

**Никита Культин**

**Основы программирования**  
**Microsoft®**  
**Visual C++ 2010**

Санкт-Петербург

«БХВ-Петербург»

2010

УДК 681.3.068  
ББК 32.973.26-018.1  
К90

**Культин Н. Б.**

К90 Основы программирования в Microsoft® Visual C++ 2010. — СПб.: БХВ-Петербург, 2010. — 384 с.: ил. + CD-ROM — (Самоучитель)

ISBN 978-5-9775-0520-8

Книга является пособием для начинающих по программированию в Microsoft Visual C++ 2010. В ней в доступной форме изложены принципы визуального проектирования и событийного программирования, на примерах показана технология создания программ различного назначения. Приведено описание среды разработки и базовых компонентов. Рассмотрены вопросы программирования графики, разработки программ работы с базами данных Microsoft Access и Microsoft SQL Server Compact Edition. Уделено внимание отладке программ, созданию справочной системы, установке созданной программы на компьютер пользователя. В справочнике приведено описание базовых компонентов и наиболее часто используемых функций. Прилагаемый компакт-диск содержит проекты, рассмотренные в книге.

*Для начинающих программистов*

УДК 681.3.068  
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>Ольги Сергиенко</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 31.05.10.

Формат 70×100<sup>1/16</sup>. Печать офсетная. Усл. печ. л. 30,96.

Тираж 2000 экз. Заказ №

"БХВ-Петербург", 190005, Санкт-Петербург, Измайловский пр., 29.

Санитарно-эпидемиологическое заключение на продукцию № 77.99.60.953.Д.005770.05.09 от 26.05.2009 г. выдано Федеральной службой по надзору в сфере защиты прав потребителей и благополучия человека.

Отпечатано с готовых диапозитивов  
в ГУП "Типография "Наука"  
199034, Санкт-Петербург, 9 линия, 12

ISBN 978-5-9775-0520-8

© Культин Н. Б., 2010  
© Оформление, издательство "БХВ-Петербург", 2010

# Оглавление

Предисловие.....	1
<b>ЧАСТЬ I. MICROSOFT VISUAL C++ 2010.....</b>	<b>3</b>
<b>Глава 1. Среда разработки Microsoft Visual C++ 2010.....</b>	<b>5</b>
Установка.....	5
Первый взгляд.....	7
Справочная система.....	13
<b>Глава 2. Первый проект.....</b>	<b>15</b>
Начало работы над проектом.....	15
Форма.....	16
Компоненты.....	20
Событие.....	30
Функция обработки события.....	31
Структура проекта.....	36
Главный модуль.....	37
Модуль формы.....	38
Компиляция.....	44
Ошибки.....	46
Предупреждения.....	47
Запуск программы.....	47
Исключения.....	48
Обработка исключения.....	49
Внесение изменений.....	54
Завершение работы над проектом.....	58
Установка приложения на другой компьютер.....	59
<b>Глава 3. Базовые компоненты.....</b>	<b>61</b>
<i>Label</i> .....	61
<i>TextBox</i> .....	64

<i>Button</i> .....	69
<i>CheckBox</i> .....	73
<i>RadioButton</i> .....	77
<i>GroupBox</i> .....	82
<i>ComboBox</i> .....	86
<i>PictureBox</i> .....	91
<i>ListBox</i> .....	98
<i>ListView</i> .....	103
<i>ImageList</i> .....	107
<i>ToolTip</i> .....	109
<i>Panel</i> .....	110
<i>CheckedListBox</i> .....	112
<i>Timer</i> .....	114
<i>NumericUpDown</i> .....	117
<i>StatusStrip</i> .....	122
<i>NotifyIcon</i> .....	125
<i>ToolStrip</i> .....	130
<i>MenuStrip</i> .....	133
<i>OpenFileDialog</i> .....	136
<i>SaveFileDialog</i> .....	139

## **ЧАСТЬ II. ПРАКТИКУМ ПРОГРАММИРОВАНИЯ ..... 149**

### **Глава 4. Графика ..... 151**

Графическая поверхность .....	153
Карандаши и кисти .....	154
Карандаш .....	154
Кисть .....	157
Графические примитивы .....	162
Линия .....	164
Ломаная линия .....	169
Прямоугольник .....	169
Точка .....	170
Многоугольник .....	171
Эллипс и окружность .....	172
Дуга .....	173
Сектор .....	173
Текст .....	179
Битовые образы .....	184
Анимация .....	187

### **Глава 5. Базы данных ..... 197**

База данных и СУБД .....	197
Локальные и удаленные базы данных .....	197
Структура базы данных .....	198

Компоненты доступа к данным .....	199
Создание базы данных .....	199
База данных Microsoft Access .....	200
Доступ к данным .....	201
Отображение данных .....	213
Выбор информации из базы данных .....	218
Работа с базой данных в режиме формы .....	221
Сервер баз данных Microsoft SQL Server Compact Edition .....	227
Среда Microsoft SQL Server Management Studio .....	228
Создание базы данных .....	228
База данных "Контакты" .....	232
Развертывание приложения работы с базой данных Microsoft SQL Server Compact Edition .....	243
<b>Глава 6. Консольное приложение .....</b>	<b>245</b>
Создание консольного приложения .....	249
Запуск консольного приложения .....	254
<b>Глава 7. Отладка программы .....</b>	<b>255</b>
Классификация ошибок .....	255
Предотвращение и обработка ошибок .....	257
Отладчик .....	260
Трассировка программы .....	260
Точки останова программы .....	261
Наблюдение значений переменных .....	262
<b>Глава 8. Справочная информация .....</b>	<b>265</b>
Справочная система HTML Help .....	265
Подготовка справочной информации .....	266
Основы HTML .....	267
Microsoft HTML Help Workshop .....	269
Файл проекта .....	269
Оглавление .....	272
Идентификаторы разделов .....	275
Компиляция .....	276
Отображение справочной информации .....	277
<b>Глава 9. Создание установщика .....</b>	<b>281</b>
Новый проект .....	281
Общая информация .....	283
Файлы .....	284
Команда запуска программы .....	286
Интерфейс .....	288
Диалоги .....	288

Построение .....	291
Установка .....	291
<b>Глава 10. Примеры программ .....</b>	<b>293</b>
Экзаменатор .....	293
Требования к программе .....	294
Файл теста .....	294
Форма .....	297
Доступ к файлу теста .....	298
Текст программы .....	301
Запуск программы .....	310
Сапер .....	312
Правила и представление данных .....	312
Форма .....	314
Игровое поле .....	316
Начало игры .....	316
Игра .....	319
Справочная информация .....	323
Информация о программе .....	324
Текст программы .....	327
<b>Глава 11. Краткий справочник .....</b>	<b>337</b>
Форма .....	337
Компоненты .....	338
<i>Button</i> .....	338
<i>ComboBox</i> .....	340
<i>ContextMenuStrip</i> .....	340
<i>CheckBox</i> .....	341
<i>CheckedListBox</i> .....	342
<i>GroupBox</i> .....	343
<i>ImageList</i> .....	344
<i>Label</i> .....	344
<i>ListBox</i> .....	345
<i>MenuStrip</i> .....	346
<i>NotifyIcon</i> .....	347
<i>NumericUpDown</i> .....	347
<i>OpenFileDialog</i> .....	348
<i>Panel</i> .....	349
<i>PictureBox</i> .....	349
<i>RadioButton</i> .....	350
<i>ProgressBar</i> .....	351
<i>SaveFileDialog</i> .....	352
<i>TextBox</i> .....	353
<i>ToolTip</i> .....	354
<i>Timer</i> .....	355

