

# **Лабораторная работа №2. Интегрированная среда разработки Microsoft Visual C++ 2010. Создание простейшего приложения**

## **1 Цель и порядок работы**

Цель работы – ознакомиться со средой разработки Microsoft Visual C++ 2010, научиться создавать, компилировать и отлаживать приложения, разобраться со структурой программы на языке C++.

Порядок выполнения работы:

- ознакомиться с описанием лабораторной работы;
- изучить возможности среды разработки Microsoft Visual C++;
- получить задание у преподавателя;
- написать программу, отладить и решить ее на ЭВМ;
- оформить отчет.

## **2 Краткая теория**

### **2.1 Введение**

Среда разработки **Microsoft Visual Studio** – это набор инструментов и средств, предназначенных для разработчиков программ, с широким набором поддерживаемых языков программирования. Visual C++ является частью Visual Studio. Интегрированную среду разработки (Integrated Development Environment, IDE) Visual Studio используют и другие средства разработки, например Microsoft C#.

Назначение среды разработки программ ясно следует из ее названия. Естественно, что любая программа сначала должна быть спроектирована, затем переложена на выбранный разработчиком для ее реализации язык программирования, после чего этот средствами конкретного языка программирования подвергается преобразованию в код, понятный микропроцессору. Сам по себе микропроцессор, "не знает" ни одного из известных нам языков программирования, кроме языка машинных кодов.

Выделим основные этапы создания программы:

1. Разработка (проектирование).
2. Выбор языка программирования.
3. Написание текста программы.
4. Перевод в язык машинных кодов (компиляция).
5. Отладка.
6. Выполнение.

О неготовности приложения к выполнению на ЭВМ Вы узнаете из предупреждений об ошибках на этапе трансляции, или же, что гораздо неприятнее, из результатов работы программы, которые могут быть совершенно неожиданными. При этом неизбежно использование средств для отладки программы.

После того, как выбран язык программирования и среда разработки, программист полностью зависит от средств языка и возможностей среды. На этапе написания программы удобно пользоваться какими-то справочными средствами по языку. И среда Visual Studio, предоставляет средства контекстной помощи, а также программу-редактор, с возможностью автодополнения и подсказкой параметров вызовов функций. Кроме этого в комплект входят компилятор и отладчик – это наиболее важные средства среды разработки.

## 2.2 Описание среды разработки Microsoft Visual Studio

Представленное ниже описание среды основано на версии Microsoft Visual C++ 2010 Professional. В других вариантах поставки некоторые варианты названия пунктов меню и элементов могут отличаться.

Запуск среды осуществляется через соответствующий пункт меню «Пуск». При запуске на экране появятся окно среды разработки (Рисунок 1.1), так называемая «Начальная страница» (Start Page), которая позволяет получить быстрый доступ к наиболее часто используемым возможностям, таким как, например, открытие недавно созданных проектов или создание нового проекта.

