

Знакомство с Visual Studio 2010

Добро пожаловать в Visual Basic.NET (в дальнейшем будем применять сокращение VB.NET). Это один из удобных языков для быстрого создания Windows приложений, а также Web приложений. Visual Studio 2010 – это платформа разработки приложений для Windows и Web. VB.NET – это один из языков, основывающийся на этой платформе. Нет такой задачи, которую можно было бы решить с помощью другого языка программирования, и нельзя решить на VB.NET.

1. Запуск системы Visual Studio 2010 и создание проекта

Создание любого проекта следует начать с создания папки (далее будем ее называть рабочей папкой), в которой будут размещаться файлы проекта.

1. Создайте свою рабочую папку. Если Вы работаете в компьютерном классе ИВЦ МЭИ, то создайте рабочую папку в корневом каталоге диска *d* компьютера. Имя папки придумайте по своему усмотрению. Оно должно быть уникальным в пределах учебной группы. Тем самым Вы подготовите пространство памяти на жестком диске, где Вы на протяжении всего занятия сможете хранить, созданные Вами файлы и папки.

2. Для запуска Visual Studio 2010 щелкните на кнопке Пуск. В последовательно открывающихся окнах выберите Все программы, Microsoft Visual Studio 2010, и еще раз выберите Microsoft Visual Studio 2010. Откроется стартовая страница (Start Page), показанная на рис. 1.

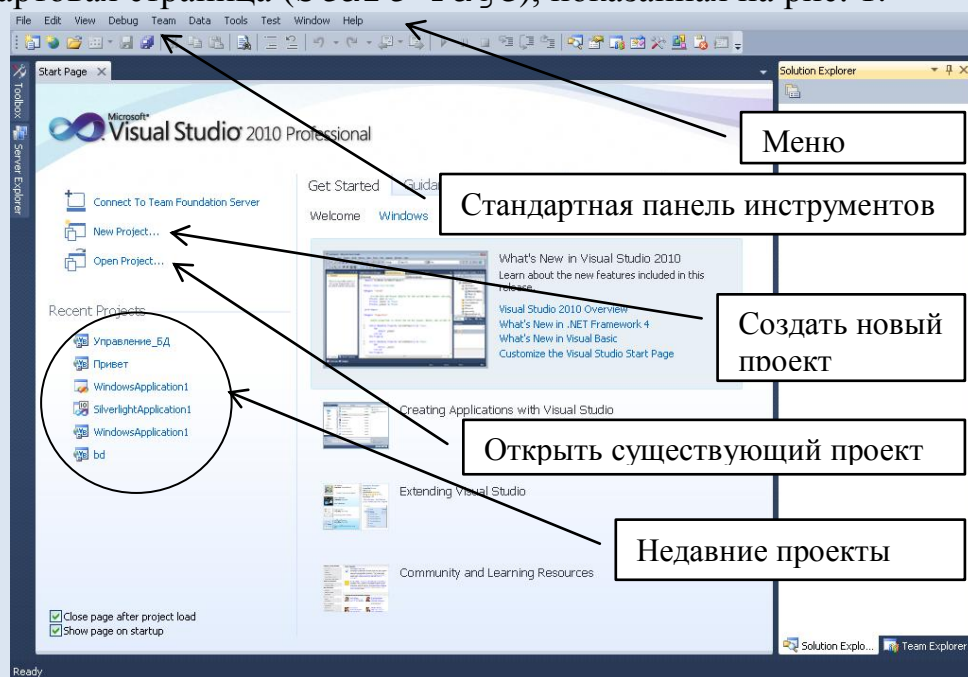


Рис. 1. Стартовая страница при запуске Visual Studio 2010

На стартовой странице ссылка `New Project` или позволяет создать новый проект. Ссылка `Open Project` позволяет открыть существующий проект. В области `Recent Project` (последние проекты) находятся ссылки, которые позволяют: открыть недавний проект.

3. Теперь, наконец, настало время приступить к непосредственному созданию нового проекта. Для этого щелкните на ссылке **New Project** на стартовой странице.