---

Графика .....	355
Графические примитивы.....	355
Карандаш.....	357
Кисть.....	358
Типы данных.....	360
Целый тип .....	360
Вещественный тип .....	360
Символьный и строковый типы .....	360
Функции .....	361
Функции преобразования.....	361
Функции манипулирования строками .....	362
Функции манипулирования датами и временем .....	364
Функции манипулирования каталогами и файлами .....	365
Математические функции.....	367
События.....	368
Исключения .....	369
<b>Приложение. Описание компакт-диска .....</b>	<b>371</b>
<b>Предметный указатель .....</b>	<b>373</b>





# Предисловие

Microsoft Visual C++ — среда программирования на языке C++ является одним из популярнейших инструментов разработки прикладных программ различного назначения. В нее на основе единого интерфейса интегрированы удобный конструктор форм, специализированный редактор кода, высокоскоростной оптимизирующий компилятор, отладчик и другие полезные инструменты.

Среда Microsoft Visual C++ ориентирована на так называемую быструю разработку, в основе которой лежит технология визуального проектирования и событийного программирования, суть которой заключается в том, что среда разработки берет на себя большую часть рутинных операций, оставляя программисту работу по созданию диалоговых окон (визуальное проектирование) и функций обработки событий (событийное программирование).

Изначально среда Microsoft Visual C++ была ориентирована на разработку Win32-приложений. С появлением технологии .NET и платформы Microsoft .NET Framework for Windows, обеспечивающей поддержку этой технологии в Microsoft Windows, в Visual C++ были интегрированы возможности, обеспечивающие разработку .NET-приложений.

Технология Microsoft .NET основана на идее универсального программного кода, который может быть выполнен любым компьютером, вне зависимости от используемой операционной системы. Универсальность программного кода обеспечивается за счет предварительной (выполняемой на этапе разработки) компиляции исходной программы в универсальный промежуточный код (CIL-код, Common Intermediate Language), который во время запуска (загрузки) программы транслируется в выполняемый. Преобразование промежуточного кода в выполняемый осуществляет JIT-компилятор (от Just In Time — в тот же момент, "на лету"), являющийся элементом виртуальной выполняющей системы (Virtual Execution System, VES). Выполнение .NET-приложений в операционной системе Microsoft Windows обеспечивает Common Language Runtime (CLR, общезыконовая исполняющая среда) — компонент Microsoft .NET Framework.

Усилия Microsoft по развитию и продвижению технологии .NET привели к тому, что в общем объеме вновь создаваемого программного обеспечения различного назначения увеличивается доля .NET-приложений, программ, ориентированных на платформу .NET. Это объясняется, прежде всего, возможностями, которые предоставляет платформа прикладным программам, и тем, что технология .NET поддерживается новейшими операционными системами — начиная с Microsoft Windows Vista, платформа .NET является неотъемлемой частью операционной системы.

Чтобы понять, что такое .NET, какие возможности она предоставляет программисту, необходимо опробовать ее в деле. Для этого нужно изучить среду, освоить технологию разработки, понять назначение и возможности компонентов, их свойства и методы.

Книга, которую вы держите в руках, посвящена практике программирования в Microsoft Visual C++ 2010, разработке приложений Windows Forms (.NET). В ней, в объеме, необходимом для начинающего программиста, приведено описание среды разработки и базовых компонентов, раскрыта суть технологии визуального проектирования и событийного программирования, показан процесс создания программ различного назначения: от простейших до программ работы с файлами, графикой и базами данных.

Состоит книга из двух частей.

Первая часть книги знакомит читателя с Microsoft Visual C++ 2010: содержит краткое описание среды разработки, демонстрирует процесс создания программы, назначение базовых компонентов.

Вторая часть книги посвящена практике. В ней рассмотрены задачи программирования графики, баз данных, создания справочной системы и установщика. Уделено внимание разработке консольных приложений. В справочнике приведено описание базовых компонентов и наиболее часто используемых функций.

Цель этой книги — научить читателя программировать, создавать программы различного назначения: от простых однооконных приложений до программ работы с базами данных. Следует обратить внимание на то, что хотя книга ориентирована на читателя, обладающего начальными знаниями и опытом в программировании, она вполне доступна и начинающим.

Научиться программировать можно только программируя. Поэтому, чтобы получить максимальную пользу от книги, вы должны работать с ней активно. Изучайте листинги (тексты программ), старайтесь понять, что и как делают программы, как они работают. Не бойтесь экспериментировать — совершенствуйте программы, вносите в них изменения. Чем больше вы сделаете самостоятельно, тем большему научитесь!

**Visual**



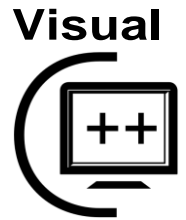
# **ЧАСТЬ I**

## **Microsoft Visual C++ 2010**

В этой части книги приведено краткое описание среды разработки Microsoft Visual C++ 2010; на примере программы "Конвертер" показан процесс разработки приложения Windows Forms; дано описание и приведены примеры использования базовых компонентов.

<b>Глава 1.</b>	Среда разработки Microsoft Visual C++ 2010
<b>Глава 2.</b>	Первый проект
<b>Глава 3.</b>	Базовые компоненты





# ГЛАВА 1

## Среда разработки Microsoft Visual C++ 2010

### Установка

Среда разработки Microsoft Visual Studio 2010, в которую входит Microsoft Visual C++, доступна в трех вариантах: Professional, Premium и Ultimate. Каждый комплект представляет собой набор инструментов, обеспечивающих разработку высокоэффективных приложений для Win32- и .NET-платформ, Web-приложений. Различие вариантов состоит в составе компонентов — чем выше уровень пакета (от Professional к Ultimate), тем больше возможностей и инструментов поддержки процесса разработки он предоставляет программисту. Так, например, в Ultimate включены инструменты UML-моделирования и анализа, в версии Premium есть только возможность просмотра UML-диаграмм, а в версии Professional работа с UML-диаграммами не поддерживается.

Microsoft Visual Studio 2010 может работать в среде операционных систем Microsoft:

- Windows XP (x86) Service Pack 3 (кроме Starter Edition);
- Windows XP (x64) Service Pack 2 (кроме Starter Edition);
- Windows Vista (x86, x64) Service Pack 1 (кроме Starter Edition);
- Windows 7 (x86, x64);
- Windows Server 2003 (x86, x64) Service Pack 2;
- Windows Server 2003 R2 (x86, x64);
- Windows Server 2008 (x86, x64) Service Pack 2;
- Windows Server 2008 R2 (x64).

