

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ**

**ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ**

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

**к курсовой работе**

**по курсу «Информатика и основы программирования»**  
для студентов дневной формы обучения направления  
7.0709 «Геодезия, картография и землеустройство», спе-  
циальностей 7.070908 «Геоинформационные системы и  
технологии», 7.070901 «Инженерная геодезия», 7.070905  
«Землеустройство и кадастр».

**У т в е р ж д е н о**  
на заседании кафедры  
геоинформатики и геодезии  
Протокол №2 от 2.03.2005

Донецк ДонНТУ-2005

## УДК 528

Методические указания к курсовой работе по курсу «Информатика и основы программирования», раздел «визуальная среда программирования Delphi» для студентов первого курса специальностей: «инженерная геодезия», «геоинформационные системы и технологии», «землеустройство и кадастр»/ Сост. Гермонова Е.А., Митрофанова Е.И..- Донецк: ДонНТУ, 2005.-56 с.

Рассмотрены вопросы программирования в среде Delphi-5 геодезических задач, порядок оформления курсовой работы, изложена методика ее выполнения, предложен пример.

Составители: Гермонова Е.А.  
Митрофанова Е.И.

Рецензенты: Грищенко Н.Н.

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ .....	4
Порядок выполнения курсовой работы.....	4
Содержание пояснительной записки.....	5
1 ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ .....	5
2 ПОСТАНОВКА ЗАДАЧИ .....	6
3 МАТЕМАТИЧЕСКИЙ АППАРАТ ФОРМУЛ ДЛЯ РЕШЕНИЯ ПОСТАВЛЕННОЙ ЗАДАЧИ .....	7
4 ОРГАНИЗАЦИЯ ИНТЕРФЕЙСА .....	11
5 СТРУКТУРА ПРОГРАММЫ .....	11
6 АПРОБАЦИЯ ПРОГРАММЫ .....	11
7 РЕКОМЕНДАЦИИ К ВЫПОЛНЕНИЮ ЗАДАНИЯ .....	12
7.1 Работа с исходными данными .....	12
7.2 Создание главного меню .....	14
7.3 Подсоединение к проекту новых форм .....	17
7.4 Проверка вводимых данных .....	22
7.5 Рекомендации по решению задачи.....	23
7.6 Результаты вычислений.....	23
7.7 Работа с графикой.....	25
СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ .....	26
ПРИЛОЖЕНИЕ А- ОРГАНИЗАЦИЯ ИНТЕРФЕЙСА .....	27
ПРИЛОЖЕНИЕ Б- ВАРИАНТ ОФОРМЛЕНИЯ ТИТУЛЬНОГО ЛИСТА .....	28
ПРИЛОЖЕНИЕ В- ТЕКСТ ЗАДАНИЯ НА КУРСОВУЮ РАБОТУ .....	29
ПРИЛОЖЕНИЕ Г- СТРУКТУРА ПРОГРАММЫ .....	31
ПРИЛОЖЕНИЕ Д- ФРАГМЕНТ КОДА МОДУЛЯ С ПОЛЬЗОВАТЕЛЬСКИМИ ПРОЦЕДУРАМИ И ФУНКЦИЯМИ	32
ПРИЛОЖЕНИЕ Е- ФРАГМЕНТ КОДА ПРОГРАММЫ ДЛЯ ГРАФИКИ .....	35
ПРИЛОЖЕНИЕ Ж – ПОДГОТОВКА ТЕСТОВОГО ПРИМЕРА В MATHCAD .....	38

## ВВЕДЕНИЕ

В предлагаемом пособии порядок выполнения курсовой работы по дисциплине **«Информатика и основы программирование»** рассматривается на конкретном примере с методическими указаниями. Образцы машинных листингов для основных блоков программы даны в приложениях 1-3.

Методические рекомендации предназначены для подготовки по информатике студентов дневной формы обучения направления 7.0709 «Геодезия, картография и землеустройство», специальностей 7.070908 «Геоинформационные системы и технологии», 7.070901 «Инженерная геодезия», 7.070905 «Землеустройство и кадастр».

### *Порядок выполнения курсовой работы*

1. Внимательно прочитать и **уяснить** условие задания, которое предстоит выполнить.
2. Ознакомиться с необходимым теоретическим материалом - см. **Список рекомендуемой литературы**.
3. Разработать структуру программы и блок-схемы для решаемой задачи.
4. Подготовить тестовые примеры для апробации программы.
5. Подготовить **свой** вариант текста программы и попробовать решить его с помощью компьютера. Обычно с первого раза решение не удается - это нормально. Нужно повторить редактирование и счет до получения нужного результата.
6. Подготовить отчет.

Отчет оформляется в виде пояснительной записки и должен содержать следующие элементы:

- ◆ титульный лист;
- ◆ задание на курсовую работу;
- ◆ реферат;
- ◆ текст пояснительной записки в машинописном виде;
- ◆ список использованной литературы;
- ◆ машинный листинг программы на языке PASCAL - в виде приложения;
- ◆ реализация данной задачи в Delphi5 в виде исходного проекта на сервере в директории студента и на дискете.

- ◆ выполненные на пленке рисунки и чертежи, необходимые для презентации и защиты курсовой работы

### *Содержание пояснительной записки*

1. Задание.
2. Список возможных сокращений.
3. Реферат.
4. Введение
5. Постановка задачи. Математический аппарат формул для решения поставленной задачи.
6. Разработка интерфейса.
7. Краткие теоретические сведения об особенностях применяемых операторов и методов (теоретическое введение).
8. Описание программы:
  - ◆ общие сведения (язык программирования, операционная система, тип процессора);
  - ◆ описание логической структуры программы;
  - ◆ описание алгоритма решения задачи (в виде блок-схем);
  - ◆ описание входных и выходных данных программы;
  - ◆ описание подпрограмм;
  - ◆ перечень аномалий и допустимых значений входных данных
  - ◆ результаты тестирования программы.
9. Заключение и выводы
10. Список литературы
11. Приложения.
12. Дискета с текстом программы.

