
Системные требования.....	556
Создание образа установочного CD	557
ЗАКЛЮЧЕНИЕ.....	559
ПРИЛОЖЕНИЯ.....	561
ПРИЛОЖЕНИЕ 1. DELPHI — КРАТКИЙ СПРАВОЧНИК.....	563
Язык программирования Delphi	563
Структура модуля	563
Основные типы данных.....	564
Инструкции выбора	565
Циклы.....	567
Безусловный переход.....	568
Объявление функции.....	568
Объявление процедуры	569
Форма.....	569
Базовые компоненты	571
<i>Label</i>	571
<i>Edit</i>	573
<i>Button</i>	574
<i>Memo</i>	575
<i>RadioButton</i>	576
<i>CheckBox</i>	577
<i>ListBox</i>	578
<i>ComboBox</i>	579
<i>StringGrid</i>	580
<i>Image</i>	581
<i>Timer</i>	583
<i>SpeedButton</i>	583
<i>UpDown</i>	585
<i>OpenDialog</i>	586

<i>SaveDialog</i>	587
<i>Animate</i>	588
<i>MediaPlayer</i>	589
Компоненты доступа к базам данных.....	590
<i>ADoConnection</i>	590
<i>ADoTable</i>	591
<i>ADoDataSet</i>	592
<i>ADoQuery</i>	593
<i>DataSource</i>	594
<i>DBText, DBEdit, DBMemo</i>	595
<i>DBGrid</i>	595
<i>DBNavigator</i>	597
Графика.....	599
<i>PaintBox</i>	599
<i>Canvas</i>	599
<i>Pen</i>	602
<i>Brush</i>	603
Цвет	603
Функции.....	604
Функции ввода и вывода.....	604
Математические функции	605
Функции преобразования.....	606
Функции манипулирования датами и временем.....	606
События	608
Исключения.....	609

ПРИЛОЖЕНИЕ 2. КОДИРОВКА СИМВОЛОВ611

ПРИЛОЖЕНИЕ 3. ПРЕДСТАВЛЕНИЕ ИНФОРМАЦИИ В ПАМЯТИ КОМПЬЮТЕРА..... 614

Десятичные и двоичные числа	614
Память компьютера	615

ПРИЛОЖЕНИЕ 4. ОПИСАНИЕ КОМПАКТ-ДИСКА 617

ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА 624

ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ..... 625

Предисловие

Delphi — что это?

В последнее время резко возрос интерес к программированию. Это связано с развитием и внедрением в повседневную жизнь информационно-коммуникационных технологий. Если человек имеет дело с компьютером, то рано или поздно у него возникает желание, а иногда и необходимость, программировать.

Среди пользователей персональных компьютеров в настоящее время наиболее популярно семейство операционных систем Windows, и, естественно, тот, кто собирается программировать, стремится писать программы, которые будут работать в этих системах.

Несколько лет назад рядовому программисту оставалось только мечтать о создании собственных программ, работающих в среде Windows, т. к. единственным средством разработки был Borland C++ for Windows, явно ориентированный на профессионалов, обладающих серьезными знаниями и опытом.

Бурное развитие вычислительной техники, потребность в эффективных средствах разработки программного обеспечения привели к появлению систем программирования, ориентированных на так называемую "быструю разработку", среди которых можно выделить Borland Delphi и Microsoft Visual Basic. В основе систем быстрой разработки (RAD-систем, Rapid Application Development — среда быстрой разработки приложений) лежит технология визуального проектирования и событийного программирования, суть которой заключается в том, что среда разработки берет на себя большую часть рутинной работы, оставляя программисту работу по конструированию диалоговых окон и функций обработки событий. Производительность программиста при использовании RAD-систем — фантастическая!

Delphi — это среда быстрой разработки, в которой в качестве языка программирования используется Object Pascal. В основе идеологии Delphi лежит технология визуального проектирования и методология объектно-ориентированного событийного программирования. Она позволяет создавать программы различного назначения: от простейших однооконных приложений до программ работы с распределенными базами данных.

В настоящее время разработчики используют разные версии Delphi, в том числе и Delphi 7. Эта версия также широко используется и в учебных заведениях: в школах, техникумах, вузах. И это не удивительно. Как показывает опыт, научившись работать в Delphi 7, программист без особых усилий сможет перейти на другие, современные версии.

Borland Delphi 7 может работать в среде операционных систем от Windows 98 до Windows XP. Особых требований, по современным меркам, к ресурсам компьютера пакет не предъявляет (это выгодно отличает его от последующих версий, требующих наличия на компьютере Microsoft .NET Framework). Процессор должен быть типа Pentium или Celeron с тактовой частотой не ниже 166 МГц (рекомендуется Pentium II 400 МГц). Компьютер должен иметь оперативную память объемом 128 Мбайт (рекомендуется 256 Мбайт). На жестком диске должно быть достаточное количество свободного пространства (для полной установки необходимо приблизительно 475 Мбайт).