Особых требований, по современным меркам, к ресурсам компьютера Microsoft Visual Studio не предъявляет: процессор должен быть с частотой не

ниже 1,6 ГГц, 1 Гбайт оперативной памяти (2 Гбайт для x64-процессора), порядка 7 Гбайт свободного места на жестком диске (для установки Visual C++ достаточно чуть больше 3 Гбайт).

Установка выполняется с DVD-диска, на котором помимо Visual Studio находятся компоненты, необходимые для установки и работы среды.

Процесс установки активизируется автоматически, после того как установочный диск будет помещен в дисковод. Если автоматический запуск программ запрещен, то надо вручную запустить установщик (файл setup.exe), который находится в корне установочного диска.

Установщик проверяет версию Windows, затем — Microsoft .NET Framework. Если версия операционной системы не соответствует необходимым требованиям, процесс установки прерывается. В случае если версия .NET Framework ниже 4, то на компьютер устанавливаются компоненты Microsoft .NET Framework 4. После этого непосредственно начинается установка Microsoft Visual Studio.

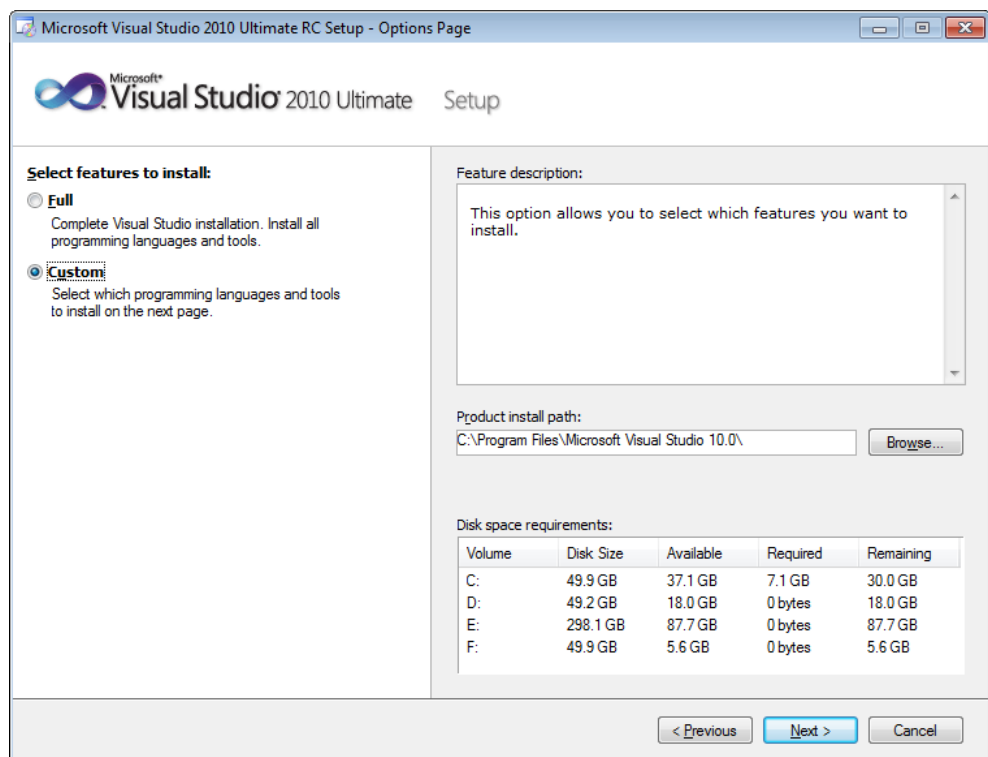


Рис. 1.1. Установка Visual C++: выбор варианта установки

Сначала на экране появляется окно лицензионного соглашения, затем установщик предлагает выбрать вариант установки: **Full** (полный) или **Custom** (частичный). Выбрав вариант установки **Custom** (рис. 1.1), в следующем окне (рис. 1.2) программист может выбрать компоненты Visual Studio, которые надо установить (точнее, отказаться от установки ненужных).

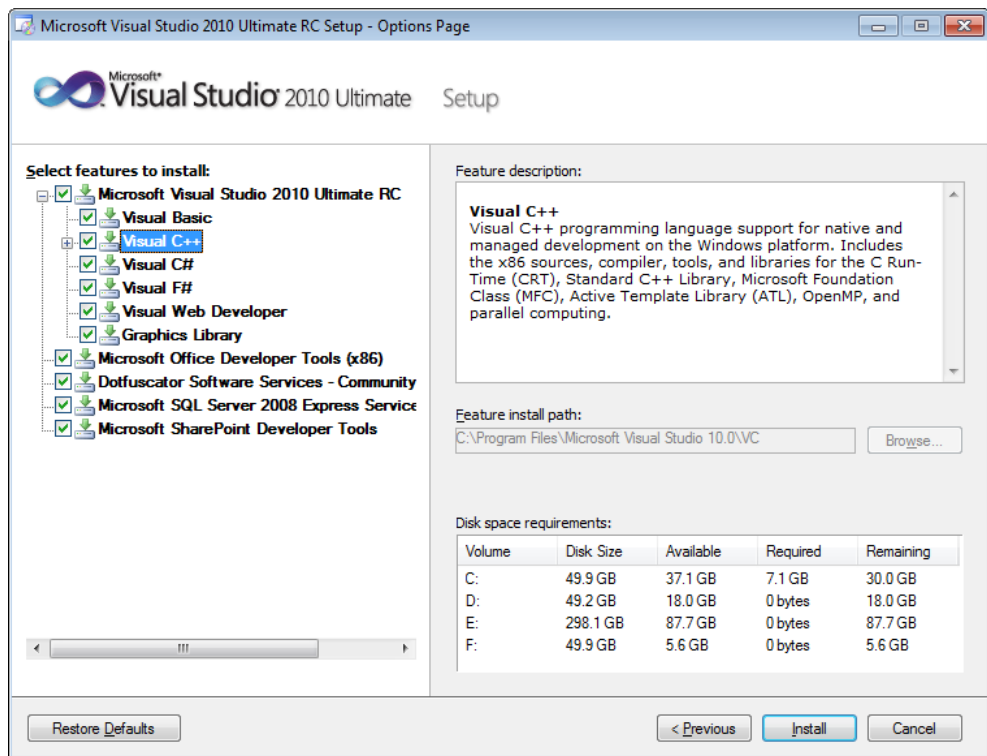
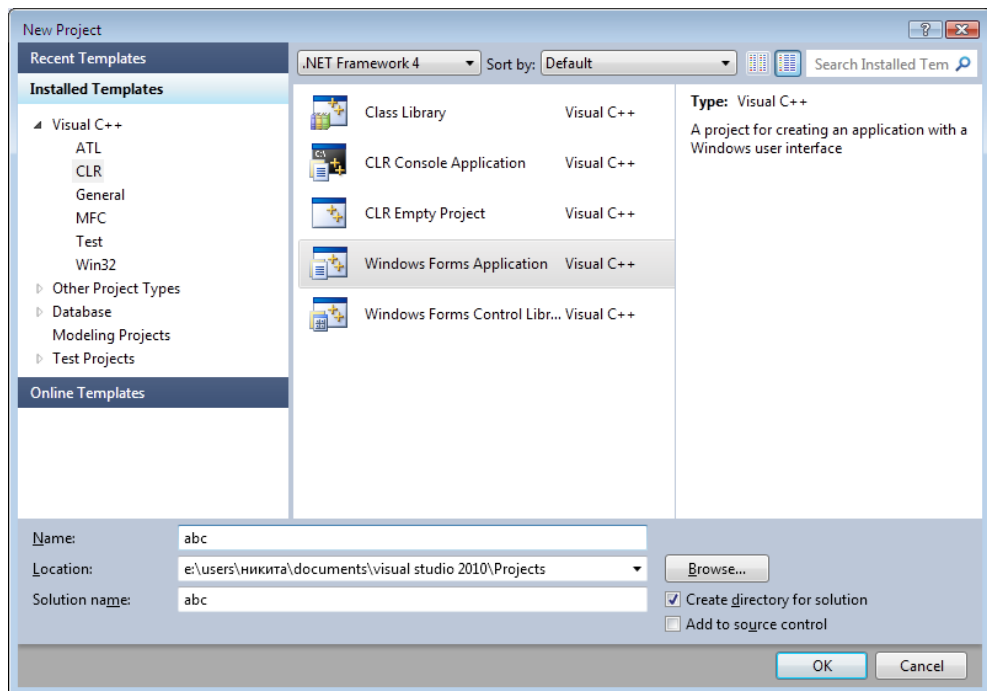