Перечисленные выше пункты пояснительной записки могут быть разделены на подпункты с учетом текста задания, приведенного ниже.

### **1 ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ**

Для своего варианта разработать алгоритм и реализовать решение задачи в среде Delphi5. При этом должны быть соблюдены следующие требования:

1. Ввод данных должен осуществляться как с клавиатуры, так и из файла.
2. Файл исходных данных может быть текстовым и типизированным (компонентным).

3. Результаты вычислений должны быть записаны как в текстовый так и в типизированный файл, а также выведены на экран.
4. В программе предусмотреть чтение результатов из типизированного и текстового файлов с последующим выводом на экран.
5. Интерфейс должен содержать пункты меню, связанные с вводом новых данных, чтением данных из файла, вычислениями, выводом результатов на экран из типизированного файла, краткую помощь работы с вашей программой, данные о программе и разработчике и др.
6. Предусмотреть графическое представление результатов работы.
7. Если необходимо, то предусмотреть обработку массивов исходных данных.
8. Предусмотреть защиту программы от некорректного ввода данных.

Выполнение всех перечисленных выше требований **ОБЯЗАТЕЛЬНО!!**

Порядок выполнения курсовой работы будет рассмотрен на примере составления программы для решения обратной геодезической засечки методом Ю.Рандярва. В методических указаниях приведены не все разделы пояснительной записки. Это следует учесть при оформлении работы.

## 2 ПОСТАНОВКА ЗАДАЧИ

Задание: написать программу для решения обратной геодезической засечки методом Ю.Рандярва. Необходимо:

- изучить математический аппарат формул для решения поставленной задачи;
- разработать структуру программы и блок-схемы для решаемой задачи;
- разработать программу для решение поставленной задачи на языке Object Pascal и реализовать ее в Delphi5.
- подготовить тестовые примеры и выполнить апробацию программы;
- подготовить отчет.

### 3 МАТЕМАТИЧЕСКИЙ АППАРАТ ФОРМУЛ ДЛЯ РЕШЕНИЯ ПОСТАВЛЕННОЙ ЗАДАЧИ\*

При создании опорных геодезических сетей геодезист сталкивается с решением обратной засечки почти на каждом шагу. На протяжении столетий ученые и инженеры предлагали различные формулы для ее решения. Наиболее часто используются формулы Кнейссля [1,2]\*\*. Но, к сожалению, эти формулы обладают тем недостатком, что контроль вычисления промежуточных величин является недостаточно жестким.

В задании на курсовую работу предлагается новый вариант решения обратной засечки, основанный на следующем принципе: как известно, положение искомого пункта М (рис.3.1) может быть определено, как место пересечения двух окружностей, проведенных через исходные пункты А, В и В, С.

Центры этих окружностей  $O_1$  и  $O_2$  лежат на диаметрах  $BA'$  и  $BC'$ , составляющих с твердыми направлениями  $BA$  и  $BC$  соответственно углы  $(\beta_1 - 90^\circ)$  и  $(90^\circ - \beta_2)$ . Линия  $A'C'$  всегда перпендикулярна линии  $BM$ .

Если координаты вспомогательных точек  $A'(X_1, Y_1)$  и  $C'(X_2, Y_2)$  известны, то нетрудно найти и координаты определяемого пункта М, решив совместно уравнения двух прямых  $A'C'$  и  $BM$ . Решение задачи будет более точным в том случае, когда измеренные углы  $\beta_i$  находятся в пределах  $60^\circ - 120^\circ$  (но не менее  $30^\circ$  и не более  $150^\circ$ ) и при расстоянии  $A'C' \geq BM$ .

#### Вывод формул.

Для вычисления вспомогательных точек  $A'(X_1, Y_1)$  и  $C'(X_2, Y_2)$  используем формулы прямой засечки, предложенные Юнгом. Рассматриваем треугольники  $ABA'$  и  $BC'C'$  (рис.1.1). В них известны все углы и одна сторона. Принимая во внимание тот факт, что один из углов при исходном пункте является прямым, получим для вычисления координат точек  $A'$  и  $C'$  формулы (1):

---

\* Приведенное описание математического аппарата формул является лишь примером для описания поставленных задач по вариантам курсовой работы.

\*\* 1. Зданович В.Г. и др. Высшая геодезия, М., 1970.

2. Маслов А.В. и др. Геодезия, М., 1972.

---

© Гермонова Е.А., Митрофанова Е.И.. Методические указания по информатике

$$\left. \begin{aligned} X_1 &= X_A + (Y_A - Y_B) * b_1 \\ Y_1 &= Y_A + (X_B - X_A) * b_1 \\ X_2 &= X_C + (Y_B - Y_C) * b_2 \\ Y_2 &= Y_C + (X_C - X_B) * b_2 \end{aligned} \right\} \quad (3.1)$$

где  $X_A, Y_A, X_B, Y_B, X_C, Y_C$  – исходные координаты точек А, В и С соответственно;

$b_1, b_2$  – котангенсы исходных углов  $\beta_1, \beta_2$ , которые можно определить по формулам:

$$b_1 = \text{ctg}\beta_1 \text{ и } b_2 = \text{ctg}\beta_2. \quad (3.2)$$

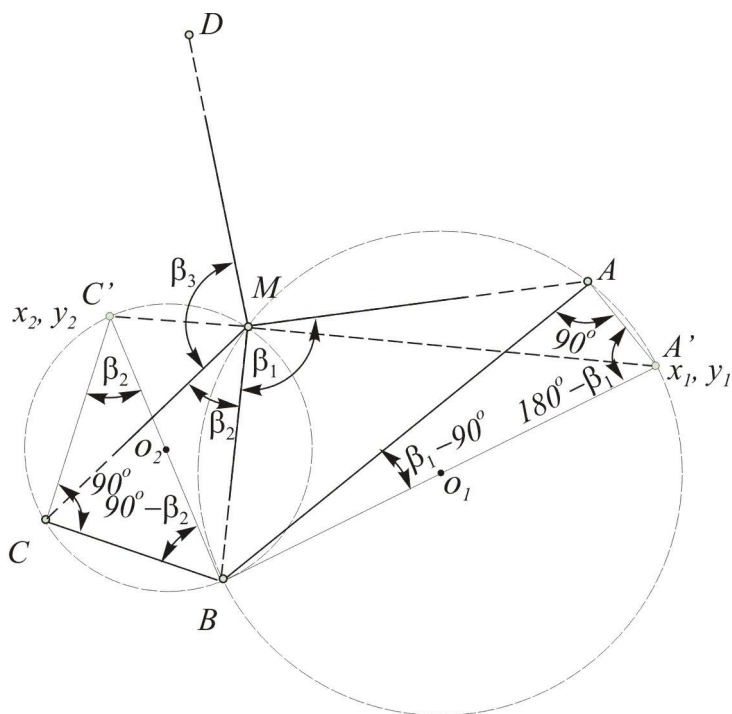


Рисунок 3.1 – Взаимное расположение исходных и определяемого пунктов



Для контроля правильности вычислений, произведенных по этим формулам, можно воспользоваться либо соотношениями:

$$\left. \begin{aligned} X_B - X_A &= \frac{b_1 * (Y_1 - Y_B) - (X_1 - X_B)}{1 + b_1^2} = \frac{D_X}{1 + b_1^2} \\ X_C - X_B &= \frac{b_2 * (Y_2 - Y_B) - (X_2 - X_B)}{1 + b_2^2} = \frac{D'_X}{1 + b_2^2} \end{aligned} \right\} \quad (3.3)$$

либо соотношениями:

$$\left. \begin{aligned} Y_B - Y_A &= \frac{b_1 * (X_1 - X_B) - (Y_1 - Y_B)}{1 + b_1^2} = \frac{D_Y}{1 + b_1^2} \\ Y_B - Y_C &= \frac{b_2 * (X_2 - X_B) - (Y_2 - Y_B)}{1 + b_2^2} = \frac{D_Y}{1 + b_2^2} \end{aligned} \right\} \quad (3.4)$$

Формулы (3.2) и (3.3) получены непосредственно из формул Юнга [2], в которых определяемыми пунктами будут а и С, а исходными – пункты В, А' и В, С'.

Поскольку направление ВМ перпендикулярно направлению А'С', для вычисления дирекционного угла (ВМ) получены формулы:

$$\operatorname{tg}(ВМ) = \frac{X_2 - X_1}{Y_1 - Y_2} = a, \quad (3.5)$$

$$\operatorname{ctg}(ВМ) = \frac{Y_1 - Y_2}{X_2 - X_1} = c. \quad (3.6)$$

Решив совместно уравнения этих взаимно перпендикулярных прямых ВМ и А'С', будут получены формулы для вычисления приращений координат от пункта В до определяемого пункта М, а затем можно получить формулы для вычисления искоемых координат.

Возможны два варианта формул:

а) если дирекционный угол (ВМ) близкий по значению 0° или 180°, координаты определяемой точки М можно вычислить по формулам:

$$\left. \begin{aligned} X_M &= X_B + \frac{S_1}{1+a^2} = X_B + \frac{S_2}{1+a^2} \\ Y_M &= Y_B + a \cdot (X_M - X_B) \end{aligned} \right\}, \quad (3.7)$$

где величины S<sub>1</sub> и S<sub>2</sub> можно вычислить по формулам:

$$\left. \begin{aligned} S_1 &= a \cdot (Y_1 - Y_B) + (X_1 - X_B) \\ S_2 &= a \cdot (Y_2 - Y_B) + (X_2 - X_B) \end{aligned} \right\} \quad (3.8)$$

б) если дирекционный угол (ВМ) близкий по значению  $90^\circ$  или  $270^\circ$ , координаты определяемой точки М можно вычислить по формулам:

$$\left. \begin{aligned} Y_M &= Y_B + \frac{S_3}{1+c^2} = X_B + \frac{S_4}{1+c^2} \\ X_M &= X_B + c \cdot (Y_M - Y_B) \end{aligned} \right\}, \quad (3.9)$$

где величины  $S_1$  и  $S_2$  можно вычислить по формулам:

$$\left. \begin{aligned} S_3 &= c \cdot (X_1 - X_B) + (Y_1 - Y_B) \\ S_4 &= c \cdot (X_2 - X_B) + (Y_2 - Y_B) \end{aligned} \right\}. \quad (3.10)$$

После вычисления координат необходимо выполнить заключительный контроль, который заключается в вычислении угла  $\beta_3^{\text{выч}}$  и его сравнения с измеренной величиной.

Угол  $\beta_3^{\text{выч}}$  можно вычислить по формуле:

$$\beta_3^{\text{выч}} = \alpha_{\text{MD}} - \alpha_{\text{MC}}, \quad (3.11)$$

где  $\alpha_{\text{MD}}$  и  $\alpha_{\text{MC}}$  - дирекционные углы направлений MD и MC соответственно, которые можно вычислить по формулам:

$$\left. \begin{aligned} \alpha_{\text{MD}} &= \arctg \frac{Y_D - Y_M}{X_D - X_M}, \\ \alpha_{\text{MC}} &= \arctg \frac{Y_C - Y_M}{X_C - X_M}. \end{aligned} \right\} \quad (3.12)$$

Контроль считается выполненным, если выполниться условие:

$$\left| \beta_3^{\text{выч}} - \beta_3^{\text{изм}} \right| \leq 3 \cdot m_\beta, \quad (3.13)$$

где  $\beta_3^{\text{изм}}$  - измеренный угол;

$\beta_3^{\text{выч}}$  - вычисленный угол;

$m_{\beta}$  - средняя квадратическая ошибка измерения угла (в данном примере  $m_{\beta} = 5''$ )

Рассмотренный выше алгоритм решения полностью реализован в MathCad. Результаты вычислений могут быть использованы в качестве тестового примера при апробации программного продукта (см. приложение Е).

