УВАЖАЕМЫЕ СТУДЕНТЫ! Изучите теоретические сведения к лабораторной работе, выполните пример и задание согласно вашему варианту.

Результаты работы, отчет, предоставить преподавателю на e-mail: r.bigangel@gmail.com до 06.02.2023.

Требования к отчету:

Отчет предоставляется преподавателю в электронном варианте и должен содержать:

- **название работы, постановку цели, вывод;**
- **ответы на контрольные вопросы, указанные преподавателем.**

При возникновении вопросов по приведенному материалу обращаться по следующему номеру телефона: (072)111-37-59, (Viber, WhatsApp), vk.com: https://vk.com/daykini

ВНИМАНИЕ!!! При отправке работы, не забывайте указывать ФИО студента, наименование дисциплины, дата проведения занятия (по расписанию).

Работа рассчитана на 4 часа, то есть, работа выполняется на 1-2 паре!!!

Лабораторная работа № 2

Тема: «Стандартный ввод-вывод»

Цель: научиться использовать среду Visual Studio 2010 для создания простейших программ по программированию на языке C++.

Теоретическая часть

Язык **C** (читается как Си) в основе своей был создан в 1972 г. как язык для операционной системы UNIX. Автором этого языка считается Деннис М. Ритчи (DENNIS M. RITCHIE).

Популярность языка **C** обусловлена, прежде всего тем, что большинство операционных систем были написаны на языке C. Его начальное распространение было задержано из-за того, что не было удачных компиляторов.

Несколько лет не было единой политики в стандартизации языка **C**. В начале 1980-х годов в Американском национальном институте стандартов (ANSI) началась работа по стандартизации языка **C**. В 1989 г. работа комитета по языку C была ратифицирована, и в 1990 г. был издан первый официальный документ по стандарту языка **C**. Появился стандарт 1989.

К разработке стандарта по языку **C** была также привлечена Международная организация по стандартизации (ISO). Появился стандарт ISO/IEC 9899:1990, или ANSI C99 языка **C**.

В данном пособии за основу принимается стандарт языка **C** от 1989 г. и написание программ будет выполняться в среде разработки Visual Studio 2010.

Язык **C** является прежде всего языком высокого уровня, но в нем заложены возможности, которые позволяют программисту (пользователю) работать непосредственно с аппаратными средствами компьютера и общаться с ним на достаточно низком уровне. Многие операции, выполняемые на языке **C**, сродни языку Ассемблера. Поэтому язык C часто называют языком среднего уровня.

Для написания программ в практических разделах данного учебного пособия будет использоваться компилятор языка **C++**, а программирование будет вестись в среде **Microsoft Visual Studio 2010**. Предполагается, что на компьютере установлена эта интегрированная среда.

Microsoft Visual Studio 2010 доступна в следующих вариантах:

Express – бесплатная среда разработки, включающая только базовый набор возможностей и библиотек.

- **Professional** – поставка, ориентированная на профессиональное создание программного обеспечения, и командную разработку, при которой созданием программы одновременно занимаются несколько человек.
- **Premium** – издание, включающее дополнительные инструменты для работы с исходным кодом программ и создания баз данных.
- **Ultimate** – наиболее полное издание Visual Studio, включающие все доступные инструменты для написания, тестирования, отладки и анализа

программ, а также дополнительные инструменты для работы с базами данных и проектирования архитектуры ПО.

Отличительной особенностью среды **Microsoft Visual Studio 2010** является то, что она поддерживает работу с несколькими языками программирования и программными платформами. Поэтому перед тем, как начать создание программы на языке С, необходимо выполнить несколько подготовительных шагов по созданию проекта и выбору, и настройке компилятора языка С для трансляции исходного кода.

После запуска **Microsoft Visual Studio 2010** появляется следующая стартовая страница, которая показана на рис. 1.1.

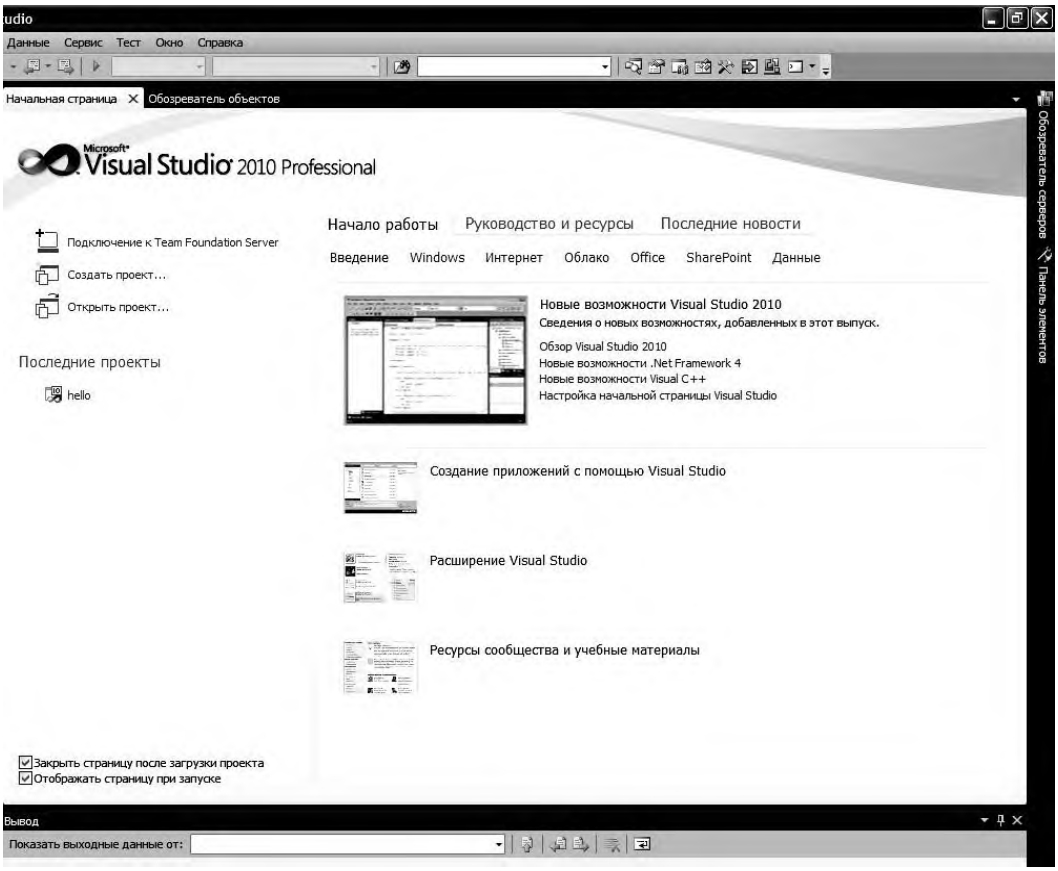

Рисунок 1.1 – Стартовая страница Visual Studio 2010

Следующим шагом является создание нового проекта. Для этого в меню **Файл** необходимо выбрать. **Создать →Проект** (или комбинацию клавиш **Ctrl + Shift + N**). Результат выбора пунктов меню для создания нового проекта показан на рис. 1.2.