Рис. 1.2. Установка Visual C++: выбор устанавливаемых компонентов

Следует обратить внимание на то, что во время инсталляции Visual Studio справочная система на компьютер не устанавливается. Установка справочной системы выполняется из среды разработки (о том, как это сделать, см. в разд. "Справочная система" далее в этой главе).

## Первый взгляд

Чтобы начать работу в Microsoft Visual C++, надо запустить Microsoft Studio — сделать щелчок на кнопке **Пуск** и в меню **Все программы** ▶ **Microsoft Visual Studio 2010** выбрать **Microsoft Visual Studio 2010**.



**Рис. 1.3.** В окне **New Project** надо выбрать язык разработки, тип приложения и задать имя проекта

Для того чтобы приступить к созданию программы в Visual C++ или, как принято говорить, начать работу над *проектом*, надо:

1. В меню **File** окна Microsoft Visual Studio выбрать команду **New ► Project**.
2. В открывшемся окне **New Project** (рис. 1.3) раскрыть список **Visual C++** и выбрать тип приложения — **CLR**.
3. Выбрать вид приложения — **Windows Forms Application Visual C++**.
4. В поле **Name** ввести имя проекта и нажать кнопку **OK**.

В результате описанных действий будет создан *проект* Visual Studio — совокупность файлов, необходимых для создания программы. Проект создается в папке проектов Visual Studio 2010 (по умолчанию это C:\Users\User\Documents\Visual Studio 2010\Projects, где *User* — имя пользователя в системе), в новой папке (обратите внимание на установленный флажок **Create directory for solution** — создать каталог для решения). Имя папки проекта определяет содержимое поля **Name** (имя проекта).

Окно среды Microsoft Visual Studio 2010 в начале работы над новым проектом приведено на рис. 1.4. В заголовке окна отображается имя проекта, над которым в данный момент идет работа.



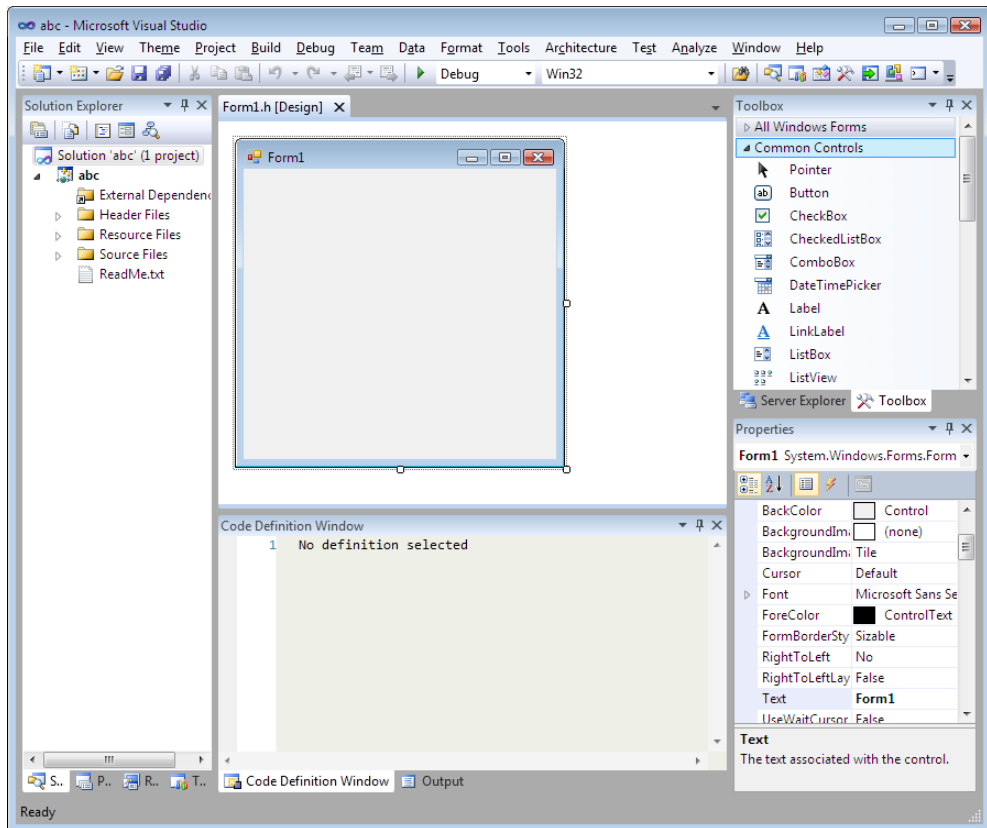


Рис. 1.4. Окно среды разработки Visual C++ в начале работы над новым проектом Windows Forms

В верхней части окна находится строка меню и область отображения панелей инструментов. По умолчанию в области отображения панелей инструментов выводится панель **Standard** (рис. 1.5). Чтобы сделать доступными другие панели инструментов, надо выбрать команду **View** ► **Toolbars** и в раскрывшемся списке сделать щелчок на имени нужной панели.

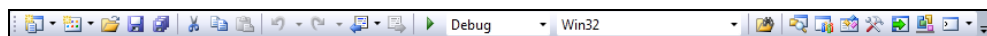


Рис. 1.5. Панель **Standard**

Центральную часть окна Visual C++ занимает окно конструктора (Designer) формы (рис. 1.6). В нем находится *форма* — заготовка окна приложения (окно программы во время его разработки принято называть формой). В окне дизайнера формы в графической форме отображаются инструкции програм-

мы (обратите внимание, в заголовке окна указано имя h-файла), обеспечивающие создание окна. Чтобы их увидеть (рис. 1.7), выберите в меню **View** команду **Code**.

