

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«ЧЕРЕПОВЕЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт информационных технологий

Кафедра математического и программного обеспечения ЭВМ

ОСНОВЫ РАБОТЫ В Microsoft Visual Studio

Учебно-методическое пособие

Направления подготовки:

01.04.02 Прикладная математика и информатика;
09.03.01 Информатика и вычислительная техника;
09.03.02 Информационные системы и технологии;
09.03.03 Прикладная информатика;
09.03.04 Программная инженерия;
09.04.01 Информатика и вычислительная техника;
09.04.04 Программная инженерия

Череповец
2014

Рассмотрено на заседании кафедры математического и программного обеспечения ЭВМ, протокол № 2 от 30.09.14.

Составители: *Е.В. Ершов* – д-р техн. наук, профессор;
В.В. Селивановских – канд. техн. наук, доцент;
И.А. Варфоломеев – канд. техн. наук

Рецензенты: *Е.В. Королева*, канд. техн. наук, менеджер Группы приема обращений пользователей СПП ОАО «Северсталь-инфоком»; *О.Ю. Лягинова*, канд. пед. наук (ЧГУ)

Научный редактор: *Е.В. Королева*, канд. техн. наук, менеджер Группы приема обращений пользователей СПП ОАО «Северсталь-инфоком»

© Ершов Е.В., Селивановских В.В.,
Варфоломеев И.А., 2014
© ФГБОУ ВПО «Череповецкий государственный университет», 2014

Введение

Данное учебно-методическое пособие составлено в соответствии с рабочей программой учебной дисциплины «Программирование», состоящей из модулей «Основы программирования», «Структурное программирование» и «Объектно-ориентированное программирование». Пособие содержит инструкции по работе с системой программирования Microsoft Visual Studio, задания, контрольные вопросы, рекомендуемую литературу для самостоятельной работы и для выполнения лабораторных работ.

В пособии рассматриваются вопросы создания и отладки программ в консольном приложении и в визуальной среде разработки Microsoft Visual Studio.

1. Создание консольного приложения

При первом запуске Visual Studio 2012, необходимо выбрать сочетание параметров, которое применяет набор предопределённых настроек к интегрированной среде разработки. Каждое сочетание параметров предназначено для упрощения разработки приложений (рис. 1).

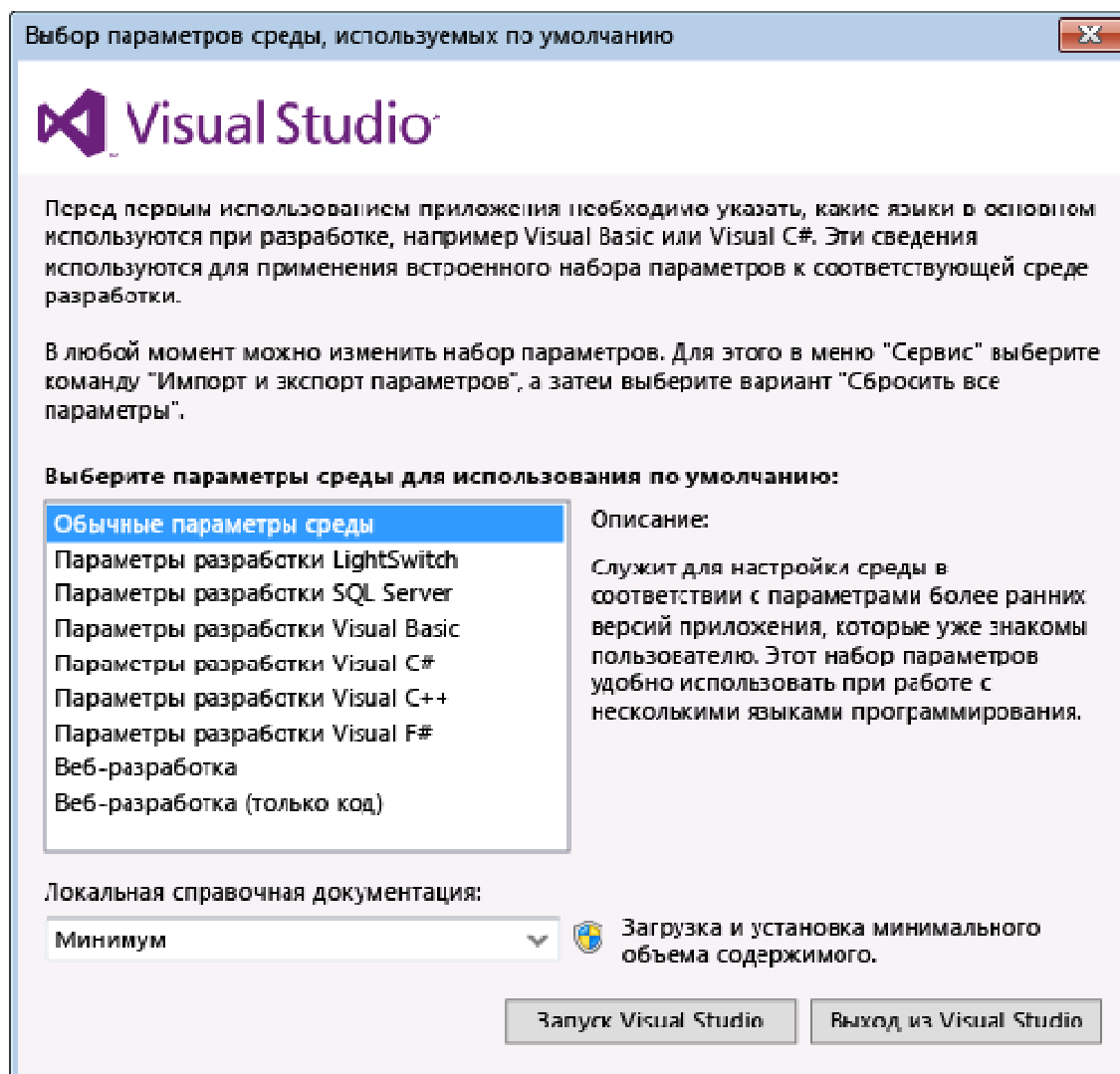


Рис. 1. Диалоговое окно «Выбор параметров среды для использования по умолчанию»

Для выбранного набора параметров «Параметры разработки «X» конфигурация определяет «X» языком программирования по умолчанию. При этом доступ к разработке приложений на других языках не

ограничивается. После выбора подходящего набора параметров открывается стартовое окно (рис. 2).

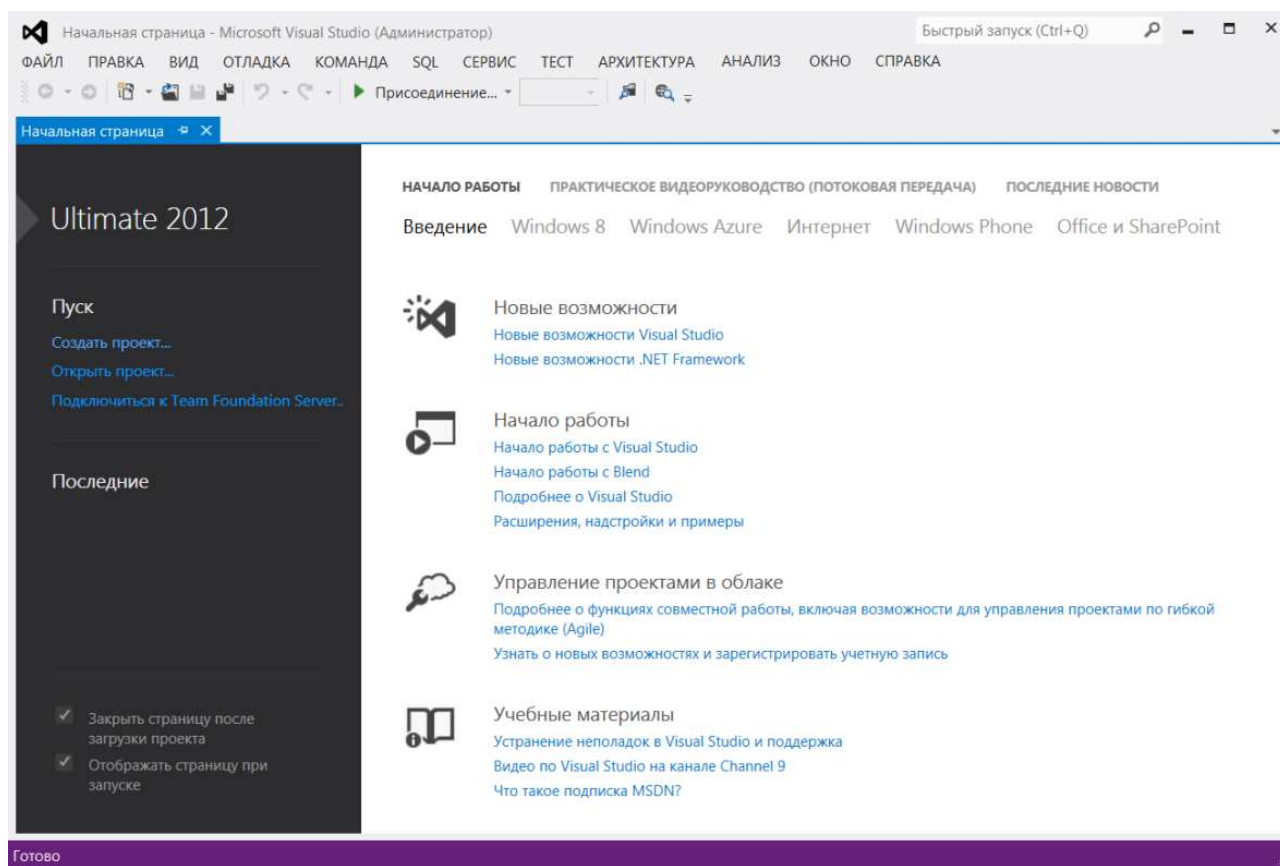


Рис. 2. Начальная страница

Создание нового проекта: Файл \ Создать Проект... (или **Ctrl** + **Shift** + **N**). Далее для создания консольного приложения необходимо выбрать пункт «Консольное приложение Win32» (рис. 3), после этого откроется «Мастер приложений Win32».

