

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
БОРИСОГЛЕБСКИЙ ФИЛИАЛ
(БФ ФГБОУ ВО «ВГУ»)

МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ ПО УЧЕБНОЙ ПРАКТИКЕ
Б2.У.1 Учебная компьютерная практика

1. Код и наименование направления подготовки:

44.03.01 Педагогическое образование

2. Профиль подготовки:

Информатика и информационные технологии в образовании

3. Квалификация (степень) выпускника:

Бакалавр

4. Форма обучения:

Заочная

5. Кафедра, отвечающая за реализацию практики:

Кафедра прикладной математики, информатики, физики и методики их преподавания

6. Составители:

В. В. Волков, кандидат физико-математических наук, доцент, М. Н. Хвостов, кандидат физико-математических наук

7. Методические указания для обучающихся, документы, определяющие процедуры оценивания результатов практики

Оценка знаний, умений и навыков, характеризующих этапы формирования компетенций, при прохождении практики проводится в ходе промежуточной аттестации. Промежуточная аттестация проводится в соответствии с Положением о промежуточной аттестации обучающихся по программам высшего образования.

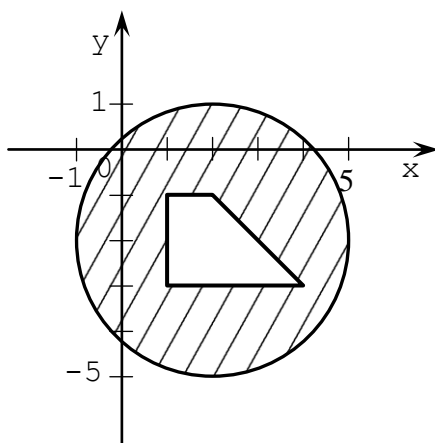
Промежуточная аттестация по практике включает подготовку и защиту отчета по практике. Результаты прохождения практики демонстрируются обучающимися руководителю в виде устного сообщения с предъявлением разработанных программ по мере выполнения. По результатам защиты представленных программ с учетом характеристики руководителя и качества представленных отчетных материалов обучающемуся выставляется соответствующая оценка.

При оценивании используется 4-балльная шкала оценок. Оценивание и учет результатов прохождения практики обучающимися проводится в соответствии Положением о порядке проведения учебной и производственной практик обучающихся в Борисоглебском филиале Воронежского государственного университета по направлениям подготовки 44.03.01 Педагогическое образование (уровень бакалавриата), 44.03.02 Психолого-педагогическое образование (уровень бакалавриата), 44.03.05 Педагогическое образование (с двумя профилями подготовки) (уровень бакалавриата).

8. Методические рекомендации по выполнению задания №1 компьютерной практики

Задание 1 (пример).

Даны два действительных числа X и Y . Составить программу, определяющую, принадлежит ли точка с координатами (X, Y) заштрихованной части плоскости.



Одним из наиболее важных вопросов, возникающих в ходе решения задания №1, является определение геометрического места точек заштрихованной части плоскости. Рассмотрим условия, которым должны соответствовать данные точки. Во-первых, точки заштрихованной области находятся внутри окружности радиусом $R=3$ с центром в точке $O(2, -2)$, т.е. их координаты должны удовлетворять неравенству $(x - x_0)^2 + (y - y_0)^2 \leq R^2$ или $(x - 2)^2 + (y + 2)^2 \leq 9$. Во-вторых, точки заштрихованной области не должны находиться внутри трапеции. Координаты точек, находящихся внутри трапеции, должны удовлетворять следующим требованиям:

- ордината должна быть больше 1,
- абсцисса должна находиться в диапазоне от -1 до -3 ,

- рассматриваемые точки должны располагаться ниже прямой, проходящей через точки $M_1(2,-1)$ и $M_2(4,-3)$.

Тогда, используя уравнение прямой проходящей через две точки $\frac{x-x_1}{x_2-x_1} = \frac{y-y_1}{y_2-y_1}$, получим систему неравенств, задающую часть плоскости, расположенную внутри рассматриваемой трапеции:

$$\begin{cases} x \geq 1, \\ y \leq -1, \\ y \geq -3, \\ y \leq -x+1. \end{cases}$$

Далее в системе программирования Lazarus создадим новый проект. На открывшейся для редактирования форме разместим объекты следующих классов, внося требуемые изменения в свойства данных объектов, используя «Инспекторе объектов»:

TPanel (для группировки управляющих элементов, задающих координату точки)

Name	Panel1
Align	alRight
Caption	<пустая строка>

TImage (для построения декартовой системы координат и заштрихованной области)

Name	Im1
Align	alClient

На панели Panel1 разместим три объекта класса TLabel

(для вывода подписи к полю ввода координаты X проверяемой точки)

Name	L1
Caption	X

(для вывода подписи к полю ввода координаты Y проверяемой точки)

Name	L2
Caption	Y

(для вывода сообщения о принадлежности проверяемой точки заштрихованной области)

Name	L3
Visible	False

На панели Panel1 разместим два объекта класса TEdit

(для ввода координаты X)

Name	EdX
Text	0

(для ввода координаты Y)

Name	EdY
------	-----

Text	0
------	---

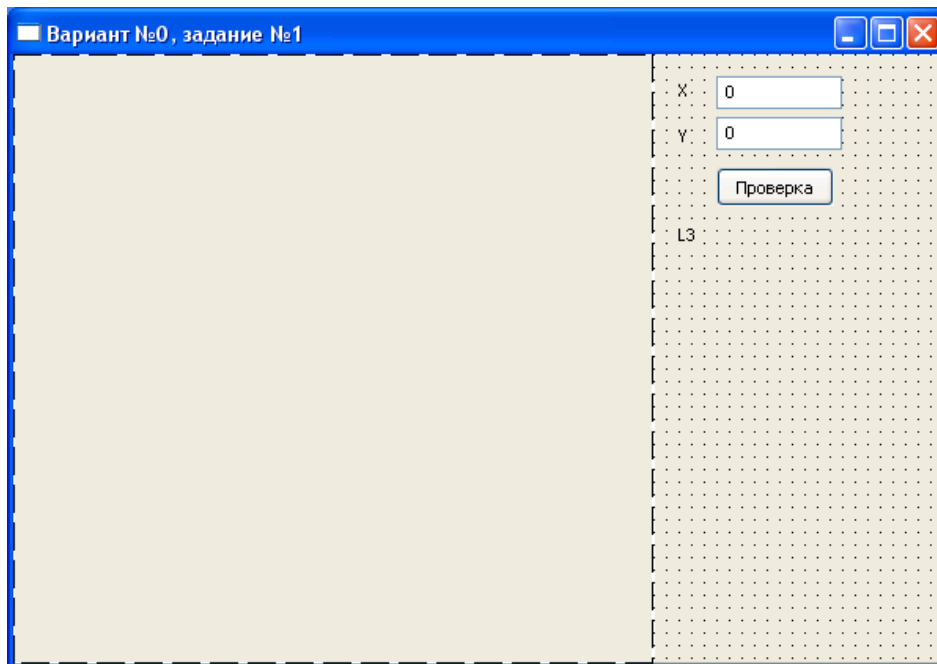
На панели Panel1 разместим объект класса TButton (для запуска процедуры проверки)

Name	Start
Caption	Проверка

У формы Form1 изменим значение свойства

Caption	Вариант №0, задание №1
---------	------------------------

Таким образом, редактируемая форма должна иметь следующий вид:



Далее, предварительно создав для формы Form1 с помощью «Инспектора объектов» (или двойным щелчком на форме) заготовку обработчика события onCreate, запишем в редакторе кода для него следующий программный код, позволяющий построить закрашенную область и графическое отображение элементов декартовой системы координат (самостоятельно объявите необходимые переменные в секции объявления глобальных переменных):

```
procedure TForm1.FormCreate(Sender: TObject);
begin
  Im1.Canvas.FloodFill(1, 1, clWhite, fsBorder);
  Im1.Canvas.Pen.Mode := pmCopy;
  Im1.Canvas.Font.Size := 12;
  Im1.Canvas.Font.Name := 'Courier New';
  Xmin := -1;
  Xmax := 5;
  Ymin := -5;
  Ymax := 1;
  ddx := 50;
  ddy := 50;
  W := Im1.Width;
  H := Im1.Height;
  dX := Xmax-Xmin;
  dY := Ymax-Ymin;
  //Определение положения точки O(0,0) на канве
  x0 := -Round((W - 2*ddx)/dX*Xmin) + Ddx;
```

```

y0 := Round((H - 2*ddy)/dY*Ymax) + Ddy;
Im1.Canvas.TextOut(x0-5-Im1.Canvas.TextWidth('0'), y0+7, '0');
//Построение оси OX
Im1.Canvas.MoveTo(10, y0);
Im1.Canvas.LineTo(w-10, y0);
Im1.Canvas.LineTo(w-40, y0-5);
Im1.Canvas.MoveTo(w-10, y0);
Im1.Canvas.LineTo(w-40, y0+5);
Im1.Canvas.TextOut(w-30, y0+7, 'X');
Kx := (W - 2*ddx)/dx;
Xt := Xmin;
while Xt<=Xmax do
begin
    Im1.Canvas.MoveTo(x0 + Round(xt*kx), Y0 - 5);
    Im1.Canvas.LineTo(x0 + Round(xt*kx), Y0 + 5);
    if xt<>0 then
        Im1.Canvas.TextOut(X0 + Round(Xt*Kx) -
Im1.Canvas.TextWidth(FloatToStr(Xt)) div 2, Y0 + 7, FloatToStr(Xt));
        Xt := Xt+1;
    end;
//Построение оси OY
Im1.Canvas.MoveTo(x0, h-10);
Im1.Canvas.LineTo(x0, 10);
Im1.Canvas.LineTo(x0-5, 40);
Im1.Canvas.MoveTo(x0, 10);
Im1.Canvas.LineTo(x0+5, 40);
Im1.Canvas.TextOut(x0 - 7 - Im1.Canvas.TextWidth('Y'), 20, 'Y');
Ky := (h - 2*ddy)/dy;
Yt := Ymin;
while yt<=Ymax do
begin
    Im1.Canvas.MoveTo(x0 - 5, y0 - Round(yt*ky));
    Im1.Canvas.LineTo(x0 + 5, y0 - Round(yt*ky));
    if yt<>0 then
        Im1.Canvas.TextOut(x0 - Round(yt*ky) -
Im1.Canvas.TextWidth(FloatToStr(yt)), y0-round(yt*ky)-
Im1.Canvas.TextHeight(FloatToStr(yt)) div 2, FloatToStr(yt));
        Yt := Yt+1;
    end;
//Построение контура фигуры
Im1.Canvas.Pen.Width := 2;
Im1.Canvas.Pen.Color := clRed;
Im1.Canvas.Arc(x0-round(kx), y0-round(ky), x0+round(5*kx),
y0+round(5*ky), 0, 0, 0, 0);
Im1.Canvas.MoveTo(X0 + round(1*kx), Y0 - round(-1*ky));
Im1.Canvas.LineTo(X0 + round(2*kx), Y0 - round(-1*ky));
Im1.Canvas.LineTo(X0 + round(4*kx), Y0 - round(-3*ky));
Im1.Canvas.LineTo(X0 + round(1*kx), Y0 - round(-3*ky));
Im1.Canvas.LineTo(X0 + round(1*kx), Y0 - round(-1*ky));
Im1.Canvas.Brush.Style := bsBDiagonal;
Im1.Canvas.Brush.Color := clBlue;
Im1.Canvas.FloodFill(x0 + round(3*kx), y0-round(-1*ky), clRed,
fsBorder);
end;

```

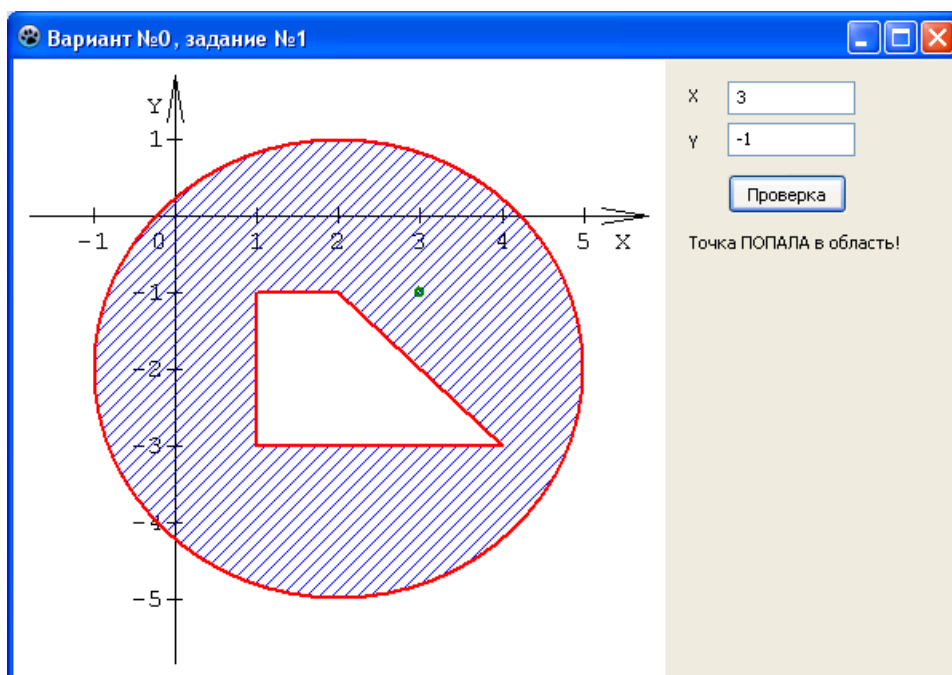
Создадим обработчик события onClick для кнопки Start (сообщение пользователю о принадлежности заданной точки заштрихованной области):

```

procedure TForm1.StartClick(Sender: TObject);
var
    X, Y: Real;
begin
    try
        X := StrToFloat(EdX.Text);
        Y := StrToFloat(EdY.Text);
        if (sqr(X-2) + sqr(Y+2) <= 9) and
            not ((X>1) and (Y<-1) and (Y>-3) and (Y<-X+1)) then
            L3.Caption := 'Точка ПОПАЛА в область!'
        else
            L3.Caption := 'Точка НЕ ПОПАЛА в область!';
            L3.Visible := True;
            Im1.Canvas.Pen.Color := clGreen;
            Im1.Canvas.Ellipse(x0+round(X*kx)-3, y0-round(Y*ky)+3,
x0+round(X*kx)+3, y0-round(Y*ky)-3);
        except
            on EConvertError do
                begin
                    MessageDlg('Ошибка при вводе координат проверяемой точки',
mtInformation, [mbOk], 0);
                    X := 0;
                    Y := 0;
                    EdX.Text := '0';
                    EdY.Text := '0';
                    L3.Visible:=False;
                end;
        end;
    end;
end;

```

Сохраним проект под именем «PrZ1» в папке Z1, а форму – «Z1». Запустим и несколько раз протестируем полученную программу, например:



9. Методические рекомендации по выполнению задания №2 компьютерной практики

Задание 2 (пример).

Дан массив $A(M,N)$, автоматически заполненный по заранее определенному закону. Удалить строки с четными номерами. Массив в исходном и преобразованном состоянии вывести на экран.

Создадим новый проект. У формы Form1 изменим значение свойства

Caption	Вариант №0, задание №2
---------	------------------------

Разместим на форме Form1 объекты управления, принадлежащие следующим классам, внося требуемые изменения в свойства данных объектов:

TPanel (для группировки управляющих элементов, формирующих и отображающих массив в исходном состоянии)

Name	Panel1
Align	alLeft
Caption	<пустая строка>

TSplitter (для изменения ширины Panel1)

Name	Splitter1
Align	alLeft

TPanel (для группировки управляющих элементов, формирующих и отображающих массив в преобразованном состоянии)

Name	Panel2
Align	alClient
Caption	<пустая строка>

Таким образом, Splitter1 должен располагаться между Panel1 и Panel2.