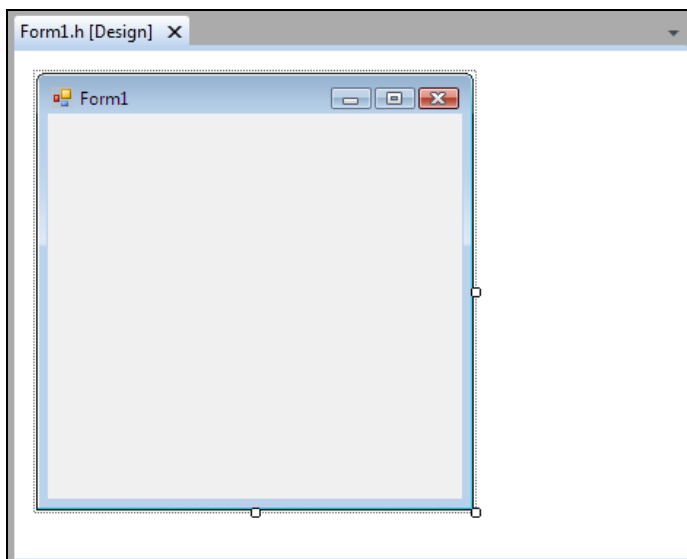


Рис. 1.6. Окно конструктора формы

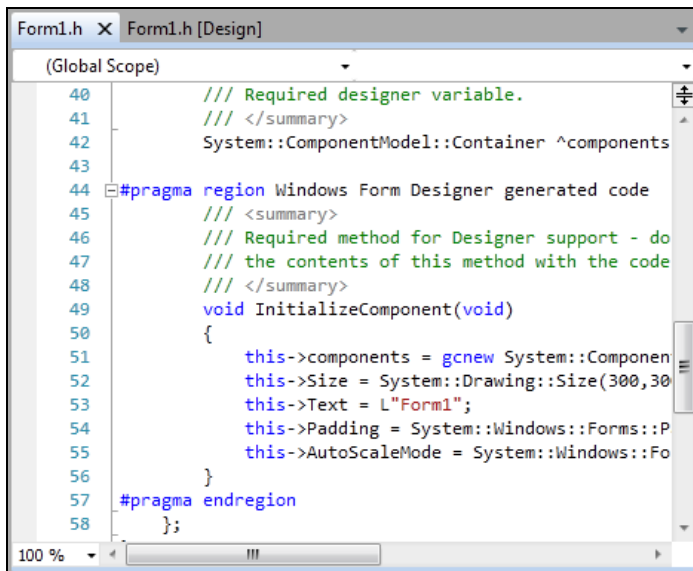


Рис. 1.7. Инструкции, обеспечивающие создание формы, находятся в секции Windows Form Designer generated code

В правой части главного окна Visual Studio обычно отображаются окна **Toolbox** и **Properties**.

В окне **Toolbox** (рис. 1.8) находятся *компоненты*, которые можно поместить (перетащить) на форму. Компонент — это объект, реализующий некоторую функциональность. Например, в группе **Common Controls** находятся компоненты, реализующие пользовательский интерфейс (Label — область отображения текста; TextBox — поле ввода/редактирования текста; Button — командная кнопка), а в группе **Data** — компоненты доступа к базам данных.

В окне **Properties** (рис. 1.9) отображаются *свойства* выбранного (выделенного) в данный момент объекта — компонента или, если ни один из элементов не выделен, самой формы. Обратите внимание, имя и тип объекта отображаются в верхней части окна **Properties**.

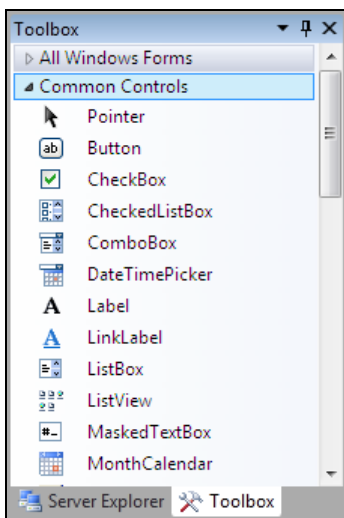


Рис. 1.8. Окно Toolbox

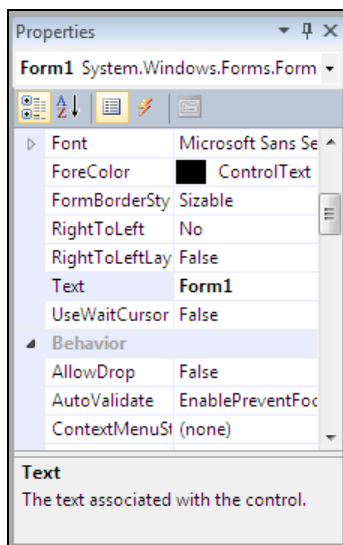


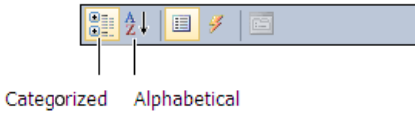
Рис. 1.9. Окно Properties

Окно **Properties** используется для редактирования значений свойств *объектов* и, как следствие, изменения их вида и поведения. Например, чтобы изменить текст в заголовке формы, надо изменить значение свойства `Text`.

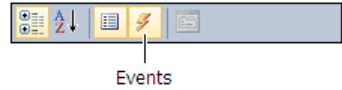
При отображении свойства могут быть объединены в группы по функциональному признаку (**Categorized**) или упорядочены по алфавиту (**Alphabetical**). Чтобы изменить способ отображения, надо сделать щелчок на соответствующей кнопке, находящейся в верхней части окна **Properties** (рис. 1.10).

Сделав в окне **Properties** щелчок на кнопке **Events** (рис. 1.11), можно увидеть *события* (рис. 1.12), которые способен воспринимать выбранный объект

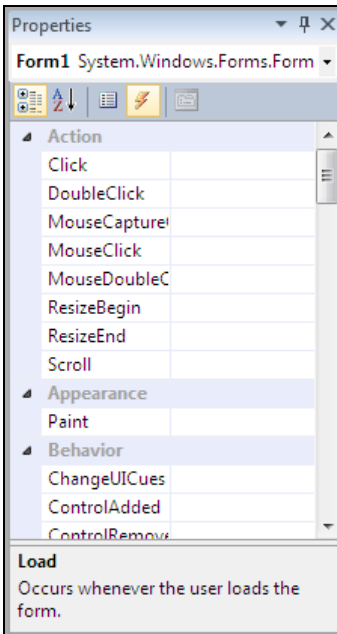
(компонент или форма). Событие — это то, что происходит во время работы программы. Например, командная кнопка может реагировать на щелчок кнопкой мыши — событие `Click`.



