

# ЛАБОРАТОРНАЯ РАБОТА №2. ПРОГРАММИРОВАНИЕ АЛГОРИТМОВ ЛИНЕЙНОЙ СТРУКТУРЫ

## 1. ЦЕЛЬ РАБОТЫ

Целью лабораторной работы является приобретение умений и навыков практического освоения современных технологий визуального объектно-ориентированного программирования в свободно распространяемой среде разработки приложений Lazarus на примере реализации способов ввода и вывода данных при создании простых приложений линейной структуры с помощью языка программирования FREE Pascal.

## 2. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

### 2.1. ОСНОВНЫЕ ТИПЫ ДАННЫХ В ЯЗЫКЕ FREE PASCAL

Основные простые типы данных представлены в таблице

Тип	Диапазон значений
<b>Целые типы</b>	
ShortInt	-128 .. 127
SmallInt	-32 768 .. 32 767
Integer	-2 147 483 648 .. 2 147 483 647
LongInt	-2 147 483 648 .. 2 147 483 647
Byte	0 .. 255
Word	0 .. 65 535
LongWord	0 .. 4 294 967 295
<b>Вещественные типы</b>	
Real	$2.9e^{-39} .. 1.7e^{38}$
Single	$1.5e^{-45} .. 3.4e^{38}$
Double	$5.0e^{-324} .. 1.7e^{308}$
Extended	$3.4e^{-4932} .. 1.1e^{4932}$
Comp	$-2^{63} .. 2^{63}$
<b>Логический тип</b>	
Boolean	True, False
<b>Символьный тип</b>	
Char	Полный набор ASCII-символов
<b>Строковый тип</b>	
String	Строки, состоящие из алфавитно-цифровых символов

Описываются переменные с помощью ключевого слова `Var`, например:

```
Var
  a,b: real;
  i,j,k: integer;
  letter: char;
  Flag: Boolean;
```

Несмотря на достаточно мощную структуру встроенных типов данных, в среде Lazarus предусмотрен механизм создания новых типов. Для этого используют служебное слово `Type`:

```
Type <новый тип данных> = <определение типа>;
```

Формат определения пользовательского типа	Описание
<b>Перечислимый тип данных</b>	
(Знач_1, Знач_2, ..., Знач_N)	Задается перечислением значений, которые он может принимать. Значения являются константами
<b>Интервальный (Тип-диапазон)</b>	
Константа_1..Константа_2;	Задаёт границы диапазона возможных значений внутри базового типа. Константы должны принадлежать одному и тому же типу.

Когда новый тип данных создан, объявляются соответствующие ему переменные:

```
Var <список переменных>: <новый тип данных>;
```

Например,

```
Type
  Days = 1..31;
Var
  RabDay, BolnDay: Days;
```

## 2.2. ОПЕРАТОР ПРИСВАИВАНИЯ

Оператор присваивания позволяет заменить текущее значение переменной, стоящей слева, новым значением, задаваемым выражением, стоящим справа. Формат оператора присваивания:

<Имя Переменной> := <Выражение>;

Выражение состоит из операндов и операций, которые выполняются над операндами.

Операндами арифметических выражений могут быть числа (константы) и переменные, функции и арифметические выражения, заключенные в круглые скобки. Операции обозначают действия, выполняемые над операндами.

В *арифметических* выражениях возможны следующие *операции*:

Операция	Действие	Тип операндов	Тип результата
+	сложение	целый/вещественный	целый/вещественный
-	вычитание	целый/вещественный	целый/вещественный
*	умножение	целый/вещественный	целый/вещественный
/	деление	целый/вещественный	целый/вещественный
div	целочисленное деление	целый	целый
mod	остаток от деления	целый	целый

*Стандартные функции* над арифметическими операндами приведены в приложении 5. Все математические функции содержатся в библиотечных модулях System и Math. Библиотечный модуль System подключается автоматически, а модуль Math необходимо подключить вручную в разделе Uses (рис. 1).

```

unit Unit1;

{$mode objfpc}{$H+}

interface

uses
  Classes, SysUtils, FileUtil, LResources, Forms,
  Controls, Graphics, Dialogs, StdCtrls, Grids, Math;

type
  { TForm1 }

```

Рис. 1. Подключение библиотечного модуля Math

Операция возведение в степень во Free Pascal может быть выполнена по-разному. Если значение степени – целое положительное число, то можно воспользоваться формулой:

$$x^n = e^{n \cdot \ln(x)},$$

которая на языке программирования имеет вид:

$$\text{exp}(n * \ln(x)).$$

Во Free Pascal имеются специальные функции возведения в степень, которые находятся в библиотечном модуле Math:

Обращение	Тип параметров	Тип результата	Описание
IntPower (y, x)	y – вещественный; x – целый	вещественный	Возведение y в целочисленную степень x
Power (y, x)	оба параметр – вещественные	вещественный	Возведение y в вещественную степень x

Результатом выполнения *логических* выражений является значение True или False. Для написания простых логических выражений используются знаки отношений (=, <, >, <=, >=, <>).