Чтобы правильно задать тип создаваемого проекта, в открывшемся окне **Создать проект**, показанном на рис. 2, выберите в окне **Установленные Шаблоны** шаблон **Windows** в разделе **Visual Basic**, а также выберите шаблон **Приложения Windows Forms**.

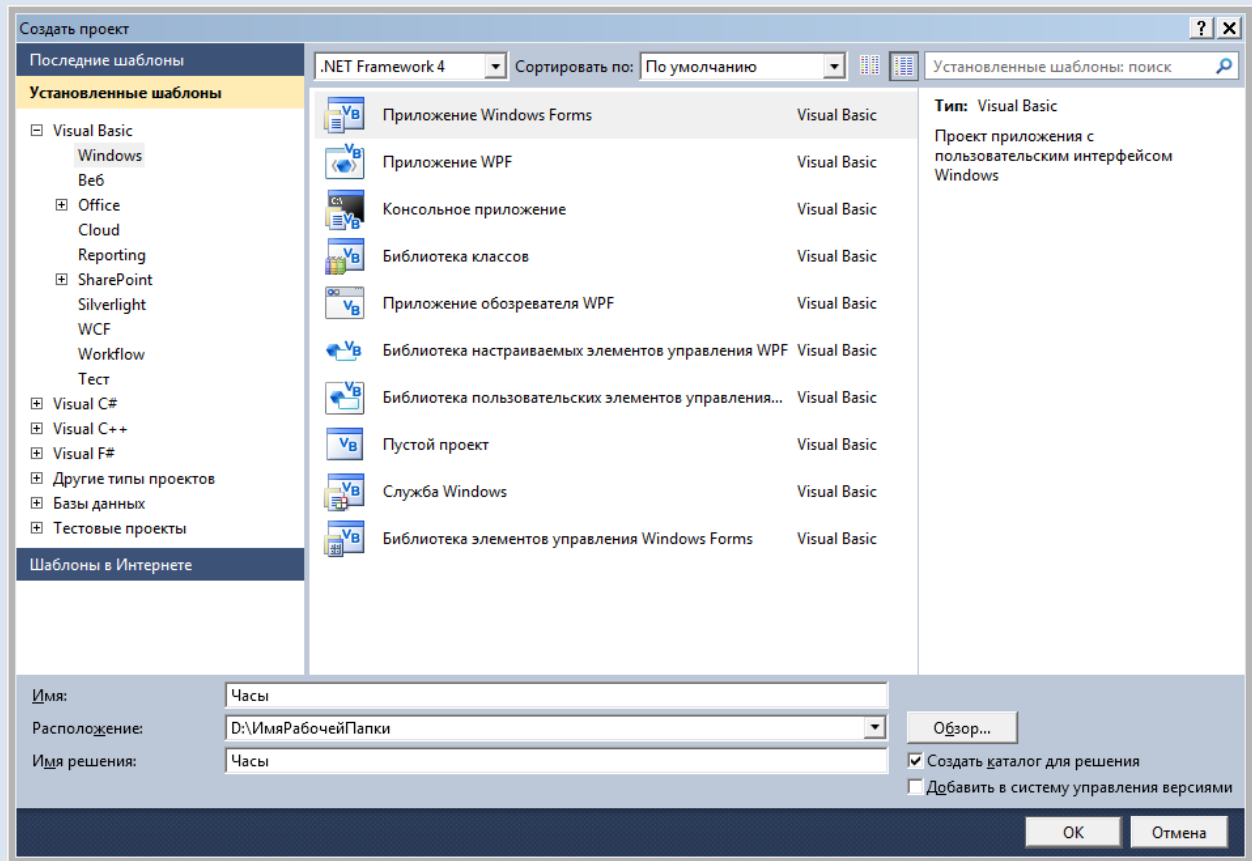


Рис. 2. Выбор типа проекта, шаблона, имени проекта и его локализации

Тем самым Вы указали, что собираетесь создать приложение для **Windows**, используя язык **Visual Basic**.

Затем в поле **Name** замените предложенное по умолчанию имя проекта именем **Часы**.