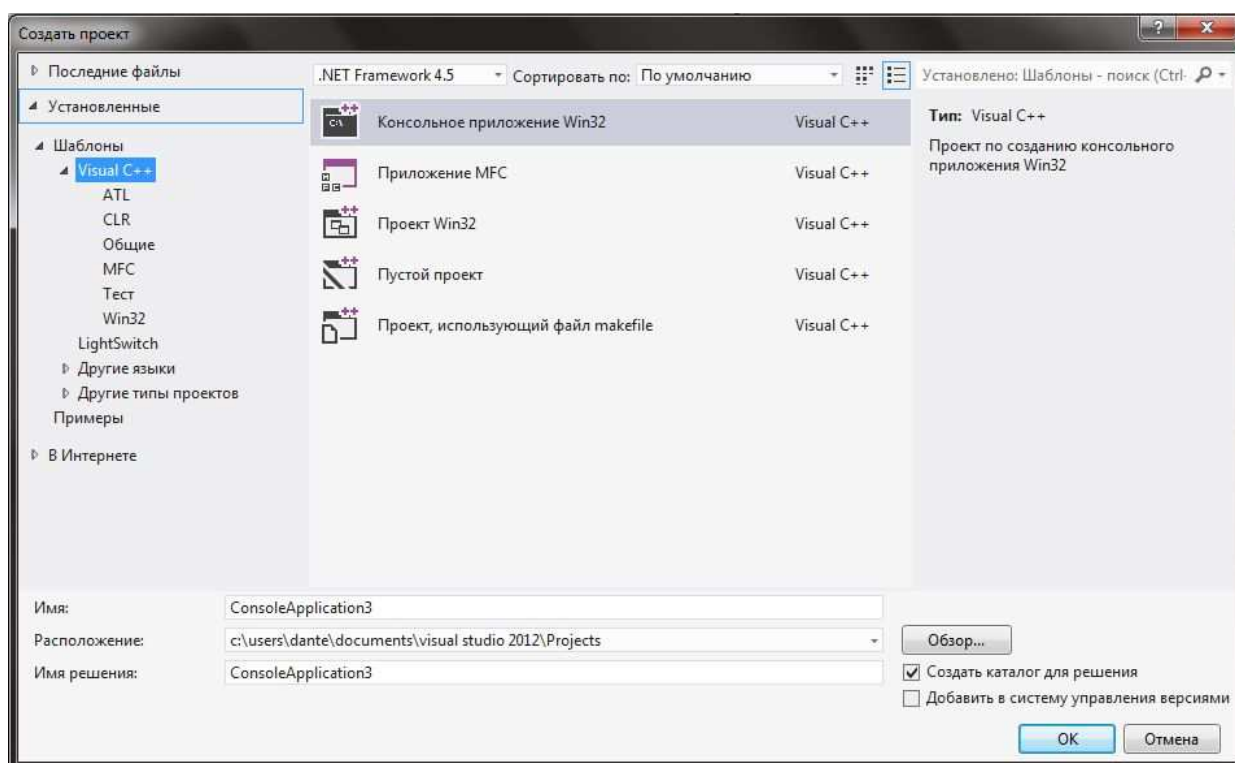


Рис. 3. Окно создания проекта

В следующем окне нужно выставить настройки: Тип приложения – Консольное приложение, Дополнительные параметры – Пустой проект (рис. 4).

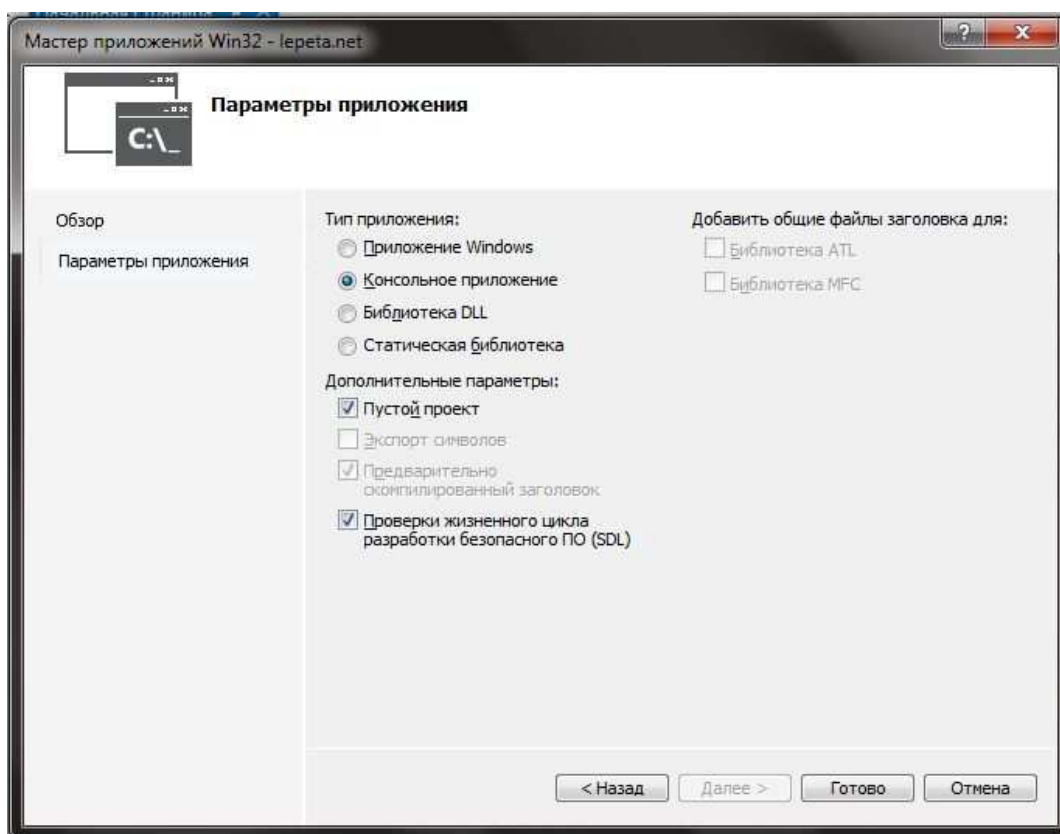


Рис. 4. Окно мастера приложений Win32

Проект создан. Панель «Обозреватель решений» выбирается с помощью пункта меню «Вид» (или **Ctrl** + **Alt** + **L**) (рис. 5).

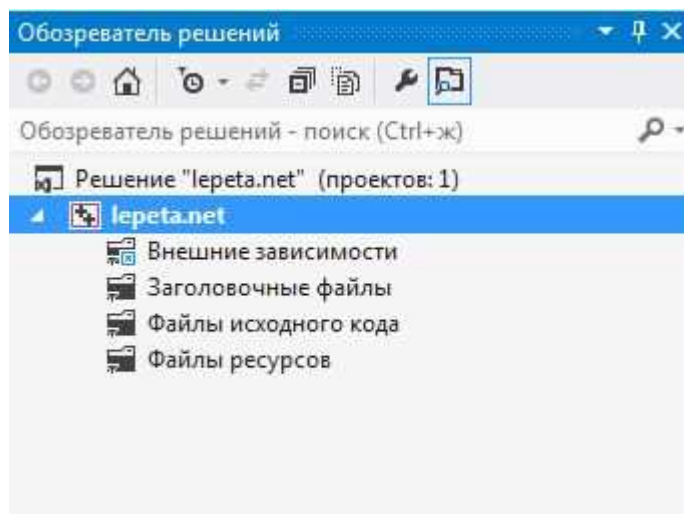


Рис. 5. Панель «Обозреватель решений»

Выбрав пункт «Файлы исходного кода» следует обратиться к его контекстному меню: Добавить \ Создать элемент (**Ctrl** + **Shift** + **A**), выбрать «Файл C++ (.cpp)» (рис. 6) и дать название файлу исходного кода.

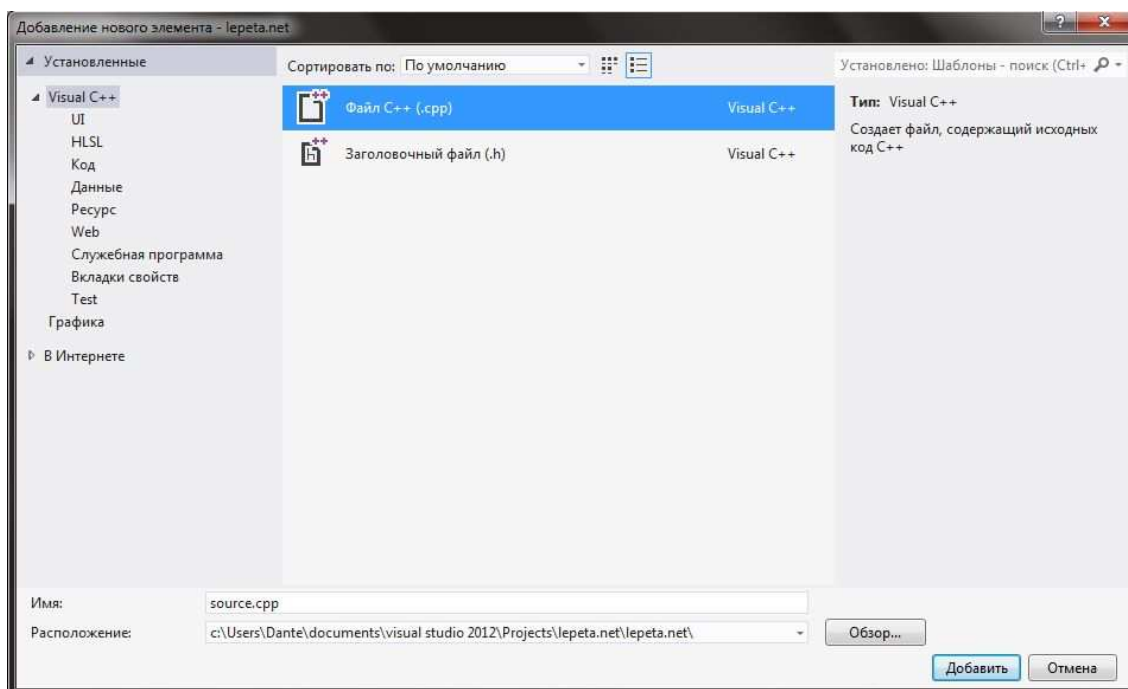


Рис. 6. Добавление нового элемента

```

/* "Hello, world!" by Dante */
#include <stdio.h>

int main(void)
{
    printf("Reader! Lepeta.net welcomes you!\n");
    getchar();
    return 0;
}

```

После написания кода в меню «Построение» необходимо выбрать пункт «Построить решение...» (или F7). Если построение пройдет успешно (т.е. нет ошибок в программе), то в окне вывода будет выведена следующая информация:

```

1>----- Построение начато: проект: lepeteta.net, Конфигурация:
Debug Win32 -----
1> source.cpp
1> lepeteta.net.vcxproj -> c:\users\dante\documents\visual studio
2012\Projects\lepeteta.net\Debug\lepeteta.net.exe
===== Построение: успешно: 1, с ошибками: 0, без измене-
ний: 0, пропущено: 0 =====

```

Для того чтобы запустить проект, следует выбрать пункт «Отладка», а затем «Запуск без отладки» (или Ctrl + F5).

В программе использована функция `getchar()`, так как если запускать exe-файл отдельно (не в Visual Studio), то программа завершится сразу же и не удастся посмотреть, что она выводит. Visual Studio же автоматически добавляет подобную функцию в конец, поэтому программа не завершается и можно видеть все, что она выводит. После выполнения последней операции появляется сообщение о том, что программа завершила работу (рис. 7).