На панели Panel1 разместим объекта класса TPanel

(для обеспечения возможности корректного размещения кнопки, формирующей массив)

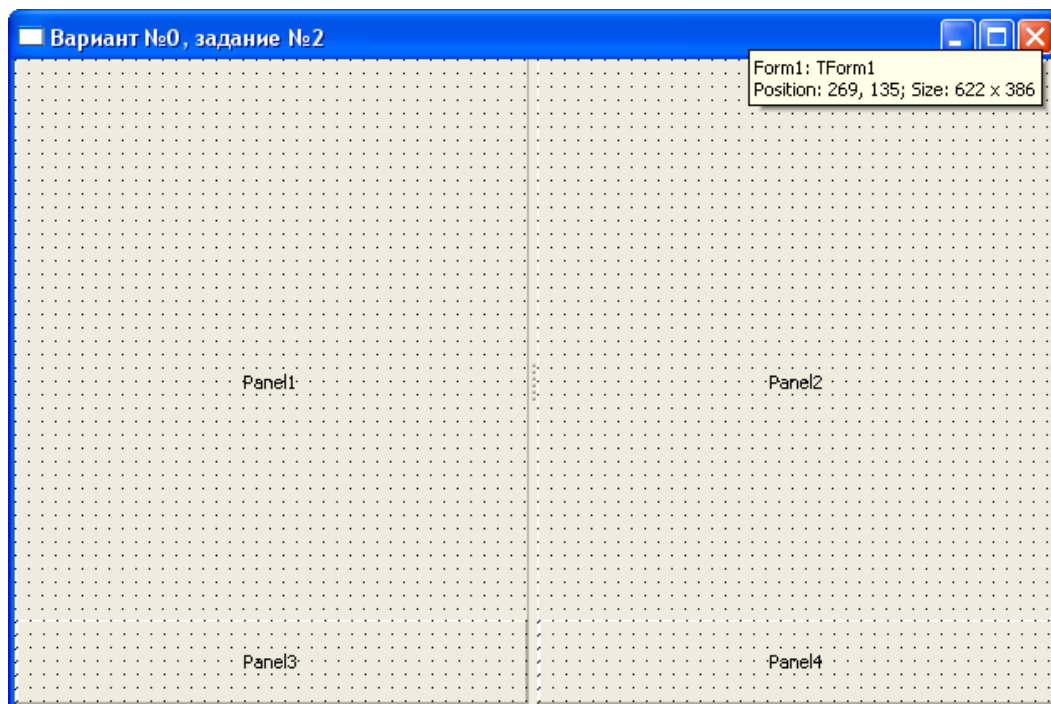
Name	Panel3
Align	alBottom
Caption	<пустая строка>

На панели Panel2 разместим объекта класса TPanel

(для обеспечения возможности корректного размещения кнопки, преобразующей массив в соответствии с заданием)

Name	Panel4
Align	alBottom
Caption	<пустая строка>

После выполнения описанных выше действий разрабатываемая экранная форма должна иметь следующий вид (подписей «Panel1» и др. будет не видно – на скриншоте они оставлены, чтобы обозначить наличие панелей):



Далее на панели Panel1 разместим объекта класса TStringGrid (для вывода элементов массива в исходном состоянии)

Name	SG1
Align	alClient
FixedCols	0
FixedRows	0
DefaultColWidth	50

На панели Panel3 разместим объекта класса TButton (для запуска процедуры генерации исходного массива)

Name	Start
Caption	Создать

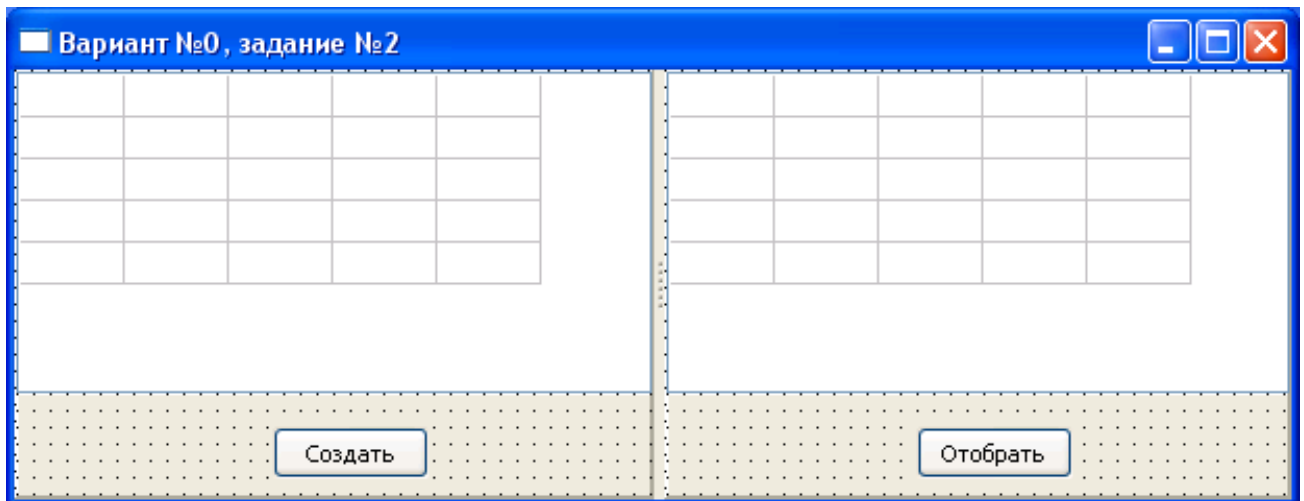
На панели Panel2 разместим объекта класса TStringGrid (для вывода элементов массива в преобразованном состоянии)

Name	SG2
Align	alClient
FixedCols	0
FixedRows	0
DefaultColWidth	50

На панели Panel4 разместим объекта класса TButton (для запуска процедуры преобразования массива в соответствии с заданием)

Name	Finish
Caption	Отобразить
Enabled	False

Таким образом, редактируемая форма должна иметь следующий вид:



Зададим размер массива в разделе описания, используя константы:

const

```
RCount = 5; //Количество строк массива
CCount = 6; // Количество столбцов массива
```

Опишем в пользовательский тип для нашего массива:

type

```
MyArray = array[1..RCount, 1..CCount] of Real;
```

В секции **var** объявим глобальные переменные для хранения текущего размера массива и самого массива:

```
M, N: Integer;
A: MyArray;
```

Запишем ряд процедур.

1. Процедура формирования массива вещественных чисел размером RCount строк на CCount столбцов. Значение элемента вычисляется по следующему правилу: «Номер строки – целая часть значения элемента, номер столбца – его сотые доли».

procedure Generation;

var

```
I, J: Integer;
```

begin

```
M := RCount;
```

```
N := CCount;
```

```
for I := 1 to M do
```

```
  for J := 1 to N do
```

```
    A[I, J] := I + J/100;
```

end;

2. Процедура, осуществляющая удаление чётных строк массива с одновременным изменением значения переменной M, отвечающей за временное хранение числа строк массива.

procedure Work;

var

```
I, J: Integer;
```

begin

```
I := 3;
```

```
while I <= M do
```

```
  begin
```

```
    for J := 1 to N do
```

```
      begin
```

```
        A[(I+1) div 2, J] := A[I, J];
```

```

    A[I, J] := 0;
  end;
  I := I + 2;
end;
M := (M+1) div 2;
end;

```

Выберем для кнопки Start в «Инспекторе объектов» событие onClick и запишем в обработчике события для него следующий программный код, обеспечивающий формирование и вывод на экран (в ячейки сетки SG1) исходного массива:

```

procedure TForm1.StartClick(Sender: TObject);
var
  I, J: Integer;
begin
  Generation;
  SG1.RowCount := M;
  SG1.ColCount := N;
  for I := 1 to M do
    for J := 1 to N do
      SG1.Cells[J-1, I-1] := FloatToStr(A[I, J]);
  Finish.Enabled := True;
end;