Если под этим полем **Имя :** отображено поле **Расположение :** (это зависит от настройки системы), то:

- Применяя кнопку **Обзор** укажите свою рабочую папку.
- Проверьте, что установлен флажок **Создать каталог для решения**.
- Для завершения создания проекта нажмите на кнопку **ОК**.

Если же под полем **Имя :** не отображено поле **Расположение :** то сразу завершите создание проекта нажатием на кнопку **ОК**. Расположение проекта Вы зададите позже в процессе его первого сохранения.

Теперь на экране монитора отображено окно интегрированной среды разработки (**Integrated Development Environment**) **Visual Studio 2010** (принятое сокращение **IDE**). Это окно приведено на рис. 3.

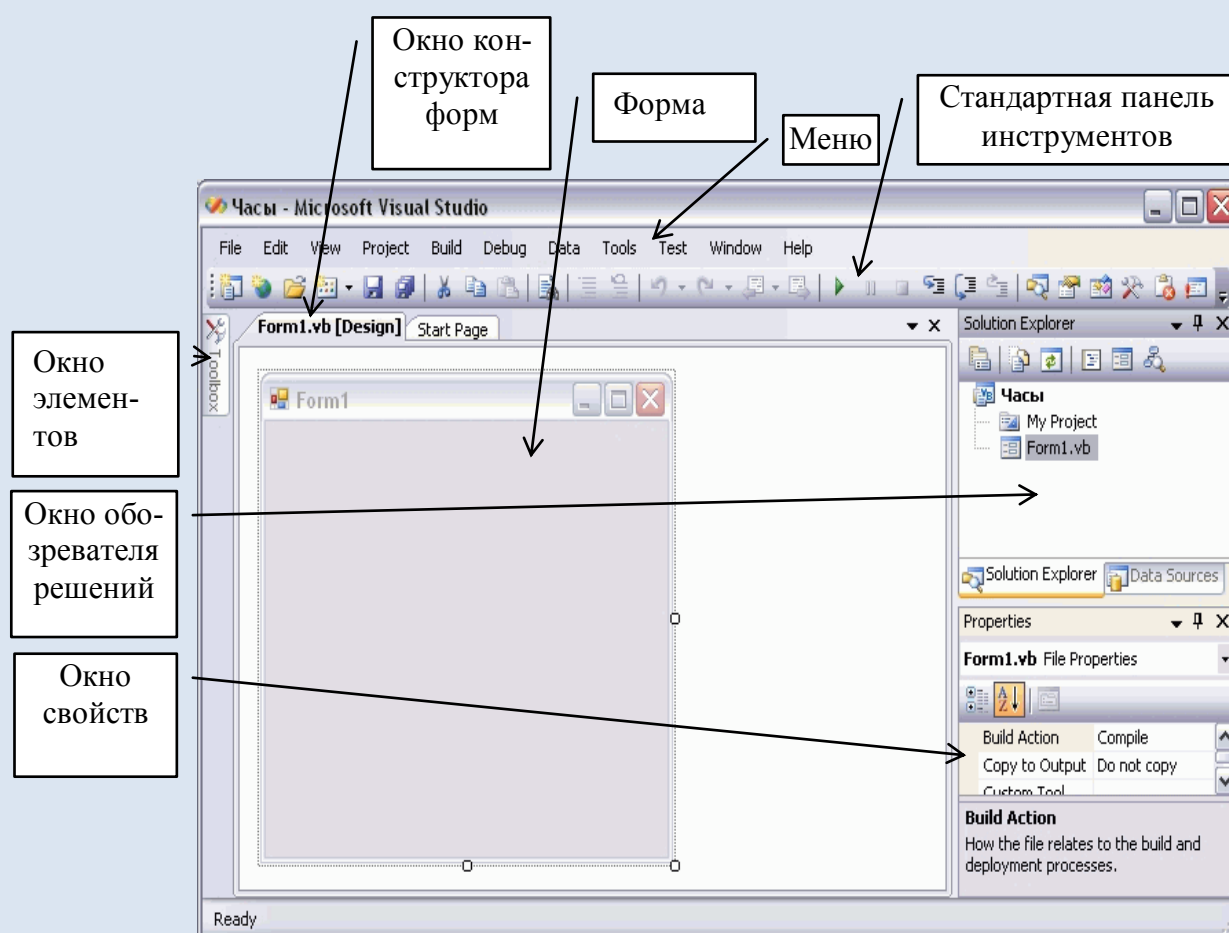



Рис. 3. Окно интегрированной среды разработки (IDE)