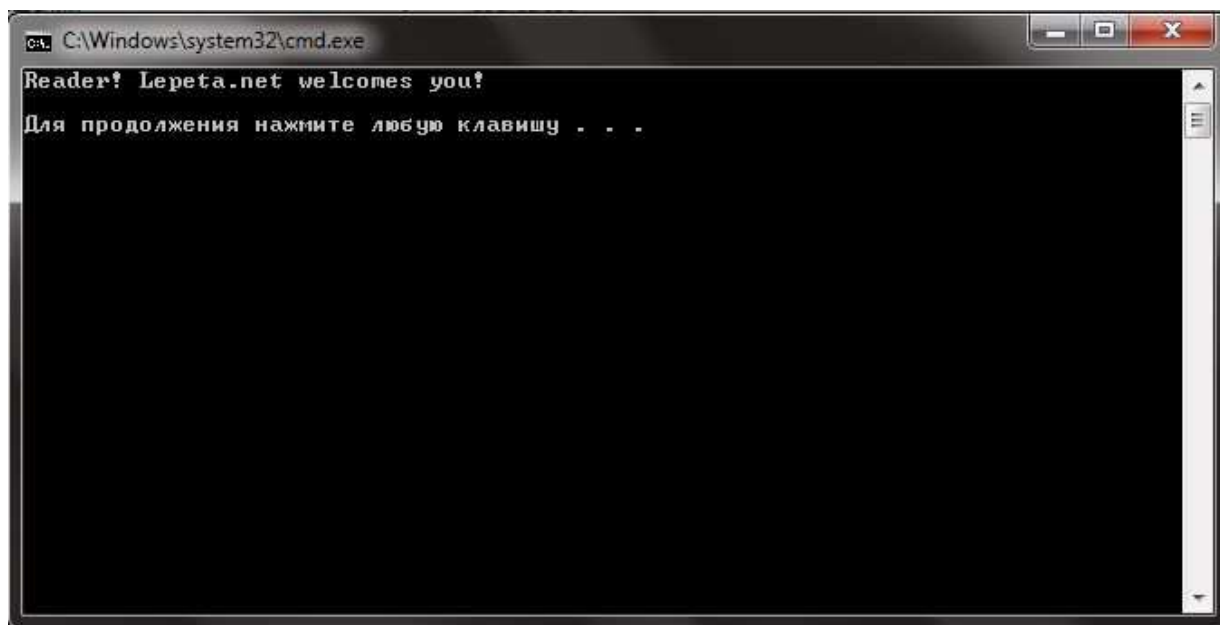


Рис. 7. Результат выполнения программы

Задания

1. Установите параметры среды Visual Studio для использования по умолчанию: «Параметры разработки Visual C++».
2. Создайте консольное приложение для задания согласно варианту, представленному в следующей таблице:

Номер варианта	Задание				
1	Даны два действительных числа. Замените первое число нулем, если оно меньше или равно второму, и оставьте числа без изменения, если это не так				
	Даны три действительных числа. Возведите в квадрат те из них, значения которых неотрицательны				
	Найдите и выведите на экран значения функций t и u при заданных x , y , z , используя стандартные математические функции. Сравните значения функций и выведите True, если $t > u$, и False, если $t < u$				
	<table border="1"> <tr> <td>$t = \left x^{y/x} - \sqrt{y/x} \right$</td><td>$x = 1,825;$</td></tr> <tr> <td></td><td>$y = 18;$</td></tr> </table>	$t = \left x^{y/x} - \sqrt{y/x} \right $	$x = 1,825;$		$y = 18;$
$t = \left x^{y/x} - \sqrt{y/x} \right $	$x = 1,825;$				
	$y = 18;$				

Номер варианта	Задание		
	<table> <tr> <td> $u = (y - x) * \frac{y - \frac{z}{y - x}}{1 + (y - x)^2}$ </td><td> $z = -3,298$ </td></tr> </table>	$u = (y - x) * \frac{y - \frac{z}{y - x}}{1 + (y - x)^2}$	$z = -3,298$
$u = (y - x) * \frac{y - \frac{z}{y - x}}{1 + (y - x)^2}$	$z = -3,298$		
2	<p>Даны два действительных числа. Меньшее из этих чисел замените их полусуммой, а большее – их удвоенным произведением</p> <p>Даны действительные числа a, b, c. Удвойте эти числа, если $a \geq b \geq c$, и замените их абсолютными значениями, если это не так</p> <p>Найдите и выведите на экран значения функций t и u при заданных x, y, z, используя стандартные математические функции. Сравните значения функций и выведите True, если $t > u$, и False, если $t < u$</p> <table> <tr> <td> $t = e^{-yz} \sin(xz + y);$ $u = y \sin((xz)^2 \cos 2z) - 1$ </td><td> $x = -0,5;$ $y = 5;$ $z = 0,44$ </td></tr> </table>	$t = e^{-yz} \sin(xz + y);$ $u = y \sin((xz)^2 \cos 2z) - 1$	$x = -0,5;$ $y = 5;$ $z = 0,44$
$t = e^{-yz} \sin(xz + y);$ $u = y \sin((xz)^2 \cos 2z) - 1$	$x = -0,5;$ $y = 5;$ $z = 0,44$		
3	<p>Текущее время (часы, минуты, секунды) задано тремя переменными: h, m, s. Округлите его до целых значений минут и часов. Например, 14 ч 21 мин 45 с можно округлить до 14 ч 22 мин или 14 ч, а 9 ч 59 мин 23 с – соответственно до 9 ч 59 мин или 10 ч</p> <p>Дано натуральное число n ($n \leq 9999$). Является ли это число палиндромом (перевертышем)? Например, числа 6116, 0440, 2222 являются палиндромами.</p> <p>Найдите и выведите на экран значения функций t и u при заданных x, y, z, используя стандартные математические функции. Сравните значения функций и выведите True, если $t > u$, и False, если $t < u$</p> <table> <tr> <td> $t = 3^{-x+1} \sin x;$ $u = \ln(y + z^2) + \sin^2 \frac{z}{y}$ </td><td> $x = -2;$ $y = 0,59;$ $z = -4,8$ </td></tr> </table>	$t = 3^{-x+1} \sin x;$ $u = \ln(y + z^2) + \sin^2 \frac{z}{y}$	$x = -2;$ $y = 0,59;$ $z = -4,8$
$t = 3^{-x+1} \sin x;$ $u = \ln(y + z^2) + \sin^2 \frac{z}{y}$	$x = -2;$ $y = 0,59;$ $z = -4,8$		

Номер варианта	Задание	
4	Используя логическую переменную, выведите True или False в зависимости от того, имеют три заданных целых числа одинаковую четность или нет	
	Дано натуральное число n ($n \leq 9999$). Верно ли, что это число содержит три одинаковые цифры, как, например, числа 6676, 4544, 0007 и т.д.?	
	<p>Найдите и выведите на экран значения функций t и u при заданных x, y, z, используя стандартные математические функции. Сравните значения функций и выведите True, если $t > u$, и False, если $t < u$</p> <table border="1"> <tr> <td> $t = x + \ln(x) - \frac{x^3}{5};$ $u = \sin(y) + \sqrt{\frac{y}{z}}$ </td><td> $x = 0,25;$ $y = 1;$ $z = 13,26$ </td></tr> </table>	$t = x + \ln(x) - \frac{x^3}{5};$ $u = \sin(y) + \sqrt{\frac{y}{z}}$
$t = x + \ln(x) - \frac{x^3}{5};$ $u = \sin(y) + \sqrt{\frac{y}{z}}$	$x = 0,25;$ $y = 1;$ $z = 13,26$	
5	Даны два действительных числа. Выведите первое число, если оно больше второго, и оба числа, если это не так	
	Выведите на экран номер четверти, которой принадлежит точка с координатами x , y или название координатной оси, если $x = 0$ или $y = 0$, или начало координат, если $x = 0$ и $y = 0$.	
	<p>Найдите и выведите на экран значения функций t и u при заданных x, y, z, используя стандартные математические функции. Сравните значения функций и выведите True, если $t > u$, и False, если $t < u$</p> <table border="1"> <tr> <td> $t = \frac{2 \cos\left(x - \frac{\pi}{6}\right)}{\frac{1}{2} + \sin^2 y};$ $u = 1 + \frac{z^2}{3 + \frac{z^2}{5}}$ </td><td> $x = 1,426;$ $y = -1,22;$ $z = 3$ </td></tr> </table>	$t = \frac{2 \cos\left(x - \frac{\pi}{6}\right)}{\frac{1}{2} + \sin^2 y};$ $u = 1 + \frac{z^2}{3 + \frac{z^2}{5}}$
$t = \frac{2 \cos\left(x - \frac{\pi}{6}\right)}{\frac{1}{2} + \sin^2 y};$ $u = 1 + \frac{z^2}{3 + \frac{z^2}{5}}$	$x = 1,426;$ $y = -1,22;$ $z = 3$	

3. Запустите приложение, введите данные, запишите полученный результат.

Контрольные вопросы

1. Если некоторый язык программирования определен языком программирования по умолчанию, то могут ли быть разработаны приложения на других языках программирования?
2. Как запустить консольное приложение на выполнение?
3. Какое расширение имеет файл с кодом программы, созданной в консольном приложении?
4. Какую библиотеку потребовалось подключить для выполнения заданий?
5. В каких случаях используется явное преобразование типов?

2. Начало работы в интегрированной среде разработки, выбор панелей инструментов

При первом запуске Visual Studio 2012, как и при создании консольного приложения (см. п. 1), необходимо выбрать сочетание параметров, которое применяет набор предопределённых настроек к интегрированной среде разработки.

После выполнения предварительных действий в центре окна приложения располагается «Начальная страница». Если загружены решение или проект, то на её месте будут отображаться редакторы и конструкторы. После открытия Visual Studio можно указать окна инструментов, меню и панели инструментов и пространство главного окна. Для работы понадобятся следующие панели инструментов: «Обозреватель решений», «Список ошибок», «Панель элементов» и «Окно свойств». Неиспользуемые панели необходимо закрыть.

Добавление необходимых панелей инструментов осуществляется с помощью опции главного меню «Вид» (рис. 8).

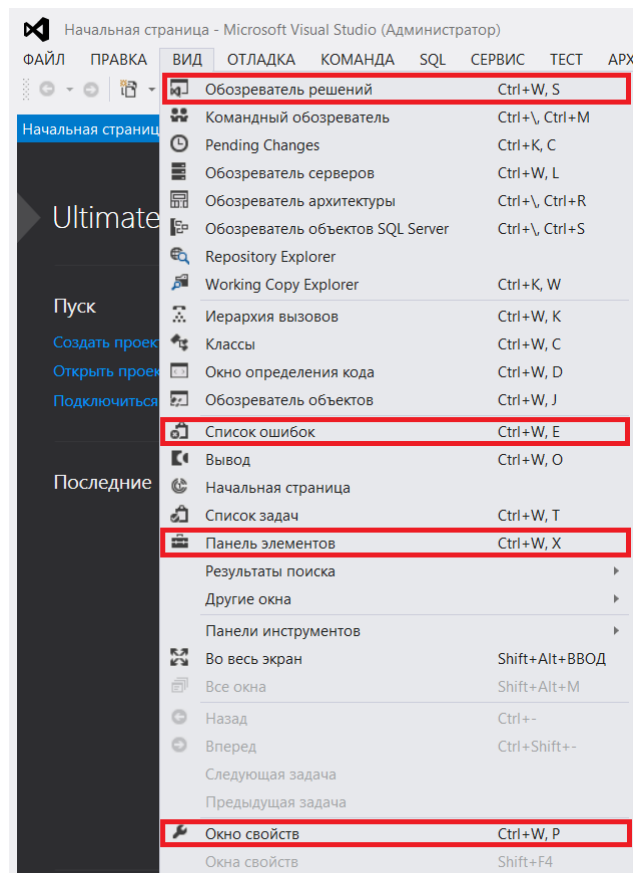


Рис. 8. Добавление панелей инструментов

Указанные панели могут быть прикреплены к удобному месту окна интегрированной среды разработки посредством их перетаскивания (панели «Обозреватель решений», «Свойства» и «Панель элементов» рекомендуется расположить по правой стороне окна, а панель «Список ошибок» – по нижней). На рис. 9 показаны возможности компоновки панелей инструментов в окне интегрированной среды разработки Visual Studio 2012.

