

# ОСНОВЫ объектно-ориентированного программирования в среде



# LAZARUS

# Урок 10



# Циклы с параметром в Pascal

```
var i: integer;  
begin  
  for i:=1 to 10 do writeln('Иванов');  
end.
```

Начальное  
значение

Конечное  
значение

- ❑ **ЗАДАНИЕ 1:**  
Вывести на экран 10 раз свою фамилию.

# Циклы с параметром в Pascal

```
var i: integer;
begin
  for i:=1 to 10 do
  begin
    writeln('Иванов');
    writeln('Петров');
  end;
end.
```

- ❑ **ЗАДАНИЕ 2:**  
Вывести на экран 10 раз свою фамилию и фамилию друга.

# Циклы с параметром

Привет, Колосова!  
Привет, Колосова!

```
var i: integer;  
    fam: string;  
begin  
    read(fam);  
    for i:=1 to 10 do  
    begin  
        writeln('Привет, ', fam, '!');  
    end;  
end.
```

## ■ ЗАДАНИЕ 3:

Ввести с клавиатуры фамилию и вывести на экран 10 раз слова приветствия.

# Циклы с параметром в Pascal

```
var i: integer;  
begin  
  for i:=1 to 10 do  
  begin  
    writeln(i:2, ' x 7 = ', i*7);  
  end;  
end.
```

```
2 x 7 = 14  
3 x 7 = 21  
4 x 7 = 28  
5 x 7 = 35  
6 x 7 = 42  
7 x 7 = 49  
8 x 7 = 56  
9 x 7 = 63  
10 x 7 = 70
```

**ЗАДАНИЕ 4:**

**Вывести таблицу умножения на 7.**

# Циклы с параметром в Pascal

```
var i, n: integer;
begin
  read(n);
  for i:=1 to 10 do
  begin
    writeln(i:2, ' x ', n, ' = ', i*n);
  end;
end.
```

## □ ЗАДАНИЕ 5:

Вывести таблицу умножения на число, введенное с клавиатуры.

# Циклы с параметром в Pascal

```
var i, n, sum: integer;
begin
writeln('введите трехзначное число');
read(n);
sum:=0;
  for i:=1 to 3 do
  begin
    sum:=sum+(n mod 10);
    n:= n div 10;
  end;
writeln('Суммы цифр числа = ', sum);
end.
```

введите трехзначное число

234

Суммы цифр числа = 9

**ЗАДАНИЕ**  
**Вывести**  
**Входные**

# Циклы с параметром в Lazarus

Компоненты (фильтр)

- Form1: TForm1
  - Memo1: TMemo
  - Edit1: TEdit
  - Button1: TButton
  - Edit2: TEdit
  - Edit3: TEdit
  - Label1: TLabel
  - Label2: TLabel
  - Label3: TLabel
  - Button2: TButton

Изучаем циклы

Фамилия

Имя

Кол-во строк

Заполнить

Очистить

ScrollBars

ssAutoVertical

# ПОВТОРЯЕМ: Тmemo – многострочный текст

## Свойство Lines

```
memo1.lines [0] := ' Это первая строка';
```

```
memo1.lines [1] := ' Это вторая строка';
```

```
memo1.lines.add(' Это первая строка');
```

```
memo1.lines.add(' Это вторая строка');
```

## Метод Clear - очистить

```
memo1.Clear;
```

# Циклы с параметром в Lazarus

```
procedure TForm1.Button1Click(Sender: TObject);  
var i, n: integer;  
begin  
  If Edit3.Text<>' ' Then n:=StrToInt(Edit3.Text) Else n:=1;  
  for i:=1 to n do  
  begin  
    mem1.lines.add('Привет ' + Edit1.Text + ' ' + Edit2.Text + '!');  
  end;  
end;
```

```
procedure TForm1.Button2Click(Sender: TObject);  
begin  
  mem1.Clear;  
  Edit1.Clear; Edit2.Clear; Edit3.Clear;  
end;
```

# Урок 11



# Операторы DIV и MOD

**DIV** – целочисленное деление

$$534 \text{ DIV } 10 =$$

**MOD** – остаток от деления

$$534 \text{ MOD } 10 =$$

# Задачи на циклы с параметром

□ Вывести все двузначные числа, в которых вторая цифра больше первой.