При написании программы следует учесть, что все промежуточные этапы вычислений следует сохранить в текстовом файле.

#### **4 ОРГАНИЗАЦИЯ ИНТЕРФЕЙСА**

Интерфейс необходимо организовать исходя из многооконного режима работы в Windows и учитывая поставленную задачу в задании. При выполнении работы предлагается для каждого пункта меню подготовить соответствующую форму. Например, для ввода данных и их сохранения в типизированный и текстовый файлы необходимо разработать форму «Исходные данные» в соответствии со структурой данных, необходимых для решения поставленной задачи.

Для предлагаемой в качестве примера задачи, схема интерфейса представлена в Приложении А (см. рис. А.1). В тексте пояснительной записки следует также дать словесное описание этой схемы.

#### **5 СТРУКТУРА ПРОГРАММЫ**

В данном разделе необходимо привести краткое описание всех форм и модулей, подсоединенных в проекте. Кроме текстового описания можно привести схему структуры программы (см. приложение Г, рис. Г.1).

В тексте пояснительной записки следует описать все применяемые компоненты и измененные Вами их свойства.

#### **6 АПРОБАЦИЯ ПРОГРАММЫ**

В этом разделе следует привести несколько примеров решения задачи другими способами ( на микрокалькуляторе, в EXEL или Math-

Cad и т.п.) и решение этих же примеров, реализованное Вашей программой (пример см. приложение Ж).

Рекомендации: для апробации необходимо применить различные исходные данные, даже такие, которые могут привести к ошибкам. Это позволит Вам устранить все неопределенности.

## 7 РЕКОМЕНДАЦИИ К ВЫПОЛНЕНИЮ ЗАДАНИЯ

### 7.1 Работа с исходными данными

Перед написанием программы следует проанализировать все исходные данные и определиться с их типом и структурой. Например, для рассматриваемой в методических указаниях задачи в качестве исходных данных будут координаты двух точек и 4-ре измеренных угла. Координаты точек - величины вещественные типа *Double*. Углы будут представлены целочисленными (тип *Integer*) переменными: градусами, минутами и секундами. Тогда структура текстового файла с исходными данными будет следующая:

1 строка- координаты точки A) XA<пробел>YA  
2 строка- координаты точки B) XB<пробел>YB  
3 строка- координаты точки C) XC<пробел>YC  
4 строка- координаты точки D) XD<пробел>YD  
5 строка- угол  $\beta_1$  ) градусы<пробел>минуты<пробел>секунды  
6 строка- угол  $\beta_2$  ) градусы<пробел>минуты<пробел>секунды  
7 строка- угол  $\beta_3$  ) градусы<пробел>минуты<пробел>секунды

Безтиповый файл будет содержать семнадцать переменных (восемь переменных типа *Double* и девять переменных типа *Integer*).

Рекомендуется файлам с исходными данными имена присваивать в диалоговом режиме, а не задавать в виде констант. Для этого, на форму, содержащую главное пользовательское меню управления Вашей программой пометите компоненту *SaveDialog* из палитры компонент *Dialogs*. Затем измените следующие свойства этого объекта: *DefaultExt*, *Filter*, *FilterIndex*, *Name*, *Options*, *Title*. Свойство *DefaultExt* позволяет установить расширение файла, которое по умолчанию присоединено к имени создаваемого файла. Так как Вы будете работать с текстовыми или типизированными файлами, то в зависимости от этого установите

эту величину либо *\*.txt*, либо *\*.tip*. Вы можете создать фильтр для работы с несколькими типами данных. Для этого дважды щелкните левой кнопкой мыши в поле свойства *Filter*, в результате откроется диалоговое окно для создания фильтра (рис.7. 2).

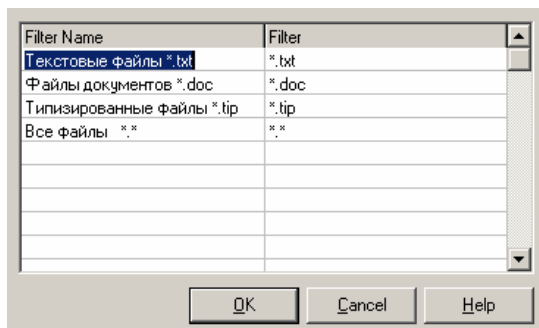


Рис.7. 2. Диалоговое окно создания фильтров

Следует заполнить поля: *Filter Name* – текстовой информацией по задаваемому фильтру и *Filter* шаблоном фильтра (см пример на рис.2).

Для того, чтобы на момент работы в диалоговом окне создания файла все существующие файлы в рабочей директории были отфильтрованы по одному шаблону, следует задать свойству *FilterIndex* номер этого шаблона в списке. Так, например, для списка, представленного на рис. 7.2 *FilterIndex:=1*, если вы хотите по умолчанию установить фильтр *\*.txt* (см. рис.7.3).

Изменив свойства *Options* Вы можете видоизменить диалоговое окно, которое применяется для создания файлов.

Если *Title:= «Сохранить как»*, то это же название появиться в диалоговом окне.

Если Вы помещаете компоненту *SaveDialog* на форму, то по умолчанию *Name=SaveDialog1*. Если на форму компонента *SaveDialog* была помещена несколько раз, то имена будут изменяться: *SaveDialog2*, *SaveDialog3...*

Свойства объектов можно изменять как непосредственно в окне Инспектора объектов, так и в тексте программы путем присваивания им значений.

Так, например, для того, чтобы при открытии *SaveDialog* в диалоговом окне сразу была подсоединена папка с данными, то в программе следует написать следующие операторы:

- 1) в событии *onCreate* главной формы поместить оператор ***directories:=ExtractFilePath(Application.EXEName);***

при этом переменную *directories* необходимо объявить глобально;  
2) в процедуру, которая будет обрабатывать *SaveDialog* диалог необходимо поместить оператор:

***OpenDialog\_data.InitialDir:=directories+'file\_data';***

при условии, что уже была создана ранее папка *file\_data* для исходных данных и результатов работы .