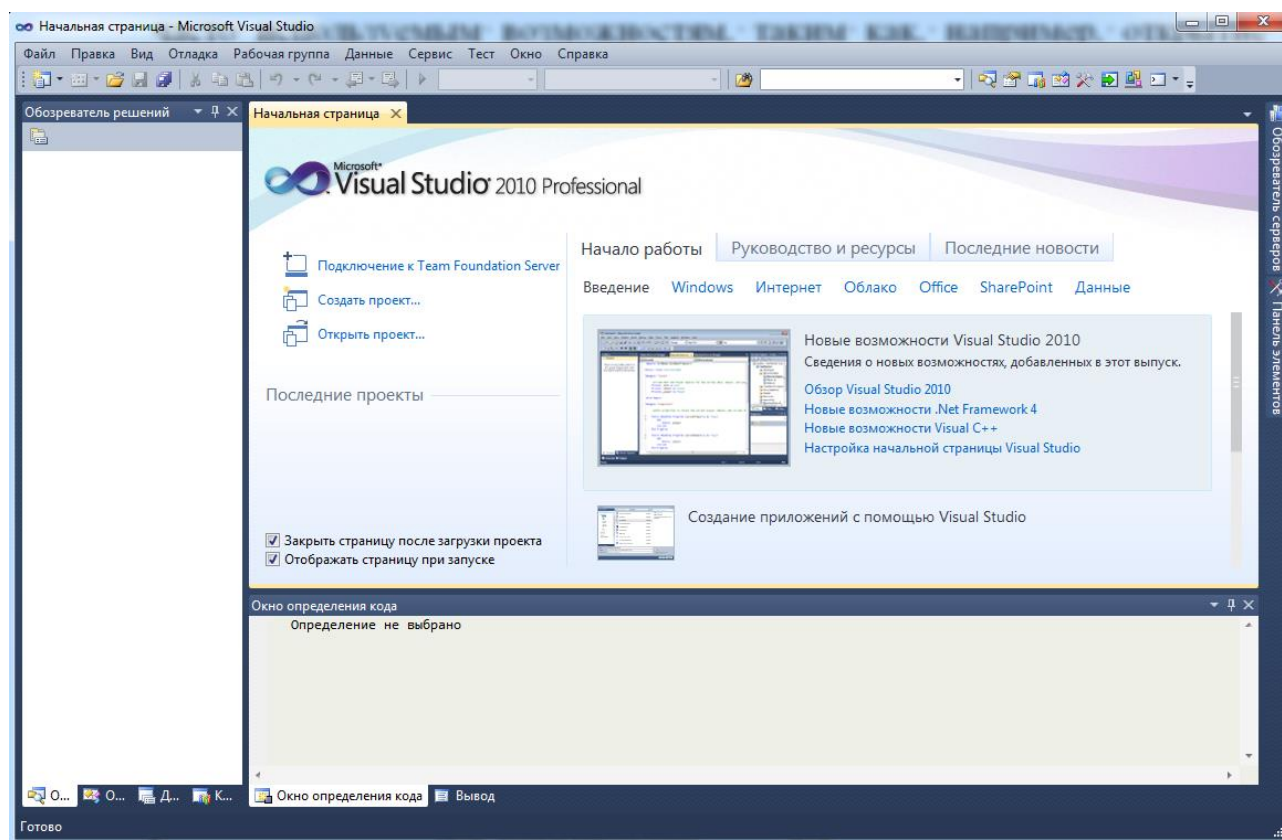


Рисунок 1.1 – Основное окно среды разработки

Для начала работы необходимо создать новый проект. В нашем случае это будет консольное приложение для платформы Win32, выводящее на экран сообщение “Hello World!”. (Win32 – это 32-х битное приложение для ОС Windows)

Для создания проекта нужно выбрать соответствующий пункт меню среды разработки (File -> New -> Project) или нажать комбинацию клавиш <Ctrl+Shift+N>. При этом появится диалоговое окно New Project, позволяющее создать все типы проектов Visual Studio.