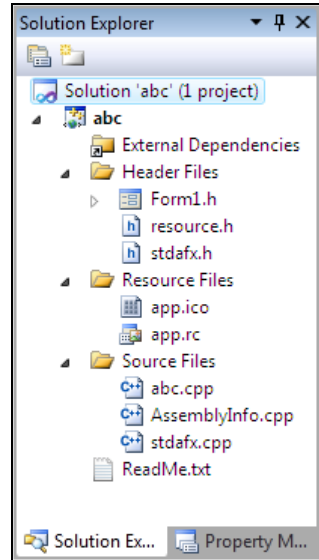
**Рис. 1.10.** Кнопки управления способом группировки свойств



**Рис. 1.11.** Чтобы увидеть события, которые способен воспринимать объект, нажмите кнопку **Events**



**Рис. 1.12.** События, которые может воспринимать объект (в данном случае — форма)



**Рис. 1.13.** В окне **Solution Explorer** отображается структура проекта

В окне **Solution Explorer** (рис. 1.13) отображается структура проекта. В нем перечислены файлы, образующие проект. Функция `main` находится в главном файле проекта (в рассматриваемом примере это `abc.cpp`), описание класса формы — в h-файле (в рассматриваемом примере — в файле `Form1.h`). Здесь же отображаются имена файлов ресурсов (`app.rc`) и значка (`app.ico`) проекта.

Окно **Solution Explorer** удобно использовать для быстрого доступа к нужному элементу проекта.

## Справочная система

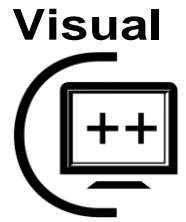
Сразу после установки Visual Studio справочная информация недоступна. Чтобы получить возможность доступа к ней, надо выполнить настройку справочной системы.

Visual Studio позволяет программисту выбрать источник справочной информации — локальная справочная информация (находящаяся на компьютере разработчика) или online (предоставляемая Microsoft через Интернет).

Чтобы справочная информация стала доступной, надо:

1. В меню **Help** выбрать команду **Manage Help Settings**.
2. В окне **Manage Help Settings** сделать щелчок на ссылке **Chose online or local help**.
3. В окне **Settings** выбрать вариант **I want to use local help** (использовать локальную справочную информацию) и нажать кнопку **OK**.
4. В окне **Manage Help Settings** сделать щелчок на ссылке **Find content on disk**.
5. В появившемся окне **Find content on disk** сделать щелчок на кнопке **Browse**, открыть находящийся на установочном диске, в папке ProductDocumentation, файл контента справочной информации (Help-ContentSetup.msha) и выбрать разделы справочной информации, которые надо скопировать с установочного диска на компьютер.





## ГЛАВА 2

# Первый проект

Процесс разработки программы в Microsoft Visual C++ рассмотрим на примере — создадим *приложение* (программы, предназначенные для решения прикладных задач, принято называть приложениями), позволяющее посчитать доход по вкладу (рис. 2.1).

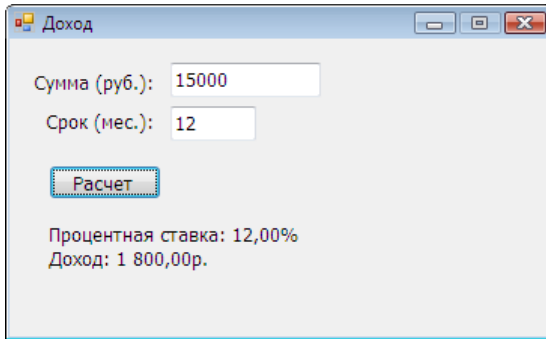


Рис. 2.1. Окно программы "Доход"

## Начало работы над проектом

Чтобы начать работу над новым приложением (проектом), надо:

1. В меню **File** выбрать команду **New ▶ Project**.
2. В открывшемся окне **New Project** в списке **Visual C++** выбрать сначала тип приложения — **CLR**, а затем вид приложения — **Windows Forms Application**.
3. В поле **Name** ввести имя проекта — **profit** и нажать кнопку **OK**.

## Форма

Работа над приложением начинается с создания стартовой *формы* — главного окна программы.

Сначала нужно установить требуемые значения свойств формы, затем — поместить на форму необходимые *компоненты* (поля ввода информации, командные кнопки, поля отображения текста и др.) и выполнить.

Настройка формы (а также компонентов) осуществляется путем изменения значений *свойств*. Свойства *объекта* (формы, компонента) определяют его вид и поведение. Например, свойство `Text` определяет текст заголовка окна, а свойство `StartPosition` — положение окна в момент появления его на экране.

Основные свойства формы (объекта `WinForm`) приведены в табл. 2.1.

**Таблица 2.1.** Свойства формы (объекта `WinForm`)

Свойство	Описание
<code>Name</code>	Имя формы
<code>Text</code>	Текст в заголовке
<code>Size</code>	Размер формы. Уточняющее свойство <code>Width</code> определяет ширину, свойство <code>Height</code> — высоту
<code>StartPosition</code>	Положение формы в момент первого появления на экране. Форма может находиться в центре экрана ( <code>CenterScreen</code> ), в центре родительского окна ( <code>CenterParent</code> ). Если значение свойства равно <code>Manual</code> , то положение формы определяется значением свойства <code>Location</code>
<code>Location</code>	Положение формы на экране. Расстояние от верхней границы формы до верхней границы экрана задает уточняющее свойство <code>Y</code> , расстояние от левой границы формы до левой границы экрана — уточняющее свойство <code>X</code>
<code>FormBorderStyle</code>	Тип формы (границы). Форма может представлять собой обычное окно ( <code>Sizable</code> ), окно фиксированного размера ( <code>FixedSingle</code> , <code>Fixed3D</code> ), диалог ( <code>FixedDialog</code> ) или окно без кнопок <b>Свернуть</b> и <b>Развернуть</b> ( <code>SizeableToolWindow</code> , <code>FixedToolWindow</code> ). Если свойству присвоить значение <code>None</code> , у окна не будет заголовка и границы
<code>ControlBox</code>	Управляет отображением системного меню и кнопок управления окном. Если значение свойства равно <code>False</code> , то в заголовке окна кнопка системного меню, а также кнопки <b>Свернуть</b> , <b>Развернуть</b> , <b>Закреть</b> не отображаются
<code>MaximizeBox</code>	Кнопка <b>Развернуть</b> . Если значение свойства равно <code>False</code> , то находящаяся в заголовке окна кнопка <b>Развернуть</b> недоступна
<code>MinimizeBox</code>	Кнопка <b>Свернуть</b> . Если значение свойства равно <code>False</code> , то находящаяся в заголовке окна кнопка <b>Свернуть</b> недоступна



Таблица 2.1 (окончание)

Свойство	Описание
Icon	Значок в заголовке окна
Font	Шрифт, используемый по умолчанию компонентами, находящимися на поверхности формы. Изменение значения свойства приводит к автоматическому изменению значения свойства Font всех компонентов формы (при условии, что значение свойства компонента не было задано явно)
ForeColor	Цвет, наследуемый компонентами формы и используемый ими для отображения текста. Изменение значения свойства приводит к автоматическому изменению соответствующего свойства всех компонентов формы (при условии, что значение свойства Font компонента не было задано явно)
BackColor	Цвет фона. Можно задать явно (выбрать на вкладке <b>Custom</b> или <b>Web</b> ) или указать элемент цветовой схемы (выбрать на вкладке <b>System</b> )
Opacity	Степень прозрачности формы. Форма может быть непрозрачной (100%) или прозрачной. Если значение свойства меньше 100%, то сквозь форму видна поверхность, на которой она отображается

Для изменения значений свойств объектов используется окно **Properties**. В левой колонке окна перечислены свойства объекта, *выбранного* в данный момент, в правой — указаны значения свойств. Имя выбранного объекта отображается в верхней части окна **Properties**.