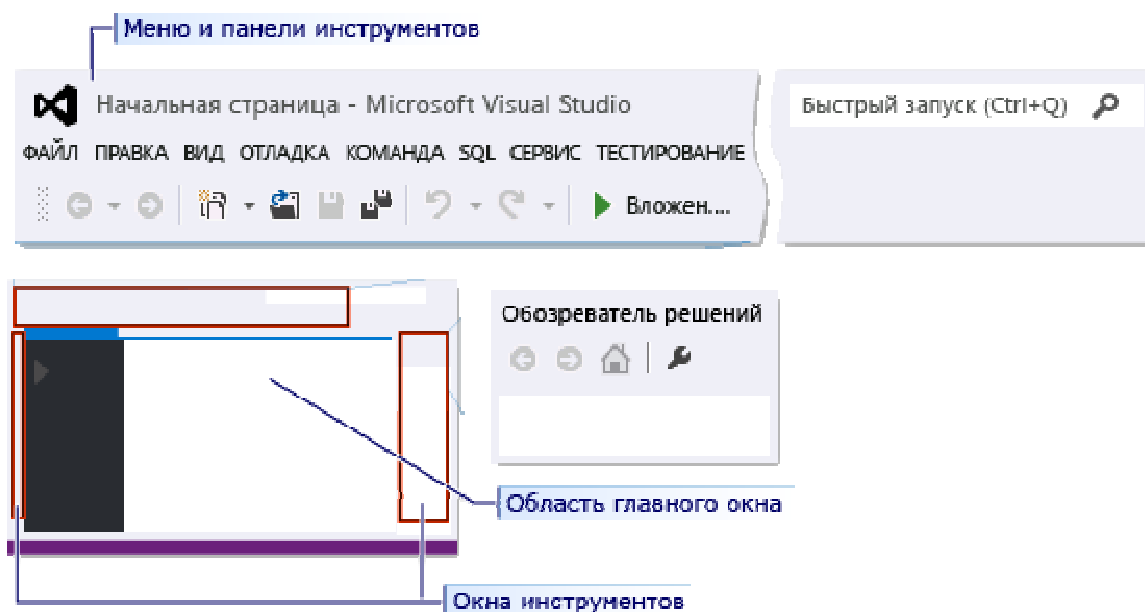


Рис. 9. Возможности компоновки окон

Кроме того, для всех вышеуказанных панелей рекомендуется задать состояние «Автоматически скрывать». Автоматическое скрывание окон позволит добиться максимального размера рабочей области. Для этого необходимо воспользоваться соответствующей командой контекстного меню окна (рис. 10).

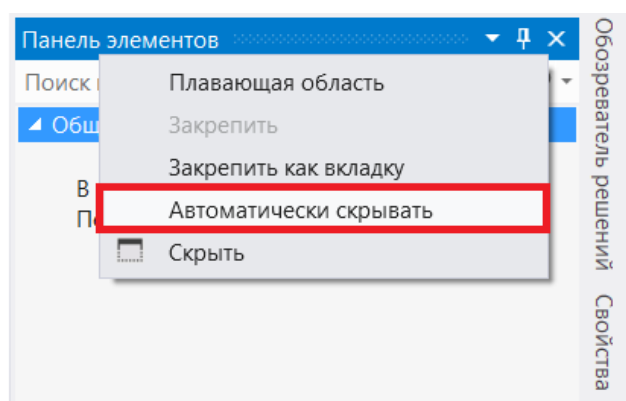


Рис. 10. Управление окнами панелей инструментов

После настройки интегрированной среды разработки можно перейти к разработке программного обеспечения.

Задания

1. Установите параметры среды Visual Studio для использования по умолчанию: «Параметры разработки Visual C++».

2. На «Начальной странице» добавьте 2–3 панели инструментов и скомпонуйте их двумя способами.

3. Для панелей инструментов задайте состояние «Автоматически скрывать».

Контрольные вопросы

1. Как осуществляется добавление панелей инструментов?

2. Как прикрепить панели инструментов к определенному месту окна интегрированной среды разработки?

3. Как закрыть неиспользуемые панели инструментов?

4. С какой целью используется «Автоматическое скрывание» окон панелей инструментов?

3. Создание проекта и решения

Чтобы эффективно управлять элементами, используемыми на этапе разработки, например ссылками, подключениями данных, папками и файлами, в Visual Studio предусмотрены два типа контейнеров: решение и проект.

Решение содержит элементы, необходимые для создания приложения. Оно содержит один или несколько проектов, а также файлы и метаданные, которые позволяют определить решение как единое целое. Visual Studio автоматически создает решение при создании нового проекта. В Visual Studio описания решения хранятся в двух файлах: «имя_решения.sln» и «имя_решения.suo».

В решении осуществляется управление, построение и отладка

элементов проектов, которые логически составляют приложение. На выходе проект, как правило, представляет собой исполняемую программу (*.exe), файл библиотеки динамической компоновки (*.dll) или модуль.

Кроме того, необходимо выделить элементы проекта и элементы решения. Элементами проекта могут являться файлы, ссылки на библиотеки (*.dll), подключения данных и папки, находящиеся в проекте. Некоторые элементы представляют собой физические элементы, помещенные в хранилище. Другие элементы могут быть ссылками и представляют собой указатели на другие элементы, которые могут находиться в службе, модуле или других компонентах. Элементы решения находятся в папке «элементы решения». Эти элементы являются проектно-независимыми файлами, созданными в дополнение к файлам проекта. Элементы решения представляют собой файлы, которые важны для разработки проектов, но не принадлежат к определенному проекту.

При выполнении лабораторных работ необходимо объединить все проекты в одно решение.

Visual Studio предоставляет несколько предварительно определенных шаблонов. Эти шаблоны можно использовать для создания основного контейнера и предварительного набора элементов, необходимых для разработки приложения, класса, элемента управления или библиотеки. Например, если необходимо создать приложение Windows, проектом будет предложен элемент формы Windows, который можно настроить.

Для создания проекта с помощью шаблона необходимо воспользоваться опцией главного меню «Файл\Создать\Проект», в результате откроется диалоговое окно (рис. 11).

В зависимости от выбранных параметров интегрированной среды разработки дерево шаблонов будет выглядеть по-разному. В данном случае среда настроена под разработку на языке C#, поэтому данный язык используется по умолчанию. Языки Visual Basic, Visual C++, Visual F# находятся в пункте «Другие языки» (1). После выбора языка Visual C++ (2) появится список доступных шаблонов.

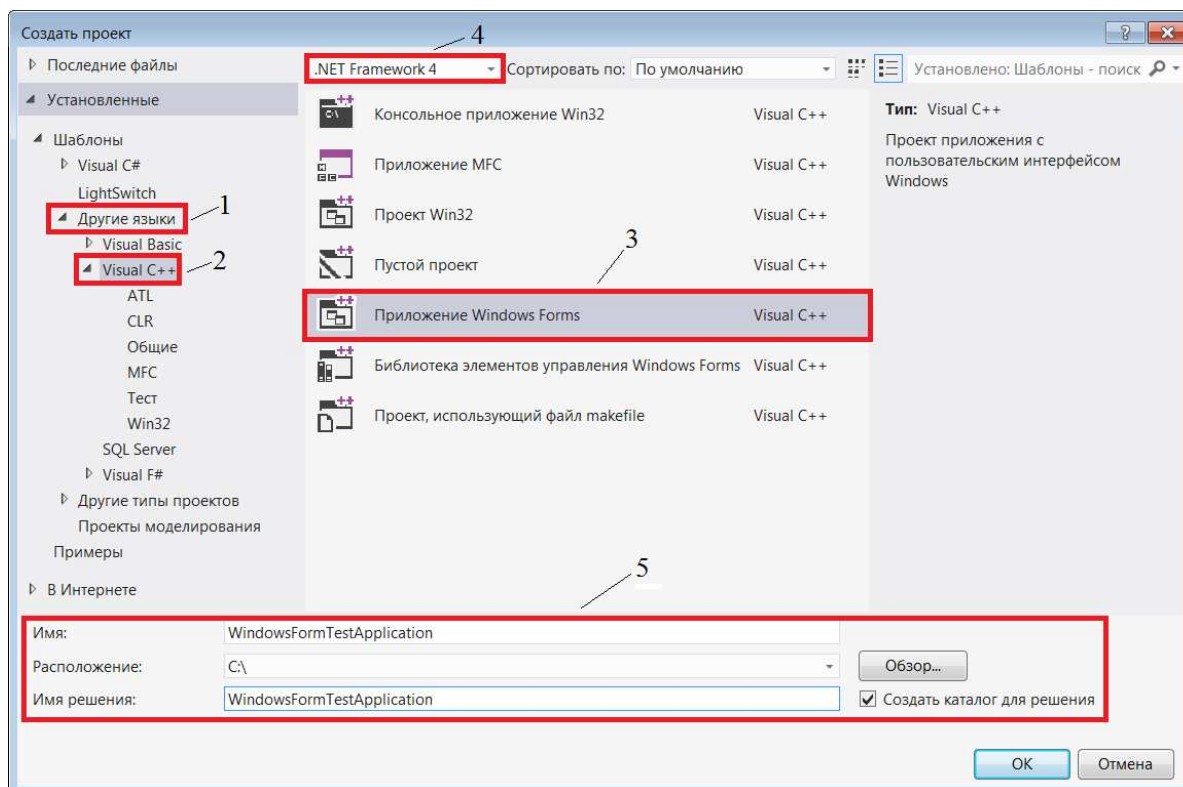


Рис. 11. Диалоговое окно создания проекта

Необходимо выбрать шаблон «Приложение Windows Forms» (3). Кроме того, необходимо задать целевую версию .NET Framework 4 (4). На заключительном шаге требуется указать имя проекта, имя решения и его расположение (5).

В том случае если для языка Visual C++ отсутствует описанный шаблон, необходимо осуществить его установку. Для этого необходимо:

1. Обратиться к системным администраторам кафедры за файлом шаблона или загрузить его по ссылке <http://www.t-hart.org/vs2012/>;
2. Скопировать файлы в каталог C:\Program Files (x86)\Microsoft Visual Studio 11.0\VC\vcprojects\vcNET\;
3. Перезапустить Visual Studio 2012.

.NET Framework – программная платформа, выпущенная компанией Microsoft в 2002 году. Основой платформы является общезыковая среда исполнения Common Language Runtime (CLR), которая подходит для разных языков программирования. Функциональные возможности CLR доступны в любых языках программирования, использующих эту среду.

После нажатия на кнопку «ОК» будет произведена загрузка проекта с автоматически созданными элементами (рис. 12).