Об этой книге

В книге, посвященной программированию в конкретной среде разработки, необходим баланс между тремя линиями: языком программирования, технологией и средой разработки. Уже при первом представлении среды разработки, описании ее возможностей у автора возникает проблема: чтобы описать процесс разработки программы, объяснить, как работает программа, нужно оперировать такими терминами, как *объект*, *событие*, *свойство*, понимание которых на начальном этапе изучения программирования весьма проблематично. Как поступить? Сначала дать описание языка, а затем приступить к описанию среды разработки и процесса программирования в Delphi? Очевидно, что это не лучшее решение. Поэтому при изложении материала принят подход, в основу которого положен принцип соблюдения баланса между описанием языка, технологией программирования и средой разработки. В начале книги некоторые понятия, без которых просто невозможно изложение материала, даются на уровне определений.

Книга, которую вы держите в руках, — это не описание языка программирования Delphi и среды разработки Delphi. Это пособие (руководство) по программированию на языке Delphi в среде Delphi 7. В нем рассмотрена вся цепочка, весь процесс создания программы: от разработки диалогового окна и функций обработки событий до создания справочной системы и установочной дискеты.

Цель этой книги может быть сформулирована так: научить программировать в среде Delphi, создавать законченные программы различного назначения: от простых однооконных приложений до вполне профессиональных программ работы с базами данных.

Научиться программировать можно, только программируя, решая конкретные задачи. При этом достигнутые в программировании успехи в значительной степени зависят от опыта. Поэтому, чтобы получить максимальную пользу от книги, вы должны работать с ней активно. Не занимайтесь просто чтением примеров, реализуйте их с помощью вашего компьютера. Не бойтесь экспериментировать — вносите изменения в программы. Чем больше вы сделаете самостоятельно, тем большему научитесь!

ГЛАВА 1



Среда программирования Delphi

В данной главе описывается процесс установки Delphi. На примере программы **Скорость**, позволяющей вычислить скорость, с которой бегун пробежал дистанцию, демонстрируется процесс разработки программы в Delphi, технология визуального проектирования и событийного программирования, вводятся основные понятия и термины.

Установка

Установка Delphi 7 на компьютер выполняется с CD, на котором находятся все необходимые файлы, а также программа инициализации установки (Delphi Setup Launcher). Программа инициализации установки запускается автоматически, как только установочный диск будет помещен в дисковод.

В результате запуска программы инициализации установки на экране появляется окно **Delphi 7 Setup Launcher** (рис. 1.1), в котором перечислены программные продукты, которые можно установить на компьютер. Это, прежде всего, Delphi 7, сервер баз данных InterBase 6.5 и InstallShield Express — утилита, предназначенная для создания установочных CD.

Для того чтобы активизировать процесс установки Delphi, следует щелкнуть на строке **Delphi 7**. Процесс установки Delphi обычный. После ввода серийного номера (Serial Number) и ключа (Authorization Key) на экране сначала появляется окно с лицензионным соглашением, затем — окно **Setup Type** (рис. 1.2), в котором можно выбрать один из возможных вариантов установки: **Typical** (Обычный), **Compact** (Компактный) или **Custom** (Выборочный, определяемый пользователем).

Обычный вариант установки предполагает, что на жесткий диск компьютера будут скопированы все компоненты Delphi.



Рис. 1.1. В окне **Delphi 7 Setup Launcher** перечислены компоненты, которые можно установить на компьютер

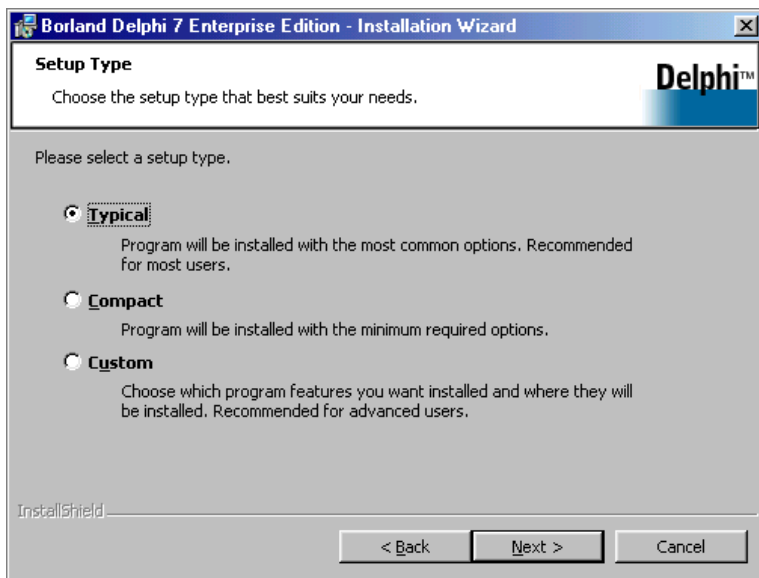


Рис. 1.2. В диалоговом окне **Setup Type** нужно выбрать вариант установки

Обычный вариант установки требует наибольшего свободного места на жестком диске компьютера, порядка 475 Мбайт (для комплекта Enterprise), однако

если на жестком диске компьютера достаточно свободного места, лучше выбрать именно этот вариант.