Сложные логические выражения строятся из простых выражений и логических операций, примененных к ним. Основные логические операции, используемые для написания логических выражений: not (отрицание), and (логическое произведение), or (логическое сложение), xor (исключающее или):

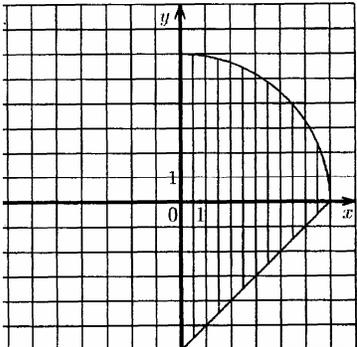
A	B	Not A	A and B	A or B	A xor B
False	False	True	False	False	False
False	True	True	False	True	True
True	False	False	False	True	True
True	True	False	True	True	False

В сложных выражениях порядок, в котором выполняются операции, соответствует приоритету операций. В языке Free Pascal приняты следующие приоритеты:

1.	not
2.	*, /, div, mod, and
3.	+, -, or, xor
4.	=, <>, >, <, >=, <=

Использование скобок в выражениях позволяет менять порядок вычислений. Примеры операторов присваивания:

Оператор присваивания	Результат
<code>i:=0;</code>	Переменной <i>i</i> присваивается значение ноль
<code>a:=b+c;</code>	Переменной <i>a</i> присваивается число, равное сумме значений переменных <i>b</i> и <i>c</i>
<code>j:=j+1;</code>	Значение переменной <i>j</i> увеличивается на единицу
<code>Label1.Caption:= 'Пример';</code>	Задается заголовок надписи (свойству <code>Caption</code> компоненты <code>Label1</code> присваивается значение <code>'Пример'</code> )
<code>Edit1.Text:=''</code>	Очищается текстовое поле (свойству <code>Text</code> компоненты <code>Edit1</code> присваивается пустая строка)
<code>Edit2.Text:= FloatToStr(a);</code>	Задается значение текстового поля (свойству <code>Text</code> компоненты <code>Edit2</code> присваивается значение переменной <i>a</i> )
<code>b:=(x&gt;=0) and (x&lt;=10);</code>	Значение логической переменной <i>b</i> будет <code>True</code> , если <i>x</i> принадлежит отрезку <code>[0; 10]</code> , и <code>False</code> если <i>x</i> не принадлежит этому отрезку

<pre> B := (x &gt;= 0) and   ((x * x + y * y) &lt;= 36) and   (y &gt;= x + 6); </pre>	<p>Значение логической переменной будет True, если точка с координатами (x, y) принадлежит заданной области, и False, если не принадлежит.</p> 
---	---

Оператор присваивания считается верным, если тип выражения соответствует, или может быть приведен к типу той переменной, которой присваивается это выражение. Например, переменной типа Real можно присвоить значение выражения типа Real или Integer. Переменной типа Integer можно присвоить значение выражения только типа Integer.

Для приведения типа переменной к соответствующему типу в Lazarus имеются функции преобразования типов. В таблице 1 приведены некоторые функции преобразования типов.

Таблица 1

Обращение	Тип параметров	Тип результата	Описание
Round (x)	вещественный	целый	округление числа (round (3.5) = 4)
Trunc (x)	вещественный	целый	отсекание дробной части числа (trunc (3.5) = 3)
Ceil (x) (в библиотечном модуле Math)	вещественный	целый	ближайшее большее целое (ceil (3.2) = 4)
Frac (x)	вещественный или целый	тип параметра	дробная часть числа (frac (3.2) = 0.2)
Int (x)	вещественный или целый	тип параметра	целая часть числа (frac (3.2) = 3)

Для ввода и вывода числовых значений (целого или вещественного типа), логических и др. в Lazarus часто используются компоненты класса TEdit, TLabel, TMemo и т.п. Обычно свойство подобных компонентов, которое используется при вводе/выводе, имеет тип String. Поэтому в случае, если типы вводимых или выводимых переменных имеют тип отличный от типа String, необходимо использовать функции преобразования строковых типов в другие типы данных. Такие функции приведены в таблице 2.

Таблица 2

Обращение	Тип параметра	Тип результата	Описание
<b><i>Преобразование строк в другие типы</i></b>			
StrToInt (s)	строковый	целый	Преобразование символов строки s в целое число
StrToFloat (s)	строковый	вещественный	Преобразование символов строки s в вещественное число
<b><i>Обратное преобразование</i></b>			
FloatToStr (x)	вещественный	строковый	Преобразование вещественного числа x в строку
IntToStr (x)	целый	строковый	Преобразование целого числа x в строку
BoolToStr (T, True)	T – логический	строковый	Преобразование логического типа в строку 'True' или 'False'.