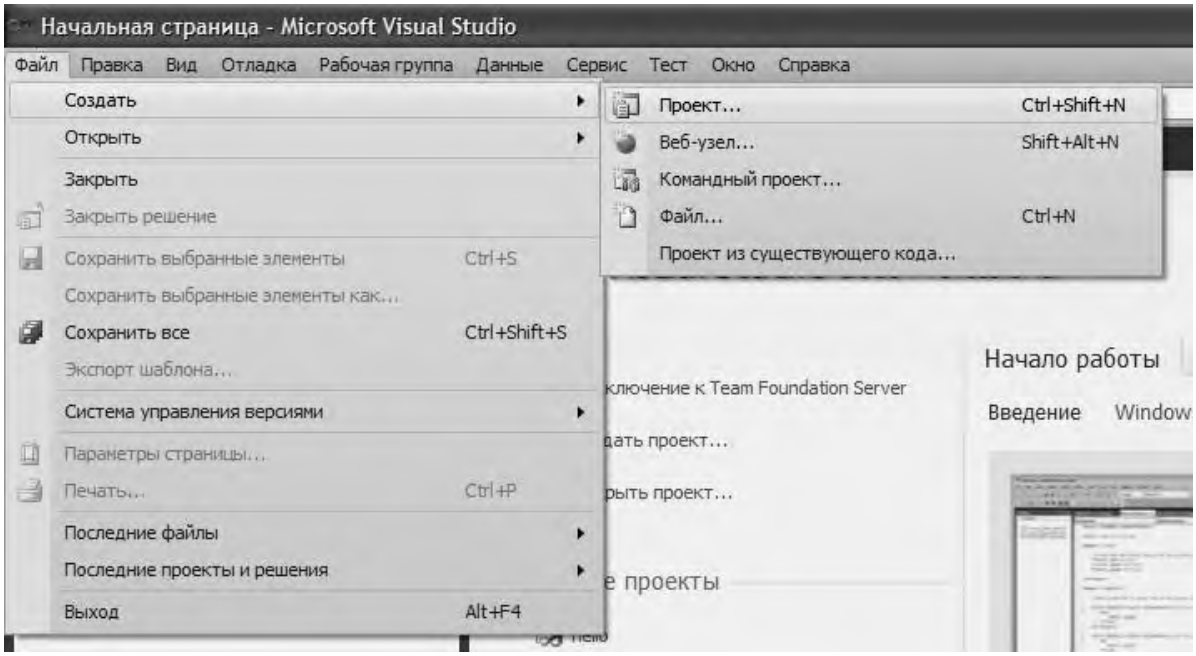

Рисунок 1.2 – Окно с выбором нового проекта

Среда Visual Studio отобразит окно. **Создать Проект**, в котором необходимо выбрать тип создаваемого проекта. Проект используется в Visual Studio для логической группировки нескольких файлов, содержащих исходный код, на одном из поддерживаемых языков программирования, а также любых вспомогательных файлов. Обычно после сборки проекта (которая включает компиляцию всех всходящих в проект файлов исходного кода) создается один исполняемый модуль.

В окне **Создать Проект** следует развернуть узел Visual C++, обратиться к пункту Win32 и на центральной панели выбрать *Консольное приложение Win32*. Выбор этой опции показан на рис. рис. 1.3.

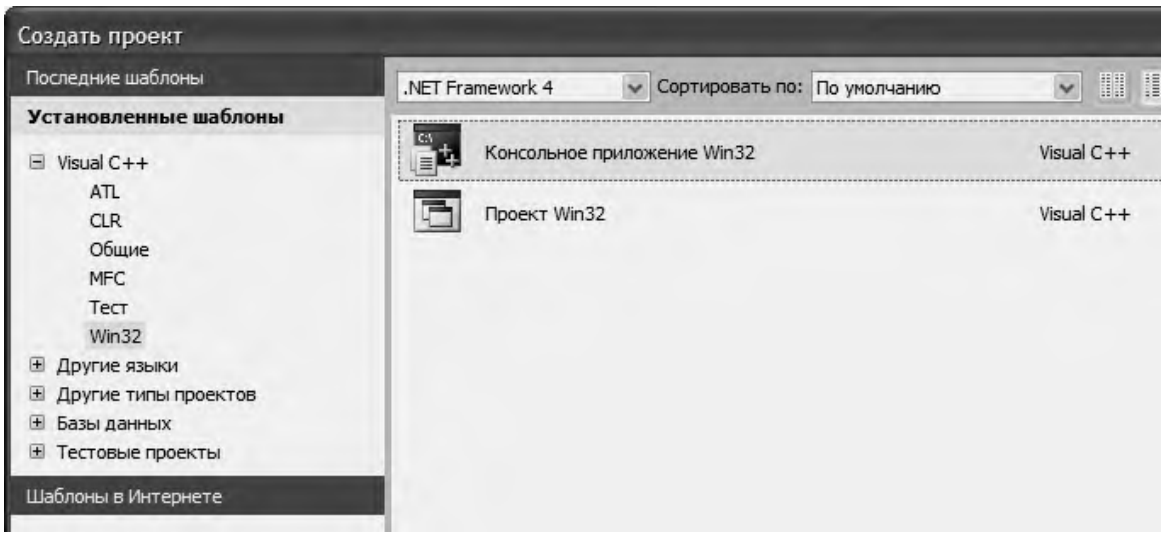

Рисунок 1.3 – Выбор типа проекта

Затем в поле редактора **Имя** (где по умолчанию имеется <Введите имя>) следует ввести имя проекта, например **hell**. В поле **Расположение** можно указать путь размещения проекта, или выбрать путь размещения проекта с помощью клавиши (кнопки) Обзор. По умолчанию проект сохраняется в специальной папке Projects. Пример выбора имени проекта по- казано на рис. 1.4.

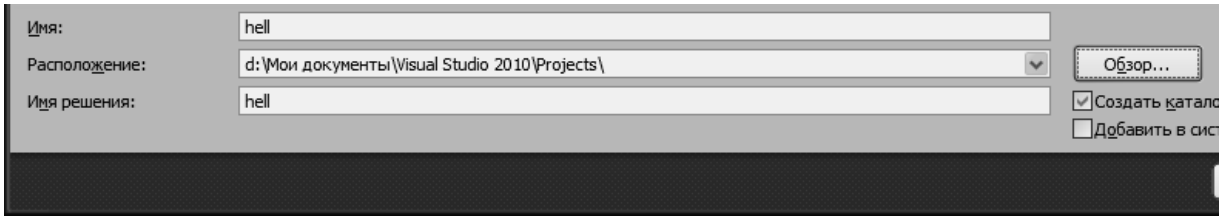

Рисунок 1.4 – Пример задания имени проекта

Одновременно с созданием проекта Visual Studio создает решение. Решение (solution) – это способ объединения нескольких проектов для организации более удобной работы с ними.

После нажатия кнопки **OK** откроется окно **Мастер приложений Win32**, показанное на рис. 1.5.