При *компактной установке* на жесткий диск компьютера копируются только самые необходимые компоненты Delphi. Компактный вариант требует наименьшего количества свободного дискового пространства. Однако в этом случае некоторые возможности среды разработки Delphi будут недоступны. В частности, при компактной установке на жесткий диск не копируются файлы справочной системы, некоторые компоненты и утилиты, примеры.

Выборочный вариант установки позволяет программисту выбрать только необходимые для работы инструменты и компоненты Delphi. Обычно этот вариант установки используют опытные программисты. Выборочный вариант можно предпочесть и в том случае, если на диске компьютера недостаточно свободного места для полной установки.

Выбрав вариант установки, нажмите кнопку **Next**. Если был выбран вариант установки **Custom**, то открывается диалоговое окно **Custom Setup** (рис. 1.3), в котором можно выбрать устанавливаемые компоненты, точнее — указать компоненты, которые устанавливать не надо. Чтобы запретить установку компонента, нужно щелкнуть на изображении диска слева от названия компонента и из появившегося меню выбрать команду **Do Not Install**.

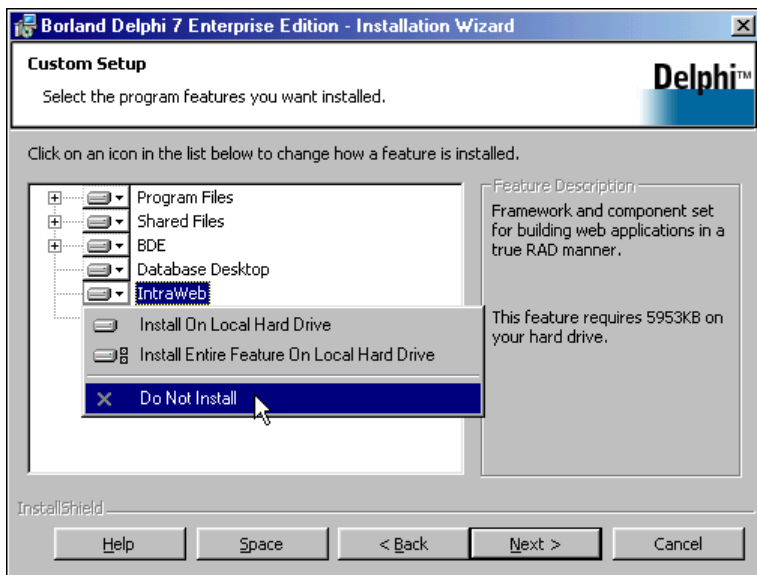


Рис. 1.3. Запрет установки компонента

Если выбран тип установки **Typical**, то в результате щелчка на кнопке **Next** открывается окно **Destination Folder**, в котором указаны каталоги, куда будет установлен пакет Delphi и его компоненты.

Очередной щелчок на кнопке **Next** открывает окно **Save Installation Database**, в котором пользователю предлагается сохранить информацию о процессе установки на жестком диске компьютера, что обеспечит возможность деинсталляции Delphi в дальнейшем без использования установочного CD. На этом процесс подготовки к установке заканчивается. На экране появляется окно **Ready To Install the Program**, щелчок на кнопке **Install** в котором активизирует процесс установки.

По окончании процесса установки на экране появляется окно с информационным сообщением о том, что установка выполнена (рис. 1.4). Щелчок на кнопке **Finish** закрывает это окно.

Теперь можно приступить к работе — запустить Delphi.

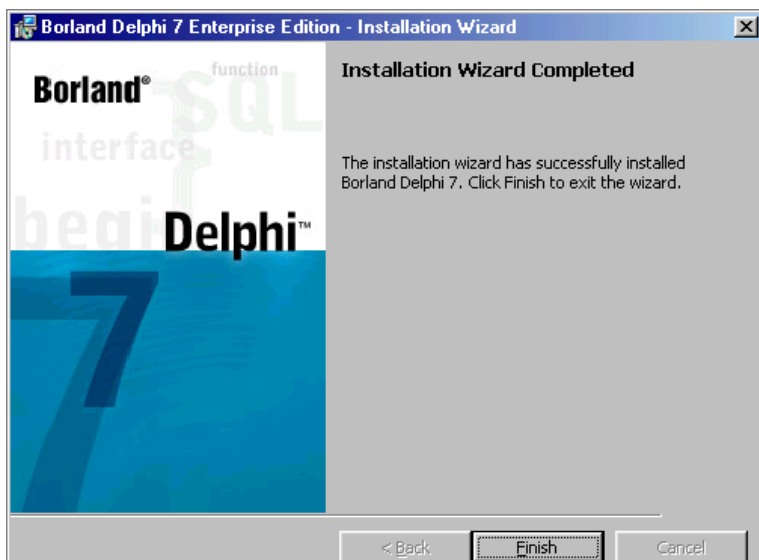


Рис. 1.4. Процесс установки завершен

Начало работы

Запускается Delphi обычным образом — выбором в меню **Пуск** команды **Все программы** ▶ **Borland Delphi 7** ▶ **Delphi 7** (рис. 1.5).

