Автор: Андрей
31.03.2009 16:16 - Обновлено 04.07.2009 13:56

[Описание программы](#)
[Расположение компонентов на форме](#)
[Свойства компонент](#)
[Код программы](#)
[Код с подробными комментариями](#)
[Советы по улучшению и расширению программы](#)

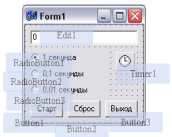
Описание программы

Это приложение выполняет функции секундомера. При этом возможен выбор шага, с которым производятся измерения - секунда, одна десятая или одна сотая секунды. При нажатии соответствующих кнопок выполняется запуск/приостановка отсчета, сброс времени на ноль (если отсчет был запущен, то он продолжится, но уже начиная с нулевого значения), выход из приложения. При смене величины шага расчет останавливается, время сбрасывается на ноль.

Никаких особых идей в базовой версии программы не используется, только простейшая работа с компонентами.

Используются следующие компоненты: Button, Edit, RadioButton, Timer.

Расположение компонентов на форме



Свойства компонент, измененные по сравнению со стандартными

Button1:
Caption: "Старт"
Button2:
Caption: "Сброс"
Button3:
Caption: "Выход"
Edit1:
Text: "0"
RadioButton1:
Caption: "1 секунда"
Checked: true

RadioButton2:
Caption: "0,1 секунды"
RadioButton3:
Caption: "0,01 секунды"

Код программы

```
    double TotalTime = 0;

void RadioButtonClick(int Step)
  {
  Form1 -> Timer1 -> Enabled = false;
  TotalTime = 0.0;
  Form1 -> Edit1 -> Text = "0";
  Form1 -> Timer1 -> Interval = Step;
  Form1 -> Button1 -> Caption = "Старт";
  }
//---------------------------------------------------------------------
void __fastcall TForm1::Button1Click(TObject *Sender)
{
Timer1 -> Enabled = !(Timer1 -> Enabled);
if (Timer1 -> Enabled == true)
  Button1 -> Caption = "Пауза";
else
  Button1 -> Caption = "Старт";
}
//---------------------------------------------------------------------
void __fastcall TForm1::Button2Click(TObject *Sender)
{
TotalTime = 0.0;
}
//---------------------------------------------------------------------
void __fastcall TForm1::Button3Click(TObject *Sender)
{
exit(0);
}
//---------------------------------------------------------------------
void __fastcall TForm1::Timer1Timer(TObject *Sender)
{
TotalTime += 0.001 * Timer1 -> Interval;
Edit1 -> Text = TotalTime;
}
//---------------------------------------------------------------------
void __fastcall TForm1::RadioButton1Click(TObject *Sender)
```

```
{
RadioButtonClick(1000);
}
//---------------------------------------------------------------------
void __fastcall TForm1::RadioButton2Click(TObject *Sender)
{
RadioButtonClick(100);
}
//---------------------------------------------------------------------
void __fastcall TForm1::RadioButton3Click(TObject *Sender)
{
RadioButtonClick(10);
}
```