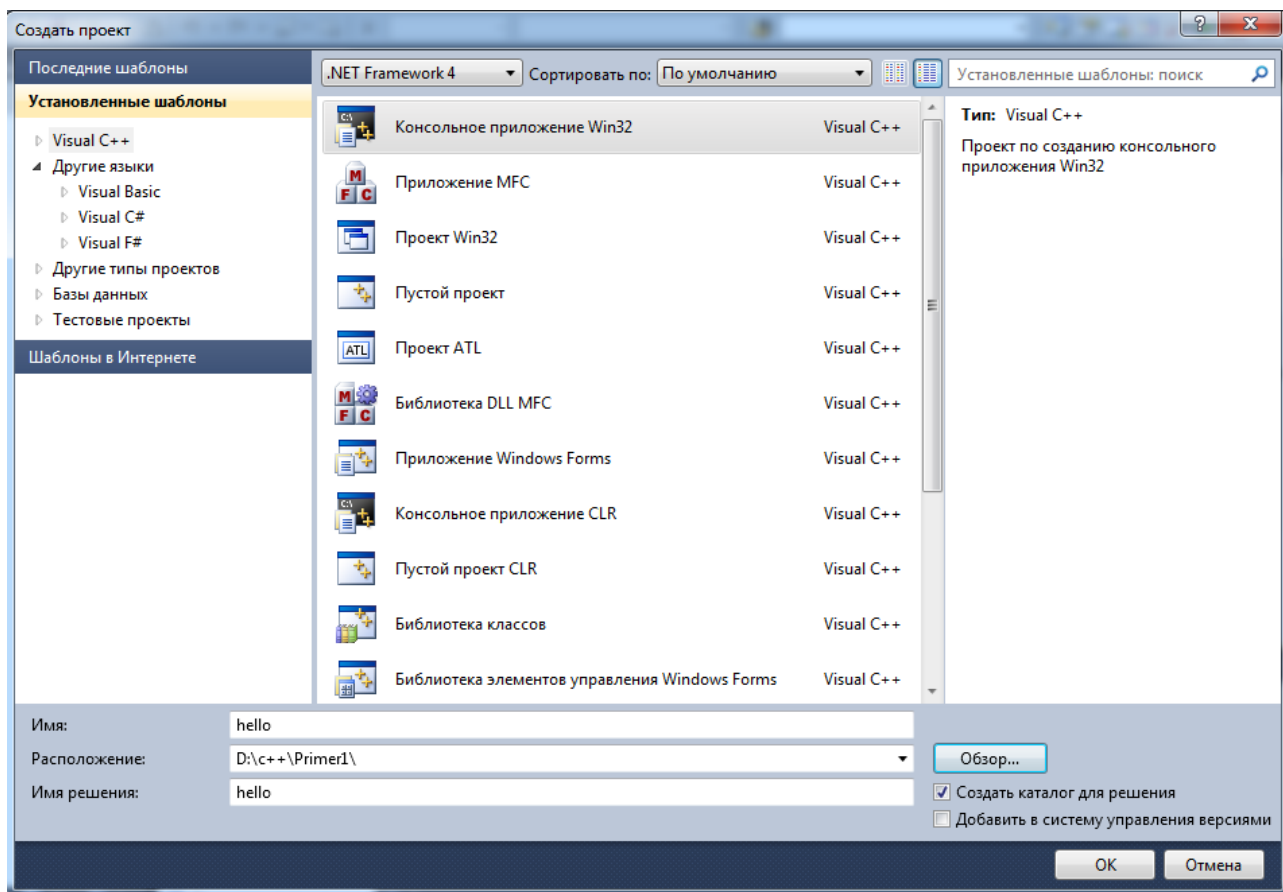


Рисунок 1.2 – Окно New Project

В начале необходимо выбрать тип проекта. В данном случае нам необходимо создать проект Visual C++. Далее выбираем необходимый подтип (Win32) и шаблон приложения (Win32 Console Application). После этого вводим имя приложения (hello), проверяем месторасположение каталога с файлами проекта (или изменяем его при помощи кнопки “Browse...”). Далее нажимаем “OK”, и попадаем в окно мастера создания приложений (Application Wizard), представленного на рисунке 1.3. Нажмем “Next >” для уточнения параметров приложения. На рисунке 1.4 представлены возможные варианты модификации параметров приложения Win32.

