

# Среда MS Visual Studio 2005

Для работы MS Visual Studio 2005 компьютер пользователя должен удовлетворять следующим аппаратным требованиям:

- процессор с частотой не ниже 600 МГц;
- ОЗУ 256 Мб;
- 3 Гб свободного места на винчестере;
- разрешение монитора не менее 800x600 (256 цветов).

## 1. Запуск программы

Запуск программы осуществляется из главного меню Пуск – Microsoft Visual Studio 2005 – Microsoft Visual Studio 2005. После этого появится окно заставки MS Visual Studio (см. рис. 1.2), в которой пользователь сможет увидеть список установленных приложений. После этого при первом запуске будет предложено выбрать язык (Basic, C#, C++ и т.д.), на работу с которым будет настроена среда MS Visual Studio. Выбираем Visual C++. При первом запуске MS Visual Studio появится окно первоначальной настройки среды. После чего появляется основное окно MS Visual Studio (рис. 1). При последующих запусках программы MS Visual Studio сразу будет появляться основное окно программы (рис. 3).

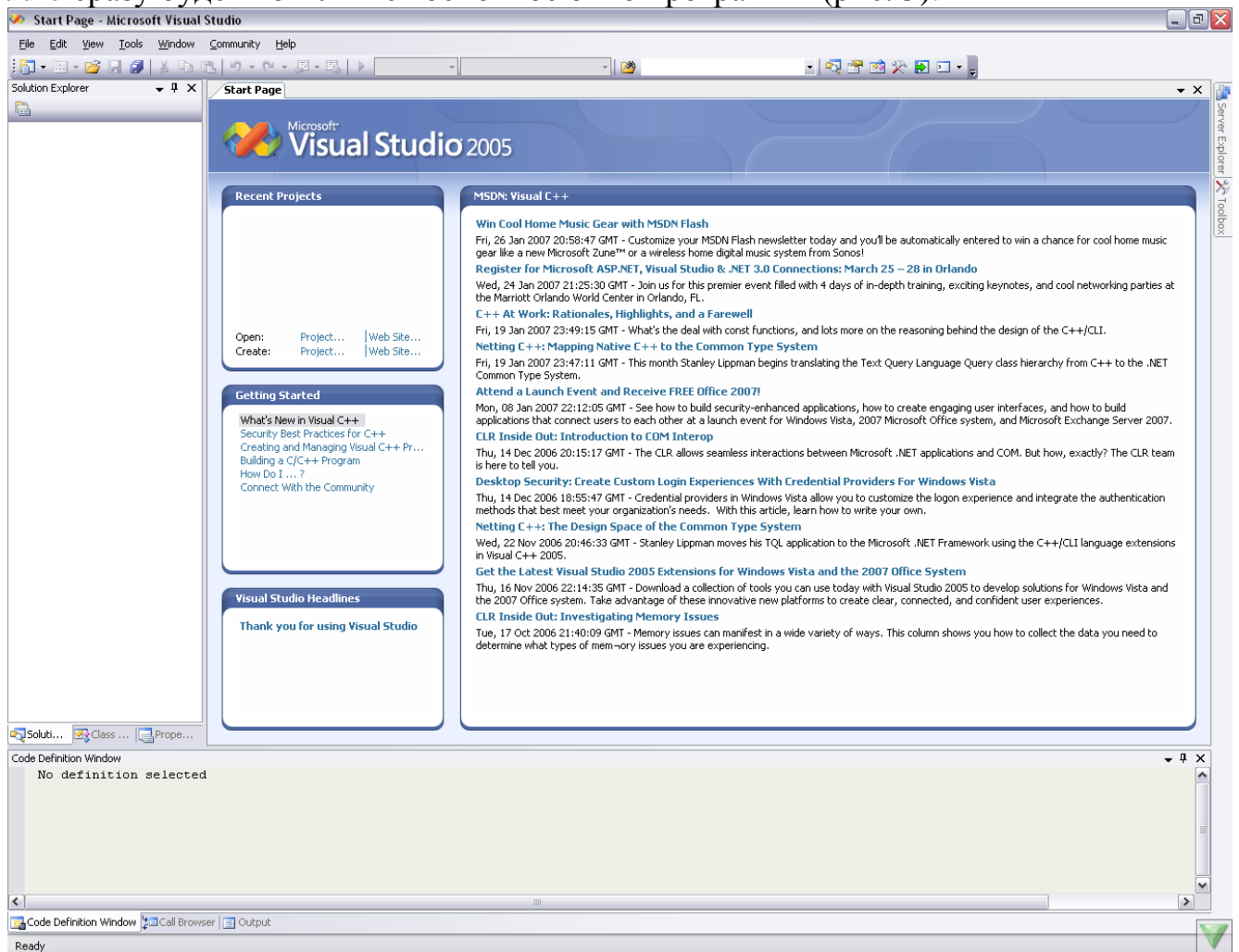


Рис. 1. Окно среды MS Visual Studio 2005

## 2. Создание консольного приложения

Под программой в среде MS Visual Studio 2005 понимается несколько взаимосвязанных между собой файлов, называемых *проектом*.

Для создания проекта консольного приложения необходимо выполнить команду **File – New – New Project** и в открывшемся диалоговом окне (рис. 2):

1. Выбрать тип проекта **Project Type – Visual C++ – Win32 – Win32 Console Application**.
2. Ввести имя проекта (поле **Name**), в нашем случае в качестве имени проекта введем `first_prime`, для хранения всех файлов проекта будет создана новая папка с именем, совпадающим с именем проекта.
3. Определить месторасположение папки с проектом – параметр **Location**.
4. Параметр **Solution Name** – определяет имя *рабочей области*, по умолчанию имя рабочей области совпадает с именем проекта, мы пока не будем его менять, но вообще говоря, в одной рабочей области может быть объединено несколько взаимосвязанных проектов.

