

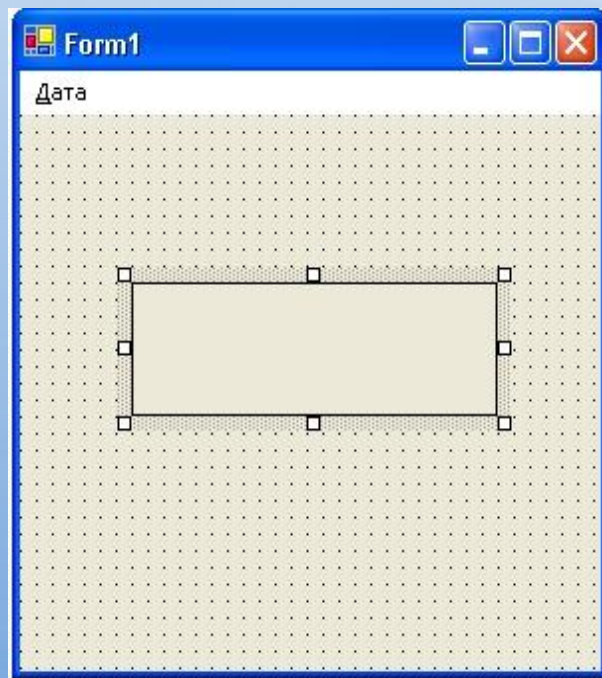


**Тема урока**

**Разработка программы с  
использованием опции меню**

# форма должна выглядеть примерно так.

Теперь добавим операторы в процедуры обработки событий для обработки команд меню **Время** и **День**



## Редактирование процедур событий меню

1. Щелкните на меню **Дата**, чтобы его раскрыть.
2. Чтобы открыть в редакторе кода процедуру обработки событий для команды **Время**, дважды щелкните мышью на этой команде. В редакторе кода появится процедура события **MenuItem3\_Click**. Имя **MenuItem3\_Click** означает, что пункт **Время** был третьим из созданных в этом проекте (вслед за **Дата** и **День**), а слово **\_Click** напоминает, что это процедура события, которая запускается при щелчке на этом элементе меню. Пока мы сохраним это название меню, хотя тут следует отметить, что оно недостаточно интуитивно понятно. С помощью свойства Name можно задавать для объектов свои собственные имена, которые описывают их назначение в программе более точно
3. Добавьте в программу следующий оператор  
• **Label1.Text = TimeString**

Этот оператор присваивает текущее время (по системным часам) свойству **Text** объекта **Label1**, которое, собственно, и показывается в виде надписи. Предыдущий текст в объекте **Label1**, если он имелся, будет заменен. **TimeString** - это свойство, в котором содержится текущее время, отформатированное для показа на экране или для печати. В программах с помощью свойства **TimeString** всегда а можно показать время с точностью до секунд. (**TimeString** заменил оператор **QuickBasic TIME\$**.)

***Совет.** Свойство **TimeString** в **Visual Basic** возвращает текущее системное время. Системное время можно установить из Панели управления **Windows** с помощью процедуры **Дата и время**. Формат системного времени можно изменить там же в **Региональных настройках**.*

4.Нажмите на клавишу со стрелкой вниз. **Visual Basic** интерпретирует строку и, если потребуется, изменит заглавные буквы и добавит или удалит пробелы. **Visual Basic** проверяет каждую строку в процессе ее ввода и ищет в них синтаксические ошибки. Набор строки можно закончить, нажав клавишу (**Enter**), стрелку вверх или стрелку вниз.