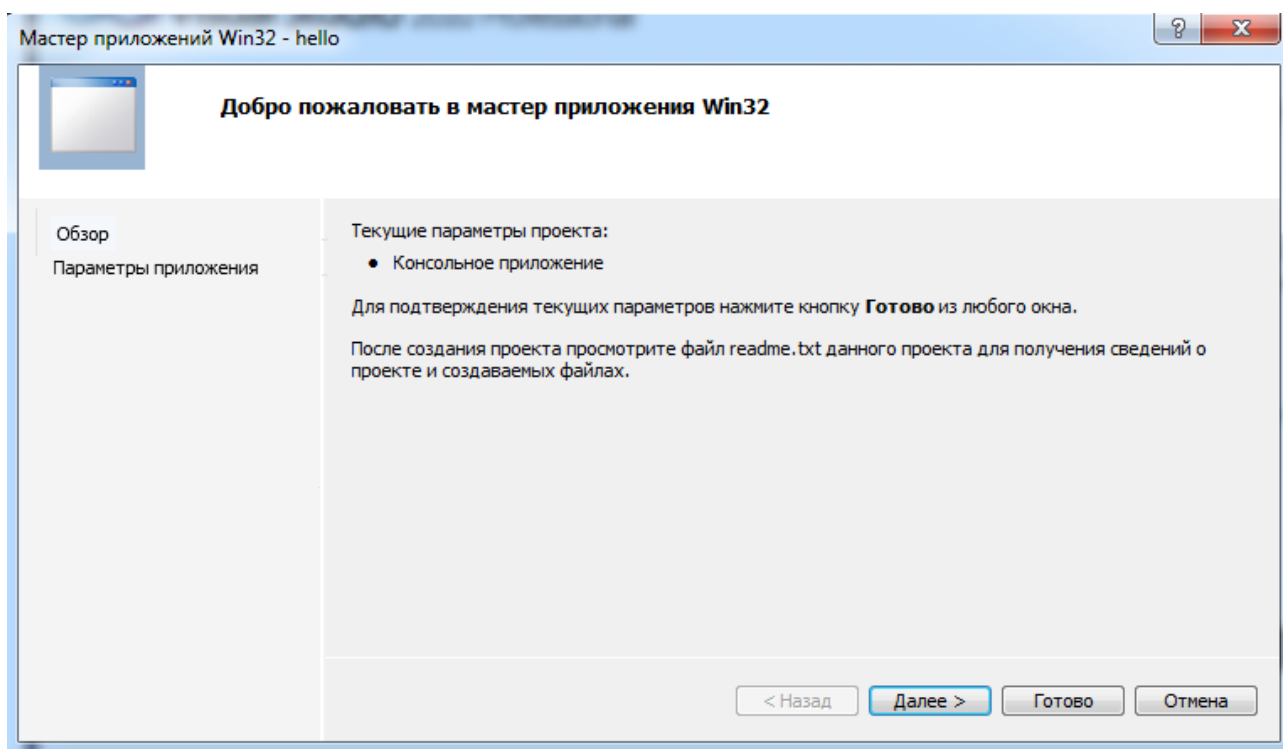


Рисунок 1.3 – Окно мастера создания приложений

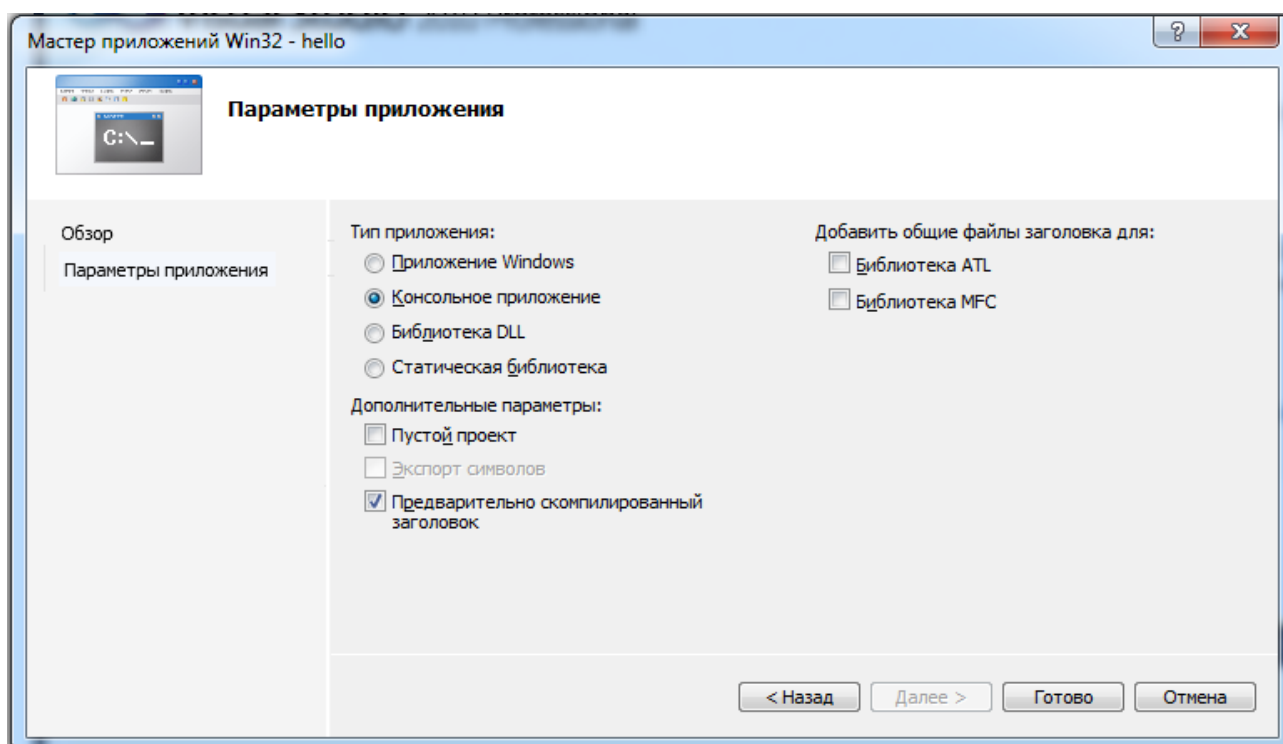


Рисунок 1.4 – Параметры приложения Win32

Здесь можно изменить тип приложения (графическое приложение ОС windows, консольное приложение, динамическая библиотека (dll) или статическая библиотека), а также некоторые другие параметры.

