

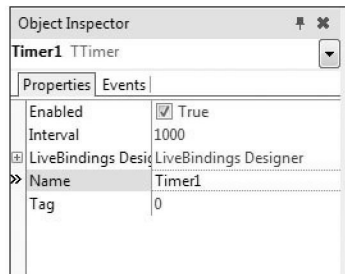
타이머^{Timer}

특정 시간 간격을 두고 실행되도록 프로그래밍할 수 있게 됩니다.
타이머를 사용하여 반짝이는 별들을 표현할 수 있습니다.

프로그래밍을 하다 보면, 프로그램 안에 있는 어떤 문장이 특정 시간 중에만 실행되어야 하는 경우가 있다. 실행되고 나서 5초가 지나면 폼을 빨강으로 칠해야 한다고 생각해보자.

이 작업을 완수하려면, *Timer*^{타이머} 컴포넌트를 사용한다. *Timer*는 *System* 그룹 안에 있다.

*Timer*를 폼 위 아무 곳이나 올려 놓는다. (프로그램이 실행되는 동안 즉, 런타임에는 눈에 보이지 않는다) 이 컴포넌트가 가진 프로퍼티들을 오브젝트 인스펙터에서 살펴보자.



Name 프로퍼티 뿐만 아니라, *Enabled*와 *Interval* 프로퍼티를 잘 보아두자. 만약 *Enabled*가 *True*이면, 타이머가 작동된다. *False*이면, 이 타이머는 꺼진 상태이다. *Interval* 프로퍼티에는 타이머가 얼마나 오래 기다렸다가 반응할 것인지를 밀리초 단위로 적는다. 기본값은

1,000 즉 1초이다.

Interval 값을 5,000 (즉, 5초)으로 설정하자.

폼에 있는 타이머를 더블 클릭하면 이 타이머의 *Timer* 이벤트핸들러가 만들어진다. 우리는 폼의 색상을 변경하는 구문을 이벤트핸들러 안에 넣을 수 있다. 아래와 같다.

```
procedure TForm1. Timer1Timer(Sender: TObject);
begin
    form1.Color:= clRed;
end;
```

프로그램을 실행하면 얼마 후 폼이 빨강으로 칠해지는 것을 보게 된다.

이 프로그램을 조금만 고쳐보자. 버튼을 하나 올리고 버튼 클릭 이벤트핸들러에 다음과 같은 구문을 추가한다.

```
procedure TForm1.Button1Click(Sender: TObject);
begin
    Timer1.Enabled:= True;
end;
```

오브젝트 인스펙터에서 *Timer1*의 *Enabled* 프로퍼티를 *False*로 설정한다.

이제 프로그램을 실행해도 아무 일도 생기지 않는다. 하지만, 버튼을 클릭하고 나서 5초가 지나면, 폼이 빨강으로 칠해진다. 즉, 이 버튼을 클릭하면, *Enabled* 프로퍼티가 *True* 설정 되므로 *Timer1*이 켜진다. 그리고 설정된 *Interval* 값인 5초 후에 *Timer1*의 *Timer* 핸들러가 실행된다.

Timer 핸들러를 다음과 같이 바꿔보자.

```
procedure TForm1.Timer1Timer(Sender: TObject);
begin
    form1.Color:= 256*256*Random(256)+256*Random(256)+Random(256);
end;
```

이제, 우리는 무작위 색상을 할당하였다.

이벤트핸들러도 바꿔보자.

```
procedure TForm1.Button1Click(Sender: TObject);
begin
    Timer1.Enabled:= True;
    Timer1.Interval:= 1500;
end;
```

이제, Timer의 간격이 1.5초로 설정되었다.

프로그램을 실행하고 버튼을 클릭해보자. 어떤 결과가 나타나는가? 매 1.5초 마다 폼의 색상이 무작위로 변경된다. 타이머는 작동되는 동안 시간 간격마다 자신의 *Timer* 이벤트핸들러를 호출한다. 타이머를 멈추려면 *Enabled* 프로퍼티를 *False*로 설정해야 한다.