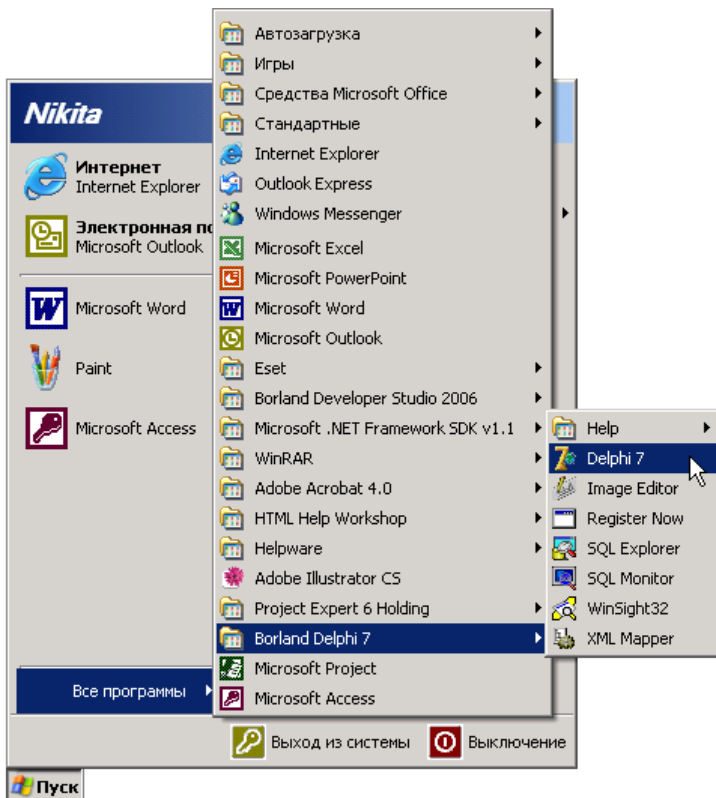


Рис. 1.5. Запуск Delphi

Вид экрана после запуска Delphi несколько необычен (рис. 1.6). Вместо одного окна на экране появляются пять:

- главное окно — **Delphi 7**;
- окно формы — **Form1**;
- окно редактора свойств объектов — **Object Inspector**;
- окно списка объектов — **Object TreeView**;
- окно редактора кода — **Unit1.pas**.

В главном окне (рис. 1.7) находится меню команд, панели инструментов и палитра компонентов.

Окно формы (**Form1**) представляет собой заготовку окна разрабатываемого приложения.