Выбор имени проекта может быть достаточно произвольным: допустимо использовать числовое значение, допустимо имя задавать через буквы

русского алфавита.

В дальнейшем будем использовать имя, набранное с помощью букв латинского алфавита и, может быть, с добавлением цифр.

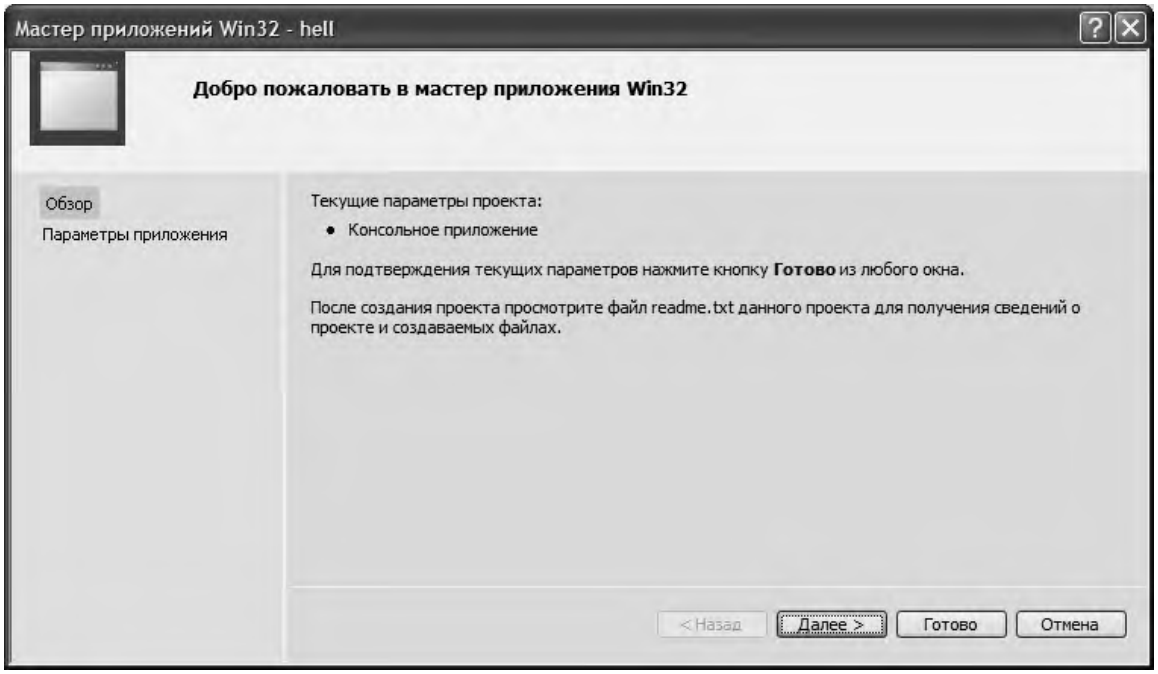

Рисунок 1.5 – Мастер создания приложения

На первой странице представлена информация о создаваемом про- екте, на второй можно сделать первичные настройки проекта. После об- ращения к странице **Параметры приложения**, или после нажатия кнопки **Далее** получим окно, показанное на рис. рис. 1.6.

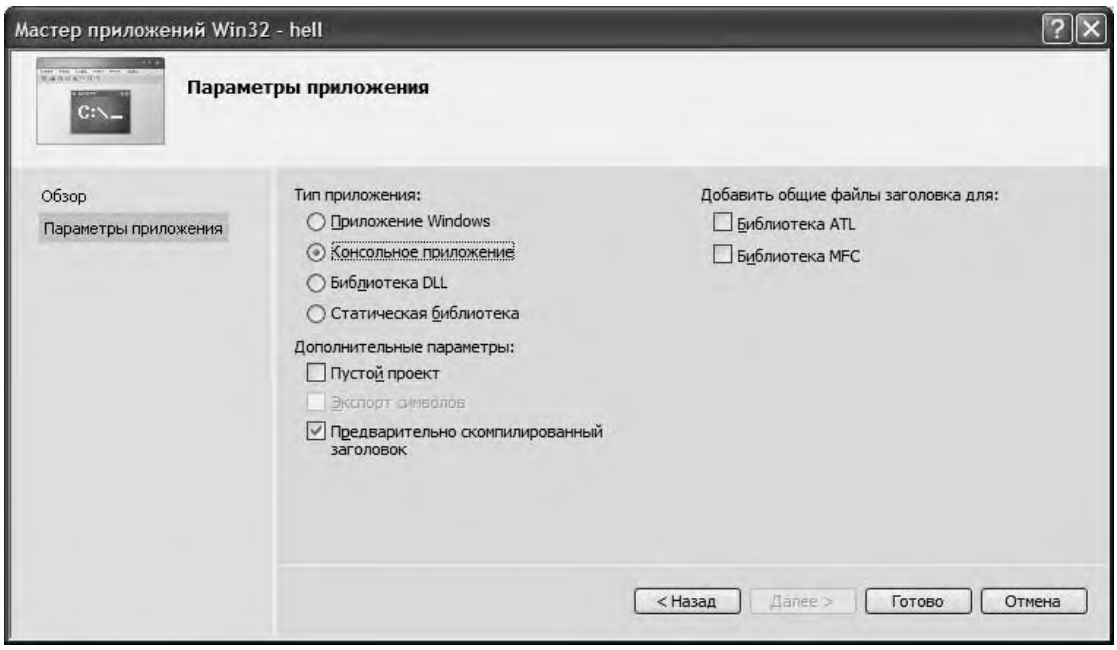

Рисунок 1.6 – Страница мастера настройки проекта по умолчанию

В дополнительных опциях (**Параметры приложения**) следует поставить галочку в поле **Пустой проект** и убрать галочку в поле **Предварительно скомпилированный заголовок**. Получим экранную форму, показанную на рис. 1.7.