```

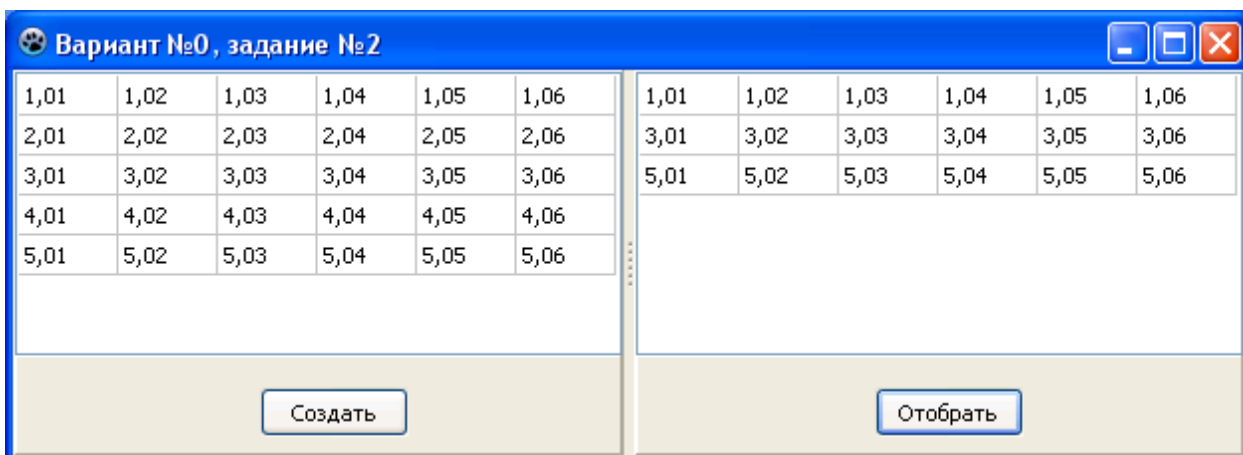
Выберем для кнопки Finish в «Инспекторе объектов» событие onClick и запишем в обработчике события для него следующий программный код, обеспечивающий удаление чётных строк массива и вывод преобразованного массива на экран (в ячейки сетки SG2):

```

procedure TForm1.FinishClick(Sender: TObject);
var
  I, J: Integer;
begin
  Work;
  SG2.RowCount := M;
  SG2.ColCount := N;
  for I := 1 to M do
    for J := 1 to N do
      SG2.Cells[J-1, I-1] := FloatToStr(A[I, J]);
end;

```

Сохраним проект под именем «PrZ2» в папке Z2, а форму – «Z2». Запустим и протестируем полученную программу, например:



10. Методические рекомендации по выполнению задания №3 компьютерной практики

Задание 3 (пример).

Сведения о стране состоят из названия государства, столицы, площади, численности населения. Пусть дан массив, содержащий сведения о нескольких (не менее 10) государствах, созданный с помощью генератора случайных чисел. Обеспечив предварительную коррекцию исходных данных, выведите все сведения о самом маленьком по площади государстве.

Создадим новый проект. У формы Form1 изменим значение свойства

Caption	Вариант №0, задание №3
---------	------------------------

Разместим на форме Form1 объекты управления, принадлежащие следующим классам, внося требуемые изменения в свойства данных объектов:

TPanel (для корректного позиционирования на форме сетки, позволяющей отображать и редактировать исходные данные)

Name	Panel1
Align	alTop
Caption	<пустая строка>

TSplitter (для изменения ширины Panel1)

Name	Splitter1
Align	alTop
Caption	<пустая строка>

TPanel (для корректного позиционирования на форме кнопки, нажатие на которую осуществляет выбор нужных сведений)

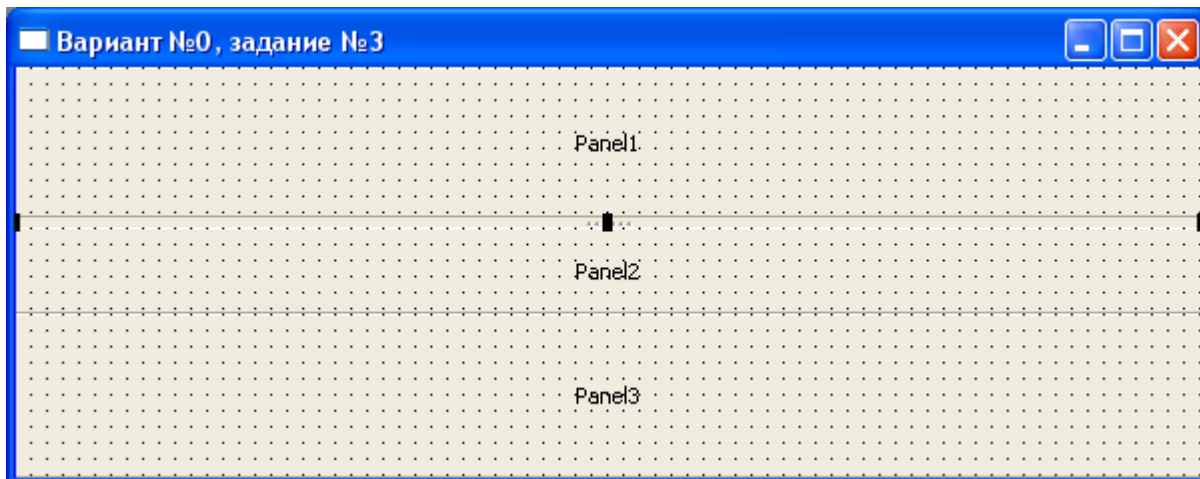
Name	Panel2
Align	alTop
Caption	<пустая строка>

TPanel (для корректного позиционирования на форме сетки, позволяющей отображать о самом маленьком государстве (или нескольких самых маленьких))

Name	Panel3
Align	alClient
Caption	<пустая строка>

Таким образом, Splitter1 должен располагаться между Panel1 и Panel2.

После выполнения описанных выше действий разрабатываемая экранная форма должна иметь следующий вид (подписи «Panel1» и др. не должны быть видны):



Далее на панели Panel1 разместим объект класса TStringGrid (для вывода и коррекции исходных данных о странах)

Name	SG1
Align	alClient
FixedCols	1
FixedRows	1
Options	goColSizing goEditing goFixedColSizing

Для установки значений свойства Options нужно щёлкнуть на стрелке (или плюсики) рядом с этим свойством и выбрать значения «True» для перечисленных в таблице подstroyств.

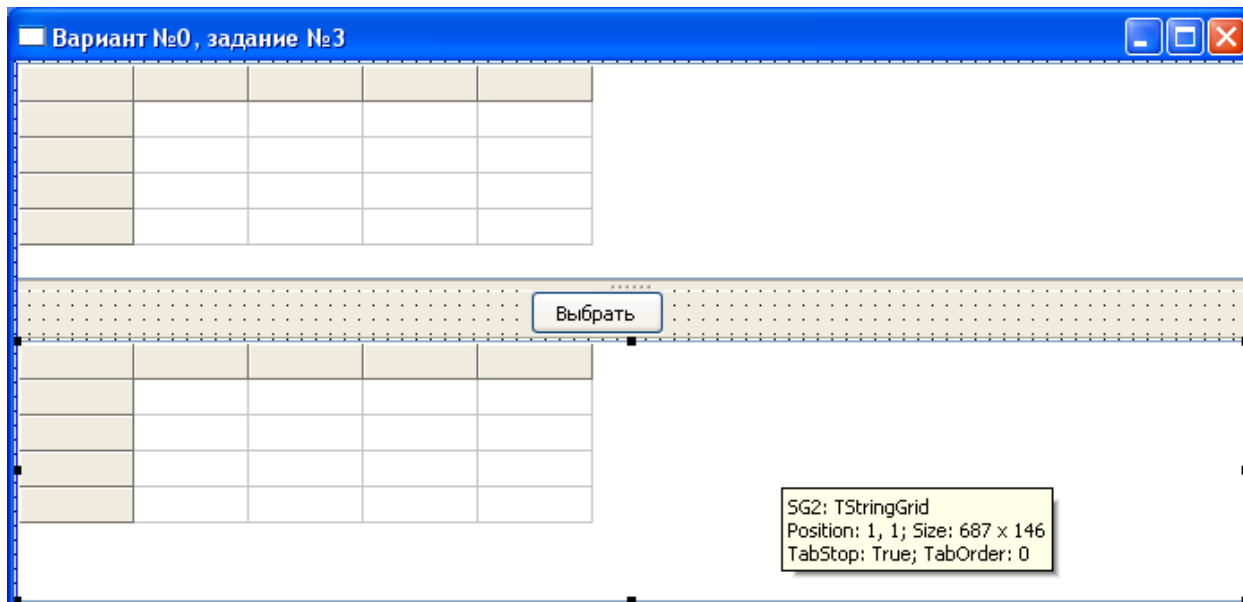
На панели Panel2 разместим объекта класса TButton (для запуска процедуры выбора нужных сведений)