Во время перевода исходной программы в исполняемую компилятор проверяет соответствие типов выражений и переменных. Если тип выражения не соответствует типу переменной, то компилятор выдает сообщение об ошибке Incompatible types (Несовместимые типы).

### 3. ПРАКТИЧЕСКАЯ ЧАСТЬ

#### 3.1. СОЗДАНИЕ, КОМПИЛЯЦИЯ И ОТЛАДКА ПРОСТОГО ПРИЛОЖЕНИЯ ЛИНЕЙНОЙ СТРУКТУРЫ

**Задание 1.** Создать приложение, которое обеспечивает ввод двух целых чисел, вычисляет их сумму и выводит значение результата.

Следуйте рекомендациям, описанным ниже.

1. Запустите ИСР Lazarus с помощью команды Главного меню Windows Пуск → Программы → Lazarus → Lazarus.
2. Создайте новый проект при помощи команды Проект → Создать проект (рис. 2), если при запуске Lazarus открылся *не новый* проект.

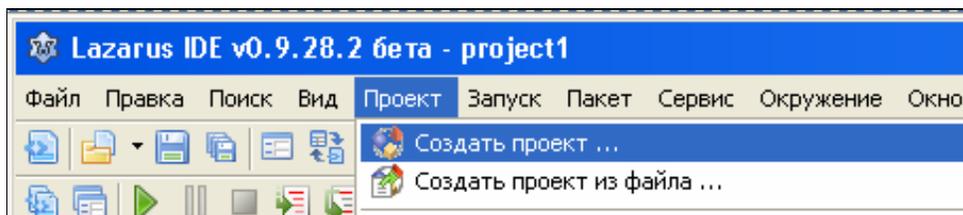


Рис. 2. Меню Проект → Создать проект

**Замечание.** Чтобы при запуске Lazarus всегда открывался новый проект нужно выбрать пункт меню Окружение → Параметры... В открывшемся окне убрать метку рядом с командой Открывать последний проект при запуске. В дальнейшем при загрузке Lazarus всегда будет создаваться *новый* проект.

3. Сохраните новый проект командой меню Проект → Сохранить проект как.... В появившемся окне Сохранить Unit1 создайте новую папку для файлов создаваемого проекта с названием «Сумма целых чисел» на Рабочем столе с помощью кнопки . Откройте созданную папку и нажмите кнопку Сохранить. После сохранения файла модуля Unit1.pas откроется окно Сохранить проект. Задайте имя файла проекта «Summa» и нажмите кнопку Сохранить.

4. Измените свойство Caption (это заголовок формы) Form1 в окне Инспектора объектов на новое значение «Сумма двух целых чисел».
5. Разместите на форме компоненты Edit1, Edit2, Edit3, Label1, Button1 группы Standard (рис. 3).
6. Выделяя каждый объект, измените некоторые их свойства с помощью Инспектора объектов:

Объект	Свойство	Значение
Form1	Caption	Сумма двух целых чисел
	Position	poScreenCenter
Button1	Caption	=
	Font	Arial 20
Label1	Caption	+
	Font	Arial 14
Edit1, Edit2, Edit3	Text	Очистите эти поля
	Font	Arial 14

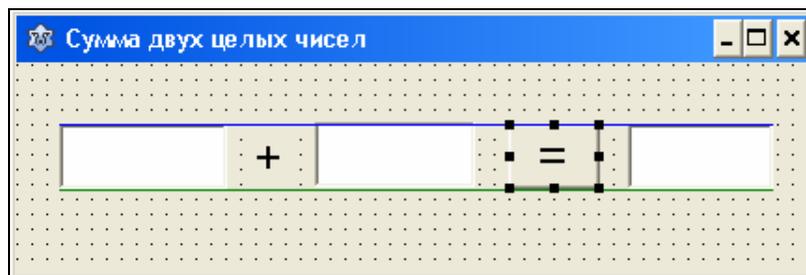


Рис. 3. Форма Form1

7. Добавьте на форму еще три объекта Label, расположите их над объектами Edit1 – Edit3 и задайте их свойствам Caption значения «Слагаемое 1», «Слагаемое 2», и «Сумма».
8. Активизируйте окно Редактора кода, нажав F12. Просмотрите сгенерированный Lazarus модуль описания формы и размещенных на ней компонентов.
9. Сохраните изменения, внесенные в проект, командой меню Проект → Сохранить проект или щелкнув на значок дискеты в главном меню.
10. Создайте код обработчика событий (событийную процедуру): при нажатии на кнопку «=» два введенных числа должны скла-

дываться. Для этого дважды щелкните на этом объекте левой кнопкой мыши. В результате откроется окно Редактора исходного кода, в котором вы увидите пустую процедуру `procedure TForm1.Button1Click(Sender:TObject.` При этом курсор будет установлен между словами `begin` и `end`. В этом месте необходимо разместить необходимые операторы.