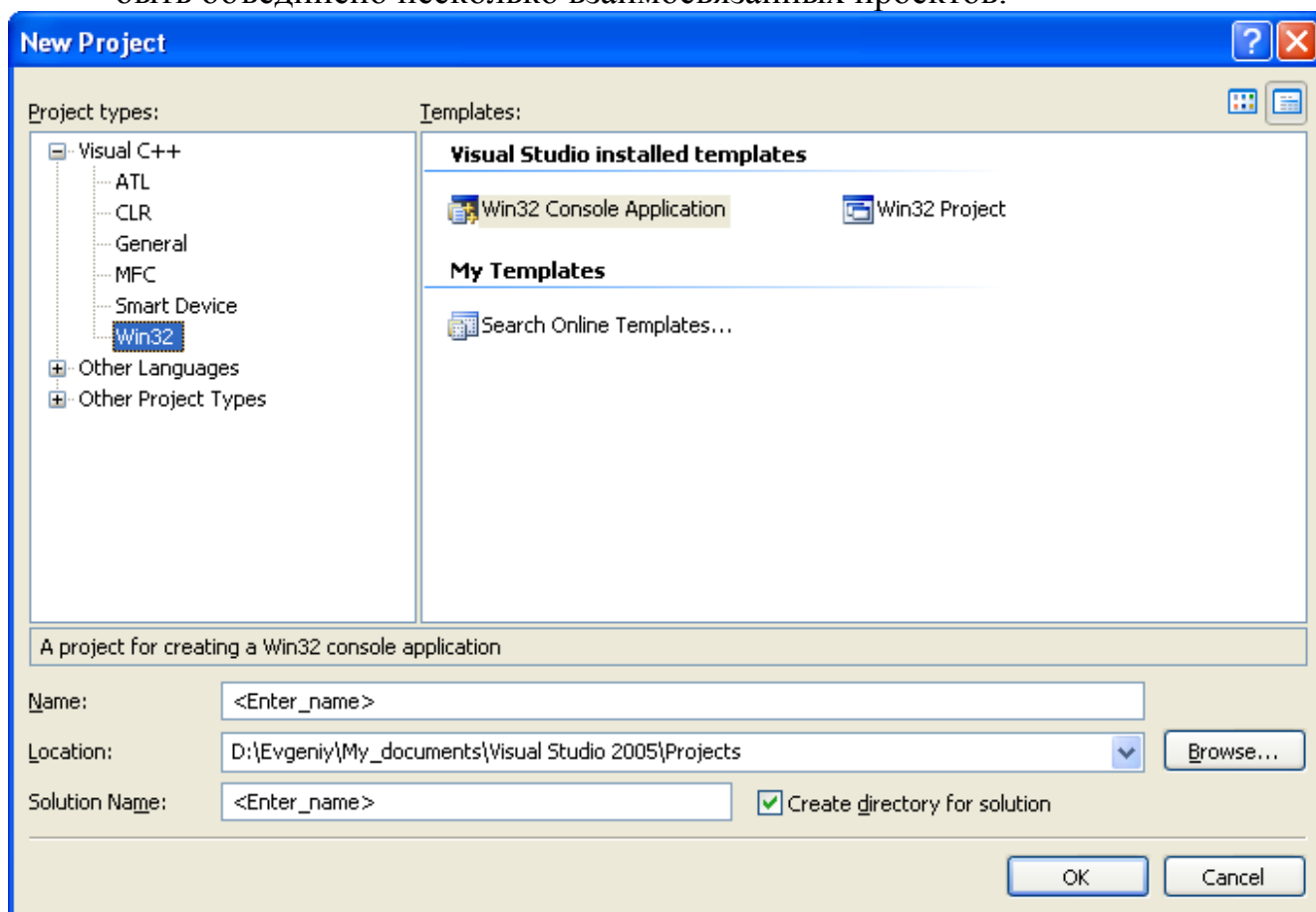


Рис. 2. Окно создания нового проекта

После этого необходимо щелкнуть по кнопке **OK**, в появившемся окне щелкнуть по кнопке **Next**, далее подтвердить тип проекта (**Console Application**) и щелкнуть по кнопке **Finish**. Окно MS Visual Studio 2005 станет таким, как показано на рис. 3.