Name	Work
Caption	Выбрать

На панели Panel3 разместим объекта класса TStringGrid (для вывода данных о выбранных странах)

Name	SG2
Align	alClient
FixedCols	1
FixedRows	1
Options	goColSizing goFixedColSizing

Таким образом, редактируемая форма должна иметь следующий вид:



Зададим размер массива в разделе описания, используя константу:

```
const
```

```
    Size = 12; //Длина массива
```

Опишем собственный тип данных («запись») для хранения информации об отдельной стране в разделе описания типов:

```
type
```

```
    Страна = record
        Nazv: string[30];
        Stolica: string[25];
        Ploshad: Real;
        Naselenie: Longint;
```

```
    end;
```

Объявим (в соответствующей секции) глобальную переменную – массив стран:

```
AStra: array [1..Size] of Страна;
```

Запишем ряд процедур.

1. Процедура формирования шапки таблицы, определения ширины столбцов и исходного количества строк для вывода сведений о странах:

```
procedure SetGridParams(SG: TStringGrid; R: Integer);
```

```
begin
```

```
    SG.RowCount := R+1;
    SG.ColWidths[0] := 50;
    SG.ColWidths[1] := 250;
    SG.ColWidths[2] := 150;
    SG.ColWidths[3] := 100;
    SG.ColWidths[4] := 100;
    SG.Cells[0,0] := '№ п/п';
    SG.Cells[1,0] := 'Страна';
    SG.Cells[2,0] := 'Столица';
    SG.Cells[3,0] := 'Площадь';
    SG.Cells[4,0] := 'Население';
```

```
end;
```

2. Процедура вывода сведений о стране из элемента массива в заданную строку указанной сетки:

```
procedure SetGridValues(SG: TStringGrid; V: Страна; N: Longint);
```

```
begin
```

```

if SG.RowCount <= N then
    SG.RowCount := N+1;
    SG.Cells[0,N] := IntToStr(N);
    SG.Cells[1,N] := V.Nazv;
    SG.Cells[2,N] := V.Stolica;
    SG.Cells[3,N] := FloatToStr(V.Ploshad);
    SG.Cells[4,N] := IntToStr(V.Naselenie);
end;

```

Затем создадим в обработчике события onCreate для формы Form1 программный код генерации и вывода сведений о странах:

```

procedure TForm1.FormCreate(Sender: TObject);
var
    I, J, K: Integer;
begin
    SetGridParams(SG1, Size);
    SetGridParams(SG2, 0);

    for I := 1 to Size do
        begin
            K := Random(11) + 10;
            AStra[I].Nazv := Chr(Random(26)+65);
            for J := 2 to K do
                AStra[I].Nazv := AStra[I].Nazv+Chr(Random(26)+97);
            Sg1.Cells[1, I] := AStra[I].Nazv;
            K := Random(11)+10;
            AStra[I].Stolica := Chr(Random(26)+65);
            for J := 2 to K do
                AStra[I].Stolica := AStra[I].Stolica+Chr(Random(26)+97);
            Sg1.Cells[1,I] := AStra[I].Stolica;
            AStra[I].Ploshad := Random(10000)/100;
            AStra[I].Naselenie := Random(10000);
            SetGridValues(SG1, AStra[I], I);
        end;
    end;

```

Выберем для кнопки Work в «Инспекторе объектов» событие onClick и запишем в редакторе кода для него следующий программный код, обеспечивающий выбор сведений о самых маленьких странах и вывод их на экран (в ячейки сетки SG2):

```

procedure TForm1.WorkClick(Sender: TObject);
var
    I, K: Integer;
    Err: Boolean;
    PMin: Real;
begin
    Err := False;
    for I := 1 to Size do
        begin
            AStra[I].Nazv := Sg1.Cells[1, I];
            AStra[I].Stolica := Sg1.Cells[2, I];
            try
                AStra[I].Ploshad := StrToFloat(Sg1.Cells[3, I]);
            except
                Err := True;
        end;
    end;

```

```

    ShowMessage('Ошибка при вводе площади страны № ' + IntToStr(I)
+ ' (' + AStra[I].Nazv + '):' + #13 + ''' + Sg1.Cells[3,I] + ''');
    Sg1.Cells[3, I] := FloatToStr(AStra[I].Ploshad);
end;
try
    Astra[I].Naselenie := StrToInt(Sg1.Cells[4,I]);
except
    Err := True;
    ShowMessage('Ошибка при вводе населения страны № ' +
IntToStr(I) + ' (' + AStra[I].Nazv + '):' + #13 + ''' +
Sg1.Cells[4,I] + ''');
    Sg1.Cells[4, I] := IntToStr(AStra[I].Naselenie);
end;
end;
if Err then
    ShowMessage('В ходе сохранения данных возникли ошибки!' + #13 +
'Еще раз проверьте данные и повторите снова!')
else
begin
    PMin := AStra[I].Ploshad;
    for I := 2 to Size do
        if AStra[I].Ploshad < PMin then
            PMin := AStra[I].Ploshad;
    K := 0;
    for I := 1 to Size do
        if AStra[I].Ploshad = PMin then
            begin
                K := K + 1;
                SetGridValues(SG2, AStra[I], K);
            end;
end;
end;
end;
end;

```

Сохраним проект под именем «PrZ3» в папке Z3, а форму – «Z3». Запустим и протестируем полученную программу, например:

№ п/п	Страна	Столица	Площадь	Население
1	Psypwowlqjilxbzhjmuvmnoky	Bicqajvyudwzwmulnu	67,88	1182
2	Qpdnytnckmgeutlfodaiqdpfqky	Rljplxscbzrqrfdti	60,78	3637
3	Oalqzycqfgezpqgkmmgqaic	Ridufyjrvcqvrccz	87,81	4686
4	Zbpltblhzdjhmdriwkbfsou	Gnncxcocyhiarvdsvh	26,47	1831
5	Poaevjadrohsthzgtokpxoq	Xyrlwzsdhtv	41,18	3965
6	Wgpiwzsprrnyfqklqpmamhrhh	Qtlwdchcojpoa	65,32	9689
7	Ilfxdjclzngxunlccxjsmzvdxwa	Nqeduwukod	40,71	332
8	Zsjksbwjzawfatjyste	Nmbifeaeumfwi	94,41	9280
9	Spaueqbpsgygprnwptsziokf	Fefyotlmzfrg	8,52	580
10	Lmiwzsjqeoawbtrmlrnp	Szmlrxgdafwtwixqpi	83,1	5197
11	Awihluxerylrmfuyftwgifynma	Fdlsjqmfhepuwtdrn	8,52	1320
12	Sskmovefdemdjsyatttcx	Ctotpbzfhgr	10,02	630

Выбрать

№ п/п	Страна	Столица	Площадь	Население
1	Spaueqbpsgygprnwptsziokf	Fefyotlmzfrg	8,52	580
2	Awihluxerylrmfuyftwgifynma	Fdlsjqmfhepuwtdrn	8,52	1320

11. Методические рекомендации по выполнению задания №4 компьютерной практики

Задание 4 (пример).

Сведения о стране состоят из названия государства, столицы, площади, численности населения, части света в которой расположено данное государство. Пусть дан файл, содержащий сведения о государствах (не менее 50), заполненный пользователем. Сформируйте новый файл, содержащий все сведения о самых маленьких по численности населения государствах в каждой части света.