Окончательно процедура обработки события щелчка на кнопке `Button1` должна выглядеть следующим образом:

```
procedure TForm1.Button1Click(Sender: TObject);  
var  
    a,b,c: integer;      {слагаемые и сумма целые числа}  
begin  
a:=StrToInt(Edit1.Text); {преобразование текстовой строки в целое число}  
b:=StrToInt(Edit2.Text);  
c:=a+b;  
Edit3.Text:=IntToStr(c); {преобразование целого числа в текстовую строку}  
end;
```

11. Сохраните изменения в проекте.
12. Запустите приложение на выполнение.
13. После проверки работы приложения, закройте его.
14. Запустите приложение из Windows, используя исполняемый файл (приложение).

**Задание 2.** В созданный проект добавьте кнопку на закрытие приложения. В редакторе кода используйте оператор `Close` (рис.4).

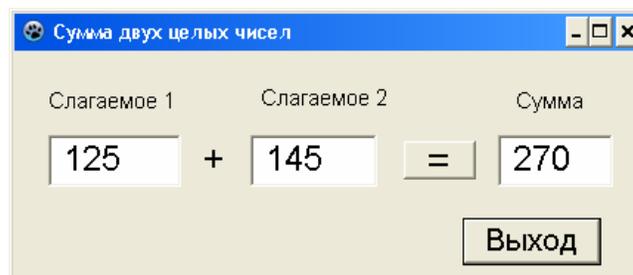


Рис. 4. Окончательная форма задания 1

**Задание 3.** Создать приложение, которое обеспечивает ввод двух вещественных чисел, вычисляет их сумму и выводит значение результата.

Следуйте рекомендациям, описанным ниже.

1. Задание 3 отличается от задания 2 только типом переменных (`integer` и `real`). Поэтому просто воспользуемся ранее созданным проектом. Для этого создайте на рабочем столе новую папку «Сумма вещественных чисел». Скопируйте в нее всё содержимое папки «Сумма целых чисел».
2. Откройте из папки «Сумма вещественных чисел» проект. Измените тип переменных на `real`. При выводе результата в текстовое окно используйте функцию преобразования *вещественного* числа в текстовую строку `FloatToStr`.
3. Сохраните изменения, внесенные в проект, командой меню Проект → Сохранить проект или щелкнув на значок дискеты в главном меню.
4. Запустите приложение на выполнение (рис. 5):

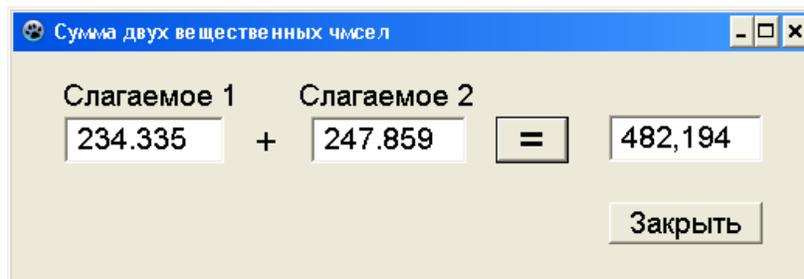


Рис. 5. Пример формы «Сумма вещественных чисел»

5. После проверки работы приложения, закройте его.
6. Запустите приложение из Windows, используя исполняемый файл (приложение).

**Задание 4.** Выполните индивидуальные задания. При выполнении заданий используйте стандартные процедуры и функции, описанные в приложении 1. При выполнении задания 2 третьего уровня сложности используйте функцию `BoolToStr(T,True)` для вывода результата.

**Замечание.** Чтобы избавиться от надоедливого сообщения «Выполнение остановлено» при каждом завершении наших программ снимите галочку с опции Показывать сообщение при остановке в окне Окружение → Параметры → Отладчик.