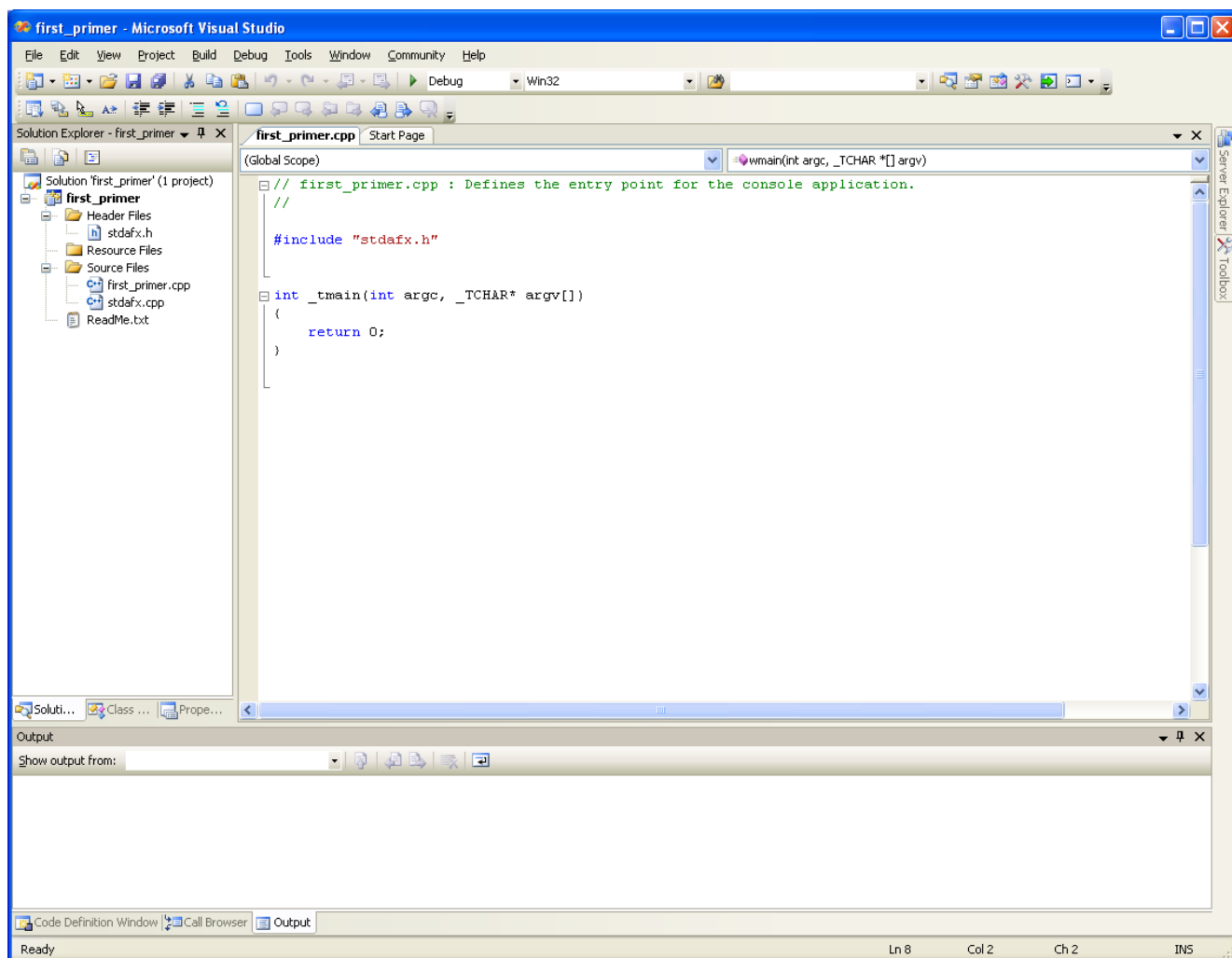


Рис. 3. Окно консольного приложения.

После этого в папке, имя которой определено параметром **Location**, будет создана папка рабочей области **first\_primer** (ее имя определяется параметром **Solution Name**). В папке рабочей области будут файлы **first\_primer.sln** (это и есть файл рабочей области) и **first\_primer.ncb** и папка **first\_primer** (имя папки определяется параметром **Name**), в которой находятся все файлы проекта, в том числе и основной файл **first\_primer.cpp**.

### 3. Открытие и закрытие рабочей области

Если рабочая область уже создана, то ее можно открыть с помощью команды **File – Open - Project/Solution**. В появившемся диалоговом окне перейти в папку, где находится рабочая область, и выбрать файл с расширением **.sln**.

Для закрытия рабочей области служит команда **File – Close Solution**. Если какие-либо файлы из рабочей области были не сохранены, то перед закрытием рабочей области появится окно с предложением их сохранить.

