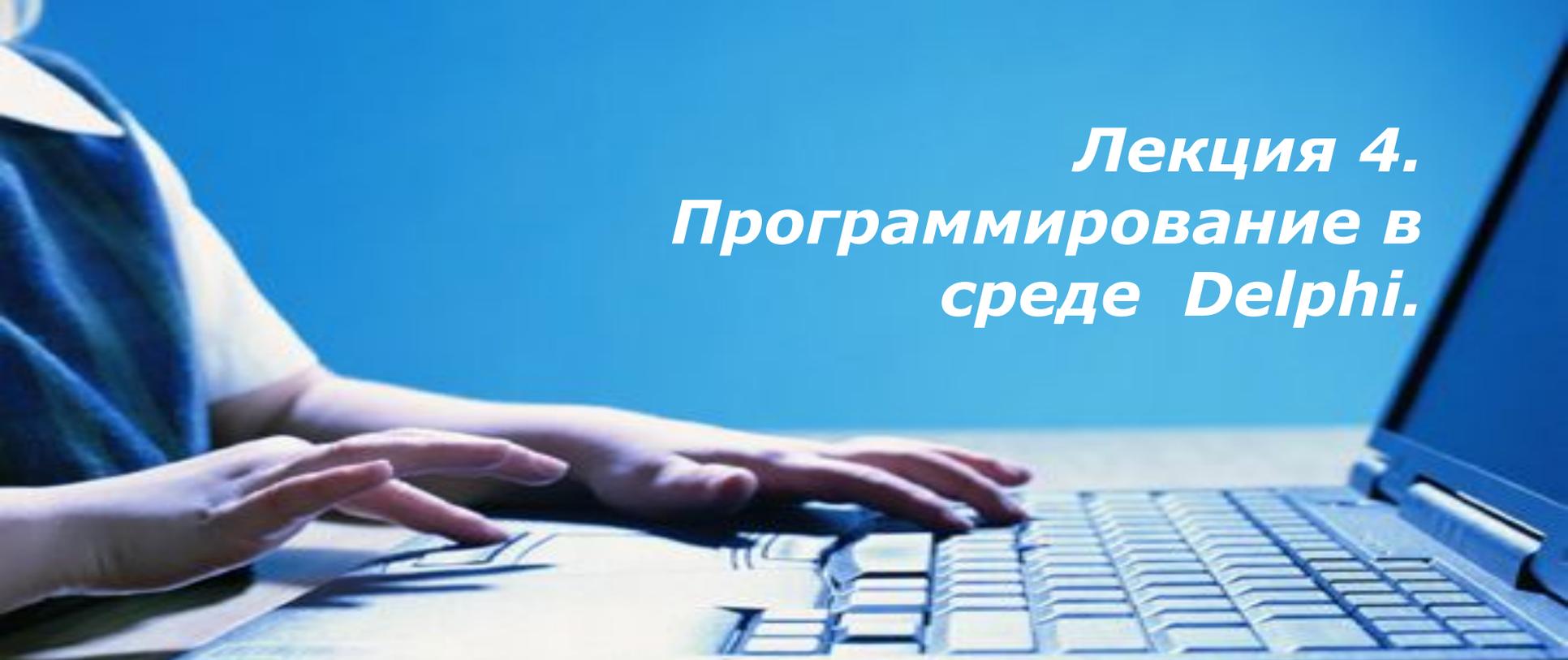


*Баранова Евгения
Васильевна,
профессор кафедры
информатики*

РГПУ им. А.И. Герцена

Дисциплина «Введение в информатику»



*Лекция 4.
Программирование в
среде Delphi.*

РГПУ им. А.И. Герцена

**Оператор присваивания.
Операторы ввода-вывода.**

Основные окна Delphi

❖ Окна среды Delphi.

The image shows the Delphi 7 IDE interface with several windows and components labeled in Russian:

- Главное окно** (Main window): Points to the main menu bar and toolbar area.
- Окно дерева объектов** (Object TreeView): Points to the window on the left showing the project structure.
- Окно инспектора объектов** (Object Inspector): Points to the window at the bottom left showing the properties of the selected form.
- Окно формы** (Form window): Points to the central design canvas.
- Окно кода программы** (Code window): Points to the code editor at the bottom of the IDE.