Если окно `Form1.vb [Design]` (окно конструктора форм) не отобразилось, то щелкните дважды на узле `Form1.vb` в окне `Solution Explorer` (обозреватель решений).

Проект создан, но пока еще не сохранен. Имеет смысл сразу решить вопрос с местом хранения файлов Вашего проекта. Для этого нажмите на кнопке  `Save All` (Сохранить все) на стандартной панели инструментов и укажите папку, где будут сохранены файлы проекта. В качестве этой папки, нажав на кнопку `Browse` (обзор), укажите в поле `Location` (расположение) путь к своей рабочей папке, в которой будет сохранен проект. Проверьте, чтобы был установлен флажок `Create directory for solution` (создать папку для решения). Завершите сохранение проекта щелчком на кнопке `Save`.


4. Выполните настройку IDE. В ближайшее время для работы потребуются стандартная панель инструментов и четыре окна (окно конструктора форм, окно обозревателя решений, окно управляющих элементов и окно свойств). Для настройки отображения панелей инструментов выполните команду меню `View` (вид), `Toolbars` (панели инструментов). В открывшемся меню оставьте галочку слева только у строки `Standard` (стандартная).

Если открыты не все требующиеся окна, то недостающие окна следует открыть, пользуясь соответствующими кнопками стандартной панели инструментов:

окно обозревателя решений – кнопкой  `Solution Explorer`;

окно свойств – кнопкой  Properties Window;

окно элементов – кнопкой  Toolbox.

Окно конструктора форм можно открыть кнопкой  View Designer, расположенной на панели инструментов окна обозревателя решений. Если эта кнопка там не отображена, то для ее отображения следует в окне обозревателя решений дважды щелкнуть на компоненте проекта `Form1.vb`.

На рис. 3 окно элементов показано в свернутом состоянии слева окна конструктора форм. Оно разворачивается автоматически при попадании в фокус мыши и сворачивается снова, если теряет фокус мыши. Чтобы оно было развернуто постоянно, достаточно на нем щелкнуть, когда оно развернуто.

5. Загляните в свою рабочую папку. Вы обнаружите, что в ней появилась папка с тем же именем, что и имя проекта (Часы). В этой папке содержаться папки и файлы Вашего проекта. Если потребуется копирование проекта, то необходимо копировать рабочую папку вместе со всем ее содержимым.

2. Разработка интерфейса

Сейчас Вы создадите Windows приложение, которое после запуска будет отображать текущую дату и время (рис. 4). Эта информация должна обновляться при каждом щелчке мышью на отображении даты и времени.



Рис. 4. Так будет выглядеть окно работающего проекта

Формулирование требований к тому, как будет выглядеть работающий проект, закладывает основу для первого шага создания проекта – разработки его интерфейса.

6. Для отображения текста в окне работающего проекта применяется управляющий элемент, называемый `Label` (надпись). Поместите надпись на форму. Для этого переместите указатель мыши на свернутое окно элементов. После автоматического разворачивания этого окна (рис. 5.) дважды щелкните на элементе `Label`. Надпись `Label1` появится на форме. Имя `Label1` надписи, установленное системой по умолчанию, позже будет Вами изменено.