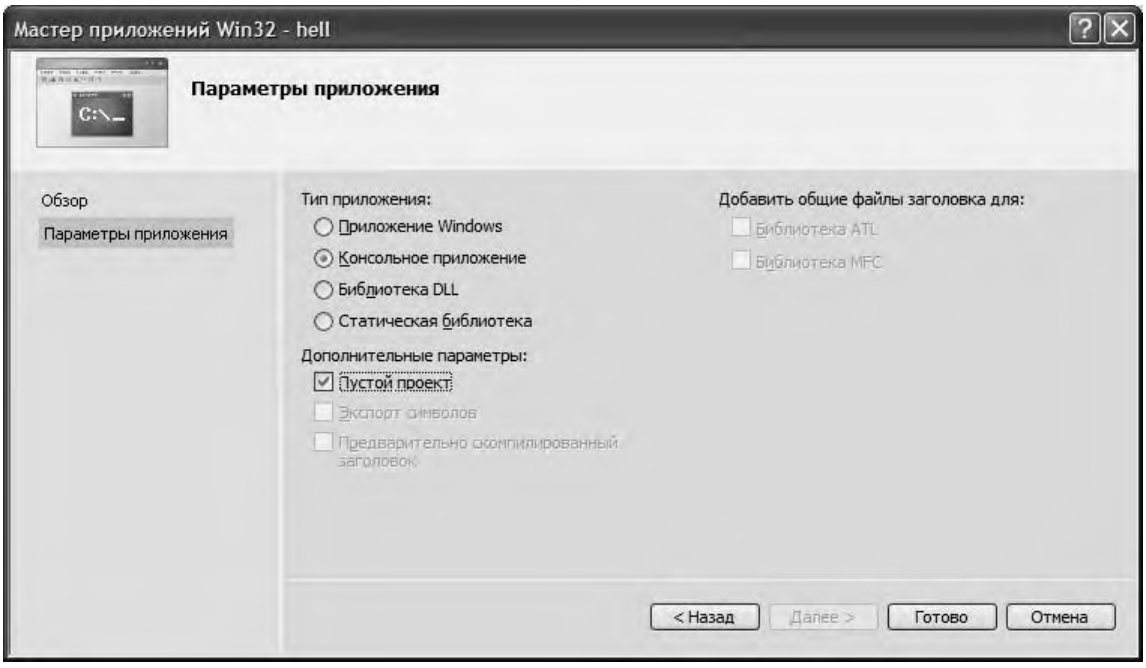

Рисунок 1.7 – Выполненная настройка мастера приложений

Здесь и далее будут создавать проекты по приведенной схеме, т.е.

проекты в консольном приложении, которые должны создаваться целиком программистом (за счет выбора **Пустой проект**). После нажатия кнопки **Готово**, получим экранную форму, показанную на рис. 1.8, где приведена последовательность действий добавления файла для создания исходного кода к проекту. Стандартный путь для этого: подвести курсор мыши к пункту **Проект**, выбрать **Добавить новый элемент**.

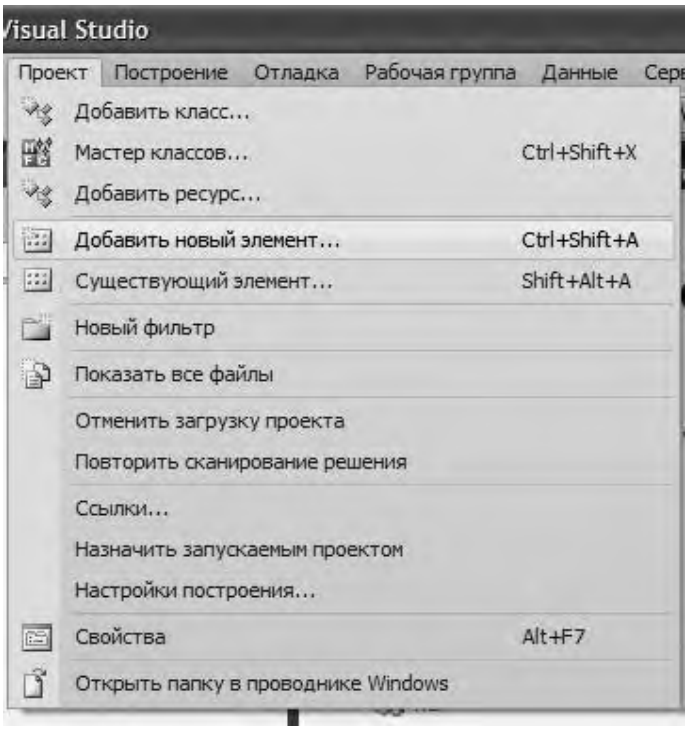

Рисунок 1.8 – Меню добавления нового элемента к проекту

После выбора (нажатия) **Новый элемент** получим окно, показанное на рис. 1.9, где через пункт меню **Код** узла **Visual C++** выполнено обращение к центральной части панели, в которой осуществляется выбор типа файлов. В данном случае требуется обратиться к закладке **C++ File** (*.cpp).

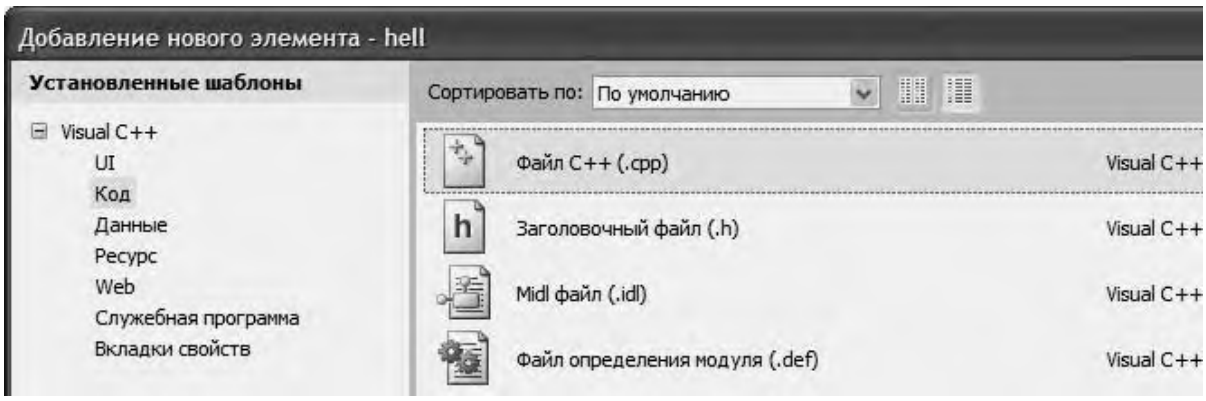

Рисунок 1.9 – Окно выбора типа файла для подключения к проекту

Теперь в поле редактора **Имя** (в нижней части окна) следует задать имя нового файла и указать расширение **".c"**. Например, **main.c**. Имя файла может быть достаточно произвольным, но имеется негласное соглашение, что имя файла должно отражать его назначение и логически описывать исходный код, который в нем содержится. В проекте, состоящем из нескольких файлов, имеет смысл выделить файл, содержащий главную функцию программы, с которой она начнет выполняться. В данном пособии такому файлу мы будем задавать имя **main.c**, где расширение **.c** указывает на то, что этот файл содержит исходный код на языке **C**, и он будет транслироваться соответствующим компилятором. Программам на языке **C** принято давать расширение **.c**. После задания имени файла в поле редактора **Name** получим форму, показанную на рис. 1.10.

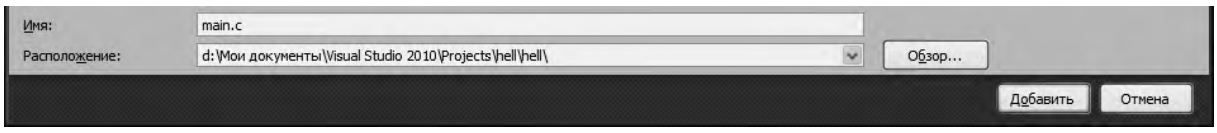

Рисунок 1.10 – Задание имени файла, подключаемому к проекту

Затем следует нажать кнопку **Добавить**. Вид среды Visual Studio после добавления первого файла к проекту показан на рис. 1.11. Для добавленного файла автоматически открывается редактор.