The Object Inspector window shows the following properties for Form1:

Property	Value
Action	
ActiveControl	
Align	alNone
AlphaBlend	False
AlphaBlendVal	255
☑ Anchors	[akLeft,akTop]
AutoScroll	True
AutoSize	False
BiDiMode	bdLeftToRight
☑ BorderIcons	[biSystemMen
BorderStyle	bsSizeable
BorderWidth	0
Caption	Form1

The code editor shows the following code:

```
var
```

Система программирования *Delphi*

◆ Структура описания процедуры

```
procedure TForm1.Button1Click(Sender:  
    TObject);  
    <блок описаний>  
begin  
    <блок реализации>  
end;
```



Оператор присваивания

- ❖ Оператор **присваивания** имеет вид:

<идентификатор > := < выражение >

- ❖ При выполнении оператора вычисляется значение указанного выражения и переменная, идентификатор которой указан в левой части оператора, получает это значение.
- ❖ Все операторы в программе отделяются символом ;



Оператор присваивания

- ❖ Переменная и выражение должны иметь **один и тот же тип**.
- ❖ Допускается только присваивание **вещественной** переменной **целого** значения.
- ❖ **Тип** переменной определяет **множество значений**, которые может принимать переменная, и совокупность **операций**, которые можно выполнять над этими значениями.



Арифметические типы данных

❖ Существует пять **целых** типов:

со знаком

shortint (-128..127, 8 бит)

integer (-32 768..32 767, 16 бит)

longint (-21 474 883 648..21 474 883 647,
32 бита);

без знака

byte (0..255,8 бит)

word (0..65 535,16 бит)



Арифметические типы данных

- ❖ **Вещественный** тип задает множество действительных чисел:
 - real** ($2.9 \cdot 10^{-39}$ - $1.7 \cdot 10^{38}$, 11 - 12 дес. цифр);
 - single** ($1.5 \cdot 10^{-45}$ - $3.4 \cdot 10^{38}$, 7 - 8 дес. цифр);
 - double** ($5.0 \cdot 10^{-324}$ - $1.7 \cdot 10^{308}$, 15 - 16 дес. цифр);
 - extended** ($1.9 \cdot 10^{-4951}$ - $1.1 \cdot 10^{4932}$, 19 дес. цифр).



Арифметические выражения

- ❖ **Арифметические выражения** содержат числовые константы и переменные арифметического типа, обращения к стандартным функциям и функциям пользователя с числовыми значениями, знаки арифметических операций, круглые скобки.
- ❖ К **арифметическим операциям** относятся: сложение (+), вычитание или унарный минус (-), умножение (*), деление (/), целочисленное деление (div), остаток от деления (mod). Две последние операции выполняются только над целыми значениями. Операции '+' и '-' имеют более низкий **приоритет** по сравнению с остальными арифметическими операциями. Для изменения последовательности действий используются **круглые скобки**.

Арифметические функции

- ❖ Sin Cos Tan Cot
- ❖ Abs Sqrt Sqr
- ❖ Exp Ln Log2 Log10

Аргументы в скобках! $Z := \text{sqrt}(a*a + b*b)$

Для вычисления степени x^y

$Z = x^y$?

$\ln z = \ln x^y$ $\ln z = y \ln x$ $z = e^{y \ln x}$

$Z := \text{exp}(y * \ln(x))$ $z := \text{power}(x, y)$ (Math)

Операторы ввода-вывода

❖ Вывод

Строковый тип данных **var s:string;**

S:='Моя первая программа в Delphi';

ShowMessage(s);

Для вывода арифметических значений
используются функции

преобразования

IntToStr (<выражение целого типа>)

FloatToStr(<выражение вещественного
типа>)

Операторы ввода-вывода

❖ Вывод

Если нужно вывести несколько строк, то они объединяются знаком **+**

```
ShowMessage(`У меня сегодня  
лекций: '+IntToStr(LectNum)+'  
начало в '+FloatToStr(LectTime));
```

Операторы ввода-вывода

❖ Вывод

```
var b:real;s:string;  
begin  
  b:=power(68,76);  
  showmessage('Если число 68 возвести  
    в степень 76, то получим значение  
    '+FloatToStr(b));  
end;
```