폼 안에 버튼 몇 개를 더 추가하고 버튼 클릭이벤트핸들러를 작성해보자.

```
procedure TForm1.Button2Click(Sender: TObject);
begin
    Timer1.Enabled:= False;
end;
```

이제 이 첫 번째 버튼을 클릭하면 색상 변경이 시작되고, 두 번째 버튼을 클릭하면 멈춘다.

타이머가 단 한번만 실행되도록 하려면, *Timer* 이벤트핸들러 안에서 *Enabled* 프로퍼티를 *False*로 설정한다. 즉 1.5초 후에 단 한번만 색상 변경되고, 이후에는 변경되지 않도록 하려면, *Timer* 이벤트핸들러를 다음과 같이 변경해야 한다.

```
procedure TForm1.Timer1Timer(Sender: TObject);
begin
    form1.Color:= 256*256*Random(256)+256*Random(256)+Random(256);
    Timer1.Enabled:= False;
end;
```

시간이 지난 뒤에 타이머가 꺼지도록 하고 싶으면, 타이머를 하나 더 쓰면 된다. 예를 들어, 20 초 후에 색상 변경이 중단되도록 하려면, 지금 있는 타이머를 다루기 위한 타이머를 새로 추가한다. (*Enabled* 프로퍼티를 *False*로 설정하는 것을 잊지 말자) 그리고 이 두 번째 타이머의 이벤트핸들러를 작성해보자.

```
procedure TForm1.Timer2Timer(Sender: TObject);
begin
    Timer1.Enabled:=False;
    Timer2.Enabled:=False;
end;
```

첫 번째 버튼의 클릭 이벤트는 다음과 같이 변경한다.

```
procedure TForm1.Button1Click(Sender: TObject);
begin
    Timer1.Enabled:=True;
    Timer1.Interval:=1500;
    Timer2.Enabled:=True;
    Timer2.Interval:=20000;
end;
procedure TForm1.Timer1Timer(Sender: TObject);
begin
    form1.Color:= 256*256*Random(256)+256*Random(256)+Random(256);
end;
```

이 프로그램을 실행한다. 첫 번째 버튼을 클릭하면, 1.5초 간격으로 폼의 색상이 변경되다가 20초 후에는 변경되는 것을 멈춘다.

시간을 이용하면, 움직이는 오브젝트^{Object}를 만들 수 있다. 예를 들어 텍스트박스를 옮겨 보자.

폼에 텍스트박스 하나를 올린다. *Timer1*의 *Timer* 이벤트핸들러를 다음과 같이 변경한다.

```
procedure TForm1.Timer1Timer(Sender: Object);
begin
    edit1.Left:=edit1.Left+15;
end;
```

프로그램을 실행하고 첫 번째 버튼을 클릭한다. 텍스트박스가 왼쪽에서 오른쪽으로 움직이는 것을 볼 수 있다.

즉, 타이머를 이용하면, 프로그램이 실행되는 중에 *Interval*에 설정된 시간에 따라 동적으로 파라미터를 변경하거나 움직이는 오브젝트^{Object} 만들기 등을 할 수 있다.

실습

Exercise 1.

타이머를 이용하여, 별이 빛나는 밤하늘을 그리는 프로그램을 적어보자. 별들은 빛났다가 얼마 후에 사라지도록 한다. (별 모양 대신, 원이나 삼각형 같은 기본 도형을 사용해도 좋다)

Exercise 2.

제한 시간 표시기를 만드는 프로그램을 적어보자.

허용할 제한 시간의 초 단위 값을 *TEdit*에서 받는다. 버튼을 클릭하면, *Timer*가 초단위로 카운트다운 한다. 시간이 0이 되자마자, *TEdit*의 배경색이 빨강으로 되고, *Timer*는 제한 시간을 0에서부터 다시 위로 세기 시작한다.

Exercise 3.

프로그램을 적어서 디지털 시계를 만들어 보자. 이 시계는 시간, 날짜, 요일을 보여준다.

Exercise 4.

바늘 시계를 흉내 낸 프로그램을 적어보자. 초침은 1초마다 이동한다. 분침은 1분마다, 시침은 1시간 마다 움직인다. (만약 시침 즉 시간을 가리키는 바늘이 부드럽게 이동하게 만들고 싶다면, 시침을 1시간 마다 움직이지 말고, 시간에서 지난 부분만큼을 미세하게 이동시키는 방법도 있다)