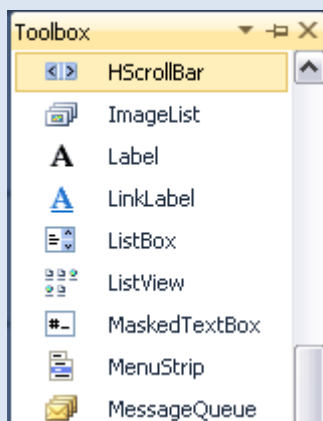


Рис. 5. Окно управляющих элементов

7. Чтобы размеры надписи можно было изменять, установите в окне свойств надписи значение свойства `AutoSize` (автоматический размер) равным `False`. Для этого достаточно дважды щелкнуть в окне свойств на названии этого свойства. При таком значении свойства `AutoSize` надписи ее размер можно изменять вручную. Если же свойство `AutoSize` имеет значение `True`, то размер надписи будет изменяться автоматически в зависимости от длины строки, являющейся значением свойства `Text` (текст) этой надписи.

Перетащите надпись мышью в центр окна формы и поочередным перетаскиванием мышью границ надписи и формы измените их размеры. Результат должен примерно соответствовать рис. 6. Размер шрифта и выравнивание текста по центру Вы установите позже.

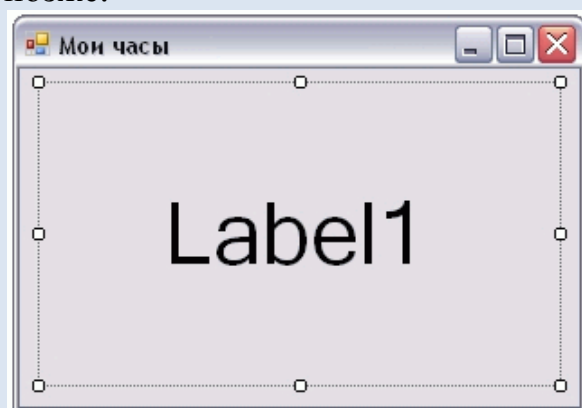


Рис. 6. Вид формы с надписью

8. Второй шаг разработки проекта – это задание значений свойств объектов, входящих в проект. В проекте два объекта: форма и надпись. Выделите форму. Для этого нужно на ней щелкнуть. В окне свойств выделенного объекта прокрутите список свойств, чтобы видеть свойство `Text`. Это свойство определяет строку, которая видна в заголовке формы. Замените значение `Form1` свойства `Text` новым значением `Мои часы`.

9. Выделите надпись. Измените имя надписи. Обычно имя объекта начинается с префикса, состоящего из трех символов. Префикс несет информацию о типе объекта. Вторая часть имени объекта должна напоминать его назначение в проекте. Имя надписи задает свойство `Name`. Замените значение `Label1` этого свойства надписи значением `lblВремя`.

10. Сделайте более крупным текст, отображаемый на надписи. Параметры текста определяет свойство Font (шрифт). Выберите это свойство в окне свойств надписи. Щелчком на появившейся в правой части этой же строки кнопке с изображенными на ней тремя точками откройте окно Шрифт (рис. 7). В поле Размер: задайте значение 36. Щелчком на кнопке ОК закройте окно Шрифт.

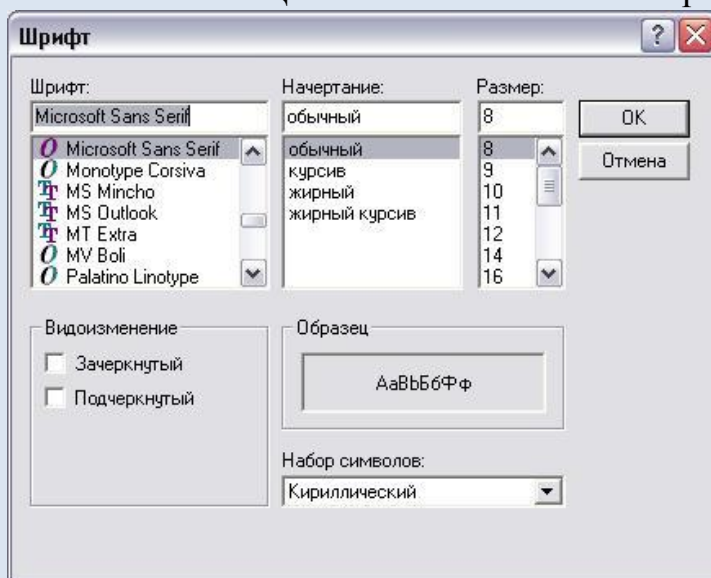




Рис. 7. Управление параметрами текста

11. Выполните выравнивание текста надписи по ее центру. Вид выравнивания определяет значение свойства TextAlign (выравнивание текста). Двойным щелчком на названии этого свойства изменяйте его значение, пока оно не примет значение MiddleCenter. При этом текст на надписи будет выровнен по центру, как по вертикали, так и по горизонтали.