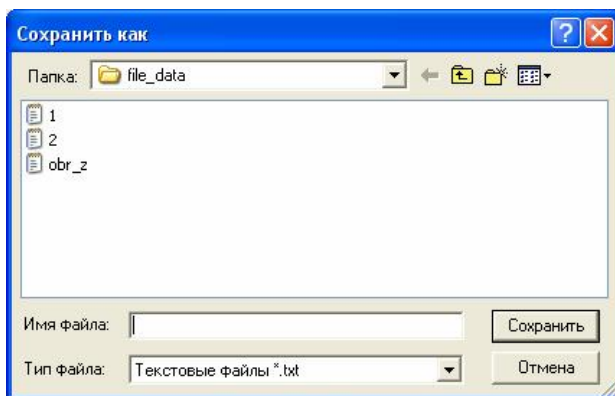


Рис. 7. 3. Диалоговое окно создания и записи файла

## 7.2 Создание главного меню

Перед написанием программы Вы должны были продумать интерфейсную часть программы. Если посмотреть рис. А.1, то можно заметить что главная форма содержит главное меню управления программой. Структура этого меню предлагается для всех по одному шаблону (см. рис.Г.2). Однако, если Вы хотите его расширить, это не запрещается. Кроме того, в задании предлагается продублировать все пункты меню на палитре инструментария управления программой.

Для создания главного меню выберите компоненту MainMenu из палитры компонент Standard и поместите ее на главную форму. Затем дважды щелкните левой кнопкой мыши на появившемся на форме значке. В результате откроется диалоговое окно создания меню (см. рис.7. 4).



Рис. 7.4. Диалоговое окно для создания меню

В окне Инспектора объектов (рис.7.5) появятся свойства и их значения по умолчанию для первого пункта меню. Переместите курсор в поле свойства *Caption* и присвойте ему значение *Файл*. Затем измените свойство *Name*=“*N\_file*” для удобства дальнейшего программирования. Остальные свойства можно не изменять.

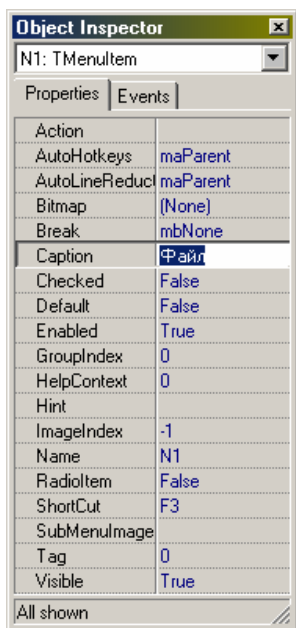


Рис.7.5. Окно Инспектора объектов

Если Вам необходимо какой-то пункт меню сделать временно не видимым, то необходимо изменить свойство *Visible=false*.

Если какой-нибудь пункт меню необходимо отключить без отключения видимости, то свойству *Enabled* присваивается значение *false*.

Действие каждого пункта меню можно продублировать нажатием функциональных и управляющих кнопок клавиатуры, для чего следует изменить свойство *Shortcut*: откройте список всех возможных кнопок и их сочетаний и выберите то, что Вам необходимо.

Если Вы хотите отделить один пункт меню от другого по вертикали, укажите в *Caption* символ “-”.

После окончания формирования главного меню закройте диалоговое окно по его созданию. На форме появится созданное Вами меню.

Затем, для дублирования функций меню поместите компоненту *ToolBar* из палитры компонент *Win32*. Кроме того, выберите компоненту *ImageList* из той же самой палитры и также пометите на форму. Затем с помощью объекта *TImageList* создайте список значков (см. рис.7.6) *ImageList1*, которые Вы хотите поместить на кнопках панели инструментов, дублирующей главное меню. Для этого сначала создайте такие значки, затем добавьте их в список (создание списка открывается при двойном нажатии левой кнопки мыши на значке объекта *ImageList*, помещенного на форму).

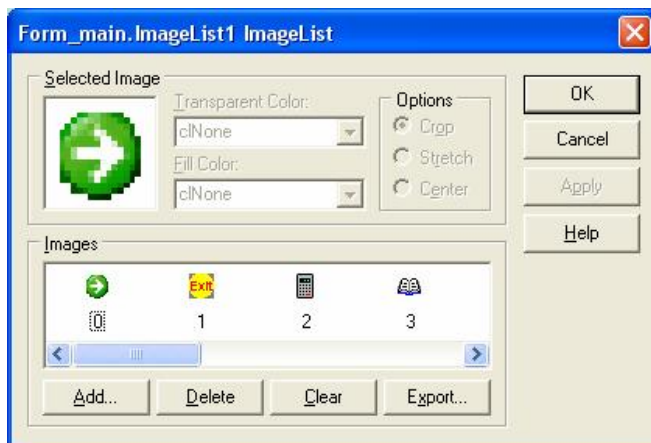


Рис. 7.6 Диалоговое окно создания списка значков

После создания списка значков следует разместить кнопки на панели *ToolBar1*. Для этого поместите курсор на эту панель, находящуюся на форме, и нажмите правую кнопку мыши. В появившемся всплывающем меню выберите пункт *New Button*. Эту операцию необходимо повторить столько раз, сколько кнопок Вы хотели бы поместить на панели.

Для помещения изображений значков из списка *ImageList1* на кнопки панели *ToolBar1* (см. рис.7.7) необходимо поместить в свойство *Images* этой панели имя *ImageList1*. Затем свойству *ImageIndex* каждой кнопки палитры инструментов присвоить номер значка в списке *ImageList1*.

После создания меню и панели инструментов необходимо обработать события на все пункты меню.