**Вывод.** Программа называется проектом. Проект хранится в отдельной папке, имя которой совпадает с именем проекта (имя проекта определяется параметром **Name** в окне создания проекта). Один или несколько проектов образуют рабочую область. Каждая рабочая область хранится в отдельной папке (имя папки определяется параметром **Solution Name** в окне создания проекта). В папке рабочей области хранится файл рабочей области с

расширением *.sln*, имя которого совпадает с именем папки, и папки всех проектов, входящих в рабочую область.

#### 4. Добавление файлов в проект

После создания проекта консольного приложения в него необходимо добавить файлы. Для этого необходимо щелкнуть по имени проекта (в нашем примере **first\_primer**) правой кнопкой мыши и в контекстном меню выбрать команду **Add – Add New Item**. Откроется окно (рис. 4), в котором в левой части выбрать **Categories – Code**, а в правой – **Templates - C++ File (.cpp)**, в поле **Name** ввести имя создаваемого файла (в нашем случае файл **test.cpp**). После этого щелкнуть по кнопке **Add**, и в окне проекта появится вкладка с пустым файлом **test.cpp**. В этом окне и следует вводить текст программы.

После ввода текста программы окно MS Visual C++ примет вид, представленный на рис. 5. Сохранить файл можно командой **File – Save**.