После выполнения всех этих действий внешний вид Вашей формы должен полностью соответствовать рис. 6.

3. Разработка программного кода

12. Перейдите к третьему шагу разработки проекта – составлению программного кода. Несмотря на то, что Вы еще не вводили программный код, система уже сгенерировала его основу. Чтобы увидеть окно программного кода формы следует на панели инструментов окна обозревателя решений (рис. 3) щелкнуть на кнопке  View Code (просмотреть код). Откроется окно с программным кодом формы, показанное на рис. 8. Как видите, там уже есть две строки. Системой подготовлена заготовка класса Form1. Весь код формы, который Вы создадите, должен находиться между этими двумя строками.

13. Щелкнув на кнопке  View Designer (просмотреть конструктор) на панели инструментов окна обозревателя решений, Вы вернетесь к отображению формы.

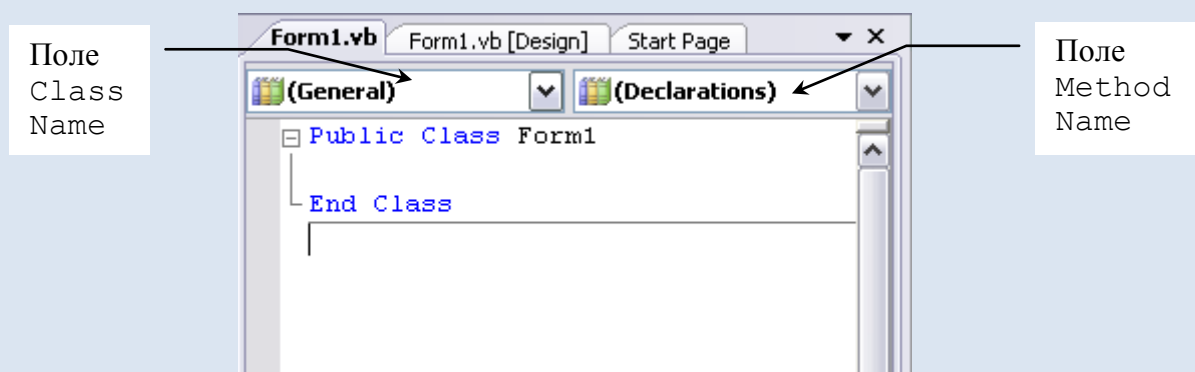


Рис. 8. Заготовка программного кода, созданного системой

Теперь займемся решением вопроса, как можно при выполнении приложения обеспечить отображение на надписи текущей даты и времени? При выполнении приложения надпись отображает строку, которая является значением свойства `Text` надписи. Следовательно, для обеспечения отображения в надписи текущей даты и времени, при выполнении проекта следует обеспечить обновление значения свойства надписи `Text`. В VB имеется стандартная функция `Now`, обращение к которой возвращает текущую дату и время. Этой функцией можно воспользоваться для решения поставленной задачи.

В VB имеются событийные подпрограммы. Каждая такая подпрограмма с одной стороны связана с конкретным управляющим элементом, и с другой стороны связана с некоторым событием. Выполняется событийная подпрограмма только в том случае, когда с управляющим элементом, с которым она связана, произойдет событие, с которым она также связана. Такие подпрограммы называются обработчиками события.