Так, например, для того чтобы выполнить команду *Выход* пункта меню *Файл* необходимо обработать событие *onClose* для исходной формы. Для этого откроем страницу *Events* (события) инспектора объектов и дважды щелкнем левой кнопкой мыши в поле события *onClose*. В открывшийся код программы впишем следующие операторы:

```
procedure TForm_main.FormClose(Sender: TObject;
var Action: TCloseAction);
begin
Action:=caFree;//оператор, дописанный программистом
end;
```



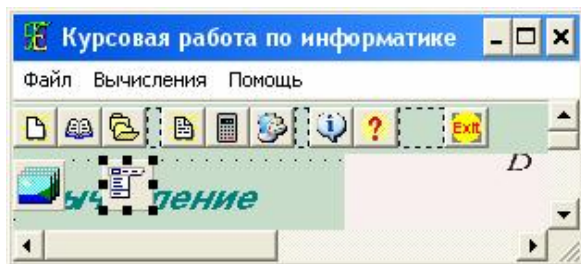


Рис. 7 Фрагмент главной формы после создания главного меню и палитры инструментов, дублирующих его

Затем обрабатываем событие *onClick* для пункта меню *Выход*. Для этого с помощью мыши можно подвести курсор на этот пункт меню и дважды щелкнуть левой кнопкой мыши. Затем следует изменить открывшийся код:

```
procedure TForm_main.N_ExitClick(Sender: TObject);  
begin  
close;           //оператор, дописанный программистом  
end;
```

Для дублирующей команду меню кнопки панели инструментов необходимо также обработать событие нажатия кнопки. Для этого можно продублировать процедуры, либо просто обратиться к уже созданной процедуре.

### 7.3 Подсоединение к проекту новых форм

Не следует в проекте создавать сразу все формы. При подсоединении к проекту новой формы в тексте файла проекта с расширением \*.dpr появиться строка

```
Application.CreateForm(T<имя формы>, <имя формы>).
```

Скопируйте эту строку в буфер обмена, прокомментируйте ее и затем вставьте в процедуру модуля, в котором эта форма будет применен. При этом не забудьте подсоединить модуль этой формы к модулю, где она будет применяться.

Для примера, предложенного в методических указаниях листинг проекта будет иметь вид:

```
program Project_germ;  
uses  
Forms,
```

```
Unit_main in 'Unit_main.pas' {Form_main},
Unit_about in 'Unit_about.pas' {AboutBox},
Unit_new_data in 'Unit_new_data.pas' {Form_new_data},
Unit_global in 'Unit_global.pas',
Unit_view in 'Unit_view.pas' {Form_view},
Unit_Titul in 'Unit_Titul.pas' {FormTitul1},
Unit_result in 'Unit_result.pas' {Form_result},
Unit_view_result in 'Unit_view_result.pas' {Form_view_result};
{$R *.RES}
begin
  Application.Initialize;
  Application.CreateForm(TFormTitul1, FormTitul1);
  FormTitul1.ShowModal;
  FormTitul1.Free;
  Application.CreateForm(TForm_main, Form_main);
  Application.Run;
end.
```

В программе следует предусмотреть создание таких форм: форма сведений о программе, форма помощи, форма создания файла исходных данных, форма просмотра файла исходных данных (старайтесь по возможности эти две формы объединить в одну), форма вычислений, форма графики.

Так, для предлагаемого варианта задачи в методических указаниях была создана форма исходных данных, приведенная на рис. 7.8.

На эту Форму были помещены компоненты из палитр компонент **Standard**: *TEdit, TLabel*; **Additional**: *TbitBtn*; **Dialogs**: *TsaveDialog, TOpenDialog*. Эта форма будет применена для выполнения двух функций меню: *Создать новые* и *Прочитать из файла*. Для этого была введена глобальная переменная *kod\_open\_data*: если *kod\_open\_data=0* то выполняется функция создания новых данных, если *kod\_open\_data=1* – функция чтения данных из файла (компонентного или текстового).

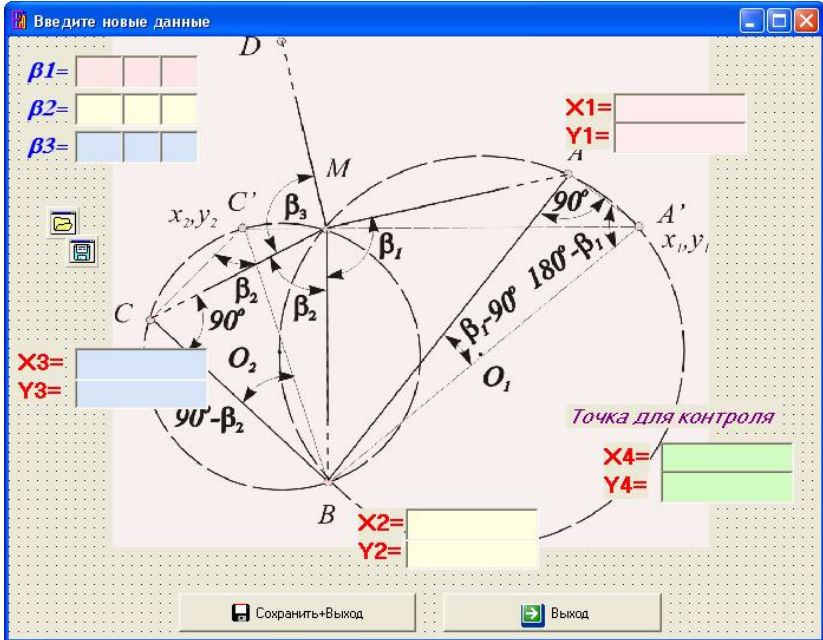


Рис. 7.8 Форма исходных данных *Form\_new\_data*

```

Код программы события onCreate формы данных имеет вид:
procedure TForm_new_data.FormCreate(Sender: TObject);
var
  file_name_txt, file_name_tip:shortstring ;
  f_txt           :textfile; f_tip           :file;
  x1,x2,x3,y1,y2,y3,x4,y4 :double;
  ug1_gr,ug1_min,ug1_sec,ug2_gr,ug2_min,ug2_sec,
  ug3_gr,ug3_min,ug3_sec :integer;
  kod             :byte;
begin
  kod:=0;
  if kod_open_data <>0 then begin
    OpenFileDialog_data.InitialDir:=directories+'file_data\';
    if OpenFileDialog_data.Execute then begin
      file_name:=OpenFileDialog_data.FileName;
      kod:=0;
      if (ExtractFileExt( File_Name)='.tip') or
        (ExtractFileExt( File_Name)='.TIP')or
  