Создадим новый проект. У формы Form1 изменим значение свойства

Caption	Вариант №0, задание №4
---------	------------------------

Разместим на форме Form1 объекты управления, принадлежащие следующим классам, внося требуемые изменения в свойства данных объектов:

TPageControl (для группировки управляющих элементов в соответствии с выполняемыми задачами)

Name	PC1
Align	alClient

Используя пункты контекстного меню для данного объекта добавьте 3 страницы TTabSheet.

Замечание: переход к страницам для настройки их свойств осуществляется через «Инспектор объектов» или щелчком на самой странице (после щелчка на её «вкладке»).

Первая страница TTabSheet (для компактного размещения управляющих элементов, предназначенных для загрузки и отображения сведений исходного файла).

Name	SourceTS
Caption	Исходные данные

Вторая страница TTabSheet (для компактного размещения управляющих элементов, предназначенных для отображения результатов отбора и выгрузки сведений в файл).

Name	ResultTS
Caption	Результаты

Третья страница TTabSheet (для группировки управляющих элементов, предназначенных для добавления сведений о новой стране).

Name	AddTS
Caption	Добавление

Добавьте на форму невидимые диалоговые компоненты.

TOpenDialog (для диалогового окна открытия файла).

Name	OD1
Filter	Файл данных *.dat
DefaultExt	.dat

TSaveDialog (для диалогового окна сохранения файла).

Name	SD1
------	-----

На странице SourceTS разместим объекта класса TPanel (для обеспечения возможности корректного размещения кнопок чтения и записи данных).

Name	Panel1
Align	alBottom
Caption	<пустая строка>

На странице SourceTS разместим объекта класса TStringGrid (для отображения данных исходного файла).

Name	DataSG
Align	alClient
FixedCols	1
FixedRows	1
Options	goColSizing goFixedColSizing

На панели Panel1 разместим два объекта класса TButton

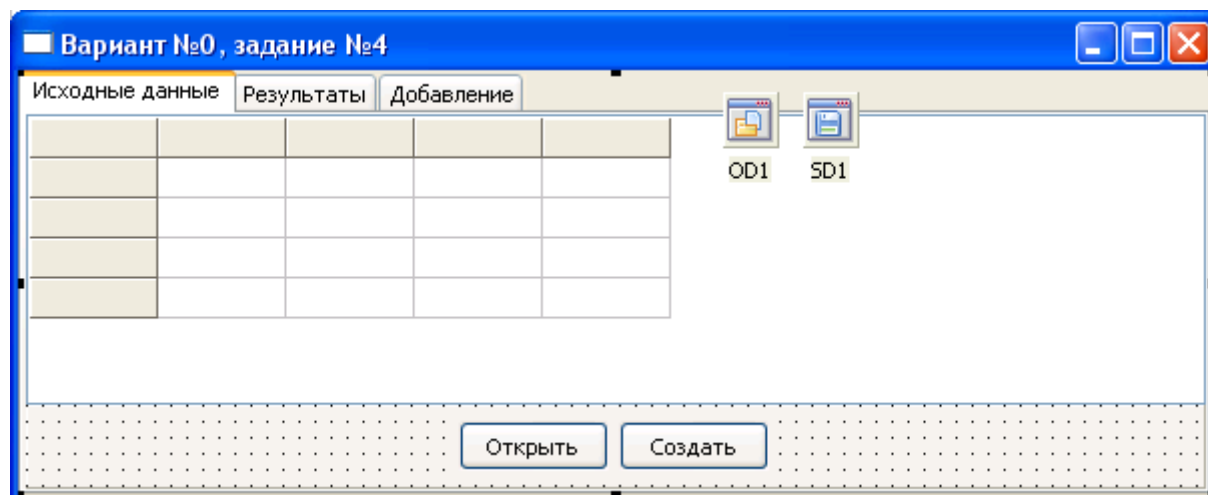
Первая кнопка (для выбора исходного файла и отображения на экране хранящихся в нём сведений).

Name	BOpen
Caption	Открыть

Вторая кнопка (для выбора файла, в котором будут храниться отредактированные исходные сведения, и сохранения данной информации в нём).

Name	BCreate
Caption	Сохранить

Таким образом, редактируемая форма с открытой страницей SourceTS должна иметь следующий вид:



На странице ResultTS разместим объект класса TPanel (для обеспечения возможности корректного размещения кнопки записи выбранных данных)

Name	Panel2
Align	alBottom
Caption	<пустая строка>

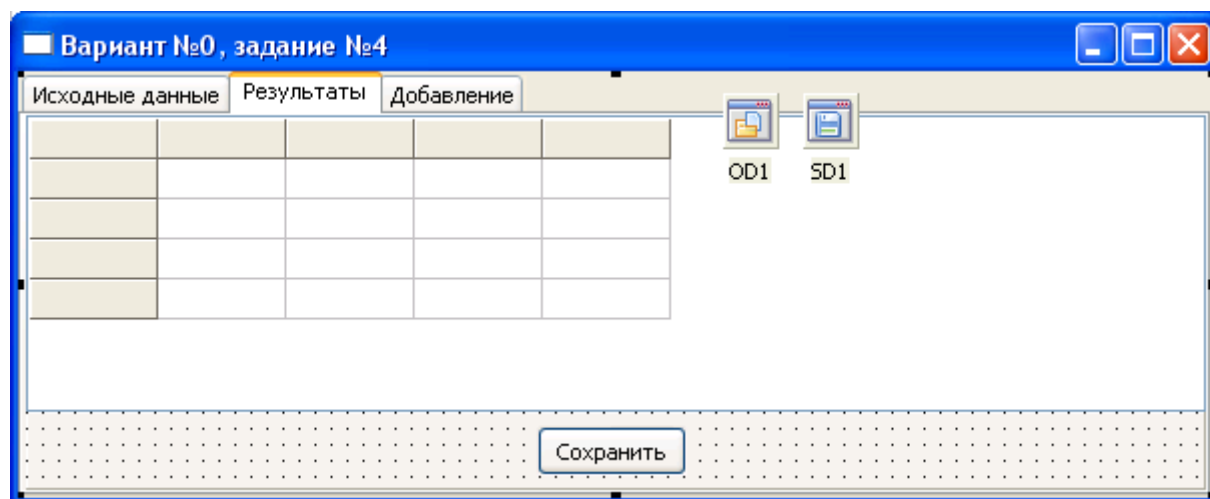
На вкладке ResultTS разместим объекта класса TStringGrid (для отображения выбранных данных)

Name	ResSG
Align	alClient
FixedCols	1
FixedRows	1
Options	goColSizing goFixedColSizing

На панели Panel2 разместим объект класса TButton (для выбора файла, в котором будут храниться выбранные сведения, и сохранения данной информации в нём)

Name	BSave
Caption	Сохранить

Таким образом редактируемая форма с открытой страницей ResultTS должна иметь следующий вид:



На вкладке AddTS разместим пять объектов класса TEdit (**замечание:** в Lazarus лучше использовать компоненты типа TLabelEdit – тогда не нужно будет добавлять пять меток, о которых будет написано ниже):

1 для ввода названия новой страны.

Name	ENazv
Text	<пустая строка>

2 для ввода столицы новой страны.

Name	EStolica
Text	<пустая строка>

3 для ввода площади новой страны.

Name	EPloshad
Text	0

4 для ввода численности населения новой страны.

Name	ENaselenie
Text	0

5 для ввода части света новой страны.

Name	ESvet
Text	<пустая строка>

На вкладке AddTS разместим пять объектов класса TLabel

1 для отображения подписи к полю ввода названия новой страны.

Name	LNazv
Caption	Название

2 для отображения подписи к полю ввода столицы новой страны.

Name	EStolica
Caption	Столица

3 для отображения подписи к полю ввода площади новой страны.

Name	EPloshad
Caption	Площадь

4 для отображения подписи к полю ввода численности населения новой страны.

Name	ENaselenie
Caption	Население

5 для отображения подписи к полю ввода части света новой страны.