Нужно, чтобы свойство `Text` надписи получало значение функции `Now` при каждом щелчке на надписи в процессе выполнения проекта. При щелчке на надписи с ней происходит событие `Click` (щелчок). Следовательно, в программный код проекта следует включить событийную подпрограмму, связанную с надписью и с событием `Click`. Имя событийной подпрограммы всегда состоит из имени объекта, с которым она связана, знака подчеркивания, и имени события, с которым она также связана.

Итак, в программный код проекта следует добавить событийную подпрограмму `lblВремя_Click`.

Заготовку событийной подпрограммы, состоящую из начальной строки `Private Sub ...` и конечной строки `End Sub` удобно создавать с помощью системы. Для этого сначала нужно выделить имя управляющего элемента в поле `Class Name` (имя класса). Это поле находится в левом верхнем углу окна кода (рис. 8). Затем при выбранном объекте в поле `Method Name` (имя метода) нужно выбрать имя нужного события. Это поле находится в правом верхнем углу окна кода (рис. 8). Так можно поступать всегда. Но одно событие предусмотрено для управляющего элемента по умолчанию. Поэтому, для создания обработчика именно этого события, предусмотренного по умолчанию для данного управляющего элемента, достаточно просто выполнить двойной щелчок на этом управляющем элементе.

14. Вставьте в проект заготовку событийной подпрограммы `lblВремя_Click`. Для этого дважды щелкните на надписи. Обработка события предусмотрена для надписи по умолчанию.

15. В появившейся в окне программного кода в заготовке подпрограммы `lblВремя_Click` между заголовком и конечной строкой подпрограммы добавьте единственную инструкцию, которая свойству `Text` надписи задает значение функции `Now`:

```
lblВремя.Text = Now
```


Такая инструкция называется инструкцией присвоения (или присваивания). Ее действие состоит в том, что записанное слева от знака равенства свойство `Text` надписи `lblВремя` получит значение, возвращаемое функцией `Now`.



Обратите внимание, в инструкции присваивания слева от знака равенства всегда стоит объект, получающий значение, а справа от знака равенства должно стоять правило, по которому это значение вычисляется. Менять местами левую и правую часть равенства нельзя.


Программный код, который в итоге должен получиться показан на листинге 1. В этом листинге слишком длинная первая инструкция подпрограммы `lblВремя_Click` с помощью знака переноса (сочетание пробели и символа подчеркивания) для удобочитаемости разбита на две части и записана в двух строках кода.

Листинг 1. Программный код проекта

```
Public Class Form1
    Private Sub lblВремя_Click(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles lblВремя.Click
        lblВремя.Text = Now
    End Sub
End Class
```

16. Наконец Вы можете запустить свой проект и проверить его работу. Для запуска приложения щелкните на кнопке  `Start Debugging` (начать отладку) стандартной панели инструментов. Проверьте, как обновляются показания времени при щелчке на надписи. Если дата и время не полностью отображаются надписью, то ее размер оказался недостаточным и должен быть увеличен.

17. Для остановки работы проекта следует закрыть окно формы выполняющегося проекта щелчком на кнопке  `Заккрыть`. Второй способ – это щелкнуть на кнопке  `Stop Debugging` (остановить отладку), находящейся на стандартной панели инструментов. Остановите выполнение проекта. Если нужно, подкорректируйте размеры надписи и формы и вновь проверьте работу проекта.


18. Сохраните проект. Снова загляните в папку, где хранятся файлы проекта. Откройте папку `Часы\bin\Debug`. В ней появился файл  `Часы.exe`. Это выполняемый файл созданного Вами приложения. Он может выполняться вне среды Visual Studio 2010.

19. Закройте систему Visual Studio 2010. Для этого выполните команду меню `File` (файл), `Exit` (выход).

20. Запустите файл Часы.exe двойным щелчком на нем. Проверьте работу созданного Вами приложения.

21. Снова запустите систему Visual Studio 2010. Теперь на стартовой странице в списке последних загружаемых проектов в верхней строке должно быть отображено имя Вашего проекта Часы. Щелкните на этом имени. Среда IDE будет готова к продолжению работы с проектом.

4. Усовершенствование проекта

Не совсем удобно, что часы обновляют свои показания лишь при щелчке на надписи. Будет удобнее, если часы будут обновлять свои показания через определенные промежутки времени, например, через каждую секунду. Для решения этой задачи можно применить управляющий элемент  таймер, который генерирует событие Tick с периодичностью, которую можно задать.

22. Добавьте на форму таймер. Он появится в окне конструирования формы не внутри формы, а снизу. Таймер при выполнении проекта не будет виден. Его назначение – периодически генерировать событие Tick. Временной интервал между соседними событиями задается в миллисекундах значением свойства Interval (интервал) таймера.

23. Замените заданное системой по умолчанию имя таймера Timer1 именем tmrСекунда. Для этого свойству Name таймера задайте значение tmrСекунда.

24. Задайте свойству Interval значение 1000. В этом случае событие Tick будет происходить с интервалом в одну секунду.

25. Дважды щелкните на свойстве Enabled (включение) таймера, чтобы его значением стало True. Иначе таймер будет отключен, и не будет генерировать события Tick.

26. Для включения в программный код проекта событийной подпрограммы tmrСекунда_Tick дважды щелкните на таймере. Заготовка подпрограммы tmrСекунда_Tick появится в программном коде проекта.

27. Теперь нужно сделать так, чтобы содержание надписи lblВремя обновлялось при каждом наступлении события Tick. Выделите в подпрограмме lblВремя_Click инструкцию lblВремя.Text = Now и отбуксируйте ее мышью в подпрограмму tmrСекунда_Tick, поместив ее между заголовком и конечной строкой заготовки этой подпрограммы.

28. Надобность в подпрограмме lblВремя_Click теперь отпала. Выделите ее и удалите. Введенный Вами программный код после выполнения всех этих изменений показан на листинге 2.

Листинг 2. Программный код проекта с таймером

```
Public Class Form1
    Private Sub tmrСекунда_Tick(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles tmrСекунда.Tick
        lblВремя.Text = Now
    End Sub
End Class
```

29. Удалите текст на надписи. Для этого выделите надпись, двойным щелчком на свойстве надписи `Text` выделите значение этого свойства `Label1` и удалите его клавишей **Delete**.

30. Сохраните проект.

31. Запустите проект. Проверьте его работу.

32. Сохраните проект до следующего занятия. Этот проект Вам еще потребуется. Имейте в виду, на пользовательском диске *d* не гарантируется сохранение Ваших файлов, если после Вас там работали другие пользователи. Для сохранения Ваших файлов до следующего занятия копируйте свою рабочую папку на собственный съемный диск (флэш-панель). При отсутствии собственного съемного диска можно копировать свою рабочую папку на сетевой диск *o* в папке Вашей учебной группы.

33. Попробуйте ответить на вопросы для контроля.

34. Продемонстрируйте преподавателю работу Вашего проекта.

35. Закройте свой проект и систему Visual Studio 2010.

36. Удалите на диске *d* свою рабочую папку.

5. Вопросы для контроля

1. Как запустить среду Visual Studio 2010?
2. Что означает аббревиатура IDE?
3. Покажите, где находится: меню, стандартная панель инструментов, окно конструктора форм, окно обозревателя решений, окно свойств?
4. Какие кнопки применяются для открытия окон: обозревателя решений, свойств, управляющих элементов, конструктора форм, программного кода?
5. Как сохранить проект?
6. Как управляющий элемент добавляется на форму?
7. Что определяет свойство `Text` формы?
8. Каково назначения управляющего элемента `Label`?
9. На что влияет значение свойства `AutoSize` надписи?
10. Что задает свойство `Name` надписи?
11. Что задает свойство `Font` надписи?
12. На что влияет значение свойства `TextAlign` надписи?
13. В чем по отношению к надписи состоит событие `Click`?
14. Как запустить проект?
15. Какими способами можно прекратить выполнение проекта?
16. Каково назначение управляющего элемента `Timer`?
17. В чем состоит событие `Tick`?
18. Что зависит от значения свойства `Interval` таймера?
19. Что зависит от значения свойства `Enabled` таймера?
20. Как запустить выполнение проекта вне среды IDE.