Чтобы в заголовке окна отображалось название программы, надо изменить значение свойства `Text`. Для этого следует щелкнуть левой кнопкой мыши в поле значение свойства `Text` (в поле появится курсор), ввести в поле редактирования текст **Доход** и нажать клавишу <Enter> (рис. 2.2).

При выборе некоторых свойств, например `FormBorderStyle`, справа от текущего значения свойства появляется значок раскрывающегося списка. Очевидно, что значение таких свойств можно задать путем выбора из списка (рис. 2.3).

Некоторые свойства являются сложными. Они представляют собой совокупность других (уточняющих) свойств. Например, свойство `Size`, определяющее размер формы, представляет собой совокупность свойств `Width` и `Height`. Перед именами сложных свойств стоит значок **▶**, в результате щелчка на котором раскрывается список уточняющих свойств (рис. 2.4). Значение уточняющего свойства можно задать (изменить) обычным образом — ввести нужное значение в поле редактирования.

Размер формы можно изменить и с помощью мыши, точно так же, как и любого окна, т. е. путем перемещения границы. По окончании перемещения

границы значения свойств `Width` и `Height` будут соответствовать установленному размеру формы.

В результате выбора некоторых свойств, например `Font`, в поле значения свойства отображается кнопка, на которой изображены три точки. Это значит, что задать значение свойства можно в дополнительном диалоговом окне, которое появится в результате щелчка на этой кнопке. Например, значение свойства `Font` можно задать путем ввода значений уточняющих свойств (`Name`, `Size`, `Style` и др.), а можно воспользоваться стандартным диалоговым окном **Шрифт**, которое появится в результате щелчка на кнопке с тремя точками (рис. 2.5).

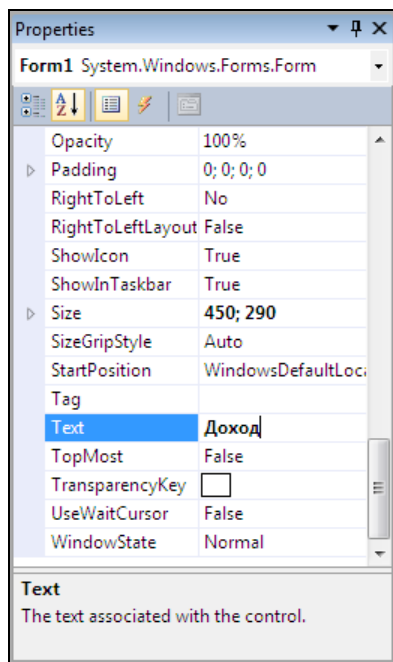


Рис. 2.2. Изменение значения свойства `Text` путем ввода нового значения

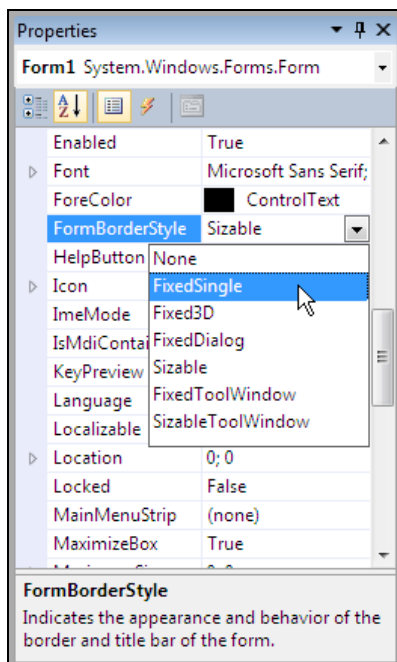


Рис. 2.3. Установка значения свойства путем выбора из списка

В табл. 2.2 приведены значения свойств формы программы "Доход". Значения остальных свойств формы оставлены без изменения и поэтому в таблице не представлены. Обратите внимание, в именах некоторых свойств есть точка. Это значит, что это значение уточняющего свойства.

После того как будут установлены значения свойств формы, она должна выглядеть так, как показано на рис. 2.6. Теперь на форму надо добавить *компоненты*.

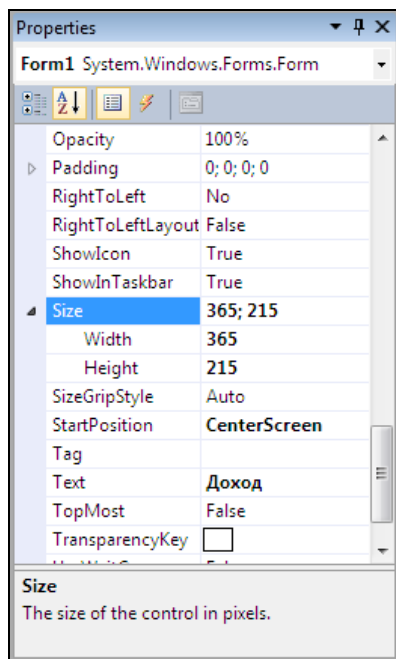


Рис. 2.4. Изменение значения уточняющего свойства

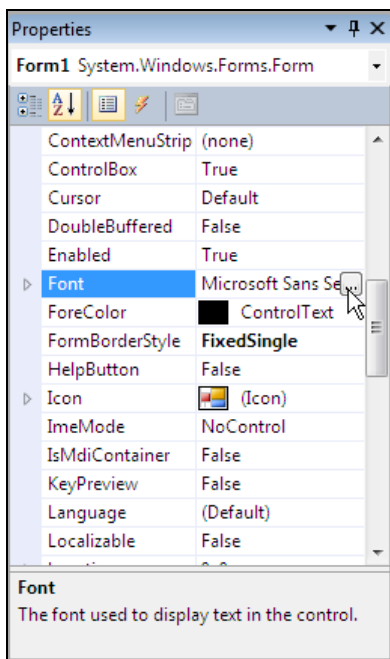


Рис. 2.5. Чтобы задать свойства шрифта, щелкните на кнопке с тремя точками

Таблица 2.2. Значения свойств стартовой формы

Свойство	Значение	Комментарий
Text	Доход	
Size.Width	365	
Size.Height	215	
FormBorderStyle	FixedSingle	Тонкая граница формы. Во время работы программы пользователь не сможет изменить размер окна путем захвата и перемещения его границы
StartPosition	CenterScreen	Окно программы появится в центре экрана
MaximizeBox	False	В заголовке окна не отображать кнопку <b>Развернуть</b>
Font.Name	Tahoma	
Font.Size	10	