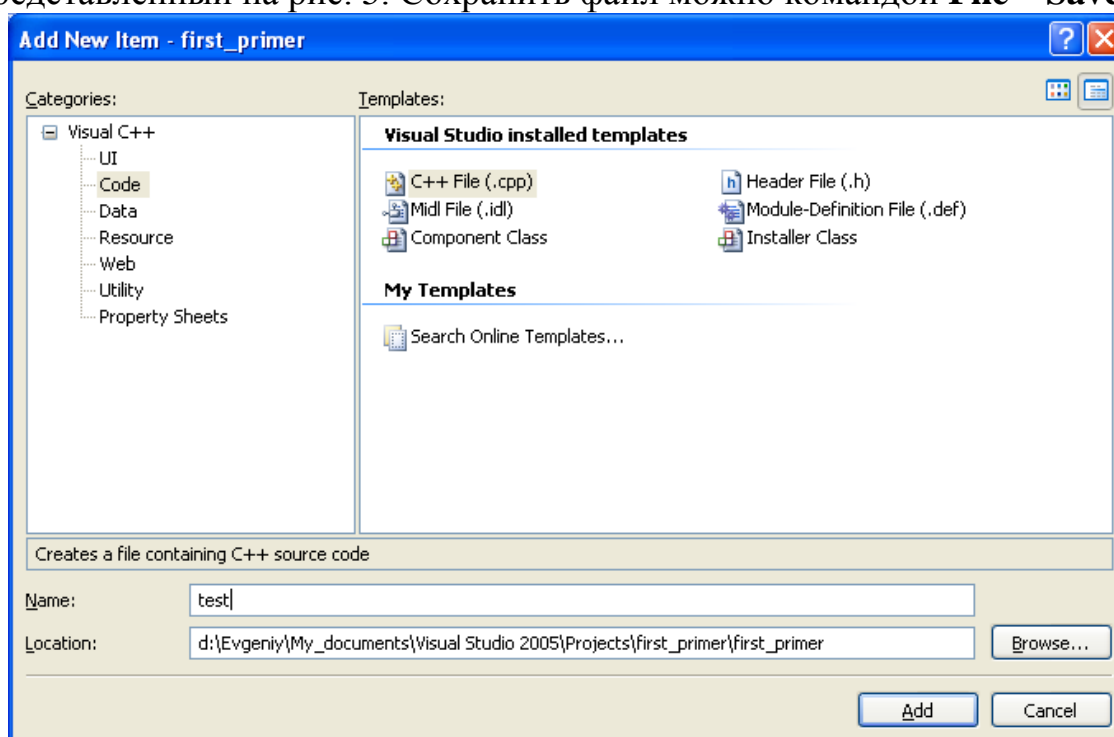


Рис. 4. Определение параметров добавляемого файла

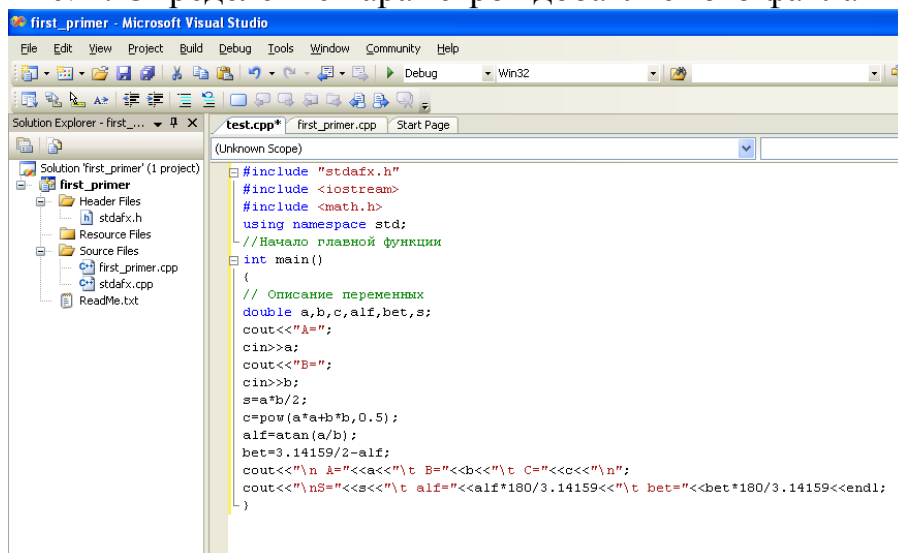


Рис. 5. Окно проекта с текстом программы

К проекту можно подключить и существующий **.cpp** файл. Для этого необходимо щелкнуть по имени проекта правой кнопкой мыши и в контекстном меню выбрать команду **Add – Existing Item**.