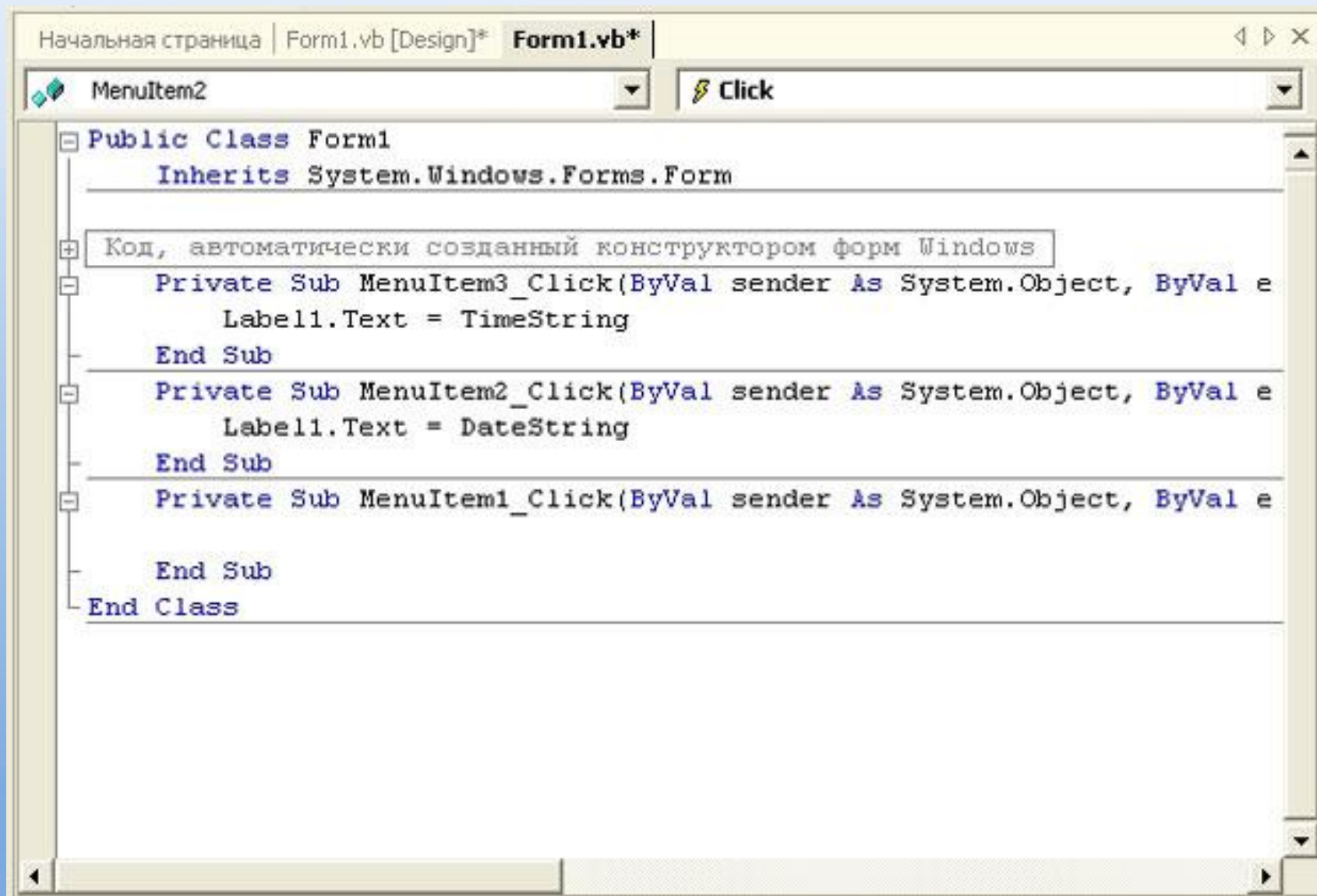
5.В **Solution Explorer** (Обозревателе решений) нажмите кнопку **View designer** (Просмотреть конструктор), а затем дважды щелкните мышью на команде **День** в меню **Дата**. В редакторе кода откроется процедура обработки событий **MenuItem2\_Click**. Эта процедура выполняется тогда, когда пользователь щелкает в меню **Дата** на команде **День**.

6. Добавьте в программу следующий оператор **Label1.Text = DateTime.Now.ToString()** Этот оператор присваивает сегодняшнее число (по системным часам) свойству **Text** объекта **Label1** , которое показывается в виде надписи. Предыдущий текст в объекте **Label1** , если он имелся, будет заменен. Свойство **DateTime.Now.ToString()** точно так же доступно для широкого использования в программах. Если в форме требуется показать текущую дату, присвойте свойству **Text** значение **DateTime.Now.ToString()**.

- ▣ **Совет.** Свойство *DateTime.Now.ToString()* в *Visual Basic* возвращает текущую системную дату. Ее можно установить в разделе *Дата и время* Панели управления; формат системного времени можно изменить там же в *Региональных настройках*.

Чтобы закончить ввод строки, нажмите клавишу со стрелкой  
ВНИЗ.

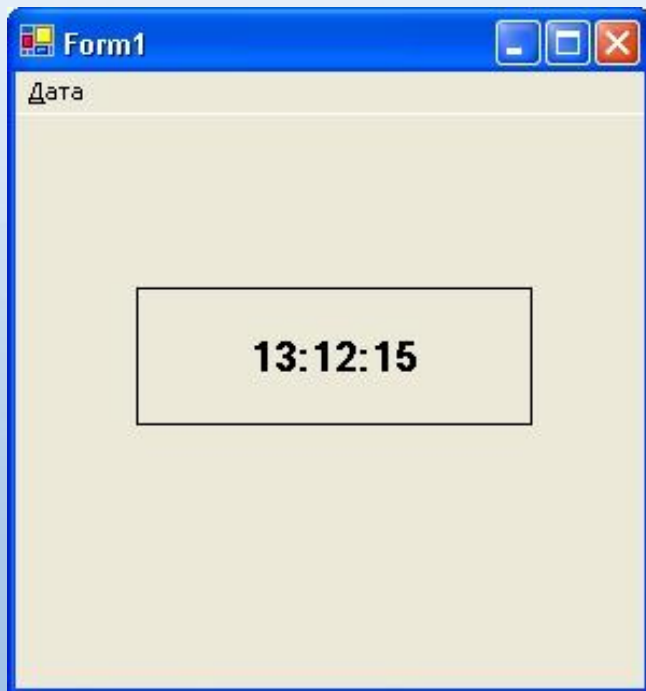
Ваш экран должен выглядеть примерно так.



```
Начальная страница | Form1.vb [Design]* Form1.vb*
MenuItem2 Click
Public Class Form1
    Inherits System.Windows.Forms.Form

    Код, автоматически созданный конструктором форм Windows
    Private Sub MenuItem3_Click(ByVal sender As System.Object, ByVal e
        Label1.Text = TimeString
    End Sub
    Private Sub MenuItem2_Click(ByVal sender As System.Object, ByVal e
        Label1.Text = DateString
    End Sub
    Private Sub MenuItem1_Click(ByVal sender As System.Object, ByVal e
    End Sub
End Class
```