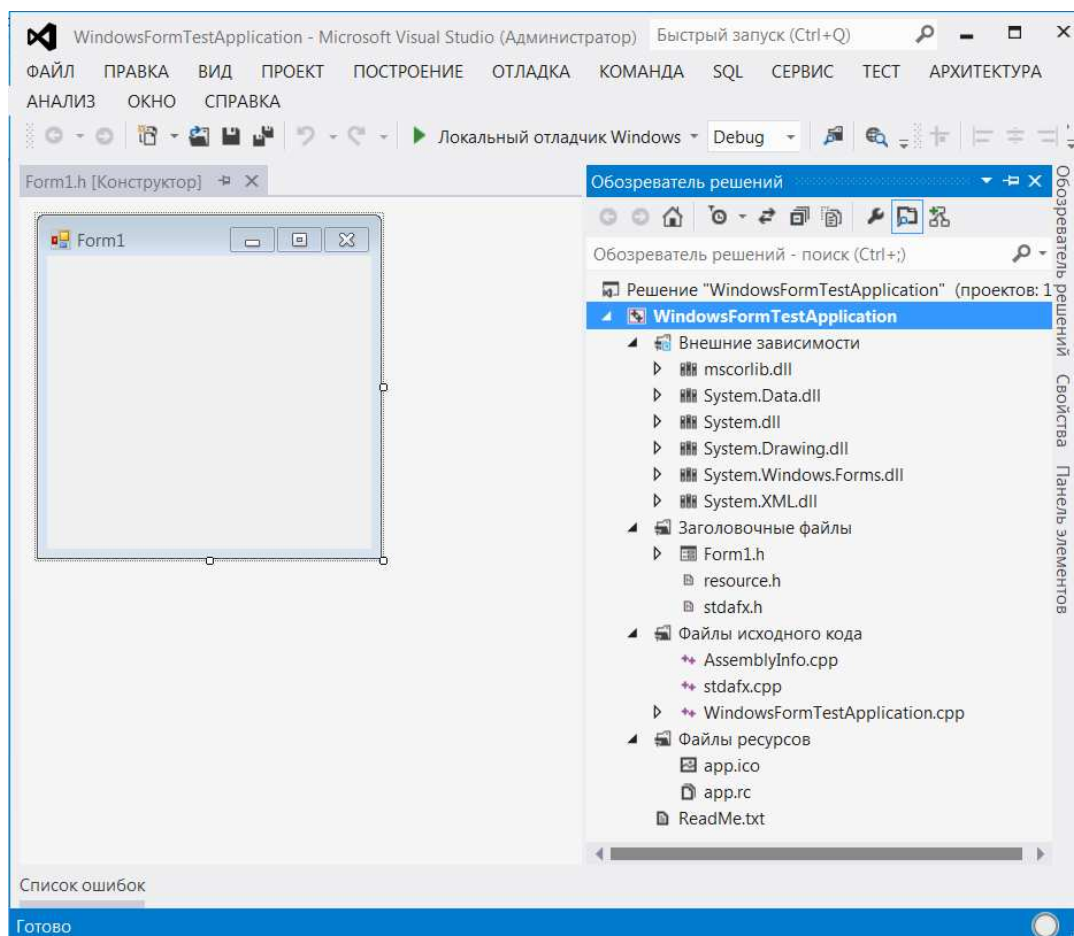


Рис. 12. Структура созданного проекта

В секции «Внешние зависимости» находятся ссылки на подключаемые библиотеки (*.dll). При создании решения или проекта с помощью шаблона автоматически подключается ряд библиотек, необходимых для работы приложения и входящих в состав .NET Framework. При этом нет необходимости в том, чтобы приложение распространялось вместе с библиотеками, входящими в .NET Framework. Достаточно, чтобы на целевом компьютере была установлена соответствующая версия .NET Framework. Кроме того, с помощью контекстного меню могут быть добавлены собственные библиотеки или библиотеки сторонних разработчиков, которые необходимо постав-

лять вместе с приложением. Для этого в свойствах библиотеки необходимо установить значение «true» для свойства «Включен в проект».

Секции «Заголовочные файлы» и «Файлы исходного кода» содержат код классов, используемых в программе. Заголовочные файлы (имя_класса.h) содержат описание класса (поля и прототипы методов). Код методов описывается в файлах (имя_класса.cpp). Секция «Файлы ресурсов» содержит прочие файлы, необходимые для разработки и выполнения приложения, например, файл растрового изображения, значка или курсора и т.д.

Зададим заголовок главной формы приложения. Для этого щелкнем левой кнопкой мыши по окну, находящемуся в режиме конструктора, и откроем панель «Свойства». Затем присвоим свойству «Text» значение «Тестовое приложение» (рис. 13).

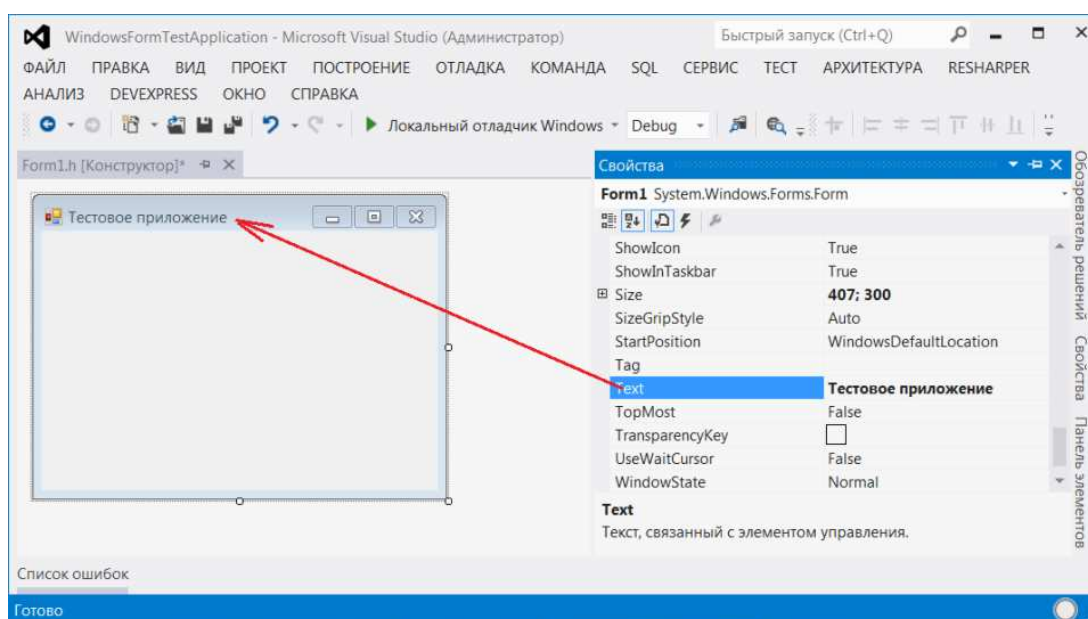


Рис. 13. Редактирование свойств окна

Аналогичным образом задаются остальные свойства объектов пользовательского интерфейса. Добавим на форму объекты пользовательского интерфейса. Для этого, используя «Панель инструментов», перетаскиванием добавим на форму три объекта для надписи (Label), объект «кнопка» (Button) и два текстовых поля ввода (TextBox) (рис. 14).

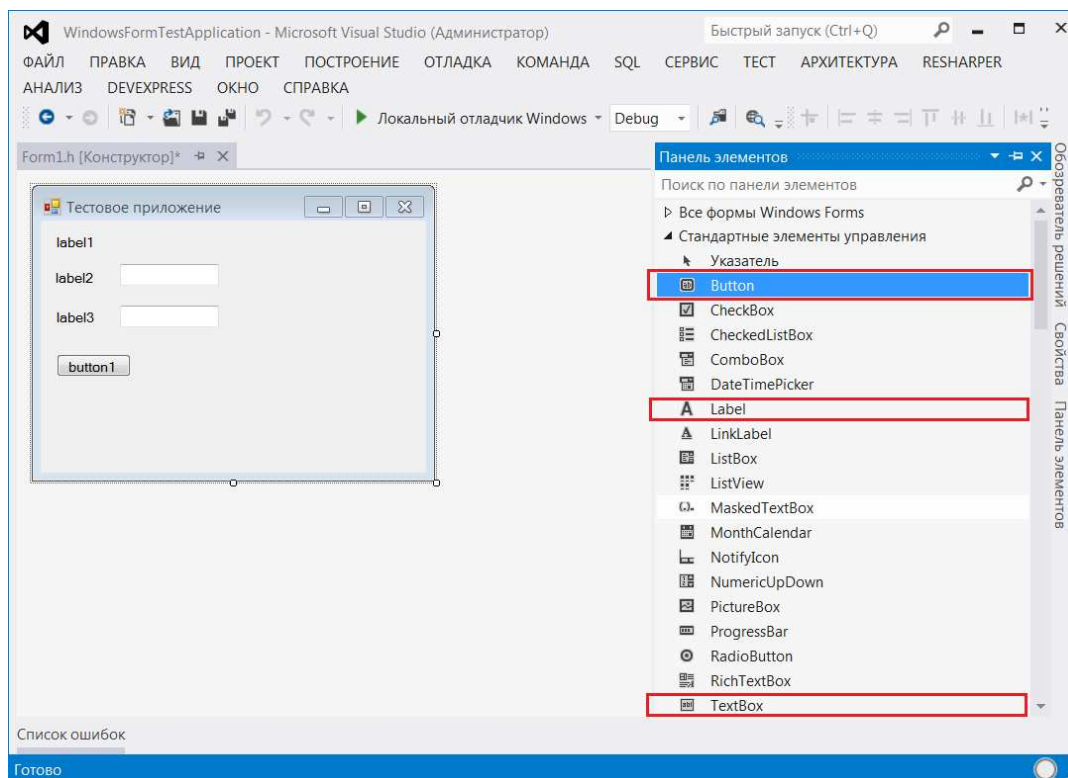


Рис. 14. Добавление объектов пользовательского интерфейса

Редактируя свойства добавленных объектов, получим следующий вид главной формы приложения (рис. 15).

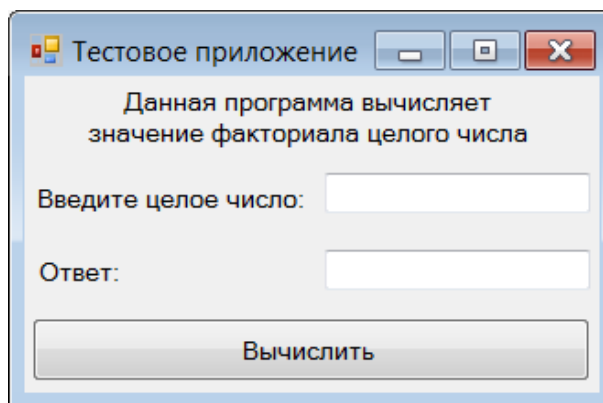


Рис. 15. Отредактированная форма приложения

Задания

1. Настройте проект (Visual C++) с помощью дерева шаблонов, выбрав шаблон «Приложение Windows Forms».
2. Опишите структуру созданного проекта.
3. Главной форме дайте название «Пример 1».

4. Добавьте на форму объекты, необходимые, по вашему мнению, для отображения входных и выходных данных при решении задач согласно варианту. Задайте их размеры, цвет и тип шрифта (см. таблицу).

Номер варианта	Задание
1	Выведите все двузначные числа, которые делятся на N или содержат цифру N , где N – цифра, введенная пользователем
	Вычислите значение выражения. Значение x вводится с клавиатуры. $Z = \prod_{n=2}^7 \frac{x^n}{2^n e^n}$
2	Определите количество четырехзначных чисел, кратных 45, две средние цифры которых 9 и 7. Выведите эти числа
	Вычислите значение выражения. Значение x вводится с клавиатуры. $Z = \prod_{n=2}^7 \frac{x^{n-1}}{n 3^n \ln(n)}$
3	Определите количество трехзначных натуральных чисел, сумма цифр которых равна введенному с клавиатуры числу N . Выведите эти числа
	Вычислите значение выражения. Значение x вводится с клавиатуры. $Y = \sum_{n=1}^{15} \left(x^n + \frac{1}{2^n x^n} \right)$
4	Среди двузначных чисел найдите и выведите те, сумма квадратов цифр которых делится на 13.
	Вычислите значение выражения. Значение x вводится с клавиатуры. $Y = \sum_{n=1}^{10} \left(x^{n-1} + \frac{x^n}{2^n} \right)$
5	Среди трехзначных чисел найдите и выведите те, квадрат которых оканчивается тремя цифрами, составляющими исходное число
	Вычислите значение выражения. Значение x вводится с клавиатуры. $Y = \sum_{n=1}^{10} \left(x^{n-1} + \frac{x^n}{2^n} \right)$

Контрольные вопросы

1. В каких файлах хранятся описания решения?
2. Дайте характеристику элементов проекта и элементов решения.
3. От чего зависит вид дерева шаблонов?
4. О каких файлах содержится информация в секции решения «Внешние зависимости»?
5. Какую информацию содержат секции «Заголовочные файлы» и «Файлы исходного кода»?