Код программы с комментариями

```
    double TotalTime = 0; //Отмеренное время в секундах

void RadioButtonClick(int Step) //Функция, вызываемая компонентами
//RadioButton - при смене величины шага.
//Аргумент - устанавливаемая величина шага в миллисекундах
  {
  Form1 -> Timer1 -> Enabled = false; //Таймер останавливается
  TotalTime = 0.0; //Время сбрасывается на ноль
  Form1 -> Edit1 -> Text = "0"; //В поле вывода Edit1 отображается
//сброс на ноль
  Form1 -> Timer1 -> Interval = Step; //Изменяем шаг измерения таймера
  Form1 -> Button1 -> Caption = "Старт"; //Устанавливаем надпись
//на кнопке - т.к. таймер остановлен
  }
//---------------------------------------------------------------------
void __fastcall TForm1::Button1Click(TObject *Sender)
//Если кнопка нажата - событие OnClick
{
Timer1 -> Enabled = !(Timer1 -> Enabled); //Меняем состояние таймера -
//запущен/остановлен...
if (Timer1 -> Enabled == true)
//...и соответствующим образом надпись на кнопке
  Button1 -> Caption = "Пауза";
else
  Button1 -> Caption = "Старт";
}
//---------------------------------------------------------------------
void __fastcall TForm1::Button2Click(TObject *Sender)
```

```
{
TotalTime = 0.0; //Сбрасываем время на ноль
}
//---------------------------------------------------------------------
void __fastcall TForm1::Button3Click(TObject *Sender)
{
exit(0); //Выход из приложения
}
//---------------------------------------------------------------------
void __fastcall TForm1::Timer1Timer(TObject *Sender)
//Выполняется каждый раз, когда таймер отсчитывает
//установленный интервал - событие OnTimer
{
TotalTime += 0.001 * Timer1 -> Interval; //Увеличиваем
//отмеренное время в секундах на величину шага
Edit1 -> Text = TotalTime; //Выводим в поле вывода текущее время
}
//---------------------------------------------------------------------
void __fastcall TForm1::RadioButton1Click(TObject *Sender)
//Если выбирается компонент RadioButton - событие OnClick
{
RadioButtonClick(1000); //Вызываем функцию установки шага - на 1000 мс
}
//---------------------------------------------------------------------
void __fastcall TForm1::RadioButton2Click(TObject *Sender)
{
RadioButtonClick(100); //Вызываем функцию установки шага - на 100 мс
}
//---------------------------------------------------------------------
void __fastcall TForm1::RadioButton3Click(TObject *Sender)
{
RadioButtonClick(10); //Вызываем функцию установки шага - на 10 мс
}
```

Советы по улучшению и расширению программы

У программы, код которой приведен здесь, есть недостаток, заключающийся в том, что время в ней отображается исключительно в секундах, а перевод в формат часы-минуты-секунды не реализован. Конечно, это не будет иметь особого значения когда вы только пишете программу и делаете тестовый запуск - если вы не заняты нечем другим, то смотреть на секундомер несколько минут не отрываясь вы все равно не будете. Однако если вы уже написали программу, зашли на кухню, поставили вариться пельмени и щелкнули по кнопке "Старт", желая не пропустить момент готовности, то информация о том, что ваш ужин варится уже 1024,32 секунды может вас изрядно

озадачить. Поэтому дописать функцию для внятного отображения времени будет полезно для практического использования.

Прежде всего необходимо определить три переменные типа int - назовем их hours, minutes, seconds (да, секунды тоже будут целым числом, а почему - сказано чуть ниже). Затем с помощью несложных арифметических действий из числа секунд получаются отдельно число часов, минут и секунд. Округление при этом производится функцией floor из библиотеки math. После чего остается лишь преобразовать три отдельных числа в пригодную для вывода строку. Выглядит это следующим образом:

```
Edit1 -> Text = AnsiString(hours) + ":" + AnsiString(minutes) + ":" + AnsiString(seconds);
```

AnsiString - строковой тип (точнее - класс), а AnsiString() - функция-конструктор, возвращающая значение типа AnsiString на основании переданных аргументов. Оператор сложения в данном случае означает объединение двух или более строк.

Вставив соответствующий код в функцию Timer1Timer, получаем то, что хотели - вывод будет производить в формате чч:мм:сс.

Теперь о том, почему же секунды должны содержаться в целой переменной.

Во-первых, не настолько уж часто вам вообще могут понадобиться доли секунды, и если уж имеет смысл выводить минуты, а то и часы, то вряд ли на таких масштабах одна сотая секунды будет важна. Во-вторых, и это здесь главное - **при маленьких значениях шага** время выполнения программой инструкций, указанных в функции таймера, становится сравнимым с величиной этого шага, что может внести существенные поправки и, следовательно, **значительно повлиять на точность измерения**

.