*Выходные данные: 12 13 14 ...*

```
var i: integer;  
begin  
  for i:=10 to 99 do  
    begin  
      if i mod 10 > i div 10 Then Write( i:4|);  
    end;  
end.
```

# Задачи на циклы с параметром

□ Вывести все трехзначные числа, в которых вторая цифра больше первой, а третья больше второй.

*Выходные данные: 123 124 125 ...*

```
var i: integer;  
begin  
  for i:=100 to 999 do  
    begin  
      if (i mod 10 > i div 10 mod 10) and (i div 10 mod 10 > i div 100) Then Write( i:5);  
    end;  
  end.
```

# Задачи на циклы с параметром

- Вывести все трехзначные числа, в которых есть две одинаковые цифры.

*Выходные данные: 100 101 110 ...*

```
var i: integer;  
begin  
  for i:=100 to 999 do  
  begin  
    if (i mod 10 = i div 10 mod 10) or (i div 10 mod 10=i div 100) Then Write( i:5);  
  end;  
end.
```

- Вывести все трехзначные числа, в которых все три цифры одинаковые.

*Выходные данные: 111 222 333 444 ...*

# Задачи на циклы с параметром

□ Вывести все четырехзначные числа, которые являются палиндромами.

*Выходные данные: 1001 1111 1221 ...*

```
var i: integer;
begin
for i:=1000 to 9999 do
begin
if (i mod 10 = i div 1000) and (i div 10 mod 10= i div 100 mod 10 ) Then Write( i:5);
end;
end.
```

# Задачи на циклы с параметром

□ Автобусный билет считается счастливым, если в его шестизначном номере сумма первых трех цифр равна сумме последних трех цифр. Подсчитайте и выведите число счастливых билетов с различными номерами (билеты от 000001 до 999999).

```
var i, kol: integer;
begin
kol:=0;
for i:=1 to 999999 do
begin
if (i mod 10 + i div 10 mod 10 + i div 100 mod 10 = i div 1000 mod 10 + i div 10000 mod 10 + i div 100000 mod 10)
Then kol:=kol+1;
end;
writeln('kol= ', kol);
end.
```

# Задачи на циклы с параметром

- Form1: TForm1
  - Edit1: TEdit
  - Edit2: TEdit
  - Edit3: TEdit
  - Button1: TButton
  - Edit4: TEdit
  - Button2: TButton

Form1

Купить билет

Проверить

□ Счастливым билет?

# Задачи на циклы с параметром

```
var
```

```
  Form1: TForm1;
```

```
  a: integer;
```

---

```
implementation
```

```
  {$R *.lfm}
```

```
□ { TForm1 }
```

```
□ procedure TForm1.Button1Click(Sender: TObject);
```

```
□ begin
```

```
  randomize;
```

```
  edit1.text:=inttostr(random(900000)+100000);
```

```
  a:=strtoint(edit1.text);
```

```
  Edit4.Text:=' ' ;
```

```
end;
```

---

# Задачи на циклы с параметром

```
procedure TForm1.Button2Click(Sender: TObject);  
begin  
    if (a mod 10 + a div 10 mod 10 + a div 100 mod 10 = a div 1000 mod 10 + a div 10000 mod 10 + a div 100000 mod 10)   
        Then Edit4.Text:='Ура, счастливый!' Else Edit4.Text:='(( Нет (( '  
end;
```

# Задачи на циклы с параметром

Счастливый билет?

Сумма первых трех цифр больше ?

Номер билета - палиндром?