4. Выполнение и отладка приложения

Зададим обработчик события «Нажатие на кнопку» (Click). Для этого необходимо дважды щелкнуть по кнопке. Обработчик события «Нажатие на кнопку» можно задать и через панель «Свойства» (рис. 16).

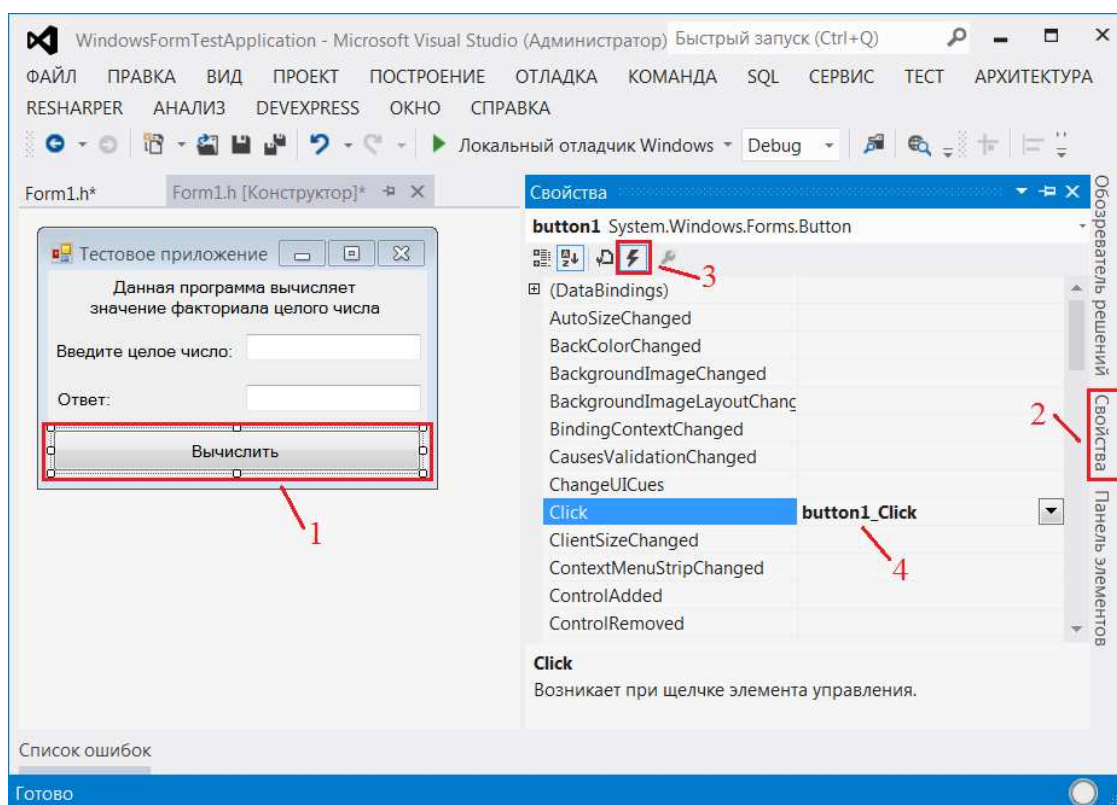


Рис. 16. Задание обработчика события


Для перехода к коду, реализующему класс формы, необходимо нажать клавишу «F7» или воспользоваться командой контекстного меню «перейти к коду».

В результате добавления обработчика события в файле Form1.h, реализующем класс формы, появится метод

```
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) {  
}
```

В данный метод необходимо добавить код обработки события. Реализуем расчет факториала введенного значения:

```
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) {  
  
    // Чтение значения из поля ввода и преобразование в целочисленный тип  
    int a = System::Convert::ToInt32(this->textBox1->Text);  
    if(a == 0) //Факториал 0 равен 1  
    {  
        textBox2->Text = "1";  
    }  
    else if(a < 0) // Факториал отрицательного числа равен 0  
    {  
        textBox2->Text = "0";  
    }  
    else // Рассчитываем факториал положительного числа  
    {  
        int result = 1;  
        for(int i=2; i<=a; i++)  
        {  
            result *= i;  
        }  
        textBox2->Text = result.ToString();  
    }  
}
```

Построим (скомпилируем) полученное приложение, используя клавишу «F5» или кнопку главного меню среды разработки  Локальный отладчик Windows. При запущенном приложении установим точки останова в строках, выполнение которых хотим отследить. Для этого необходимо щелкнуть левой кнопкой мыши на серой полосе слева от текста программы (рис. 17).


```

137 private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) {
138
139     // Чтение значения из поля ввода и преобразование его в целочисленный тип
140     int a = System::Convert::ToInt32(this->textBox1->Text);
141
142     if(a == 0) //Факториал 0 равен 1
143     {
144         textBox2->Text = "1";
145     }
146     else if(a < 0) // Факториал отрицательного числа равен 0
147     {
148         textBox2->Text = "0";
149     }
150     else // Расчитываем факториал положительного числа
151     {
152         int result = 1;
153         for(int i=2; i<=a; i++)
154         {
155             result *= i;
156         }
157         textBox2->Text = result.ToString();
158     }
159 }

```

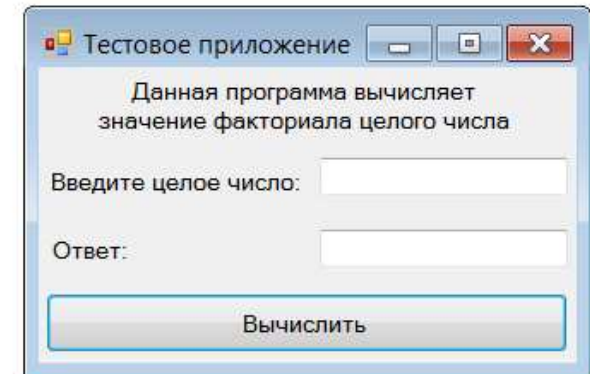
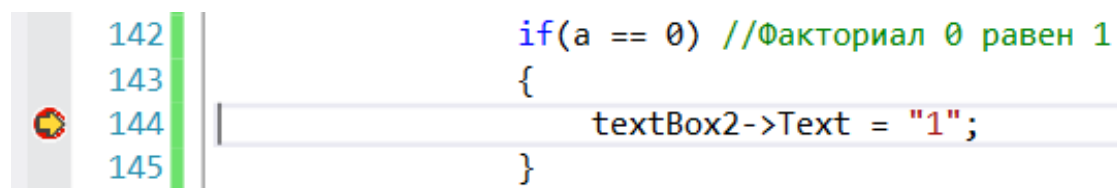


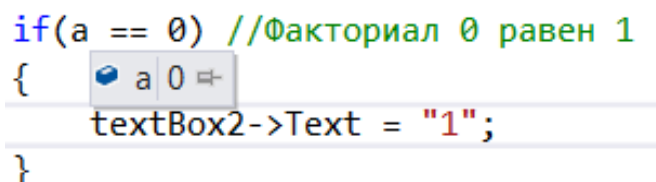
Рис. 17. Установка точек останова

В окне программы введем число 0 и нажмем кнопку «Вычислить». Выполнение программы остановится на строке:



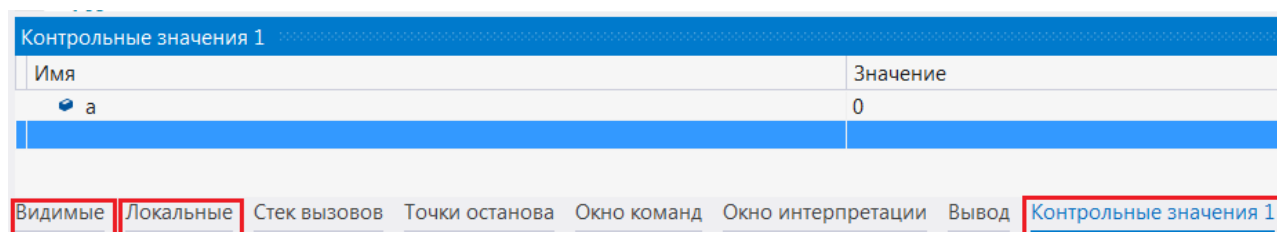
```
142                                     if(a == 0) //Факториал 0 равен 1
143                                     {
144                                     textBox2->Text = "1";
145                                     }
```

Значения переменных можно отследить несколькими способами: навести указатель мыши на переменную в тексте программы (рис. 18, а) или воспользоваться панелями «Видимые», «Локальные», «Контрольные значения» (рис. 18, б).



```
if(a == 0) //Факториал 0 равен 1
{
  a 0
  textBox2->Text = "1";
}
```

а)



б)

Рис. 18. Отслеживание значений переменной

Для работы с приложением в режиме отладки необходимо использовать команды опции «Отладка» главного меню интегрированной среды разработки (рис. 19).

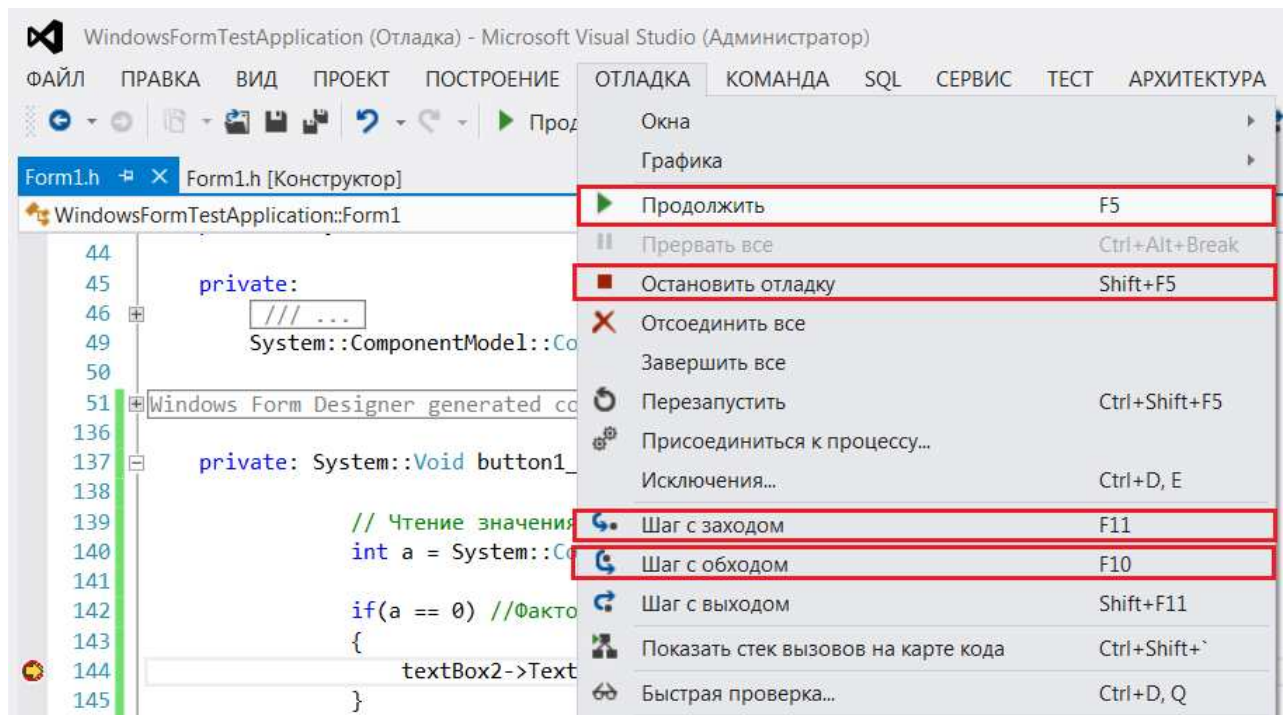


Рис. 19. Основные команды, используемые для отладки приложения

Введем в программу значение «0.1» и нажмем на кнопку «Вычислить». Выполнение программы будет остановлено из-за ошибки преобразования строки в целое число (рис. 20).