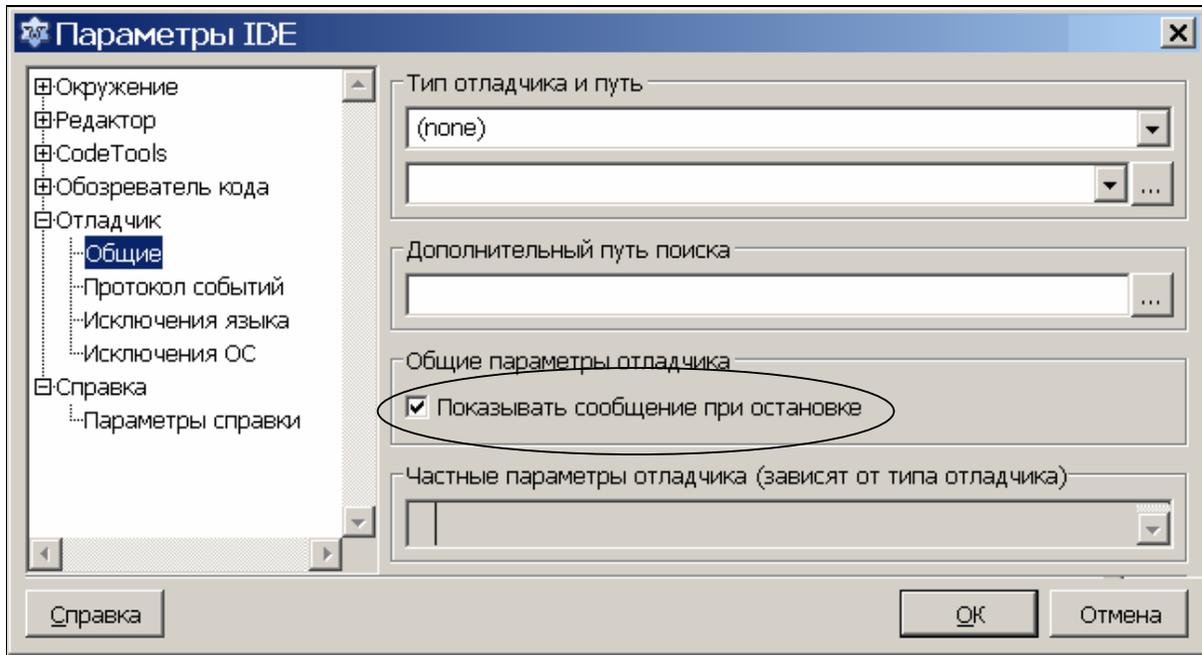


Рис. 6. Опция отладчика Показывать сообщение при остановке

### 3.2. КОМПОНЕНТЫ LAZARUS, ИСПОЛЬЗУЕМЫЕ ДЛЯ НАГЛЯДНОСТИ И УКРАШЕНИЯ ОКОН

Компонент **Image** (графический образ) . Позволяет отображать рисунок, загруженный из графического файла. Относится к группе Additional.

Основные свойства компонента Image :

Свойство	Описание
Align	Задаёт режим выравнивания расположения объекта Image внутри формы.
Picture	Задаёт имя файла графического файла с рисунком с помощью диалогового окна Открыть файл изображения (рис. 7).
Stretch	Задаёт разрешение на автоматическое масштабирование рисунка относительно Image.

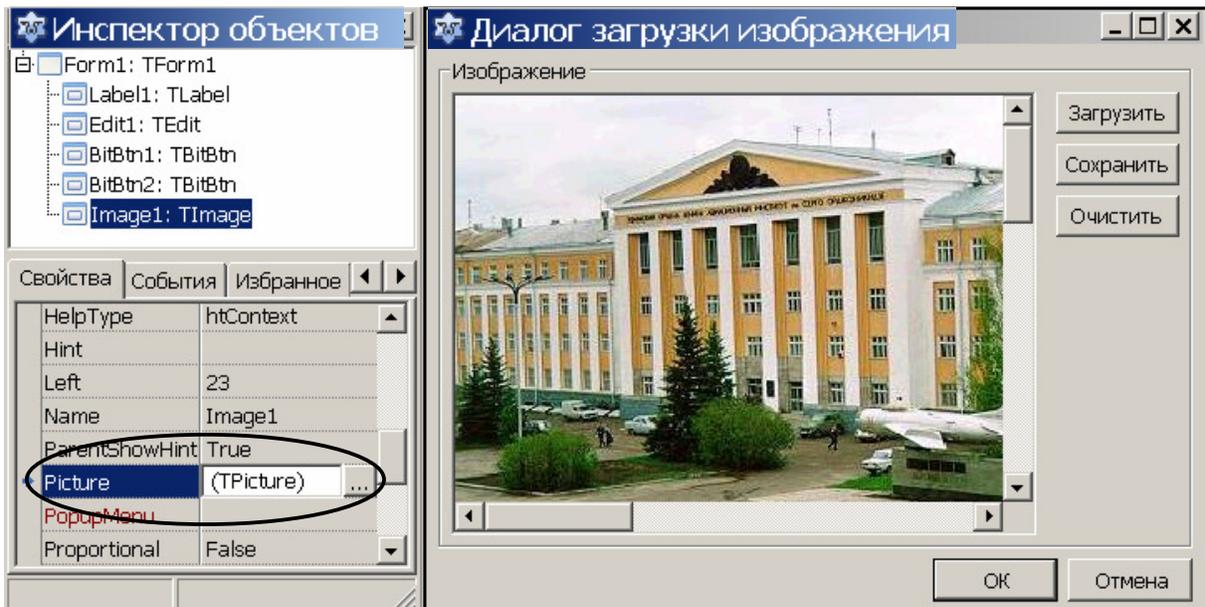


Рис. 7. Открытие диалогового окна Открыть файл изображения

Компонент **BitBtn** (кнопка с изображением) . Используется как обычная кнопка для инициирования некоторого события, но может содержать графическое изображение. Относится к группе `Additional`.