Система программирования Delphi

❖ Описание процедуры

```
procedure TForm1.Button1Click(Sender: TObject);  
// блок описаний  
var b:real;s:string;  
begin  
// блок реализации  
    b:=power(68,76);  
    showmessage('Если число 68 возвести в степень  
76, то получим значение '+FloatToStr(b));  
end;
```



Система программирования Delphi

- ❖ **Комментарий** - это пояснительный текст, который можно записать в одной строке программы, указав в начальных символах строки символы `//`. Текст комментария не анализируется транслятором.

```
procedure TForm1.Button1Click(Sender: TObject);  
// блок описаний  
var b:real;s:string;  
begin  
// блок реализации  
    b:=power(68,76);  
    showmessage('Если число 68 возвести в степень  
76, то получим значение '+FloatToStr(b));  
end;
```



Операторы ввода-вывода

❖ Ввод

Используется диалоговое окно, которое вызывается функцией

InputBox(<название окна ввода>, <текст у строки ввода>, <значение по умолчанию>)

Результат – строка.

Введем строку и распечатаем два раза через пробел.

Операторы ввода-вывода

❖ Ввод

```
var s:string;
```

```
begin
```

```
  s:=InputBox('Окно для ввода строк',  
              'Введите строку','');
```

```
  showMessage('А теперь строки уже  
  две: '+s+' '+s)
```

```
end; 1. Операторы ввода-вывода  
(строки)
```

Операторы ввода-вывода

❖ Ввод

Для ввода арифметических значений используются функции преобразования

StrToInt (<строка, содержащая значение целого типа>)

StrToFloat (< строка, содержащая значение вещественного типа >)

Операторы ввода-вывода

❖ Программа, вычисляющая сумму двух вещественных чисел

```
var x,y,z:real;
```

```
begin
```

```
  x:=StrToFloat(InputBox(' Окно ввода', 'Введите  
  значение x',''));
```

```
  y:=StrToFloat(InputBox(' Окно ввода ',  
  ' Введите значение y',''));
```

```
  z:=x+y;
```

```
  showmessage('А вот и сумма: ' + FloatToStr(z));
```

```
end; 2. Операторы ввода-вывода  
(арифметические значения)
```

Функции преобразования типов

❖ *Арифметические значения в строковые*

IntToStr (<выражение целого типа>)

FloatToStr(<выражение вещественного типа>)

```
var a: integer; b: real; s: string;
```

```
begin
```

```
.....
```

```
s := IntToStr(a); ... ; s := FloatToStr(b);
```

```
.....
```

```
end;
```

Любое арифметическое выражение можно преобразовать к ***строковому!***

Функции преобразования типов

❖ Строковые значения в арифметические

StrToInt (<выражение строкового типа>)

StrToFloat(<выражение строкового типа>)

```
var a: integer; b: real; s: string;
```

```
begin
```

```
.....
```

```
a:=StrToInt(s);... ;b:=StrToFloat(s);
```

```
.....
```

```
end;
```

Если строка не содержит арифметическое значение, при таких преобразованиях происходит **ошибка!**

❖ **Оператор *try***

try

<операторы, которые выполняются, если не происходит ошибки>

except

<операторы, которые выполняются, если произошла ошибка>

end;

Оператор *try*

- ❖ **Программа, вычисляющая сумму двух вещественных чисел с обработкой ошибок ввода**