В окне текстового редактора доступны все основные операции с текстовыми файлами, которые доступны из меню **Edit**.

**Вывод.** Для того, чтобы ввести консольную программу в MS Visual Studio 2005, необходимо:

1. Создать проект с помощью команды **File – New – New Project** и в открывшемся диалоговом окне (рис. 1.6) выбрать в качестве типа проекта консольное приложение (**Project Type – Visual C++ – Win32 – Win32 Console Application**).
2. Добавить в него файл, для чего необходимо щелкнуть по имени проекта правой кнопкой мыши и выбрать команду **Add – Add New Item**. В открывшемся окне выбрать тип (**C++ File**) и имя файла.
3. В созданном файле ввести текст программы и сохранить его.

## 5. Трансляция и запуск программы

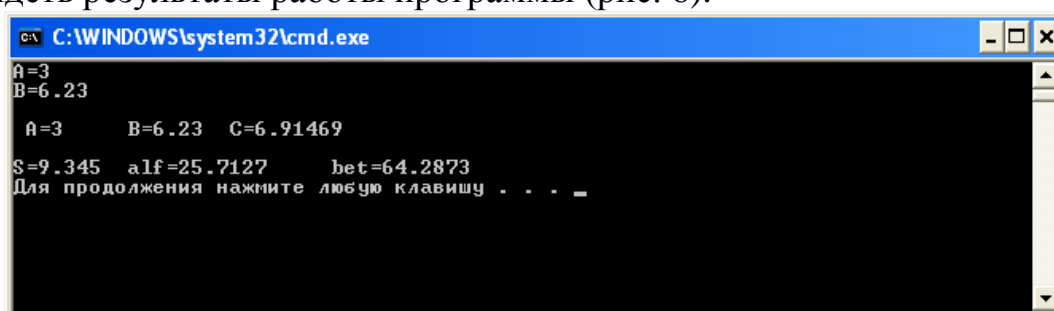
После ввода программы в компьютер ее необходимо откомпилировать. Для компиляции программы необходимо сделать активным проект, в дереве проекта выбрать компилируемый файл и в контекстном меню выбрать команду **Compile**.

При отсутствии ошибок в окне **Output** среды MS Visual Studio 2005 появится сообщение об успешной компиляции. Если компилятор обнаружит синтаксические ошибки, то при компиляции в окне **Output** пользователь увидит соответствующие сообщения. Если щелкнуть в окне **Output** по сообщению об ошибке, то в окне программы будет выделена строка с ошибкой, которую следует исправить и повторно откомпилировать файл. Результатом трансляции является файл с расширением **.obj**, который будет создан в папке **Debug** каталога проекта.

После успешной компиляции всех файлов, входящих в проект, необходимо собрать (скомпоновать) весь проект из объектных (откомпилированных) файлов с расширением **.obj**. Для этого следует выполнить команду **Build – Build (F7)**. В результате компоновки будет сгенерирован исполняемый файл (файл с расширением **.exe**) программы.

Для автоматической компиляции всех файлов проекта и его компоновки служит команда **Build – Rebuild (Ctrl+Alt+F7)**.

После компоновки проекта программу можно запустить на выполнение, для чего следует выполнить команду **Debug – Start Without Debugging (Ctrl+F5)**. После чего откроется окно, в котором можно будет вводить исходные данные и видеть результаты работы программы (рис. 6).



```
C:\WINDOWS\system32\cmd.exe
A=3
B=6.23
A=3    B=6.23    C=6.91469
S=9.345  a1f=25.7127  bet=64.2873
Для продолжения нажмите любую клавишу . . . -
```

Рис. 6. Результат работы программы

Если запустить нескомпонованную программу, то перед запуском будет выведено сообщение о необходимости сборки программы, после подтверждения произойдет автоматическая компиляция всех файлов проекта и его сборка. Если компиляция и компоновка проекта пройдет успешно, то программа запустится на выполнение.