Основные свойства компонента `BitBtn`:

Свойство	Описание
<code>Caption</code>	Задаёт текст надписи на кнопке
<code>Kind</code>	Предлагает на выбор десять predefined типов кнопок
<code>Glith</code>	Создание собственного типа кнопки с помощью окна Диалог загрузки изображения (рис. 8).

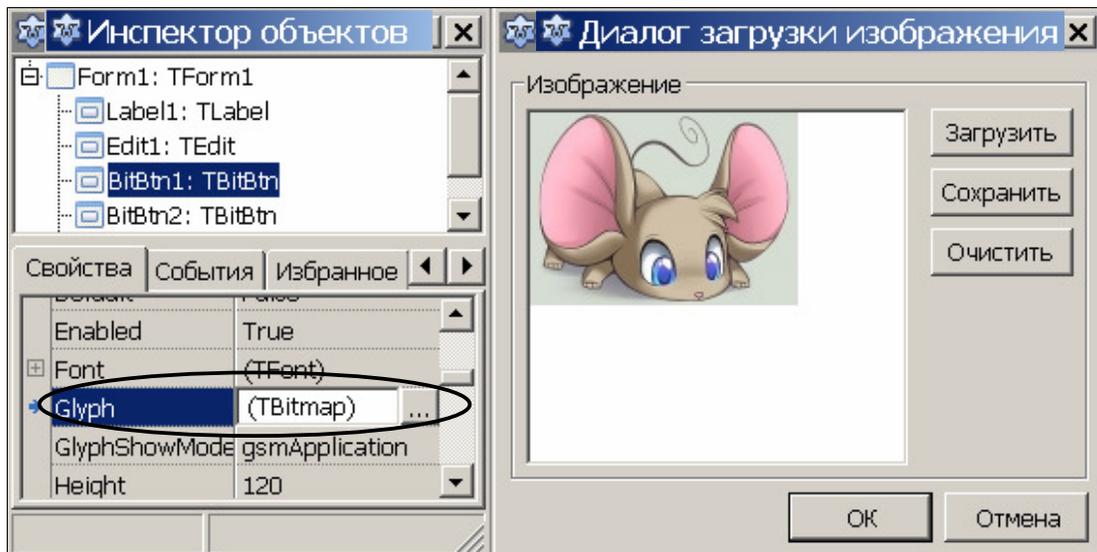


Рис. 8. Открытие окна Диалог загрузки изображения

Пример формы с компонентами BitBtn и Image – на рис. 9.



Рис. 9. Пример формы с компонентами BitBtn и Image

## 4. ЗАДАНИЯ К ЛАБОРАТОРНОЙ РАБОТЕ №2

### 4.1. ПЕРВЫЙ УРОВЕНЬ СЛОЖНОСТИ

Создать приложение, вычисляющее значения переменных по заданным расчетным формулам и наборам исходных данных. Исходные данные вводите с помощью компонент класса TEdit. Ввод исходных данных и вывод результатов вычислений сопровождайте поясняющими комментариями.

№ вар.	Расчетные формулы	Значения исходных данных
1.	$a = \frac{2 \cos(x - \pi/6)}{1/2 + \sin^2 y}$ $b = 1 + \frac{z^2}{3 + z^2/5}$	$x = 1.426$ $y = -1.22$ $z = 3.5$
2.	$\gamma = \left  x^{y/x} - \sqrt[3]{y/x} \right $ $\psi = (y - x) \frac{y - z/(y - x)}{1 + (y - x)^2}$	$x = 1.825$ $y = 18.225$ $z = -3.298$
3.	$s = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!}$ $\psi = x(\sin x^3 + \cos^2 y)$	$x = 0.335$ $y = 0.025$
4.	$y = e^{-bt} \sin(at + b) - \sqrt{ bt + a }$ $s = b \sin(at^2 \cos 2t) - 1$	$a = -0.5$ $b = 1.7$ $t = 0.44$
5.	$\omega = \sqrt{x^2 + b} - b^2 \sin^3(x + a)/x$ $y = \cos^2 x^3 - x/\sqrt{a^2 + b^2}$	$a = 1.5$ $b = 15.5$ $x = -2.9$
6.	$s = x^3 \operatorname{tg}^2(x + b)^2 + a/\sqrt{x + b}$ $Q = \frac{bx^2 - a}{e^{ax} - 1}$	$a = 16.5$ $b = 3.4$ $x = 0.61$

7.	$R = x^2(x+1)/b - \sin^2(x+a)$ $s = \sqrt{xb/a} + \cos^2(x+b)^2$	$a = 0.7$ $b = 0.05$ $x = 0.5$
8.	$y = \sin^3(x^2+a)^2 - \sqrt{x/b}$ $z = \frac{x^2}{a} + \cos(x+b)^3$	$a = 1.1$ $b = 0.004$ $x = 0.2$
9.	$f = \sqrt[3]{mctb +  c \sin t }$ $z = m \cos(bt \sin t) + c$	$m = 2$ $c = -1$ $t = 1.2$ $b = 0.7$
10.	$y = abx^2 - \frac{a}{\sin^2(x/a)}$ $d = ae^{-\sqrt{a}} \cos(bx/a)$	$a = 3.2$ $b = 17.5$ $x = -4.8$
11.	$f = \ln(a+x^2) + \sin^2(x/b)$ $z = e^{-cx} \frac{x + \sqrt{x+a}}{x - \sqrt{ x-b }}$	$a = 10.2$ $b = 9.2$ $c = 0.5$ $x = 2.2$
12.	$y = \frac{a^{2x} + b^{-x} \cos(a+b)x}{x+1}$ $R = \sqrt{x^2 + b} - b^2 \sin^3(x+a)/x$	$a = 0.3$ $b = 0.9$ $x = 0.61$
13.	$z = \sqrt{ax \sin 2x + e^{-2x}(x+b)}$ $\omega = \cos^2 x^3 - x/\sqrt{a^2 + b^2}$	$a = 0.5$ $b = 3.1$ $x = 1.4$
14.	$U = \frac{a^2x + e^{-x} \cos bx}{bx - e^{-x} \sin bx + 1}$ $f = e^{2x} \ln(a+x) - b^{3x} \ln(b-x)$	$a = 0.5$ $b = 2.9$ $x = 0.3$

15.	$z = \frac{\sin x}{\sqrt{m^2 + \sin^2 x}} - cm \ln mx$ $s = e^{-ax} \sqrt{x+1} + e^{-bx} \sqrt{x+1.5}$	$m = 0.7$ $c = 2.1$ $x = 1.7$ $a = 0.5$ $b = 1.08$
-----	--	--

#### 4.2. ВТОРОЙ УРОВЕНЬ СЛОЖНОСТИ

Создать приложение, реализующее решение задач. Ввод и вывод пояснить комментариями.

№ вар.	Формулировка задачи
1.	Вычислить площадь и периметр прямоугольника, если задана длина одной стороны ( $a$ ) и коэффициент $n$ (%), позволяющий вычислить длину второй стороны ( $b=n*a$ ).
2.	Дана длина ребра куба. Найти площадь грани, площадь полной поверхности и объем этого куба.
3.	Вычислить периметр и площадь прямоугольного треугольника по заданным длинам двух катетов $a$ и $b$ .
4.	Вычислить площади геометрических фигур: прямоугольника и треугольника по заданным сторонам.
5.	По известному радиусу вычислить объем и площадь поверхности шара.
6.	Даны два числа. Найти среднее арифметическое кубов этих чисел и среднее геометрическое модулей этих чисел.
7.	Даны два числа. Вычислить их сумму, разность, произведение и частное.
8.	Известен объем информации в байтах. Выразить его в мегабайтах и гигабайтах.
9.	Длина выражена в сантиметрах. Выразить ее в дюймах. (1 дюйм=2.5 см)
10.	Перевести значение веса, выраженное в граммах, в унции (1 унция =28.3 г)
11.	Три сопротивления соединены последовательно. Найти сопротивление соединения.
12.	Вычислить путь, пройденный лодкой по течению, если известна ее скорость в стоячей воде, скорость течения реки и время движения.
13.	Вычислить расстояние между двумя точками с заданными координатами.

№ вар.	Формулировка задачи
14.	Известна длина окружности. Найти площадь круга, ограниченного этой окружностью.
15.	Известны внутренний и внешний диаметры кольца. Найти его площадь.

### 4.3. ТРЕТИЙ УРОВЕНЬ СЛОЖНОСТИ

Создать приложение, реализующее решение задач. Ввод и вывод пояснить комментариями.