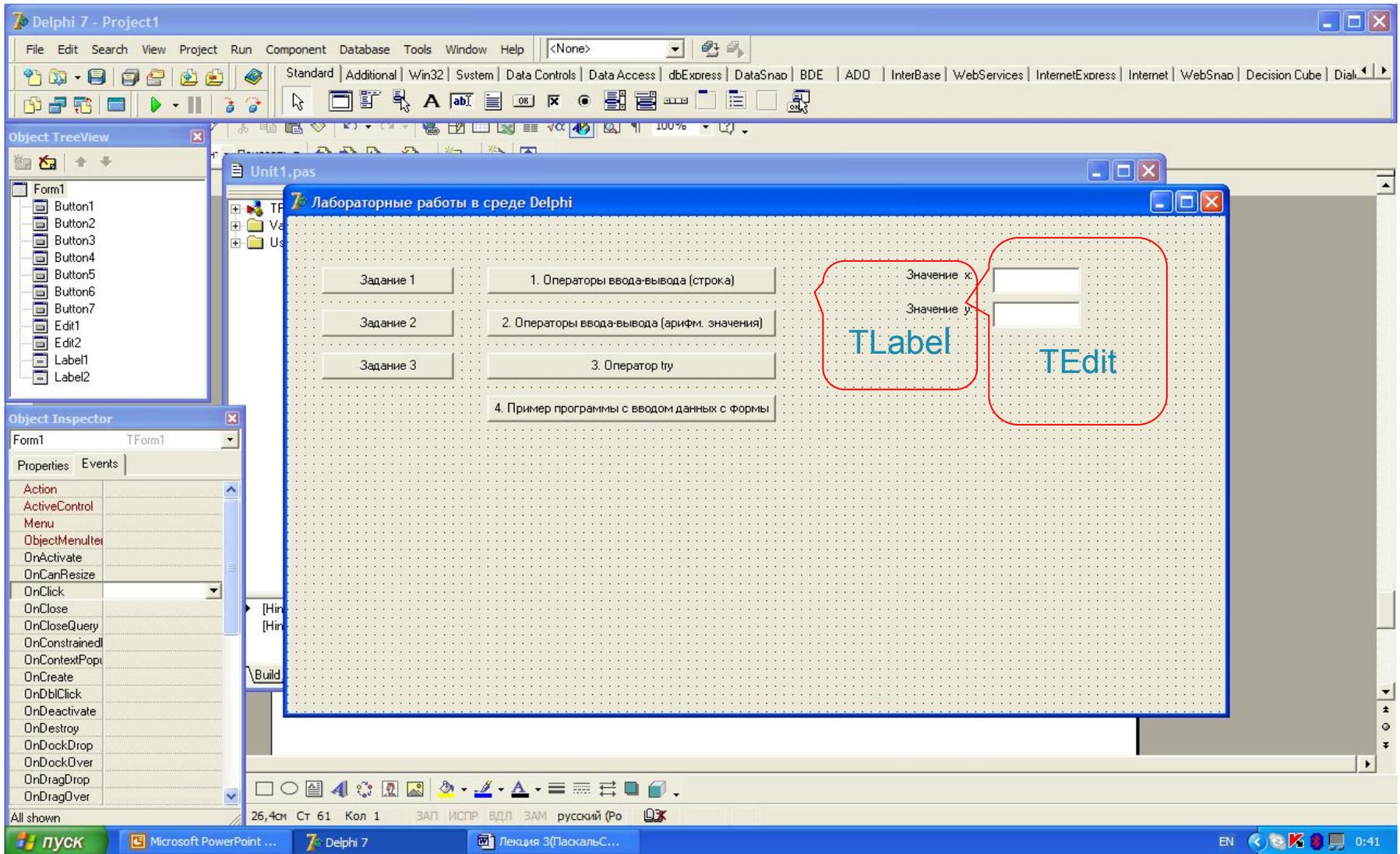
```
var x,y,z:real;  
begin  
  try  
    x:=StrToFloat(TextBox(' Окно ввода', 'Введите значение  
    x',''));  
    y:=StrToFloat(TextBox(' Окно ввода ', 'Введите значение y',''));  
    z:=x+y;  
    showMessage('А вот и сумма: ' + FloatToStr(z));  
  except  
    showMessage('Вводите числа правильно!')  
  end  
end; 3. Оператор try
```

Пример программы с вводом информации с формы

❖ Ввод данных с формы

- 1.** Поставим на форму два компонента класса **TEdit**, которые служат для ввода данных. По умолчанию они будут иметь имена Edit1 и Edit2 (свойство **Name** компонента TEdit). По эти именам будем к ним обращаться.
- 2.** Удалим из свойства **Text** компонентов значения, чтобы при запуске программы строки ввода были пустые.
- 3.** Поставим на форму два компонента класса **TLabel**, которые служат для вывода строковых значений. Свойству **Caption** компонентов дадим значения 'Значение x:' и 'Значение y:' соответственно.