Name	ESvet
Caption	Часть света

На вкладке AddTS разместим два объекта класса TButton

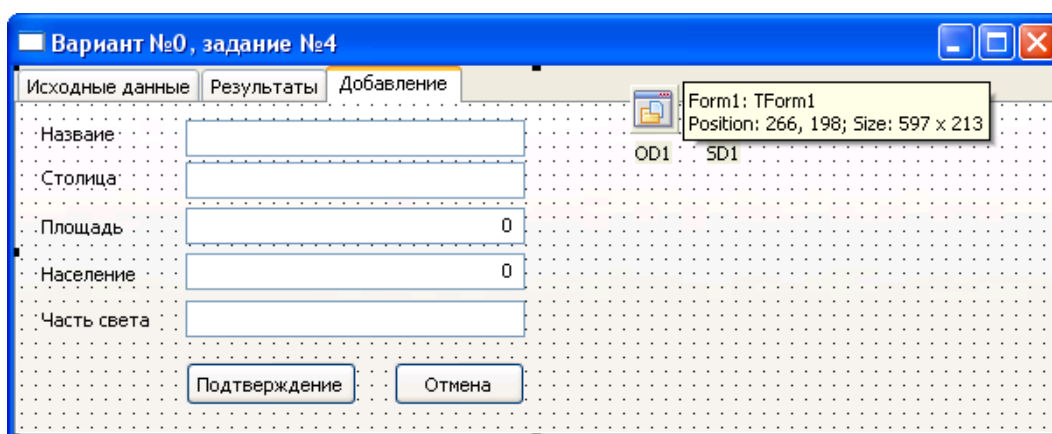
1 для подтверждения отправки размещенных в текстовых полях данных в объект для отображения исходных сведений (DataSG) в виде новой строки.

Name	BYes
Caption	Подтверждение

2 для очистки текстовых полей для ввода новых сведений.

Name	BNo
Caption	Отмена

Таким образом, редактируемая форма с открытой страницей AddTS должна иметь следующий вид:



Объявим в секции объявления типов пользовательский тип для хранения сведений о стране:

```
Strana = record
  Nazv: string[60];
  Stolica: string[50];
  Ploshad: Real;
  Naselenie: Longint;
  Svet: string[40];
end;
```

Опишем в секции объявления переменных соответствующую глобальную переменную (будет использоваться для обработки каждой записи файла):

```
St: Strana;
```

Объявим глобальные переменные для доступа к исходному файлу (он должен быть типизированным) и файлу с результатом (текстовый файл):

```
FIn: file of Strana;
FOut: Text;
```

Также объявим глобальные переменные для хранения имени исходного и результирующего файла:

```
FInName, FOutName: string;
```

Запишем ряд подпрограмм.

1. Процедура формирования шапки таблицы и установки ширины столбцов для вывода сведений о странах:

```
procedure SetGridParams(SG: TStringGrid);
begin
  SG.ColCount := 6;
  SG.RowCount := 1;
  SG.ColWidths[0] := 50;
  SG.ColWidths[1] := 250;
  SG.ColWidths[2] := 150;
  SG.ColWidths[3] := 100;
  SG.ColWidths[4] := 100;
  SG.ColWidths[5] := 150;
  SG.Cells[0,0] := '№ п/п';
  SG.Cells[1,0] := 'Страна';
  SG.Cells[2,0] := 'Столица';
  SG.Cells[3,0] := 'Площадь';
  SG.Cells[4,0] := 'Население';
  SG.Cells[5,0] := 'Часть света';
end;
```

2. Процедура вывода сведений о стране из записи в заданную строку указанной таблицы:

```
procedure SetGridValues(SG: TStringGrid; V: Strana; N: Longint);
begin
  if SG.RowCount<=N then
    SG.RowCount := N + 1;
  SG.Cells[0,N] := IntToStr(N);
  SG.Cells[1,N] := V.Nazv;
  SG.Cells[2,N] := V.Stolica;
  SG.Cells[3,N] := FloatToStr(V.Ploshad);
  SG.Cells[4,N] := IntToStr(V.Naselenie);
  SG.Cells[5,N] := V.Svet;
```

end;

3. Функция чтения строки сведений из таблицы с данными:

```
function LoadFromGrid(SG: TStringGrid; N: Integer): Strana;  
var
```

```
    Res: Strana;
```

```
begin
```

```
    Res.Nazv := SG.Cells[1, N];
```

```
    Res.Stolica := SG.Cells[2, N];
```

```
    Res.Ploshad := StrToFloat(SG.Cells[3, N]);
```

```
    Res.Naselenie := StrToInt(SG.Cells[4, N]);
```

```
    Res.Svet := SG.Cells[5, N];
```

```
    Result := Res;
```

```
end;
```

Создадим для формы Form1 обработчик события onCreate и запишем в него программный код, задающий внешний вид таблиц со сведениями:

```
procedure TForm1.FormCreate(Sender: TObject);
```

```
begin
```

```
    SetGridParams(DataSG);
```

```
    SetGridParams(ResSG);
```

```
    Form1.Width := 850;
```

```
end;
```

Для кнопки BOpen в обработчик события onClick запишем программный код, обеспечивающий открытие исходного файла и отображение его содержимого:

```
procedure TForm1.BOpenClick(Sender: TObject);
```

```
var
```

```
    K: Integer;
```

```
begin
```

```
    if OD1.Execute then
```

```
        begin
```

```
            FInName := UTF8ToSys(OD1.FileName); // Для Lazarus
```

```
            //FInName := OD1.FileName; // Для Delphi
```

```
            try
```

```
                try
```

```
                    AssignFile(FIn, FInName);
```

```
                    Reset(FIn);
```

```
                    K := 0;
```

```
                    while not (Eof(FIn)) do
```

```
                        begin
```

```
                            Read(FIn, St);
```

```
                            K := K + 1;
```

```
                            SetGridValues(DataSG, St, K);
```

```
                        end;
```

```
                    except
```

```
                        ShowMessage('Ошибка загрузки данных из файла ' + #13 + '"' + FInName + '"');
```

```
                    end;
```

```
                finally
```

```
                    CloseFile(FIn);
```

```
                end;
```

```
            end;
```

```
end;
```

Для кнопки BCreate в обработчик события onClick запишем программный код, обеспечивающий создание (при необходимости) файла и сохранение сведений, расположенных в таблице, отображающей исходные данные (DataSG), в него:

```

procedure TForm1.BCreateClick(Sender: TObject);
var
    I: Integer;
begin
    SD1.Filter := 'Файл данных|*.dat';
    SD1.DefaultExt := '.dat';
    if SD1.Execute then
    begin
        FInName := UTF8ToSys(SD1.FileName); // Для Lazarus
        //FInName := SD1.FileName; // Для Delphi
        try
            try
                AssignFile(FIn, FInName);
                Rewrite(FIn);
                for I := 1 to DataSG.RowCount-1 do
                    Write(FIn, LoadFromGrid(DataSG, I));
            except
                ShowMessage('Ошибка сохранения данных в файл ' + #13 + ''' +
                    FInName + '''');
            end;
        finally
            CloseFile(FIn);
        end;
    end;
end;

```

Сохраним проект под именем «PrZ4» в папке Z4, а форму – «Z4». Запустим и протестируем полученную программу в части работоспособности элементов управления первой вкладки, например:

№ п/п	Страна	Столица	Площадь	Население	Часть света
1	Австралия	Канберра	7692024	23130931	Австралия и Океания
2	Маршалловы Острова	Маджуро	181	53158	Австралия и Океания
3	Новая Зеландия	Веллингтон	268680	4596700	Австралия и Океания
4	Папуа - Новая Гвинея	Порт-Морсби	462840	7334638	Австралия и Океания
5	Соломоновы Острова	Хониара	28450	515870	Австралия и Океания
6	Федеративные Штаты Микронезии	Паликир	702	106104	Австралия и Океания
7	Самоа	Апиа	2832	187820	Австралия и Океания
8	Тонга	Нукуалофа	748	103252	Австралия и Океания
9	Фиджи	Сува	18274	849000	Австралия и Океания
10	Индонезия	Джакарта	1919440	253609643	Азия
11	Пакистан	Исламабад	803940	188444000	Азия
12	Бангладеш	Дакка	144000	166280712	Азия
13	Япония	Токио	377944	126910000	Азия
14	Филиппины	Манила	299764	101108300	Азия
15	Вьетнам	Ханой	331210	92477857	Азия
16	Иран	Тегеран	1648000	78108412	Азия
17	Турция	Анкара	783562	77695904	Азия
18	Таиланд	Бангкок	514000	70498494	Азия
19	КНДР	Пхеньян	120540	24720407	Азия
20	Мексика	Мехико	1972550	121736809	Америка

(У вас сетка с данными будет пока пустой, но кнопки «Открыть» и «Сохранить» уже должны работать)

Замечание: кнопка «Открыть» будет работать не совсем корректно (увидеть это можно будет несколько позже, когда появится возможность добавлять данные): если в открываемом файле меньше строк, чем уже загружено в таблице, то после загрузки останутся «лишние» строки. Исправьте этот недочёт самостоятельно.