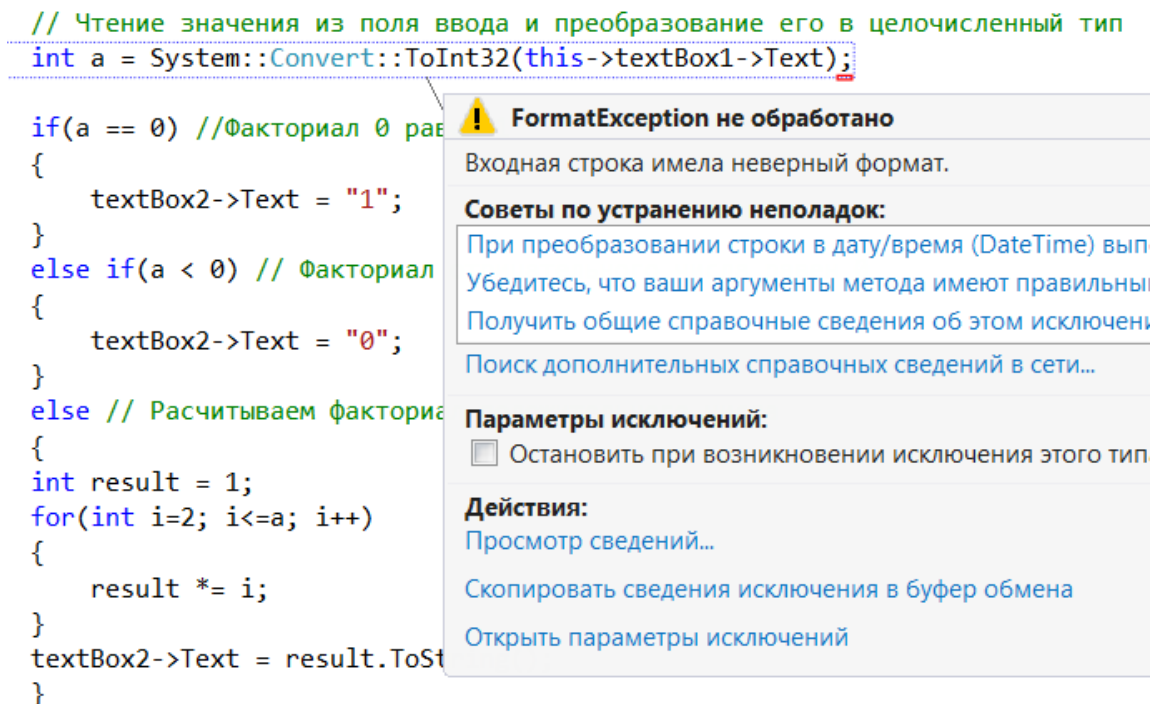


Рис. 20. Ошибка работы программы при отладке

Завершим отладку («Shift»+«F5»). Реализуем обработчик исключительных ситуаций (рис. 21):

```
try {  
    // Чтение значения из поля ввода и преобразование его в цело-  
    численный тип  
    int a = System::Convert::ToInt32(this->textBox1->Text);  
  
    if(a == 0) //Факториал 0 равен 1  
    {  
        textBox2->Text = "1";  
    }  
    else if(a < 0) // Факториал отрицательного числа равен 0  
    {  
        textBox2->Text = "0";  
    }  
    else // Расчитываем факториал положительного числа  
    {  
        int result = 1;  
        for(int i=2; i<=a; i++)  
        {  
            result *= i;  
        }  
        textBox2->Text = result.ToString();  
    }  
}  
catch (...)  
{  
    MessageBox::Show("Ошибка вычисления факториала");  
}
```

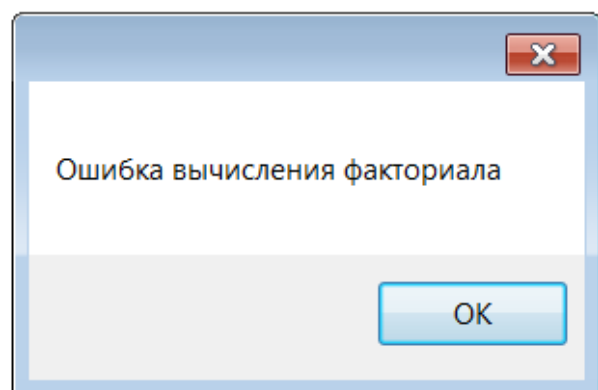
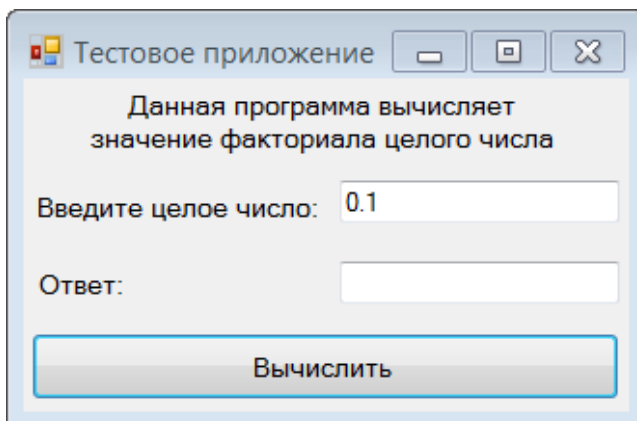


Рис. 21. Обработка исключительной ситуации

Задания

1. Задайте обработчик события нажатия на кнопку в созданных в предыдущем разделе приложениях.
2. Постройте (скомпилируйте) полученные приложения.
3. Установите точки останова.
4. Введите исходные данные.
5. Отследите значения переменных разными способами.
6. Завершите отладку приложений.

Контрольные вопросы

1. Какими способами можно задать обработчик события?
2. Назовите два способа построения приложения.
3. Какими способами можно отследить значения переменных в процессе отладки приложения?
4. Каким образом можно установить контрольные точки перед отладкой приложения?
5. Как можно пошагово выполнить приложение?

5. Создание класса

Создадим класс, который будет осуществлять расчет факториала двумя способами и выводить результат на форму. Для добавления нового класса в проект необходимо воспользоваться контекстным меню секции «Заголовочные файлы» (рис. 22).

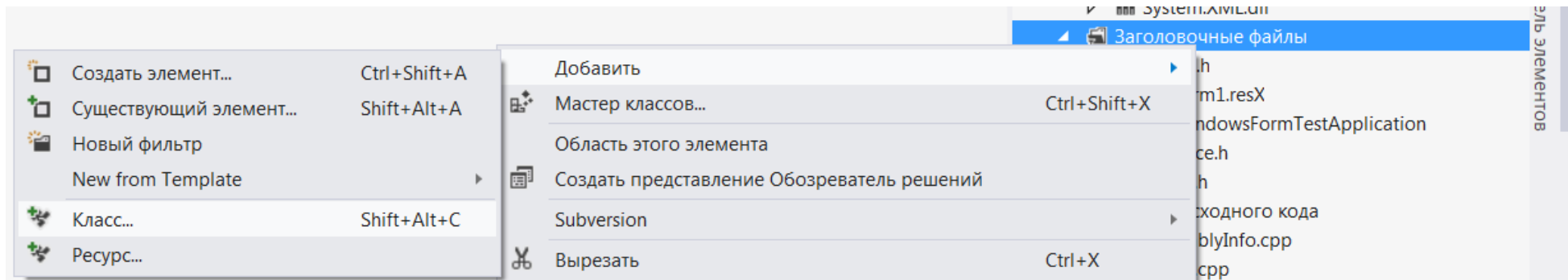


Рис. 22. Вызов диалогового окна добавления класса

В открывшемся окне необходимо указать «Класс C++» и выбрать пункт «Добавить». В результате откроется диалоговое окно (рис. 23), в котором необходимо указать имя класса, его область видимости, имя базового класса и имена заголовочного файла и файла исходного кода.

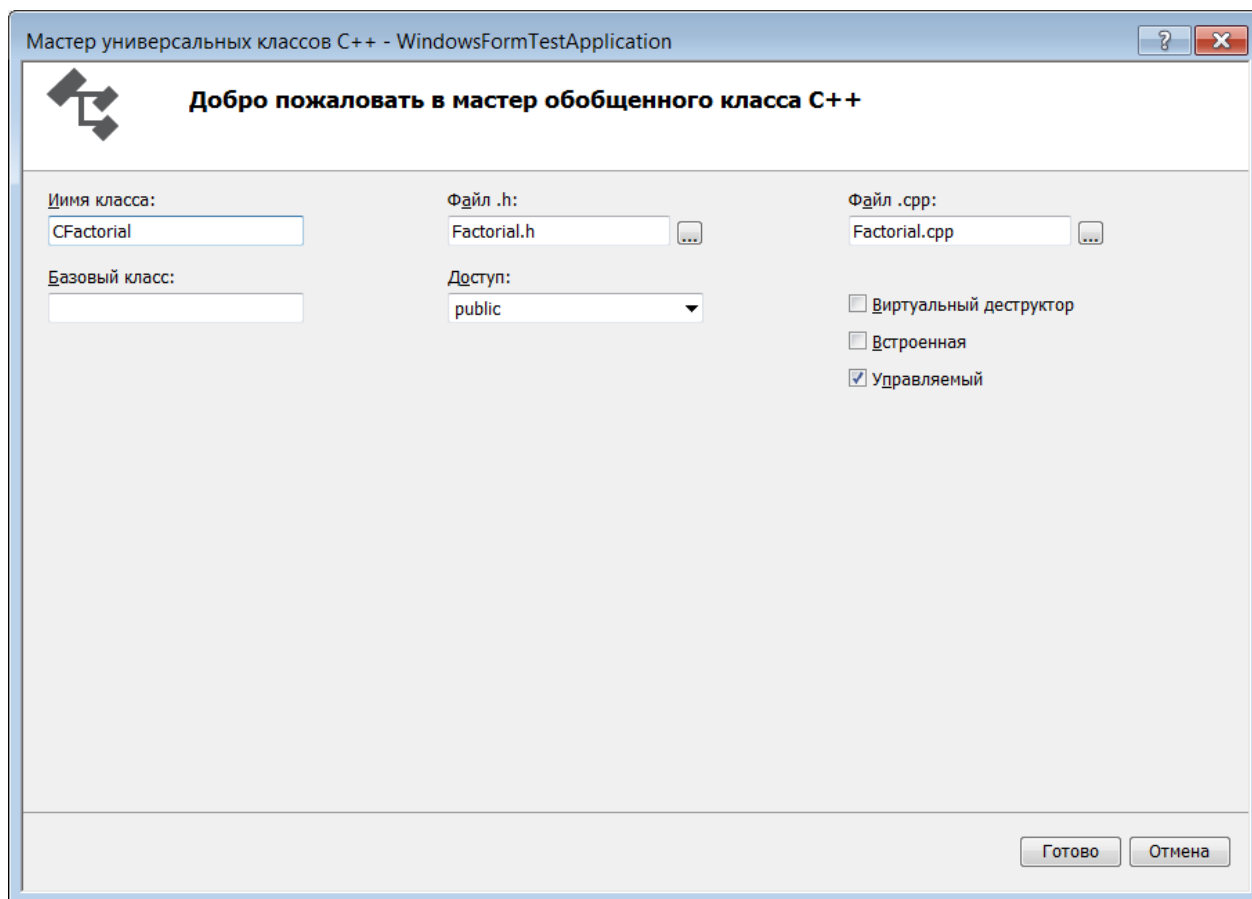


Рис. 23. Мастер универсальных классов C++

После нажатия на кнопку «Готово» в проект будут добавлены 2 файла (заголовочный файл и файл исходного кода) (рис. 24).

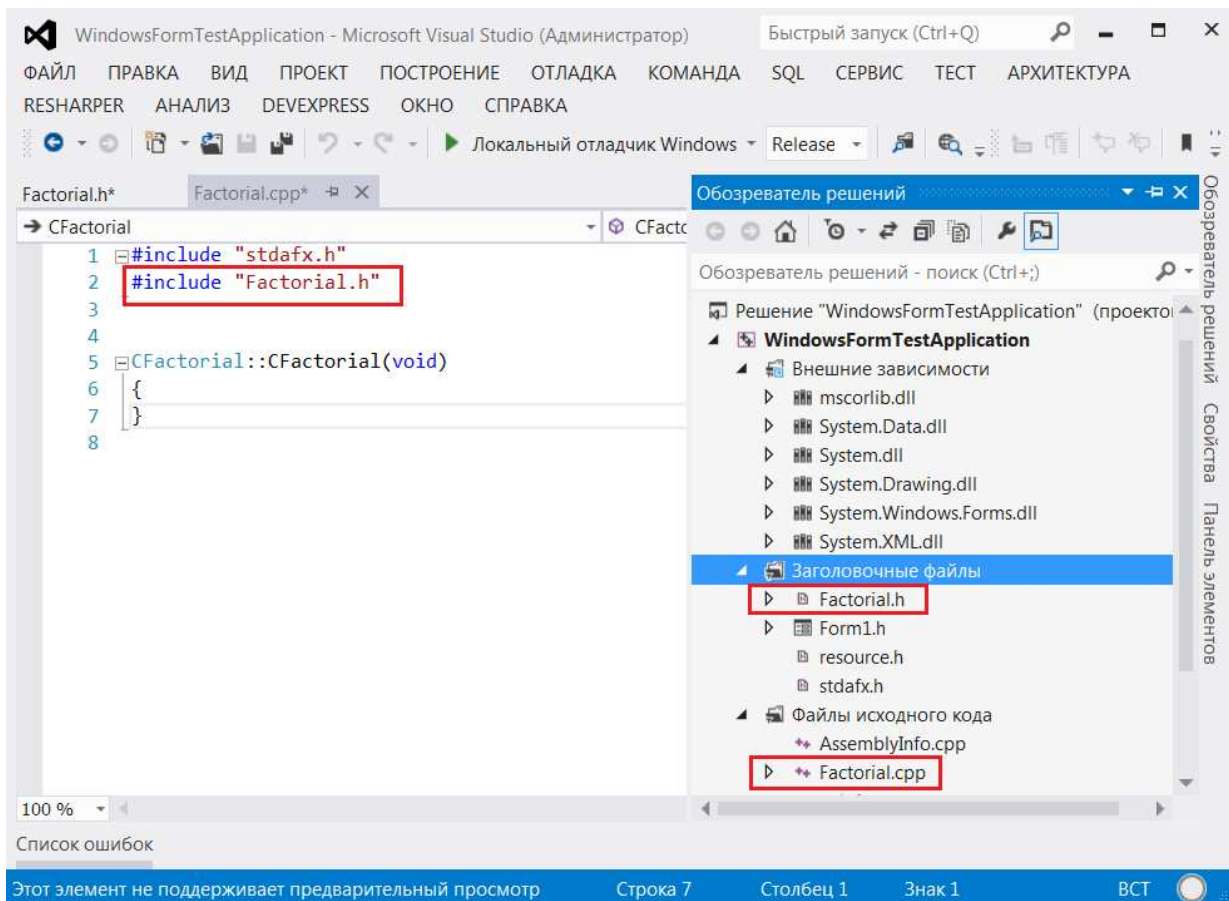


Рис. 24. Структура проекта после добавления файлов

Директива «include» указывает препроцессору, что содержимое заданного файла необходимо обработать так, как если бы оно находилось в исходной программе в той точке, в которой располагается эта директива. Соответствующие заголовочные файлы и файлы исходного кода на этапе компиляции (построения) приложения обрабатываются как единый файл, например, Factorial.h и Factorial.cpp.

Добавим поле и прототипы методов в заголовочный файл класса CFactorial (Factorial.h).

```
class CFactorial
{
private:
    // Значение факториала
    long unsigned int answer;

public:
```

```

// Конструктор по умолчанию
CFactorial(void);
// Вычисление факториала в цикле
long unsigned int CountLoop(int number);
// Вычисление факториала рекурсивно
long unsigned int CountRecursion(int number);
// Вывод результата в текстовое поле вывода
void OutputAnswer(System::Windows::Forms::TextBox^ textBox);
};

```

Реализуем объявленные методы в файле исходного кода класса CFactorial (Factorial.cpp):

```

#include "stdafx.h"
#include "Factorial.h"

CFactorial::CFactorial(void)
{
}

long unsigned int CFactorial::CountLoop(int number)
{
    if(number < 0) // Факториал отрицательного числа равен 0
    {
        return answer = 0;
    }
    else // Расчитываем факториал положительного числа и 0
    {
        int result = 1;
        for(int i=2; i<=number; i++)
        {
            result *= i;
        }
        return answer = result;
    }
}

long unsigned int CFactorial::CountRecursion(int number)
{
    if(number == 0)
    {
        return answer = 1;
    }
}

```

```

    }
    else if (number < 0)
    {
        return answer = 0;
    }
    else
    {
        return answer = number * CountRecursion(number-1);
    }
}

void CFactorial::OutputAnswer(System::Windows::Forms::TextBox^ textBox)
{
    textBox->Text = answer.ToString();
}

```

Далее добавим кнопку («Button») на главную форму приложения так, чтобы он выглядел следующим образом (рис. 25).

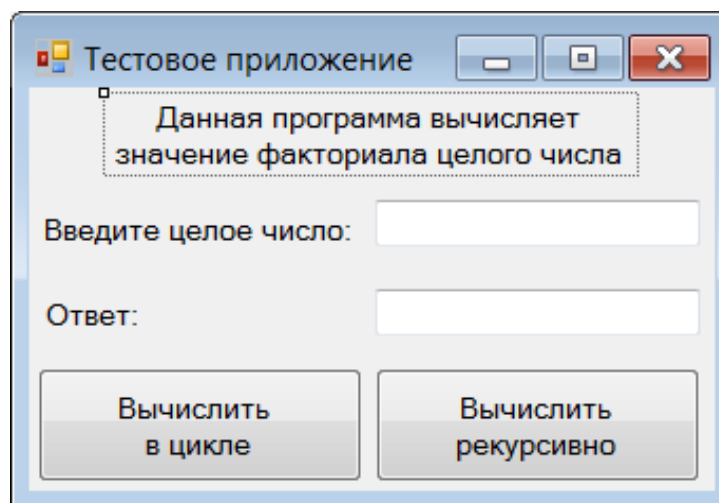


Рис. 25. Отредактированное главное окно приложения

Для того чтобы в классе формы главного окна можно было использовать объекты класса «CFactorial», необходимо указать директиву #include «Factorial.h».

Зададим следующие обработчики событий путем нажатия на кнопки:

```
//Вычисление циклом
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) {
    try {
        CFactorial* f = new CFactorial();
        f->CountLoop(System::Convert::ToInt32(textBox1->Text));
        f->OutputAnswer(this->textBox2);
    }
    catch(...)
    {
        MessageBox::Show("Ошибка вычисления факториала");
    }
}

//Рекурсивное вычисление
private: System::Void button2_Click(System::Object^ sender, System::EventArgs^ e) {
    try {
        CFactorial* f = new CFactorial();
        f->CountRecursion(System::Convert::ToInt32(textBox1->Text));
        f->OutputAnswer(this->textBox2);
    }
    catch(...)
    {
        MessageBox::Show("Ошибка вычисления факториала");
    }
}
```

Построим (скомпилируем) приложение в режиме «Release» (рис. 26).

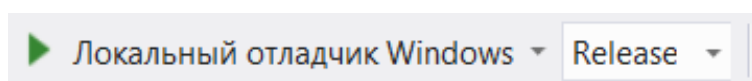


Рис. 26. Компиляция приложения в режиме «Release»

Скомпилированный исполняемый файл готового приложения находится в каталоге «Каталог_решения\Release\».

Задания

1. Создайте класс, который будет осуществлять расчет корней квадратного уравнения.
2. Добавьте прототипы методов в заголовочный файл класса.
3. Реализуйте объявленные методы в файле исходного кода класса.
4. Укажите соответствующую директиву, задайте обработчики событий нажатия на кнопки.
5. Скомпилируйте приложение в режиме «Release».

Контрольные вопросы

1. Как добавить класс в новый проект?
2. Какие действия необходимо выполнить, чтобы в классе формы главного окна можно было использовать объекты класса?

Литература

1. Введение в Windows Forms – пишем первую программу. – URL: <http://code-live.ru/post/first-windows-form/>
2. Getting Started with Windows Forms using Visual C++ Tutorial. – URL: <http://holowczak.com/getting-started-with-windows-forms-using-visual-c-tutorial/>
3. Начало работы с C++ в Visual Studio. – URL: <http://msdn.microsoft.com/ru-ru/library/jj620919.aspx>
4. Дальнейшее обучение (C++). – URL: [http://msdn.microsoft.com/ru-ru/library/vstudio/bb384836\(v=vs.100\).aspx](http://msdn.microsoft.com/ru-ru/library/vstudio/bb384836(v=vs.100).aspx)
5. Microsoft. Microsoft Visual Studio 2012 Product Guide. – URL: <http://www.microway.com.au/microsoft/Visual-Studio-2012-Product-Guide.pdf>

Содержание

Введение	3
1. Создание консольного приложения.....	4
Задания	9
Контрольные вопросы	12
 2. Начало работы в интегрированной среде разработки, выбор панелей инструментов.....	 12
Задания	14
Контрольные вопросы	14
 3. Создание проекта и решения	 14
Задания	19
Контрольные вопросы	21
 4. Выполнение и отладка приложения.....	 21
Задания	27
Контрольные вопросы	27
 5. Создание класса	 27
Задания	34
Контрольные вопросы	34
Литература	34

Корректурa: *Н.А. Бачурина*
Изготовление оригинала-макета: *Т.С. Камыгина*
Лицензия А № 165724 от 11.04.06 г.

Подписано к печати 22.12.14. Тир. 4.
Уч.-изд. л. 2,17. Усл. п. л. 2,09. Формат 60 × 84 ¹/₁₆.
Гарнитура Таймс. Зак.

ФГБОУ ВПО «Череповецкий государственный
университет»
162600 г. Череповец, пр. Луначарского, д. 5.