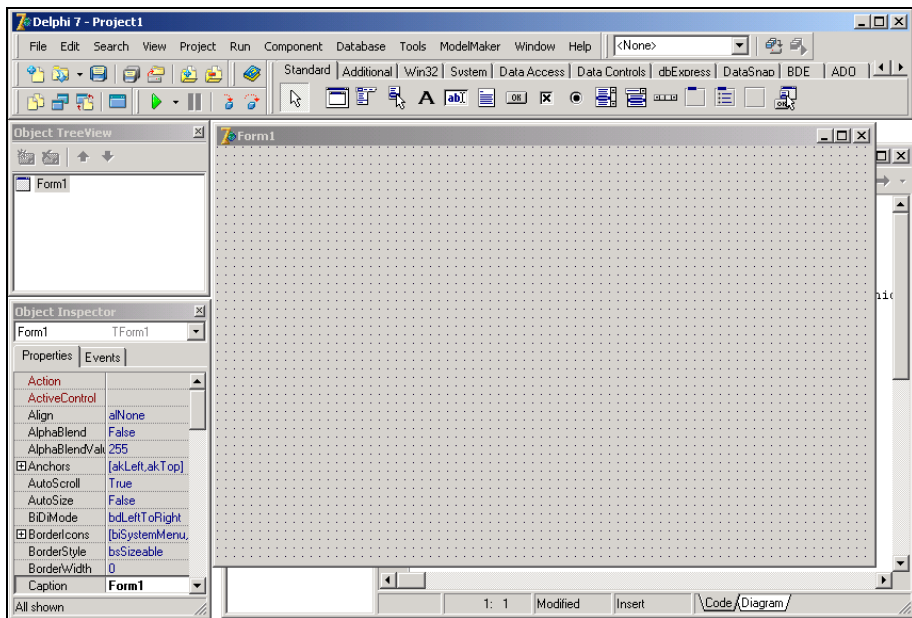


Рис. 1.6. Вид экрана после запуска Delphi

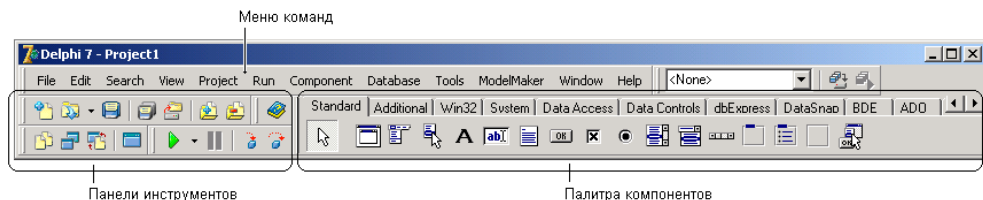


Рис. 1.7. Главное окно

ЗАМЕЧАНИЕ

Программное обеспечение принято делить на *системное* и *прикладное*. Системное программное обеспечение — это все то, что составляет операционную систему. Остальные программы, предназначенные для решения *прикладных* задач, принято называть *прикладными* или *приложениями*.

Окно **Object Inspector** (рис. 1.8) предназначено для редактирования значений свойств объектов. В терминологии визуального проектирования *объекты* — это диалоговые окна и элементы управления (поля редактирования, поля отображения текста, командные кнопки, переключатели и др.).

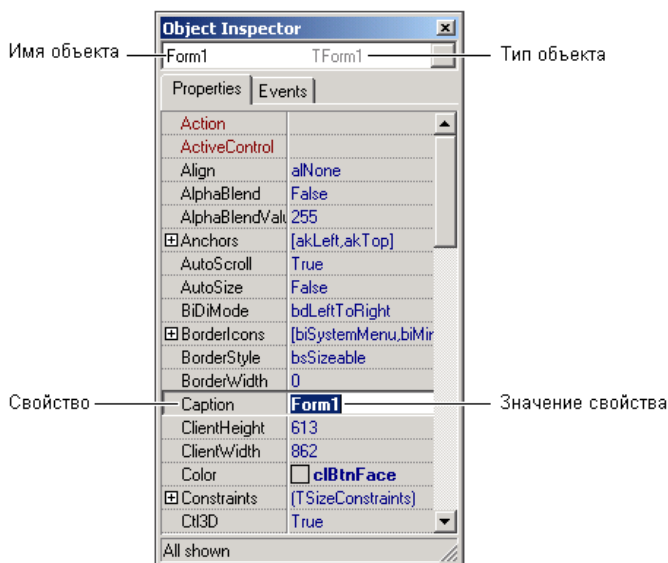


Рис. 1.8. На вкладке **Properties** перечислены свойства объекта и указаны их значения

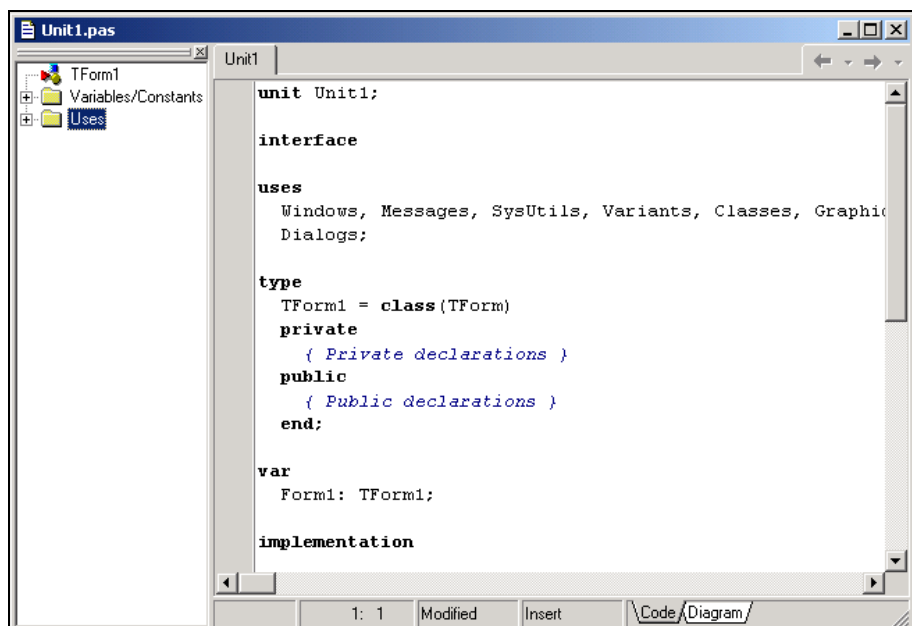


Рис. 1.9. Окно редактора кода

Свойства объекта — это характеристики, определяющие вид, положение и поведение объекта. Например, свойства `Width` и `Height` задают размер (ширину и высоту) формы, свойства `Top` и `Left` — положение формы на экране, свойство `Caption` — текст заголовка.

Окно редактора кода (рис. 1.9), которое можно увидеть, отодвинув в сторону окно формы или нажав функциональную клавишу <F12>, в начале работы над новым проектом содержит сформированный Delphi шаблон программы.