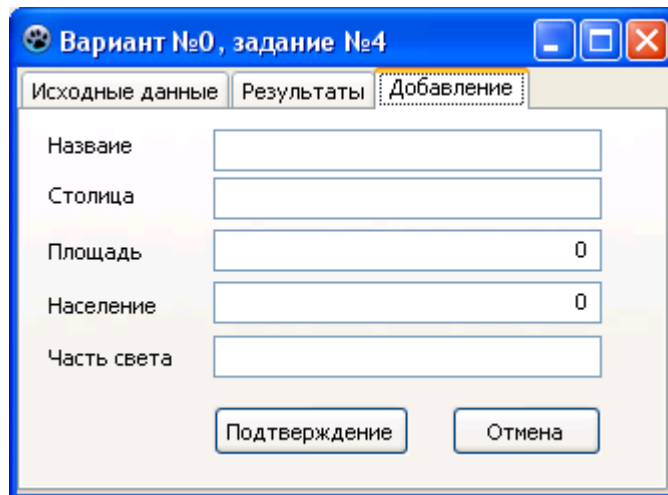
Закроем программу и продолжим редактирование исходного кода. Для кнопки BNo в обработчик события onClick, запишем программный код, обеспечивающий очистку полей ввода сведений о стране:

```
procedure TForm1.BNoClick(Sender: TObject);  
begin  
    ENazv.Text := '';  
    EStolica.Text := '';  
    EPloshad.Text := '0';  
    ENaselenie.Text := '0';  
    ESvet.Text := '';  
end;
```

Для кнопки BYes в обработчик события onClick запишем программный код, обеспечивающий размещение новых сведений о стране в таблице для отображения данных исходного файла (DataSG):

```
procedure TForm1.BYesClick(Sender: TObject);  
var  
    Err: Boolean;  
begin  
    Err := False;  
    St.Nazv := ENazv.Text;  
    St.Stolica := EStolica.Text;  
    try  
        St.Ploshad := StrToFloat(EPloshad.Text);  
    except  
        Err := True;  
        ShowMessage('Ошибка при вводе площади страны!');  
    end;  
    try  
        St.Naselenie := StrToInt(ENaselenie.Text);  
    except  
        Err := True;  
        ShowMessage('Ошибка при вводе населения страны!');  
    end;  
    St.Svet := ESvet.Text;  
    if Err then  
        ShowMessage('При вводе данных возникли ошибки!' + #13 + 'Внесите в данные необходимые исправления и повторите попытку!')  
    else  
        begin  
            SetGridValues(DataSG, St, DataSG.RowCount);  
            BNoClick(Self);  
        end;  
    end;
```

Запустим и протестируем полученную программу в части работоспособности элементов управления третьей вкладки:



Замечание: обратите внимание, что добавление данных происходит в таблицу DataSG, а не в файл, поэтому, чтобы в дальнейшем программа могла использовать добавленные данные для выбора и отображения в таблице на вкладке «Результаты», необходимо будет предварительно сохранить данные в файл с помощью соответствующей кнопки на странице «Исходные данные». Более правильное решение: сразу сохранять данные в файл (при необходимости – запросить выбор файла) и обновлять таблицу «Исходные данные». Реализуйте такой подход самостоятельно (для оценки «хорошо» и выше).

Закроем форму и продолжим редактирование программы. Для страницы ResultTS создадим обработчик события onShow и запишем в него программный код, обеспечивающий выбор сведений о самых маленьких по численности населения государствах в каждой части света и размещение этих сведений в сетке, отображающей выбранные данные (ResSG).

Так как использовать вспомогательные массивы запрещено, для решения задачи потребуется два прохода по файлу: за первый проход находим минимальные значения населения для каждой из частей света, а во время второго прохода уже выводим искомые значения в результирующую таблицу. Для хранения значения минимального населения для частей света объявим пользовательский тип Schet.

```

procedure TForm1.ResultTSShow(Sender: TObject);
type
  Schet = record
    Svet: string[40];
    Naselenie: Longint;
  end;
var
  I, K, N: Integer;
  Flag: Boolean;
  M: array[1..100] of Schet;
begin
  N := 0; // Количество частей света
  if FileExists(FInName) then
    try
      try
        AssignFile(FIn, FInName);
        // Первый проход: Определение параметров отбора
        // (минимум населения для каждой части света)
        Reset(FIn);
        while not Eof(FIn) do
          begin

```



```

Read(FIn, St);
Flag := True; // Такой части света ещё нет в массиве
I := 1;
while (I <= N) and Flag do
begin
  if St.Svet = M[I].Svet then
  begin
    Flag := False;
    if St.Naselenie < M[I].Naselenie then
      M[I].Naselenie := St.Naselenie;
  end;
  I := I + 1;
end;
if Flag then
begin
  N := N + 1;
  M[N].Svet := St.Svet;
  M[N].Naselenie := St.Naselenie;
end;
end;
// Второй проход: выбор с использованием параметров
Reset(FIn);
K := 0;
while not Eof(FIn) do
begin
  Read(FIn, St);
  for I := 1 to N do
  if (St.Svet = M[I].Svet) and
    (St.Naselenie = M[I].Naselenie) then
  begin
    K := K + 1;
    SetGridValues(ResSG, St, K);
  end;
end;
except
  ShowMessage('Ошибка при работе с файлом!');
end;
finally
  CloseFile(FIn);
end;
end;

```

Для кнопки BSave в обработчик события onClick запишем программный код, обеспечивающий сохранение выбранных сведений (из таблицы ResSG) в текстовый файл:

```

procedure TForm1.BSaveClick(Sender: TObject);
var
  I: Integer;
begin
  SD1.Filter := 'Текстовый файл|*.txt';
  SD1.DefaultExt := '.txt';
  if SD1.Execute then
  begin
    FOutName := UTF8ToSys(SD1.FileName); // Для Lazarus
    // FOutName := SD1.FileName; // Для Delphi
  end;

```

```

try
  try
    AssignFile(FOut, FOutName);
    Rewrite(FOut);
// Для Lazarus самостоятельно добавьте преобразование кодировки
Write(FOut, '№ п/п':5);
Write(FOut, 'Страна':30);
Write(FOut, 'Столица':30);
Write(FOut, 'Площадь':20);
Write(FOut, 'Население':20);
Writeln(FOut, 'Часть света':20);
  for I := 1 to ResSG.RowCount-1 do
  begin
    Write(FOut, ResSG.Cells[0, I]:5);
    Write(FOut, ResSG.Cells[1, I]:60);
    Write(FOut, ResSG.Cells[2, I]:50);
    Write(FOut, ResSG.Cells[3, I]:25);
    Write(FOut, ResSG.Cells[4, I]:25);
    Writeln(FOut, ResSG.Cells[5, I]:40);
  end;
except
  ShowMessage('Ошибка сохранения данных в файл ' + #13 + '"' +
FOutName + '"');
end;
finally
  CloseFile(FOut);
end;
end;
end;

```

Сохраним форму. Запустим и протестируем полученную программу в части работоспособности элементов управления второй вкладки, предварительно открыв данные исходного файла, например:

