Компонент **Delphi Timer** очень простой компонент, который не виден на экране, но, тем не менее,**Timer Delphi** выполняет очень важные функции в программе. **Delphi Timer** позволяет вводить необходимые задержки между выполнением тех или иных действий.  
  
   Компонент **Timer** имеет всего четыре свойства и одно событие, и работать с компонентом **Delphi Timer** очень просто.

|  |  |
| --- | --- |
| Delphi Timer |  |
| **СВОЙСТВО** | **НАЗНАЧЕНИЕ** |
| **Enabled** | "Включение-выключение" таймера |
| **Interval** | Интервал срабатывания (в миллисекундах) |
| **Name** | Имя компонента в программе |
| **Tag** | Произвольный числовой параметр |

   Помещаем компонент **Delphi Timer** на Форму. Задаём в свойстве Interval нужный интервал времени (измеряется в миллисекундах). Переходим на вкладку **Events** и видим единственное событие, поддерживаемое компонентом Delphi Timer: **OnTimer**. Выполнив по нему двойной щелчёк, или также двойной щелчёк по самому компоненту, мы попадём в сформированный средой Delphi обработчик события, где и введём код, предусматривающий выполнение тех или иных действий.  
  
   Так как по умолчанию сойство **Enabled** установлено в **True**, то в программе через установленный в свойстве **Interval** промежуток времени таймер сработает, то есть выдаст событие **OnTimer**. Будут выполнены необходимые действия. Иногда же запланированные действия должны произойти не автоматически при старте программы, а при выполнении каких-либо других действий. В этом случае необходимо свойство **Enabled** в Инспекторе Объектов установить в **False**. Затем в необходимый момент нужно выполнить команду:  
  
Timer1.Enabled:=True;  
  
   Учтите, что пока **Timer1.Enabled** равно **True**, компонент продолжит генерировать событие**OnTimer** по истечении каждого промежутка времени, равного значению свойcтва **Interval**. Поэтому, если нужно только единичное срабатывание, то таймер нужно остановить, причём сразу же:  
  
**procedure** TForm1.Timer1Timer(Sender: TObject);  
**begin**  
  Timer1.Enabled:=False;  
  { Необходимые действия }  
**end**;   
  
   Если же запрограммировать "выключение" таймера после выполнения предусмотренных в программе действий, то при достаточно малом по сравнению с продолжительностью необходимых действий значении свойства **Interval** таймер вновь сработает, и это может помешать ходу программы.  
  
   Компонент **Delphi Timer** не является очень точным и не подходит для измерения малых промежутков времени. Его точность порядка 50 миллисекунд. В качестве примера приведу простую программу, отображающую текущее время.  
  
   На форме будут только компонент **[Label](http://www.delphi-manual.ru/label.php" \o "Компонент Delphi Label)** и собственно наш компонент **Delphi Timer**. Свойство Timer можно оставить равным 1000, но раньше на более медленных компьютерах приходилось отображать время несколько раз в секунду, так как из-за влияния других выполняемых компьютером процессов выводимые раз в секунду показания "плавали", что было хорошо заметно на глаз. Размер шрифта возьмём побольше - 50, и подберём более гладкий, я взял **Bell MT**. Растянем на всю Форму: **Align=alClient**. Ну и, собственно, сам код:  
  
**procedure** TForm1.Timer1Timer(Sender: TObject);  
**begin**  
  Label1.Caption:=TimeToStr(Now);  
**end**;   
  
  Вот и всё! Delphi **Now** - это системная функция, возвращающая текущую дату-время в соответствующем формате **TDateTime**. Если преобразовывать её в строку функцией **TimeToStr**, то она вернёт текущее время, если **DateToStr**, то текущую дату. Если мы хотим отображать в нашей программе и дату тоже, достаточно поставить ещё один компонент **Label**, и функцией **DateToStr**передавать в него дату.