```

```
(ExtractFileExt( File_Name)='.Tip') then
begin
kod:=1;
file_name_tip:=file_name;
assignFile(f_tip,file_name_tip); Reset(f_tip,1);
blockread(f_tip,x1,8);          lockread(f_tip,y1,8);
blockread(f_tip,x2,8);          ockread(f_tip,y2,8);
blockread(f_tip,x3,8);          ockread(f_tip,y3,8);
blockread(f_tip,x4,8);          ockread(f_tip,y4,8);
blockread(f_tip,ug1_gr,4);      blockread(f_tip,ug1_min,4);
blockread(f_tip,ug1_sec,4);    blockread(f_tip,ug2_gr,4);
blockread(f_tip,ug2_min,4);    blockread(f_tip,ug2_sec,4);
blockread(f_tip,ug3_gr,4);    blockread(f_tip,ug3_min,4);
blockread(f_tip,ug3_sec,4);    closefile(f_tip);
end else
if (ExtractFileExt( File_Name)='.txt') or
(ExtractFileExt( File_Name)='.TXT') or
(ExtractFileExt( File_Name)='.Txt') then
begin
kod:=2;
file_name_txt:=file_name;
assignFile(f_txt,file_name_txt); Reset(f_txt);
readln(f_txt,x1,y1);   readln(f_txt,x2,y2);
readln(f_txt,x3,y3);   readln(f_txt,x4,y4);
readln(f_txt,ug1_gr,ug1_min,ug1_sec);
readln(f_txt,ug2_gr,ug2_min,ug2_sec);
readln(f_txt,ug3_gr,ug3_min,ug3_sec);
closeFile(f_txt);
end          end;
if kod=0 then begin
showmessage('Исходные данные не были прочитаны!');
close;
end;
end;
edit_x1.Text:=FloatToStrF(x1,ffFixed,14,3);
edit_y1.Text:=FloatToStrF(y1,ffFixed,14,3);
edit_x2.Text:=FloatToStrF(x2,ffFixed,14,3);
edit_y2.Text:=FloatToStrF(y2,ffFixed,14,3);
edit_x3.Text:=FloatToStrF(x3,ffFixed,14,3);
edit_y3.Text:=FloatToStrF(y3,ffFixed,14,3);
edit_x4.Text:=FloatToStrF(x4,ffFixed,14,3);
```

```
edit_y4.Text:=FloatToStrF(y4,ffFixed,14,3);  
edit_ug1_deg.Text:=inttostr(ug1_gr);  
edit_ug1_min.Text:=inttostr(ug1_min);  
edit_ug1_sec.Text:=inttostr(ug1_sec);  
edit_ug2_deg.Text:=inttostr(ug2_gr);  
edit_ug2_min.Text:=inttostr(ug2_min);  
edit_ug2_sec.Text:=inttostr(ug2_sec);  
edit_ug3_deg.Text:=inttostr(ug3_gr);  
edit_ug3_min.Text:=inttostr(ug3_min);  
edit_ug3_sec.Text:=inttostr(ug3_sec);  
showmodal;  
end;
```

Для подсоединения к проекту формы сведений о программе можно использовать шаблон. Для этого выберите из пункта меню *File* в Delphi5 команду *New* и в открывшемся списке шаблонов на странице *Forms* выберите шаблон *Aboutbox*. К проекту автоматически добавиться форма, показанная на рис. 7.9.

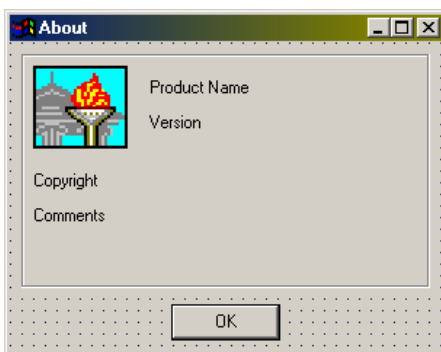


Рис. 7.9 Шаблон формы *About*

Затем удалите из кода проекта строку *Application.CreateForm(TAboutBox1, AboutBox1)*, предварительно скопировав ее в буфер обмена. Затем в код события *onClick* для пункта меню *О программе* строку:

```
Application.CreateForm(TAboutBox, AboutBox);
```

Форму в дальнейшем можно видоизменить меняя свойства объектов, помещенных на нее.

### 7.4 Проверка вводимых данных

На рис.7.8 приведена форма для ввода исходных данных, где данные будут вводиться в окна компонент *TEdit*. Известно, что данная компонента работает только со строковыми величинами, а для решения задачи потребуется, чтобы вводимые переменные имели числовой тип (*Integer, Real, Double...*). Следовательно в программе будут присутствовать операторы перевода строковой величины в числовую. А если вдруг окажется, что при вводе пользователь программы ошибется, и вместо число введет буквенное значение? Что тогда? Как осуществить контроль вводимых величин? Существуют различные способы выполнения такого контроля. Здесь будет рекомендован один из них.

Для того, чтобы проконтролировать вводимую величину, можно в событии *OnChange* компоненты *TEdit* написать следующий фрагмент программы:

```
procedure TForm_new_data.Edit_x1Change(Sender: TObject);  
var  
kontrol :string;  
x1 :real;  
kod :integer;  
begin  
kontrol:= Edit_x1.Text;  
if kontrol<>" then begin  
val(kontrol,x1,kod);  
if kod<>0 then begin  
showmessage('Ошибка при вводе X1!'+  
' Повторить ввод значения!');  
delete(kontrol,length(kontrol),1);  
edit_x1.Text:=kontrol;  
edit_x1.SelLength:=length(kontrol);  
end; end; end;  
//жирным шрифтом выделены строки программы, которые  
//разработчик программы пишет самостоятельно
```

Тогда, например, при вводе в окно *Edit\_X1* координаты точки *A* будет осуществляться контроль и если случайно пользователь вместо цифры нажмет кнопку на клавиатуре с какой-нибудь буквой или недопустимым символом, при выполнении события *OnChange* эта ошибка будет устранена.