# Запуск программы Menu



1. На стандартной панели инструментов нажмите кнопку Start (Начать). Программа Menu запустится в среде разработки.
2. В строке меню выберите пункт **Дата**. Появится меню **Дата**.
3. Выберите команду **Время**. В поле надписи появится текущее системное время, как показано ниже.

## Теперь посмотрим, какое сегодня число.

1. Нажмите и отпустите клавишу (Alt). В строке меню выделится меню **Дата**.
2. Чтобы раскрыть меню **Дата**, нажмите **Д**. Меню появится на экране.
3. Чтобы показать сегодняшнее число, нажмите **Ч**. В поле надписи появится дата.
4. Чтобы остановить программу, нажмите кнопку **Закреть** в строке заголовка программы.

# Свойства и функции системных часов

- Чтобы получить от системных часов информацию о времени, можно использовать их различные свойства и функции. Информация о времени может потребоваться в программах при создании собственных календарей, часов или оповещений. В следующей таблице содержится перечень наиболее полезных функций системных часов.

Свойство или функция	Описание
<b>TimeString</b>	Возвращает от системных часов текущее время.
<b>DateString</b>	Возвращает от системных часов текущую дату.
<b>Now</b>	Возвращает закодированное значение, содержащее текущие дату и время. Наиболее полезно как аргумент для других функций системных часов.
<b>Hour(time)</b>	Возвращает количество часов для указанного времени (от 0 до 24).
<b>Minute(time)</b>	Возвращает количество минут для указанного времени (от 0 до 59).
<b>Second(time)</b>	Возвращает количество секунд для указанного времени (от 0 до 59).
<b>Day(date)</b>	Возвращает целое число, представляющее собой день месяца (от 1 до 31).
<b>Month(date)</b>	Возвращает целое число, представляющее собой месяц (от 1 до 12).
<b>Year(date)</b>	Возвращает год для указанной даты.
<b>Weekday(date)</b>	Возвращает целое число, представляющее собой день недели (по американской системе: 1 - это воскресенье, 2 - это понедельник, и т.д.).



- **Использование элементов управления для диалоговых окон**
- В Visual Studio на закладке Windows Forms окна области элементов имеется семь стандартных элементов управления для диалоговых окон. Они предоставляются в готовом виде, так что вам вряд ли потребуется создавать собственные диалоговые окна для типичных задач, таких как открытие файлов, сохранение файлов или печать. Во многих случаях нужно написать код, который подключает эти диалоговые окна к программе, но пользовательский интерфейс уже сделан, и он соответствует стандартам для общих задач в приложениях Windows. Все семь имеющихся элементов управления для стандартных диалоговых окон перечислены в следующей таблице. Они во многом соответствуют объектам, которые в Visual Basic 6 предоставлялись в составе элемента управления CommonDialog, но имеют и некоторые отличия. Так, вы не найдете в таблице элемент управления PrintPreviewControl, но в случае необходимости его заменит элемент управления PrintPreviewDialog.

<b>Название элемента управления</b>	<b>Назначение</b>
<code>OpenFileDialog</code>	<b>Получает названия диска, папки и файла для существующего файла.</b>
<code>SaveFile (Файл) Dialog</code>	<b>Получает названия диска, папки и файла для нового файла.</b>
<code>FontDialog</code>	<b>Позволяет выбрать новый шрифт и его стиль.</b>
<code>ColorDialog</code>	<b>Позволяет выбрать цвет из палитры.</b>
<code>PrintDialog</code>	<b>Позволяет задать параметры печати.</b>
<code>PrintPreviewDialog</code>	<b>Отображает диалоговое окно предварительного просмотра материала для печати, так, как это делает Microsoft Word.</b>
<code>PageSetupDialog</code>	<b>Позволяет управлять параметрами страницы: полями, размером бумаги и ее ориентацией.</b>