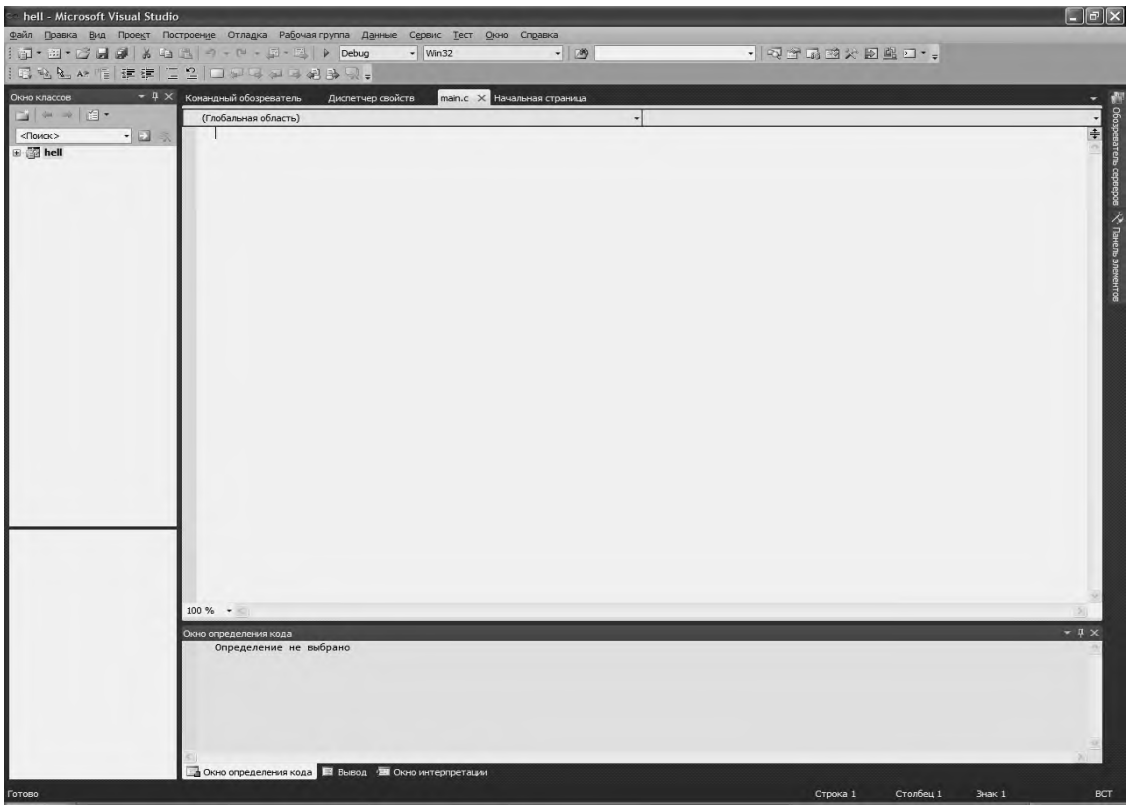

Рисунок 1.11 – Подключение файла проекта

В папке проекта отображаются файлы, включенные в проект в папках. Приведем описание.

Папка **Файлы исходного кода** предназначена для файлов с исходным кодом. В этой папке отображаются файлы с расширением **.c**.

Папка **Заголовочные файлы** содержит заголовочные файлы с расширением **.h.**

Папка **Файлы ресурсов** содержит файлы ресурсов, например изображения и т. д.

Папка **Внешние зависимости** отображает файлы, не добавленные явно в проект, но использующиеся в файлах исходного кода, например включенные при помощи директивы #include. Обычно в папке **Внешние зависимости** присутствуют заголовочные файлы стандартной библиотеки, использующиеся в проекте.

Следующий шаг состоит в настройке проекта. Для этого в меню **Проект** главного меню следует выбрать **Свойства hell** (или с помощью

последовательного нажатия клавиш Alt+F7). Пример обращения к этому пункту меню показан на рис. 1.12.

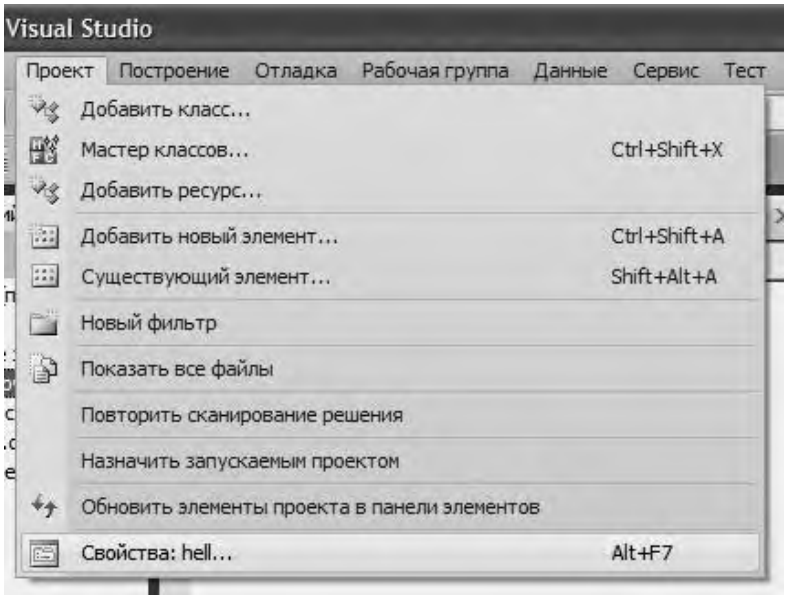

Рисунок 1.12 – Обращение к странице свойств проекта

После того как произойдет открытие окна свойств проекта, следует обратиться (с левой стороны) к **Свойства конфигурации**. Появится ниспадающий список, который показан на рис. 1.13. Выполнить обращение к узлу **Общие**, и через него в левой панели выбрать **Набор символов**, где установить свойство **Использовать многобайтовую кодировку**. Настройка **Набор символов** позволяет выбрать, какая кодировки символов – ANSI или UNICODE – будет использована при компиляции программы. Для совместимости со стандартом C89 мы выбираем **Использовать многобайтовую кодировку**. Это позволяет использовать многие привычные функции, например функции по выводу информации на консоль.