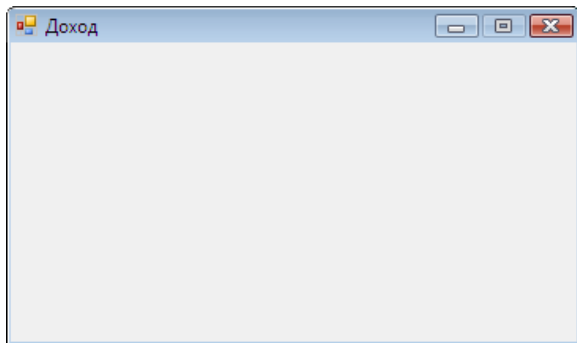


Рис. 2.6. Форма после изменения значений ее свойств

## Компоненты

Поля ввода/редактирования, поля отображения текста, командные кнопки, списки, переключатели и другие элементы, обеспечивающие взаимодействие пользователя с программой, называют *компонентами пользовательского интерфейса*. Они находятся в окне **Toolbox** на вкладке **Common Control**.

Программа "Доход" должна получить от пользователя исходные данные — сумму и срок вклада. Для ввода данных с клавиатуры предназначен компонент `TextBox`. Таким образом, на форму разрабатываемого приложения нужно поместить два компонента `TextBox`.

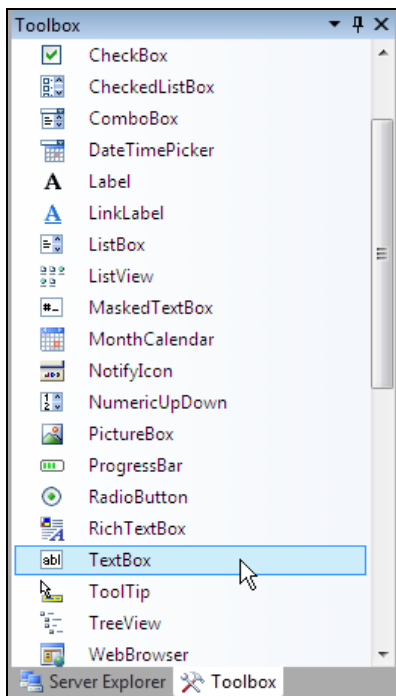
Чтобы на форму добавить компонент `TextBox`, надо:

1. В палитре компонентов (окно **Toolbox**) раскрыть вкладку **Common Control**. Сделать щелчок на значке компонента `TextBox` (рис. 2.7).
2. Установить указатель мыши в ту точку формы, в которой должен быть левый верхний угол компонента, и сделать щелчок левой кнопкой мыши.

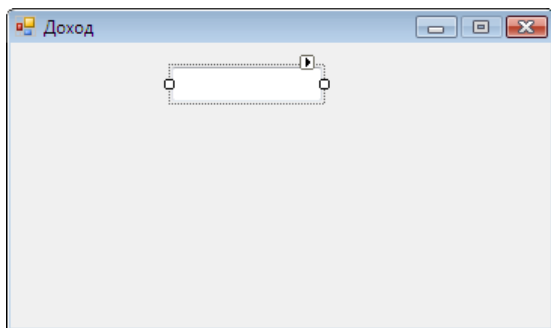
В результате на форме появляется поле ввода/редактирования — компонент `TextBox` (рис. 2.8).

Каждому добавленному компоненту среда разработки присваивает имя, которое состоит из названия компонента и его порядкового номера. Например, первый добавленный на форму компонент `TextBox` получает имя `textBox1`, второй — `textBox2`. Программист путем изменения значения свойства `Name` может поменять имя компонента. Однако в простых программах имена компонентов, как правило, не меняют.

Основные свойства компонента `TextBox` приведены в табл. 2.3.



**Рис. 2.7.** Выбор компонента в палитре (компонент `TextBox` — поле редактирования)



**Рис. 2.8.** Результат добавления на форму компонента `TextBox`

**Таблица 2.3.** Свойства компонента `TextBox`

Свойство	Описание
Name	Имя компонента. Используется для доступа к компоненту и его свойствам
Text	Текст, который находится в поле редактирования
Location	Положение компонента на поверхности формы
Size	Размер компонента
Font	Шрифт, используемый для отображения текста в поле компонента
ForeColor	Цвет текста, находящегося в поле компонента
BackColor	Цвет фона поля компонента
BorderStyle	Вид рамки (границы) компонента. Граница компонента может быть обычной ( <code>Fixed3D</code> ), тонкой ( <code>FixedSingle</code> ) или отсутствовать ( <code>None</code> )

Таблица 2.3 (окончание)

Свойство	Описание
TextAlign	Способ выравнивания текста в поле компонента. Текст в поле компонента может быть прижат к левой границе компонента (Left), правой (Right) или находиться по центру (Center)
MaxLength	Максимальное количество символов, которое можно ввести в поле компонента
Multiline	Разрешает (True) или запрещает (False) ввод нескольких строк текста
ReadOnly	Разрешает (True) или запрещает (False) редактирование отображаемого текста
Lines	Массив строк, элементы которого содержат текст, находящийся в поле редактирования, если компонент находится в режиме MultiLine. Доступ к строке осуществляется по номеру. Строки нумеруются с нуля
ScrollBars	Задаёт отображаемые полосы прокрутки: Horizontal — горизонтальная; Vertical — вертикальная; Both — горизонтальная и вертикальная; None — не отображать

На рис. 2.9 приведен вид формы программы "Доход" после добавления двух полей ввода-редактирования. Один из компонентов *выбран* — выделен рамкой. Свойства именно этого (выбранного) компонента отображаются в **Properties**. Чтобы увидеть и, если надо, изменить свойства другого компонента, нужно этот компонент выбрать — щелкнуть левой кнопкой мыши на изображении компонента в форме или выбрать его имя в раскрывающемся списке, который находится в верхней части окна **Properties** (рис. 2.10).

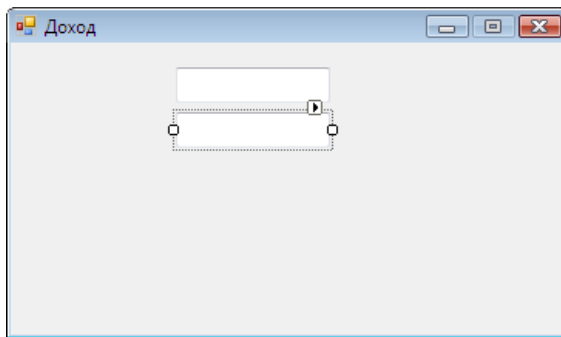


Рис. 2.9. Форма с двумя компонентами TextBox

Значения свойств компонента, определяющих размер и положение компонента на поверхности формы, можно изменить с помощью мыши.