В нашем случае вносить какие-либо изменения не нужно, по этому просто нажимаем на кнопку “Finish” и попадаем в основное окно среды разработки (Рисунок 1.5). Рассмотрим его подробнее.

Проекты для Visual Studio состоят из большого количества файлов, и для удобства работы с ними существуют так называемые «представления», доступные через пункт меню View.

Слева в верхнем углу располагается «Проводник решений» (Solution Explorer), который отображает все файлы, связанные с данным проектом. Файлы разделены на несколько групп: файлы заголовков (Header Files), файлы ресурсов (Resource Files) и файлы исходного кода (Source Files).

Справа располагается окно редактора, поддерживающего автодополнение и подсветку синтаксиса.

Внизу располагается панель, куда будут выводиться результаты компиляции.

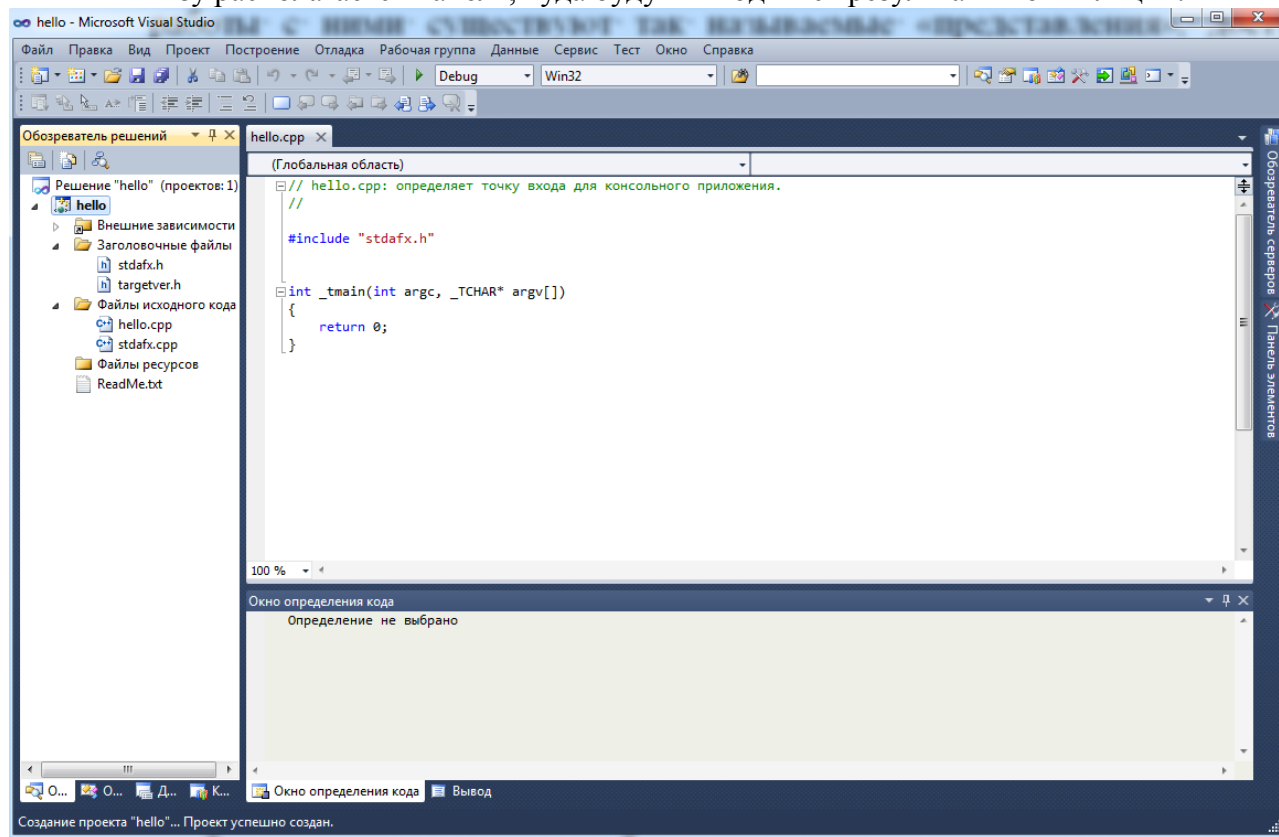


Рисунок 1.5 – Основное окно среды разработки

Для того чтобы скомпилировать и запустить приложение существует несколько способов. Это и соответствующие пункты меню, и кнопки на панели инструментов, и сочетания горячих клавиш, ускоряющих работу программиста. Рассмотрим доступные варианты подробнее (в скобках указано сочетание клавиш).