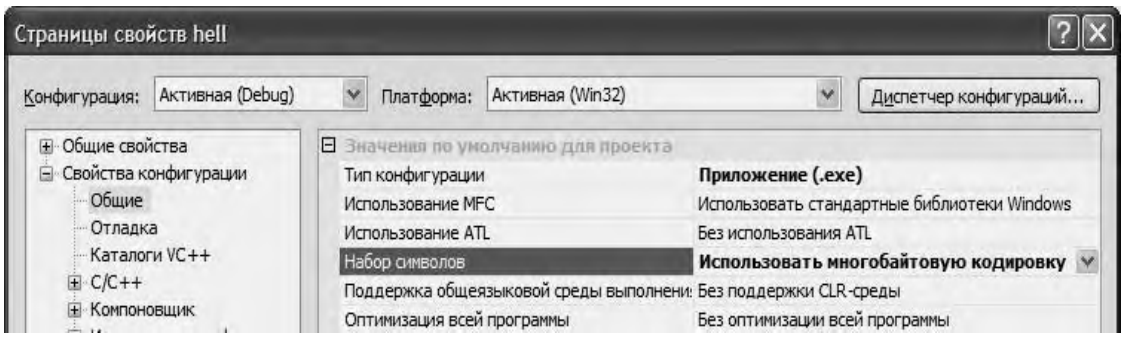

Рисунок 1.13 – Меню списка свойств проекта

После сделанного выбора, показанного на рис. 1.13, следует нажать кнопку Применить. Затем следует выбрать узел **C/C++** и в ниспадающем меню выбрать пункт **Создание кода**, через который следует обратиться в правой части панели к закладке **Включить C++ исключения**, для которой установить **Нет** (запрещение исключений C++). Результат установки выбранного свойства показан на рис. 1.14. После произведенного выбора нажать кнопку **Применить**.

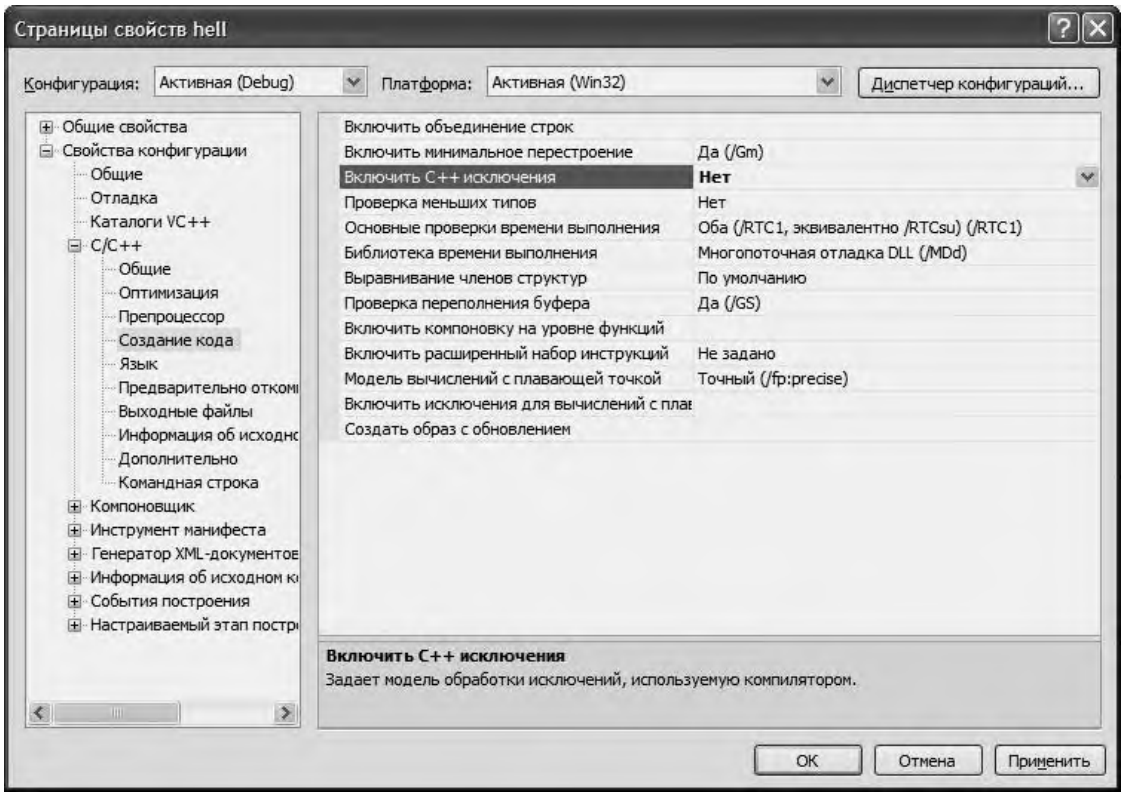

Рисунок 1.14 – Страница свойств для запрещения исключений C++

Далее в ниспадающем меню узла **C/C++** необходимо выбрать пункт **Язык** и через него обратиться в правую часть панели, где установить следующие свойства: свойство **Отключить расширение языка** (дополнительные языковые расширения фирмы Microsoft) в **Да (/Za)**, свойство **Считать wchar_t встроенным типом** установить в **Нет (/Zc:wchar_t–)**, свойство **Обеспечение согласования видимости переменных, объявленных в заголовке оператора цикла for** установить в да(/Zc:forScope), свойство **Включить информацию о типах время выполнения** установить в **Нет (/GR–)**, свойство **Поддержка Open MP** (разрешить расширение Open MP – используется при написании программ для многопроцессорных систем) установить в **Нет(/openmp–)**.

Результат выполнения этих действий показан на рис. 1.15.