Первый проект

Процесс создания программы в Delphi рассмотрим на примере — создадим приложение, используя которое можно вычислить скорость, с которой спортсмен пробежал дистанцию. Вид окна программы во время ее работы приведен на рис. 1.10.

Для начала работы над новым приложением выберите в меню **File** команду **New ▶ Application**.

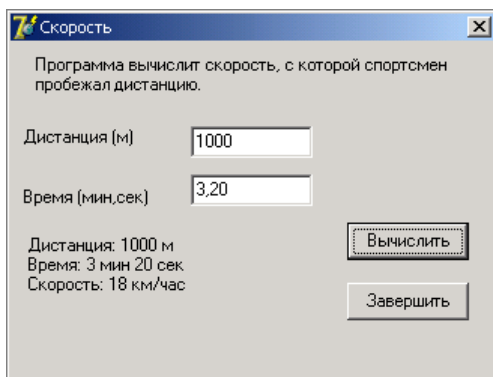


Рис. 1.10. Окно программы вычисления скорости

Форма

Работа над *новым проектом*, так принято называть разрабатываемую программу, начинается с создания стартовой формы.

Форма создается путем изменения значений ее свойств и добавления к форме необходимых компонентов (полей редактирования, полей отображения текста, командных кнопок и др.).

Свойства формы (табл. 1.1) определяют ее вид: размер, текст заголовка, вид рамки, а также положение на экране.

Для просмотра и изменения значений свойств формы и ее компонентов используется окно **Object Inspector**. В верхней части окна **Object Inspector** указано имя объекта, значения свойств которого отображается в данный момент. В левой колонке вкладки **Properties** (Свойства) перечислены свойства объекта, а в правой указаны их значения.

Таблица 1.1. Свойства формы (объекта TForm)

Свойство	Описание
Name	Имя формы. Используется для доступа к форме
Caption	Текст заголовка
Width	Ширина формы
Height	Высота формы
Position	Положение окна в момент первого его появления на экране (poDesktopCenter — в центре рабочего стола; poCenterScreen — в центре экрана; poOwnerFormCenter — в центре родительского окна; poDesigned — положение окна определяют значения свойств Top и Left)
Top	Расстояние от верхней границы формы до верхней границы экрана
Left	Расстояние от левой границы формы до левой границы экрана
BorderStyle	Вид границы. Граница может быть обычной (bsSizeable), тонкой (bsSingle) или отсутствовать (bsNone). Если у окна обычная граница, то во время работы программы пользователь может при помощи мыши изменить размер окна. Изменить размер окна с тонкой границей нельзя. Если граница отсутствует, то на экран во время работы программы будет выведено окно без заголовка. Положение и размер такого окна во время работы программы изменить нельзя
BorderIcons	Кнопки управления окном. Значение свойства определяет, какие кнопки управления окном будут доступны пользователю во время работы программы. Значение свойства задается путем присвоения значений уточняющим свойствам biSystemMenu, biMinimize, biMaximize и biHelp. Свойство biSystemMenu определяет доступность кнопки системного меню, biMinimize — кнопки Свернуть , biMaximize — кнопки Развернуть , biHelp — кнопки Справка

Таблица 1.1 (окончание)

Свойство	Описание
Icon	Значок в заголовке диалогового окна, обозначающий кнопку вывода системного меню
Color	Цвет фона. Цвет можно задать, указав название цвета или привязку к текущей цветовой схеме операционной системы. Во втором случае цвет определяется текущей цветовой схемой, выбранным компонентом привязки и меняется при изменении цветовой схемы операционной системы
Font	Шрифт, используемый "по умолчанию" компонентами, находящимися на поверхности формы. Изменение свойства Font формы приводит к автоматическому изменению свойства Font компонента, располагающегося на поверхности формы. То есть компоненты наследуют свойство Font от формы (имеется возможность запретить наследование)

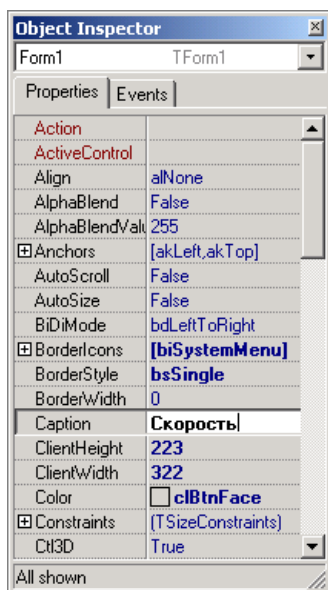


Рис. 1.11. Установка значения свойства путем ввода значения

При создании формы в первую очередь следует изменить значение свойства `Caption` (Заголовок). В нашем примере надо заменить текст `Form1` на `Скорость`. Чтобы это сделать, нужно в окне **Object Inspector** щелкнуть мышью в

строке `Caption` (в результате в поле значения свойства появится курсор) и ввести текст `Скорость` (рис. 1.11).