TEdit



Ввод информации с формы

```
var x,y,z: real;  
begin  
  try  
    x:=StrToFloat(Edit1.Text);  
    y:=StrToFloat(Edit2.Text);  
    z:=x+y;  
    showmessage('А вот и сумма: '+FloatToStr(z));  
  except  
    showmessage('Вводите числа правильно!')  
  end  
end; 4. Ввод информации с формы
```

Пример отчета в документе MS Word по выполнению задания

Задание №.

- ❖ Текст задания**
- ❖ Алгоритм решения**
- ❖ Текст процедуры**
- ❖ Значения входных и выходных данных для 3-4 расчетов**

Примеры на оператор присваивания

Задание 1.1. Оператор присваивания

Текст задания. Рассчитать значение функции $f=(x*x+y*y)^{1/2}$ при заданных значениях входных данных x и y .

Текст процедуры.

```
var x,y,z: real;  
begin  
  try  
    x:=StrToFloat(Edit1.Text);  
    y:=StrToFloat(Edit2.Text);  
    z:=sqrt(x*x+y*y);  
    showmessage('Значение функции: '+FloatToStr(z));  
  except  
    showmessage('Вводите числа правильно!')  
  end  
end;
```

Значения входных и выходных данных для 3-4 расчетов

Входные данные: $x=3$ $y=4$ Выходные данные: $z=5$

Примеры на оператор присваивания

Задание 1.2. Оператор присваивания

1. Текст задания. Даны длины двух сторон треугольника и величина угла, находящегося между этими сторонами. Найти периметр этого треугольника.

2. Решение.

Пусть a и b длины сторон треугольника. По теореме косинусов длина третьей стороны определяется по формуле:

$$c^2 = a^2 + b^2 - 2 * a * b * \cos \alpha$$

Следовательно, периметр треугольника p будет определяться по формуле:

$$p = (a^2 + b^2 - 2 * a * b * \cos \alpha)^{1/2} + a + b$$

Примеры на оператор присваивания

Задание 1.2. Оператор присваивания

Текст процедуры.

```
var a,b,alfa,p:real;  
begin  
  try  
    a:=StrToFloat(Edit3.Text);  
    b:=StrToFloat(Edit4.Text);  
    alfa:=StrToFloat(Edit5.Text);  
    p:=sqrt(a*a+b*b-2*a*b*cos(alfa))+a+b;  
    showmessage('Периметр треугольника: '+FloatToStr(p));  
  except  
    showmessage('Вводите числа правильно!')  
  end  
end;
```

Значения: прямоугольный треугольник с катетами $a=1$, $b=1$, следовательно угол $\text{alfa}=\pi/2 \approx 1,5707$. Результат: 3,414 (1+1+1,414).

Примеры на оператор присваивания

Задание 1.3. Оператор присваивания

1. Текст задания. Дано трехзначное число, вычислить сумму его цифр.

2. Решение.

Пусть a – число, которое вводится, $a_1a_2a_3$ представление числа в виде последовательности цифр.

Например, если $a=354$, то $a_1=3$, $a_2=5$, $a_3=4$.

Для нахождения младшей цифры надо найти остаток от деления числа на 10: $a_3=a \bmod 10$ ($a_3=4$).

Далее: $a=a \operatorname{div} 10$ ($a=35$), $a_2=a \bmod 10$ ($a_2=5$);
 $a=a \operatorname{div} 10$ ($a=3$), $a_1=a$ ($a_1=3$).

Примеры на оператор присваивания

Задание 1.3. Оператор присваивания

Текст процедуры.

```
var a1,a2,a3: integer;  
begin  
  try  
    a:=StrToInt(Edit3.Text); a3:=a mod 10;  
    a:=a div 10; a2:=a mod 10;  
    a:=a div 10; a1:=a;  
    showmessage('Сумма цифр числа '+IntToStr(a)+' равна  
      '+IntToStr(a1+a2+a3))  
  except  
    showmessage('Вводите числа правильно!')  
  end  
end;
```

Значения входных и выходных данных:

Входные данные: a=354 Выходные данные: 12