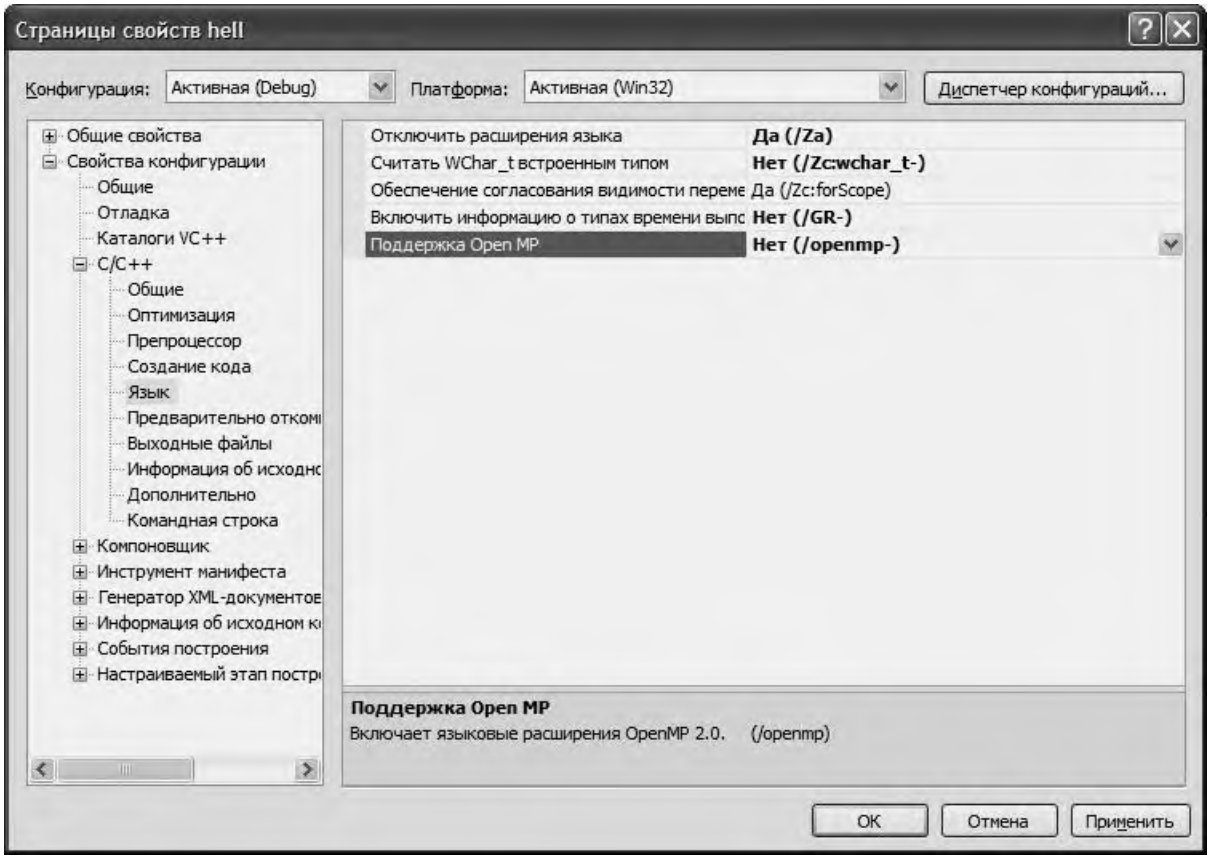

Рисунок 1.15 – Страница свойств закладки Язык

После выполнения указанных действий следует нажать клавишу **Применить**. Далее в ниспадающем списке узла **C/C++** следует выбрать пункт

Дополнительно и в правой панели изменить свойство **Компилировать как** в свойство компиляции языка **C**, т.е. **Компилировать как код C (/TC)**. Результат установки компилятора языка **C** показан на рис. 1.16.

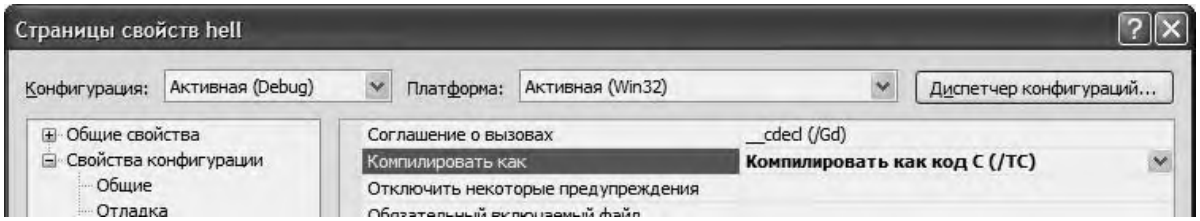

Рисунок 1.16 – Результат выбора режима компиляции языка C

После нажатия клавиш **Применить** и **ОК** сначала откроется подготовленный проект с пустым полем редактора кода, в котором можно начать писать программы. В этом редакторе наберем программу, выводящую традиционное приветствие "Hello World". Для компиляции созданной программы можно обратиться в меню **Построение**, или, например, набрать клавиши Ctr+F7. В случае успешной компиляции получим следующую экранную форму, показанную на рис. 1.17.

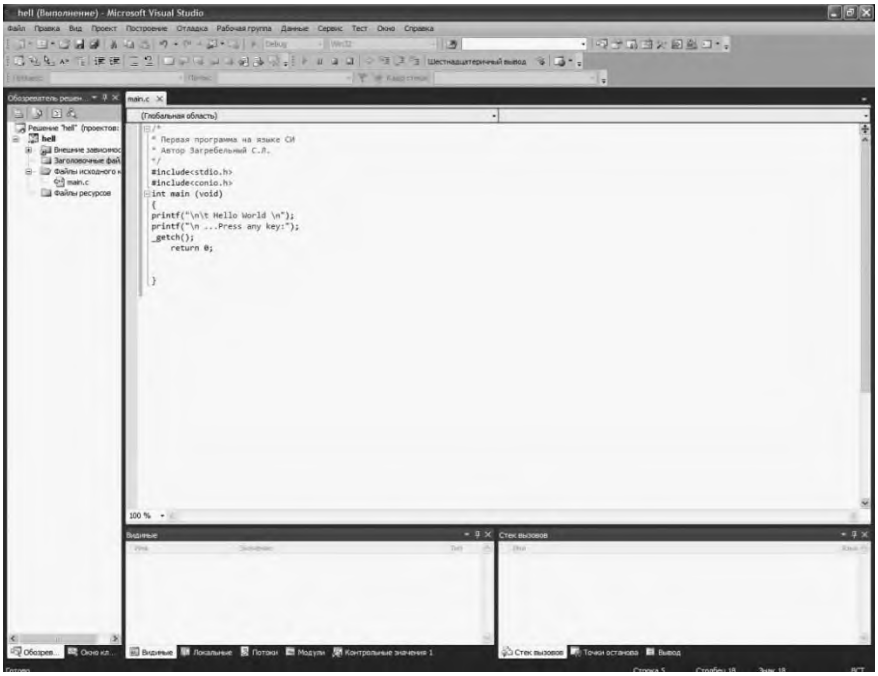

Рисунок 1.17 – Успешно откомпилированная первая программа на языке C

Для приведенного кода программы запуск на ее исполнение из окна редактора в Visual Studio 2010 можно нажать клавишу **F5**. На рис. 1.18 показан результат исполнения первой программы.

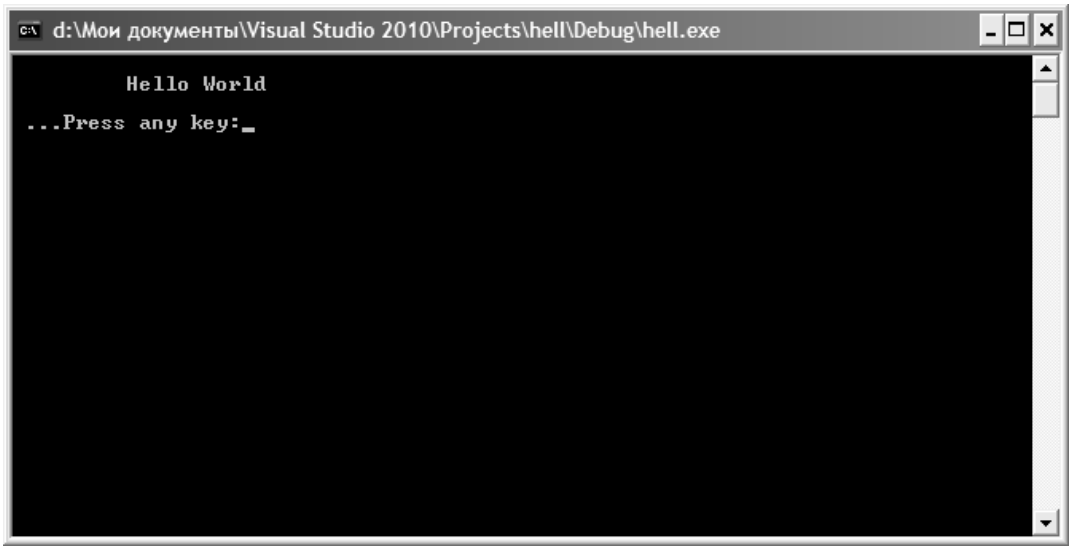

Рисунок 1.18 – Консольный вывод первой программы на языке С

Примечание. Вывод требуемой информации осуществляется с помощью букв латинского алфавита. Комментарии в программе могут быть сделаны после символа "//" или внутри комбинации символов "/* */".