Аналогичным образом следует установить значения свойств `Height` и `Width`, которые определяют размер формы — соответственно, высоту и ширину.

Размер формы и ее положение на экране, а также размер компонентов и их положение на поверхности формы задают в пикселах, т. е. точках экрана. Свойствам `Height` и `Width` надо присвоить значения 250 и 330 соответственно.

Форма — это обычное окно. Поэтому ее размер можно изменить точно так же, как любого другого окна, т. е. захватом и перемещением (с помощью мыши) границы. По окончании перемещения границ автоматически изменяются значения свойств `Height` и `Width`. Они будут соответствовать установленному размеру формы.

По умолчанию положение окна на экране сразу после запуска программы соответствует положению формы во время разработки программы, которое определяется значением свойств `Top` (отступ от верхней границы экрана) и `Left` (отступ от левой границы экрана). Значения этих свойств также можно задать путем перемещения окна формы при помощи мыши.

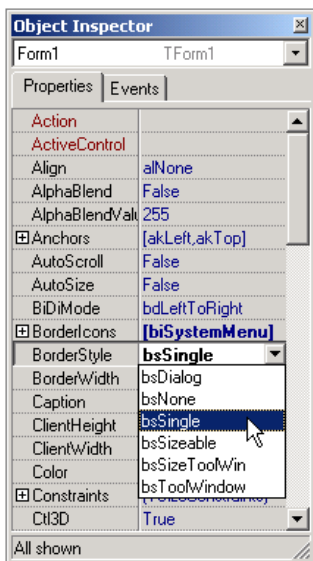


Рис. 1.12. Установка значения свойства путем выбора из списка

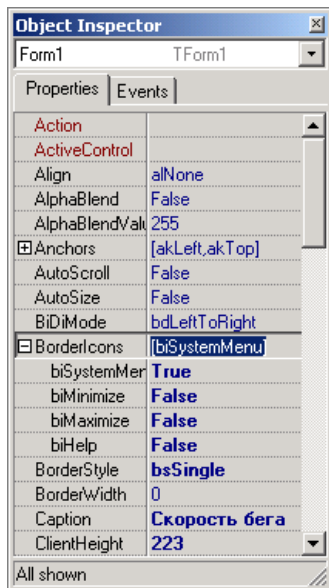


Рис. 1.13. Раскрытый список вложенных свойств сложного свойства `BorderIcons`

При выборе некоторых свойств, например `BorderStyle`, справа от текущего значения свойства появляется значок раскрывающегося списка. Очевидно, что значение таких свойств можно задать путем выбора из списка (рис. 1.12).

Некоторые свойства являются сложными, т. е. их значение определяется совокупностью значений других (уточняющих) свойств. Перед именами сложных свойств стоит значок "+", при щелчке на котором раскрывается список уточняющих свойств (рис. 1.13). Например, свойство `BorderIcons` определяет, какие кнопки управления окном будут доступны во время работы программы. Так, если свойству `biMaximize` присвоить значение `False`, то во время работы программы кнопки **Развернуть** в заголовке окна не будет.

Рядом со значениями некоторых свойств отображается командная кнопка с тремя точками. Это значит, что для задания значения свойства можно воспользоваться дополнительным диалоговым окном. Например, значение сложного свойства `Font` можно задать путем непосредственного ввода значений уточняющих свойств, а можно воспользоваться стандартным диалоговым окном выбора шрифта.

В табл. 1.2 перечислены свойства формы разрабатываемой программы, которые следует изменить. Остальные свойства оставлены без изменения и в таблице не приведены.

Таблица 1.2. Значения свойств формы

Свойство	Значение
<code>Caption</code>	Скорость
<code>Height</code>	250
<code>Width</code>	330
<code>Position</code>	<code>poDesktopCenter</code>
<code>Font.Name</code>	Tahoma
<code>Font.Size</code>	10
<code>BorderStyle</code>	<code>bsSingle</code>
<code>BorderIcons.biMinimize</code>	<code>False</code>
<code>BorderIcons.biMaximize</code>	<code>False</code>

В приведенной таблице в именах некоторых свойств есть точка. Это значит, что надо задать значение уточняющего свойства. После того как будут уста-

новлены значения свойств главной формы, она должна выглядеть так, как показано на рис. 1.14.

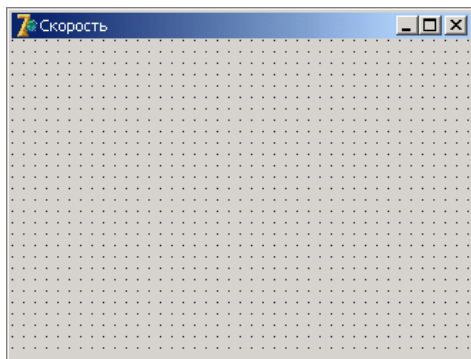


Рис. 1.14. Так выглядит форма после установки значений свойств

Компоненты

Программа вычисления скорости бега должна получить от пользователя исходные данные — длину дистанции и время, за которое спортсмен пробежал дистанцию. В подобных программах данные с клавиатуры, как правило, вводят в поля редактирования. Поэтому в форму надо добавить компонент `Edit` — поле редактирования.