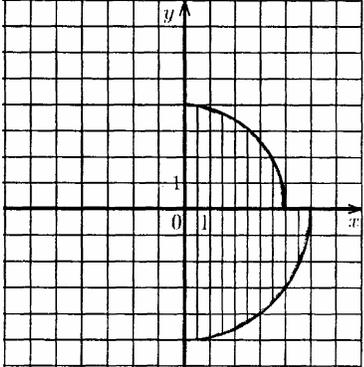
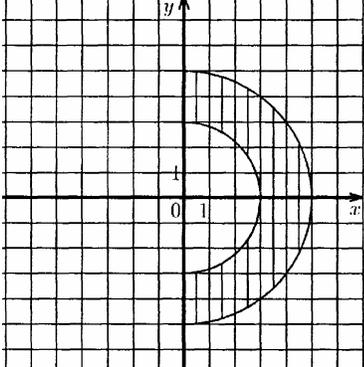
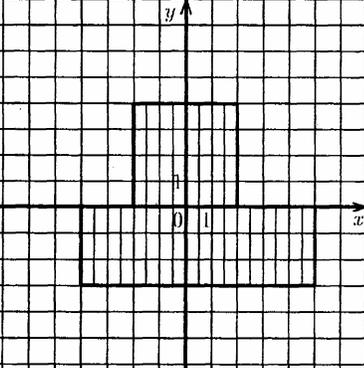
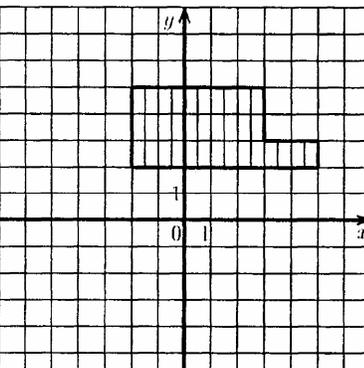
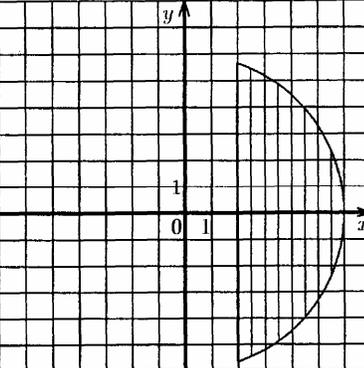
#### Задание 1.

№ вар.	Формулировка задачи
1.	Вычислить объем призмы, боковые грани которой квадраты, а основанием служит равносторонний треугольник, вписанный в круг радиуса $R$ .
2.	Треугольник задан тремя сторонами. Вычислить его медианы.
3.	В шар радиуса $R$ вписан конус с углом $\alpha$ при вершине в осевом сечении конуса. Определить объем и полную поверхность конуса.
4.	Вычислить диагональ и площадь прямоугольника, вписанного в окружность радиуса $R$ , если отношение его сторон равно $n$ .
5.	Даны две стороны треугольника и угол между ними. Определить третью сторону, площадь треугольника и радиус описанной окружности.
6.	Определить плату за квартиру, если известны: площадь квартиры, количество проживающих и плата за содержание жилья, отопление, водопровод, канализацию, подогрев воды, коллективную антенну, лифт, вывоз мусора.
7.	Определить плату за электроэнергию, если известны: старое и новое показания счетчика, стоимость одного квт/часа электроэнергии, количество просроченных дней и размер пени за один день просрочки.

№ вар.	Формулировка задачи
8.	$V_1$ литров воды нагревается на электроплите от температуры $t_0$ до температуры $t_1$ , а $V_2$ литров – до температуры $t_2$ . Сколько будет стоить затраченная на это электроэнергия? КПД электроплитки и стоимость 1 квт час электроэнергии известны (удельная теплоемкость воды $C=4190$ Дж/кгК).
9.	Пусть смешано $V_1$ литров воды температуры $t_1$ с $V_2$ литрами воды температуры $t_2$ и $V_3$ литрами воды температуры $t_3$ . Вычислить объем и температуру образовавшейся смеси.
10.	Торговая фирма закупила $n$ количество меховых изделий по цене $C_1$ для продажи. Фирма облагается налогами: 20% налог от прибыли в местный бюджет, 28% налог в пенсионный фонд, 5% от объема продажи идет на формирование зарплаты. Определить розничную цену товара, если планируемая прибыль предприятия 15%.
11.	Длина отрезка задана в дюймах (1 дюйм = 2.54 см). Перевести значение длины в метрическую систему, то есть выразить ее в метрах, сантиметрах и миллиметрах. Например, 21 дюйм = 0 м 53 см 3,4 мм.
12.	Заданы моменты начала и конца некоторого промежутка времени в часах, минутах и секундах (в пределах одних суток). Найти продолжительность этого промежутка в тех же единицах.
13.	Текущее время (часы, минуты, секунды) задано тремя переменными: $h, m, s$ . Округлить его до целых значений минут и часов. Например, 14 ч 21 мин 45 с преобразуется в 14 ч 22 мин или 14 ч, а 9 ч 59 мин 23 с – соответственно в 9 ч 59 мин или 10 ч.
14.	Угол $\alpha$ задан в радианах. Найти его величину в градусах, минутах и секундах.
15.	Длина некоторого отрезка составляет $p$ метров. Перевести ее в русскую не метрическую систему. 1 верста = 500 сажень; 1 сажень = 3 аршина; 1 аршин = 16 вершков; 1 вершок = 44,45 мм.

Задание 2.

Используя логическое выражение, создайте проект, в котором выводится сообщение 'True', если заданная точка принадлежит заштрихованной области и 'False', если нет. Используйте функцию `BoolToStr(T,True)` для вывода результата.

№ вар	Область	№ вар	Область
1.		2.	
3.		4.	
5.		6.	