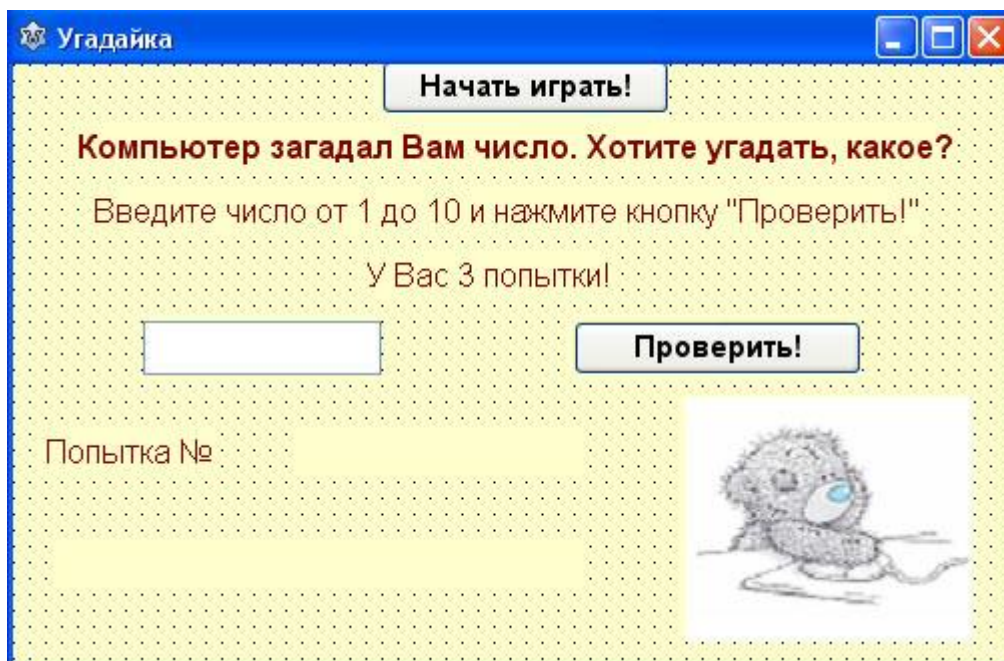


## LAZARUS



### Задание № 12. «Угадайка». Генератор случайных чисел

1. Запустите Lazarus.
2. Закройте свой последний проект. Создайте новый.
3. Создайте форму по образцу:



4. Поле ввода Edit1 и надписи Label5 и Label6 очистите от текста (сотрите всё в значении свойства Text для Edit1 и Caption для надписей).
5. Код кнопки «Играть»:

```
implementation
  {$R *.lfm}
var
  x, k, R: Integer;
{ TForm1 }
procedure TForm1.Button1Click(Sender: TObject);
begin
  Randomize;
  R := Random(10)+1; //случайное число от 1 до 10 включительно
  k := 1;
  Label6.Caption := ''; //очистка текста реакции на попытку
  Label5.Caption := IntToStr(k); //номер текущей попытки
  Edit1.Text := ''; //очистка текстового поля ввода числа
end;
```

Не забудьте описать переменные R – загаданное число, k – счетчик попыток, x – загаданное число.

## 6. Код кнопки «Проверить»:

```

30 .
. procedure TForm1.Button2Click(Sender: TObject);
. begin
.     Label5.Caption:=IntToStr(k);
.     x:=StrToInt(Edit1.Text);
55 If R < x Then
. begin
.     Label6.Caption:='Ваше число больше';
.     Edit1.Text:=''
. end;
60 If R > x Then
. begin
.     Label6.Caption:='Ваше число меньше';
.     Edit1.Text:=''
. end;
65 k:=k+1; //счётчик попыток
. If k=4 Then
. begin
.     Label6.Caption:='Вы проиграли. Загаданное число:' + IntToStr(R);
.     k:=1
70 end;
. If R=x Then
. begin
.     Label6.Caption:='Вы победили';
.     k:=1
75 end;
. end;

```

7. Запустите построение и отладку проекта. Проверьте его работу. Сохраните проект.
8. Попробуйте изменить проект так, чтобы количество попыток и предел для загадываемого числа мог задавать сам игрок.