Наиболее часто используемые компоненты находятся на вкладке **Standard** (рис. 1.15).

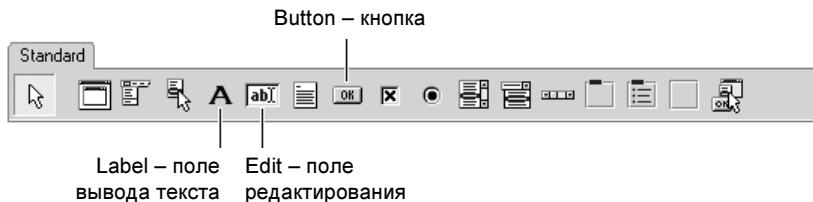


Рис. 1.15. На вкладке **Standard** находятся наиболее часто используемые компоненты

Для того чтобы добавить в форму компонент, необходимо в палитре компонентов выбрать этот компонент, щелкнув левой кнопкой мыши на его пикто-

грамме, далее установить курсор в ту точку формы, в которой должен быть левый верхний угол компонента, и еще раз щелкнуть левой кнопкой мыши. В результате в форме появляется компонент стандартного размера.

Размер компонента можно задать в процессе его добавления к форме. Для этого надо после выбора компонента из палитры поместить курсор мыши в ту точку формы, где должен находиться левый верхний угол компонента, нажать левую кнопку мыши и, удерживая ее нажатой, переместить курсор в точку, где должен находиться правый нижний угол компонента, затем отпустить кнопку мыши. На форме появится компонент нужного размера.

Каждому компоненту Delphi присваивает имя, которое состоит из названия компонента и его порядкового номера. Например, если к форме добавить два компонента `Edit`, то их имена будут `Edit1` и `Edit2`. Программист, путем изменения значения свойства `Name`, может изменить имя компонента. В простых программах имена компонентов, как правило, не изменяют.

На рис. 1.16 приведен вид формы после добавления двух компонентов `Edit` — полей редактирования, предназначенных для ввода исходных данных. Один из компонентов выделен. Свойства выделенного компонента отображаются в окне **Object Inspector**. Чтобы увидеть свойства другого компонента, надо щелкнуть левой кнопкой мыши на изображении нужного компонента. Можно также выбрать имя компонента в окне **Object TreeView** или из находящегося в верхней части окна **Object Inspector** раскрывающегося списка объектов.

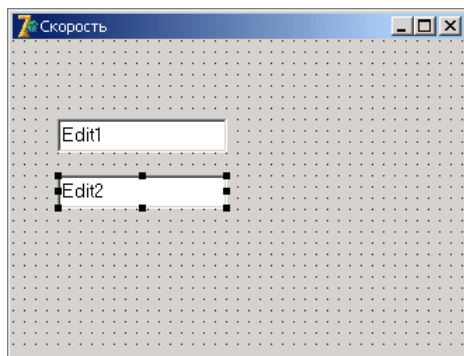


Рис. 1.16. Форма после добавления компонентов `Edit`

В табл. 1.3 перечислены основные свойства компонента `Edit` — поля редактирования.

Таблица 1.3. Свойства компонента Edit

Свойство	Описание
Name	Имя компонента. Используется для доступа к компоненту и его свойствам
Text	Текст, находящийся в поле ввода и редактирования
Left	Расстояние от левой границы компонента до левой границы формы
Top	Расстояние от верхней границы компонента до верхней границы формы
Height	Высота поля
Width	Ширина поля
Font	Шрифт, используемый для отображения вводимого текста
ParentFont	Признак наследования компонентом характеристик шрифта формы, на которой находится компонент. Если значение свойства равно True, то при изменении свойства Font формы автоматически меняется значение свойства Font компонента

Delphi позволяет изменить размер и положение компонента при помощи мыши.

Для того чтобы изменить положение компонента, необходимо установить курсор мыши на его изображение, нажать левую кнопку мыши и, удерживая ее нажатой, переместить контур компонента в нужную точку формы, затем отпустить кнопку мыши. Во время перемещения компонента (рис. 1.17) отображаются текущие значения координат левого верхнего угла компонента (значения свойств Left и Top).

Для того чтобы изменить размер компонента, необходимо его выделить, установить указатель мыши на один из маркеров, помечающих границу компонента, нажать левую кнопку мыши и, удерживая ее нажатой, изменить положение границы компонента. Затем отпустить кнопку мыши. Во время изменения размера компонента отображаются текущие значения свойств Height и Width (рис. 1.18).

Свойства компонента так же, как и свойства формы, можно изменить при помощи **Object Inspector**. Для того чтобы свойства требуемого компонента были выведены в окне **Object Inspector**, нужно выделить этот компонент (щелкнуть мышью на его изображении). Можно также выбрать компонент из