## 6. Отладка программы

*Отладка программ* – это процесс поиска и устранения ошибок. Все ошибки, встречающиеся в программе, можно разделить на следующие группы:

1. *Синтаксические ошибки.*
2. *Ошибки времени выполнения.*
3. Ошибки, возникшие на этапе разработки алгоритма (так называемые *алгоритмические ошибки*).

Синтаксические ошибки обнаруживаются и исправляются еще на этапе трансляции программы.

Ошибки времени выполнения обнаруживаются при первых запусках программы. При таких ошибках выполнение программы прерывается, пользователю следует анализировать текст программы и искать ошибку.

Самыми сложными являются алгоритмические ошибки. Программа запускается, но выдает неправильный результат. В среде MS Visual Studio 2005 есть мощные средства для выявления и исправления таких ошибок: есть возможность останавливать выполнение программы на определенных участках (трассировка программы) и видеть значение переменных в процессе выполнения программы.

Для приостановки выполнения программы на определенных участках отладчик предоставляет следующие возможности:

1. *Выполнение программы до курсора*, программа остановится в той строке, где находится курсор (если это возможно). Для этого необходимо нажать комбинацию клавиш **Ctrl+F10**.
2. *Построчное выполнение программы*, в этом режиме выполняется одна строка программы. Для активизации этого режима необходимо выполнить команду **Step over** из пункта меню **Debug** или нажать клавишу **F10**.
3. *Построчное выполнение программы с заходом в функцию*. Главное отличие этого режима от предыдущего состоит в том, что если в текущей строке встретился оператор обращения к подпрограмме, то начинается построчное выполнение подпрограммы, для выполнения программы в этом режиме необходимо вызвать команду **Step into** из пункта меню **Debug** или нажать клавишу **F11**. Для выхода из режима построчного выполнения функции существует команды **Step Out** из меню **Debug** (комбинация клавиш **Shift+F11**).
4. *Точки останова*. В этом режиме после запуска программы на выполнение она останавливается в определенных строках, которые помечены как точки останова. Для того чтобы установить точку останова в строке, необходимо в этой строке вызвать контекстное меню, в котором выбрать команду **BreakPoint – Insert Breakpoint**. После этого в левой части строки появится красный круг, который будет свидетельствовать о наличии точки останова. Точку останова

можно удалить или запретить с помощью команд контекстного меню **BreakPoint – Delete Breakpoint** и **BreakPoint – Disable Breakpoint** соответственно.

Но одна приостановка выполнения программы в определенных точках не позволит найти ошибку. Для поиска алгоритмической ошибки полезно знать значения переменных в той или иной точке. Можно, конечно, для этого вставлять операторы вывода и просматривать значение переменной. Но MS Visual Studio 2005 позволяет просматривать значения переменных в специальном окне. Для этого необходимо выполнить команду **Debug - Quick Watch**, и в появившемся окне ввести имя или выражения, значения которых нужны. После нажатия **Enter** эти значения появятся в окне. Если в окне **Quick Watch** щелкнуть по кнопке **Add Watch**, то введенное выражение будет помещено в специальное многовкладочное окно **Watch1** среды MS Visual Studio 2005, которое будет доступно и после закрытия окна **Quick Watch**. Любую переменную или выражение можно удалить из окна **Watch1** с помощью клавиши **Delete**.

Обратите внимание, что в окне **Watch 1** есть вкладка **Locals**, в которой находятся все переменные проекта.

Для поиска ошибок необходимо останавливать программу в определенных точках и просматривать значения переменных. Совместное использование приостановки программы и просмотра значений переменных в любой точке программы позволит найти большинство алгоритмических ошибок.

Для выхода из режима отладки следует использовать команду **Debug – Stop Debugging (Shift+F5)**. Обратите внимание, что при этом закрывается окно **Watch1** со значениями переменных.