# Урок 12



# Условный цикл в Pascal

```
While <Условие> do  
begin  
  <Действия>;  
end;
```

Пока

условие  
истинно

# Условный цикл в Pascal

Вводится целое число  $n$ . Вывести сумму цифр данного числа.

```
var n, sum: integer;
begin
  read(n);
  sum:=0;
  While n>0 do
    begin
      sum:=sum+n mod 10;
      n:= n div 10;
    end;
```

# Условный цикл в Pascal

■ С помощью алгоритма Евклида найти НОД двух чисел.

1. Даны 2 числа  $N$  и  $M$ .

2. Сравнить данные числа и наибольшее из них заменить на разность большего и меньшего.

3. Повторять до тех пор, пока числа не станут равными.

# Условный цикл в Pascal

■ С помощью алгоритма Евклида найти НОД двух чисел.

1. Даны 2 числа  $N$  и  $M$ .

2. Сравнить данные числа и наибольшее из них заменить на разность большего и меньшего.

3. Повторять до тех пор, пока числа не станут равными.

# УСЛОВНЫЙ ЦИКЛ В PASCAL

```
var n, m: integer;
```

```
begin
```

```
  read(n); read(m);
```

```
  While n<>m do
```

```
    begin
```

```
      if m>n then m:=m-n else n:=n-m;
```

```
    end;
```

```
  writeln(m);
```

```
end.
```

# Работа с файлами в Lazarus

Компоненты, которые умеют работать с файлами:

- ✓ **ListBox**
- ✓ **ComboBox**
- ✓ **Memo**

Они читают и сохраняют своё содержимое, строки типа **String**, в файл текстового формата.

# Работа с файлами в Lazarus

Каждая строка компонентов

**ListBox** и **ComboBox** является объектом **Items[i]**

Каждая строка **Memo** является объектом **Lines[i]**

где **i** - номер строки, который отсчитывается от нуля.

```
ComboBox1.Items[0] := 'Первая строка  
изменилась';
```

```
Listbox1.Items[1] := 'Вторая строка изменилась';
```

# Работа с файлами в Lazarus

Добавление строк в компоненты выполняется методами **Add** и **Insert**

Метод **Add** добавляет новую строку в конец.

Метод **Insert** имеет дополнительный параметр, указывающий, после какой строки разместить новую строку.