Произведем разбор первой программы. Во-первых, надо отметить, что в языке **С** нет стандартных инструкций (операторов) для вывода сообщений на консоль (окно пользователя). В языке **С** предусматриваются специальные библиотечные файлы, в которых имеются функции для этих целей. В приведенной программе используется заголовочный файл с именем stdio.h (стандартный ввод–вывод), который должен быть включен в начало программы. Для вывода сообщения на консоль используется функция printf(). Для работы с консолью включен также заголовочный файл conio.h., который поддерживает функцию _getch(), которая извлекает символ из потока ввода, т. е. она предназначена для приема сообщения о нажатии какой-либо (почти любой) клавиши на клавиатуре. С другими компиляторами, возможно, потребуется getch(), т. е. без префиксного нижнего подчеркивания. Строка

программы ***int main (void)***

сообщает системе, что именем программы является main() – главная функция, и что она возвращает целое число, о чем указывает аббревиатура **"int"**. Имя main() – это специальное имя, которое указывает, где программа должна начать выполнение. Наличие круглых скобок после слова main() свидетельствует о том, что это имя функции. Если содержимое круглых скобок отсутствует или в них содержится служебное слово **void**, то это означает, что в функцию main() не передается никаких аргументов. Тело функции main() ограничено парой фигурных скобок. Все утверждения программы, заключенные в фигурные скобки, будут относиться к функции main().

В теле функции main() имеются еще три функции. Во-первых, функции printf() находятся в библиотеке компилятора языка **C**, и они печатают или отображают те аргументы, которые были подставлены вместо параметров. Символ "\n" составляет единый символ newline (новая строка), т.е. с помощью этого символа осуществляется перевод на новую строку. Символ "\t" осуществляет табуляцию, т.е. начало вывода результатов программы с отступом вправо.

Функция без параметров _getch() извлекает символ из потока ввода (т.е. ожидает нажатия почти любой клавиши). С другими компиляторами, возможно, потребуется getch(), т.е. без префиксного нижнего подчеркивания.

Последнее утверждение в первой программе ***return 0;***

указывает на то, что выполнение функции main() закончено и что в систему возвращается значение 0 (целое число). Нуль используется в соответствии с соглашением об индикации успешного завершения программы.

В завершение следует отметить, что все действия в программе завершаются символом точки с запятой.

Все файлы проекта сохраняются в той папке, которая сформировалась после указания в поле Location имени проекта (hell). На рис. 1.19 показаны папки и файлы проекта Visual Studio 2010.

\

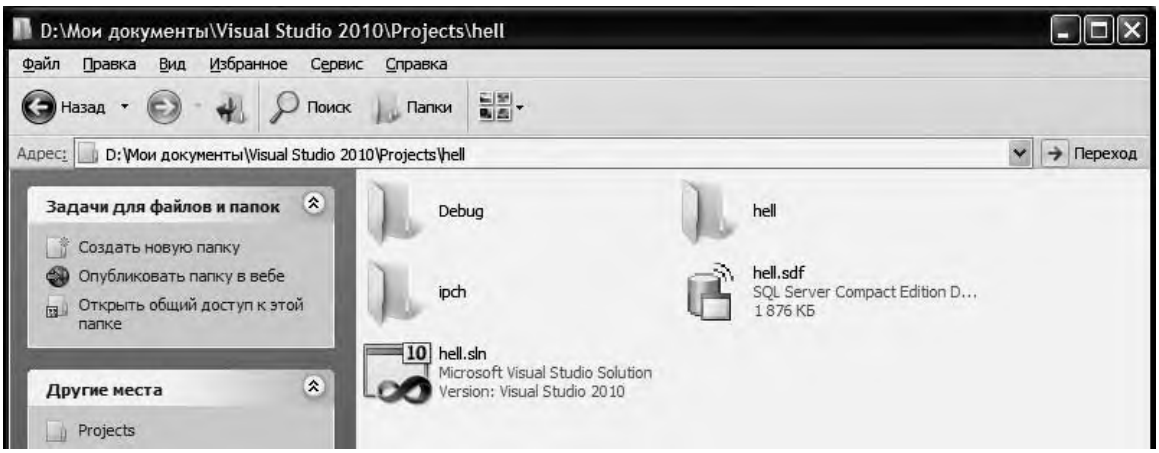

Рисунок 1.19 – Файлы и папки созданного проекта

На рис. 1.19 файлы с полученными расширениями означают:

hell.sln – файл решения для созданной программы. Он содержит информацию о том, какие проекты входят в данное решение. Обычно, эти проекты расположены в отдельных подкаталогах. Например, наш проект находится в подкаталоге hell;

hell.suo – файл настроек среды Visual Studio при работе с решением, включает информацию об открытых окнах, их расположении и прочих пользовательских параметрах.

hell.sdf – файл, содержащий вспомогательную информацию о проекте, который используется инструментами анализа кода Visual Studio, такими как IntelliSense для отображения подсказок об именах и т. д.

Файлы папки **Debug** показаны на рис. 1.20.

Рисунок 1.20 – Файлы папки Debug

Рассмотрим файлы в соответствии с рис. 1.20.

hell.exe – исполняемый файл проекта;

hell.ilk – файл "incremental linker", используемый компоновщиком для ускорения процесса компоновки;

hell.pdb – отладочная информация/информация об именах в исполняемых файлах, используемая отладчиком.

Файлы папки **hell** показаны на рис. 1.21.

Рисунок 1.21 – Содержимое папки hell

Характеристика содержимого папки **hell**: **main.c** – файл исходного программного кода, **hell.vcxproj** – файл проекта,

hell.vcxproj.user – файл пользовательских настроек, связанных с проектом,

hell.vcxproj.filters – файл с описанием фильтров, используемых Visual Studio Solution Explorer для организации и отображения файлов с исходным кодом.

Для настройки Visual Studio 2015:

https://www.youtube.com/watch?v=NEuUxy4Skpc&t=56s&ab_channel=%23SimpleCode

Задание к лабораторной работе

Разобраться с программой Visual Studio 2010 и суметь создать консольную программу по выше показанному примеру.

Ответить на контрольные вопросы.

Контрольные вопросы

1 Назовите компиляторы языка **C**?

2 Какое имя имеет исполняемый файл созданного проекта?

3 Объясните назначение заголовочных файлов stdio.h, conio.h.

4 Как будет работать программа без заголовочного файла conio.h?

5 В каком месте программы находится точка ее входа?

6 Как осуществляется табуляция строки на консоли и на сколько позиций выполняется отступ от левого края?

7 Какое значение имеет главная функция проекта main() в программах на языке **C**?