Build Solution (<F7>) – собрать проект. При этом перекомпилируются все файлы проекта.

Rebuild Solution (<Ctrl + Alt + F7>) – пересобрать проект.

Clean Solution – очистить проект. При этом удаляются все лишние файлы, необходимые на момент разработки и отладки, но не нужные в конечном продукте.

Compile (<Ctrl + F7>) – скомпилировать проект. При этом перекомпилируются только измененные файлы проекта.

Start Debugging (<F5>) – начать отладку. Запускает программу под отладчиком.

Start without Debugging (<Ctrl + F5>) – запустить без отладчика. Просто осуществляется запуск откомпилированной программы.

Step Into (<F11>) – Пошаговое выполнение с заходом в процедуру.

Step Over (<F10>) – Пошаговое выполнение без захода в процедуру.

Toggle Breakpoint (<F9>) – Установить/снять точку останова.

Breakpoints (<Alt + F9>) – показать текущие точки останова.

Также среда обладает возможностью контекстной справки, для этого необходимо навести каретку на интересующий оператор и нажать сочетание клавиш <Ctrl + F1>. При этом справа от окна редактора появится окно с вариантами выбора контекстной помощи.

## 2.3 Создание простейшего приложения

Теперь попробуем написать первое приложение на языке C++ и скомпилировать его. Откройте в окне редактора файл `hello.cpp` (дважды кликнув на нем в окне проводника решений) и введем текст программы, представленный в примере 1.

Пример 1. Простейшая программа на языке C++

```
#include "stdafx.h"
#include <iostream>

using namespace std;

void main()
{
    cout << "Hello World!" << endl;
}
```

Попытаемся скомпилировать данный пример и запустить его и протестировать. В общем виде структура программы выглядит следующим образом:

```
директивы_препроцессора
void main()
{
    исполняемые_операторы;
}
```

`main` – это имя главной функции программы. С функции `main` всегда начинается выполнение. У функции есть имя (`main`), после имени в круглых скобках перечисляются аргументы или параметры функции (в данном случае у функции `main` аргументов нет). У функции может быть результат или возвращаемое значение. Если функция не возвращает никакого значения, то это обозначается ключевым словом `void`. В фигурных скобках записывается тело функции – действия, которые она выполняет.

Вслед за заголовком функции `main` в фигурных скобках размещается тело функции, которое представляет последовательность определений, описаний и исполняемых операторов. Как правило, определения и описания размещаются до исполняемых операторов. Каждое определение, описание и оператор завершается «;».

## 3 Контрольные вопросы

1. Назовите основные этапы создания проекта.
2. Как можно просмотреть вывод Вашей программы?
3. Как компилировать и выполнять программу в MS Visual C++?
4. Что такое контекстно-зависимая помощь?
5. Как получить контекстно-зависимую подсказку?

## 4 Задание

1. Изучить возможности интегрированной среды разработки Microsoft Visual C++.
2. Создать новый проект консольного приложения.
3. Набрать, отладить простейшую консольную программу, выводящую несколько символов на экран.
4. Проверить работоспособность программы.
5. Модифицировать программу, изменив, например, текст выводимого сообщения.
6. Отладить и протестировать программу.
7. Просмотреть программу и попытаться понять, что она делает, не запуская ее на выполнение.

```
#include "stdafx.h"
#include <iostream>

using namespace std;

int main()
{ int x = 5;
  int y = 7;
  cout << "\n";
  cout << x + y << " " << x * y ;
  cout << "\n" << endl;
  return 0;
}
```

8. Введите следующую программу и скомпилируйте ее. Какие сообщения об ошибках вы получили?

```
#include "stdafx.h"
#include <iostream>

int main()
{
  cout << "Hello World!\n";
  return 0;
}
```