# Работа с файлами в Lazarus

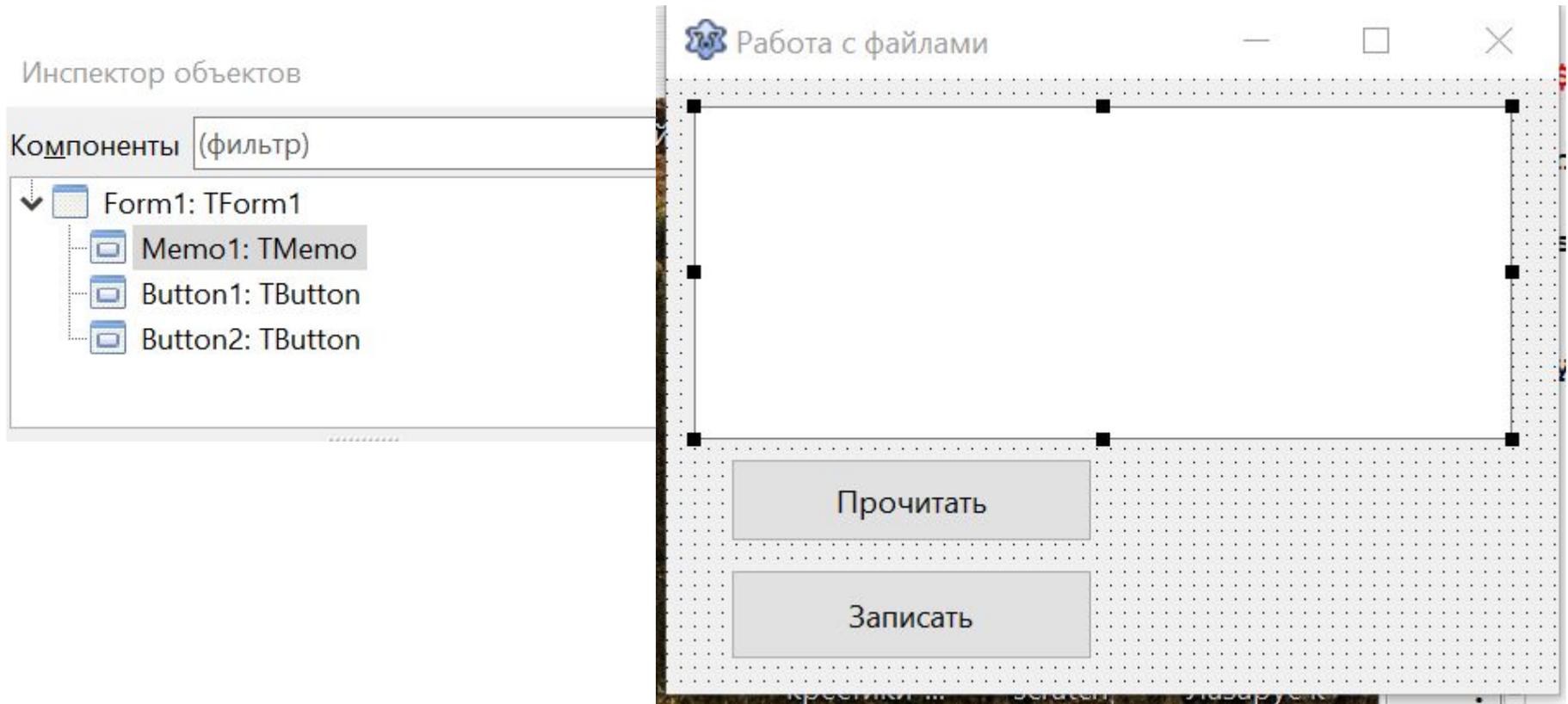
Сохранение содержимого в файл:

```
ListBox1.Items.SaveToFile('Имя_файла.txt');
```

Загрузка из файла:

```
ListBox1.Items.LoadFromFile('Имя_файла.txt');
```

# Работа с файлами в Lazarus



В папке **11\_работа с файлами** создать:

**input.txt**

**output.txt**

# Работа с файлами в Lazarus

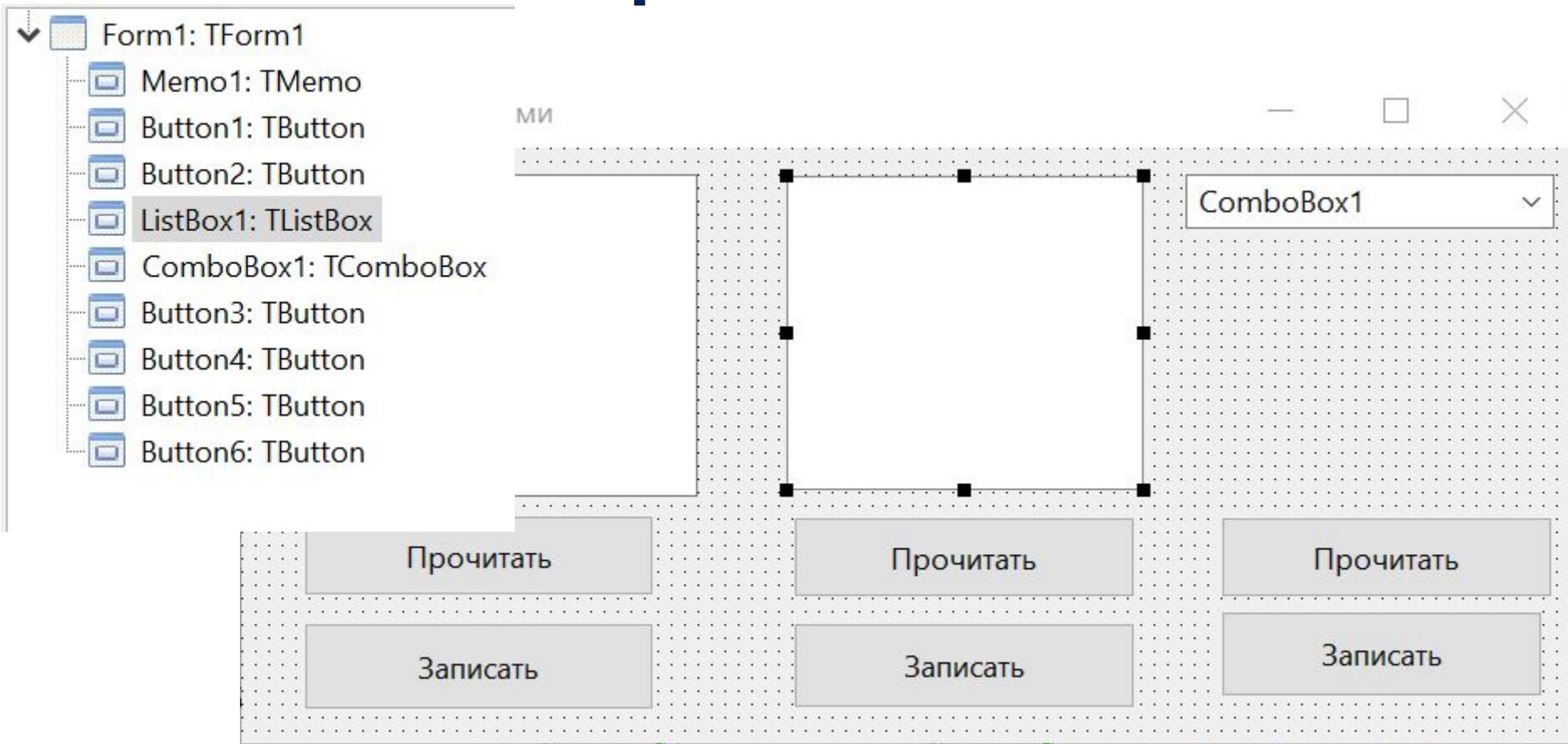
## Прочитать из файла в поле Мемо:

```
procedure TForm1.Button1Click(Sender: TObject);  
begin  
    mem1.Lines.LoadFromFile('input.txt');  
end;
```

## Записать в файл из поля Мемо:

```
procedure TForm1.Button2Click(Sender: TObject);  
begin  
    mem1.Lines.SaveToFile('output.txt');  
end;
```

# Работа с файлами в Lazarus



# Работа с файлами в Lazarus

## Прочитать из файла в поле ListBox:

```
procedure TForm1.Button3Click(Sender: TObject);  
begin  
    ListBox1.Items.LoadFromFile('input.txt');  
end;
```

## Записать в файл из поля ListBox :

```
procedure TForm1.Button4Click(Sender: TObject);  
begin  
    ListBox1.Items.SaveToFile('output.txt');  
end;
```

# Работа с файлами в Lazarus

## Прочитать из файла в поле ListBox:

```
procedure TForm1.Button5Click(Sender: TObject);  
begin  
    ComboBox1.Items.LoadFromFile('input.txt');  
end;
```

## Записать в файл из поля ListBox :

```
procedure TForm1.Button6Click(Sender: TObject);  
begin  
    ComboBox1.Items.SaveToFile('output.txt');  
end;
```

# Работа с файлами в Lazarus

## Прочитать из файла в поле ListBox:

```
procedure TForm1.Button5Click(Sender: TObject);  
begin  
    ComboBox1.Items.LoadFromFile('input.txt');  
end;
```

## Записать в файл из поля ListBox :

```
procedure TForm1.Button6Click(Sender: TObject);  
begin  
    ComboBox1.Items.SaveToFile('output.txt');  
end;
```