### **7.5 Рекомендации по решению задачи**

При написании программы следует очень внимательно проанализировать весь аппарат формул для решения поставленной задачи с целью разбиения вычислительного процесса на ряд процедур и функций.

Рекомендация: все процедуры и функции, связанные с вычислительным процессом разместить в отдельном модуле, например, в том, где будут описываться глобальные переменные и пользовательские типы данных.

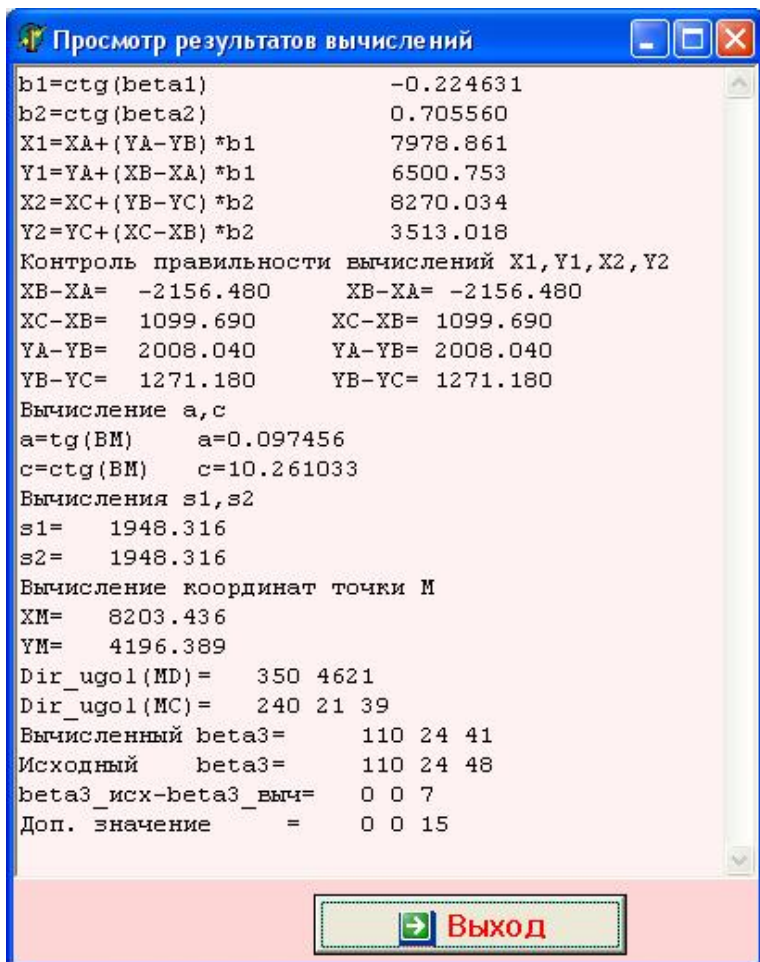
Для решения задачи, приведенной в качестве примера, весь вычислительный процесс был поделен на такие подпрограммы:

- преобразование угла из градусной меры в радианную;
- преобразование угла из радианной меры в градусную;
- вычисление дирекционного угла;
- вычисления координат вспомогательных точек по формулам (3.1);
- контроль по формулам (3.2);
- вычисление котангенса угла в градусной мере;
- вычисление величин  $a$  и  $c$  по формулам (3.5-3.6);
- вычисление координат определяемой точки по формулам (3.7) и (3.9);
- вычисление вспомогательных длин по формулам (3.8) и (3.10).

В приложении Д приведен листинг модуля, в котором были реализованные все перечисленные выше подпрограммы.

### **7.6 Результаты вычислений**

В программе необходимо предусмотреть вывод в файл и на экран всех промежуточных этапов вычисления, а не только показать окончательные результаты. Так, например, для рассматриваемой задачи результаты могут быть выведены в следующем виде:



```

Просмотр результатов вычислений
b1=ctg(beta1)          -0.224631
b2=ctg(beta2)          0.705560
X1=XA+(YA-YB)*b1       7978.861
Y1=YA+(XB-XA)*b1       6500.753
X2=XC+(YB-YC)*b2       8270.034
Y2=YC+(XC-XB)*b2       3513.018
Контроль правильности вычислений X1, Y1, X2, Y2
XB-XA= -2156.480      XB-XA= -2156.480
XC-XB= 1099.690      XC-XB= 1099.690
YA-YB= 2008.040      YA-YB= 2008.040
YB-YC= 1271.180      YB-YC= 1271.180
Вычисление a, c
a=tg(BM)      a=0.097456
c=ctg(BM)     c=10.261033
Вычисления s1, s2
s1= 1948.316
s2= 1948.316
Вычисление координат точки M
XM= 8203.436
YM= 4196.389
Dir_ugol(MD)= 350 4621
Dir_ugol(MC)= 240 21 39
Вычисленный beta3= 110 24 41
Исходный beta3= 110 24 48
beta3_исх-beta3_выч= 0 0 7
Доп. значение = 0 0 15

[Выход]

```

Рис.7.10

В каком виде должны быть представлены результаты вычислений, решает сам разработчик программы. Поэтому не обязательно, чтобы форма с результатами выглядела именно так, как показано на рис.7.10. Единственное требование, чтобы можно было легко просмотреть весь вычислительный процесс.



## 7.7 Работа с графикой

В задании курсовой работы предусмотрена работа с графикой. Так, например, в предложенной задаче необходимо изобразить две исходных точки, две вставляемой точки, соединить их линиями и подписать вершины и их координаты. При просмотре этого чертежа на экране монитора необходимо предусмотреть изменение масштаба, например, уменьшение или увеличение.

Для просмотра графики также необходимо подготовить форму, например, вида, показанного на рис.7.1.

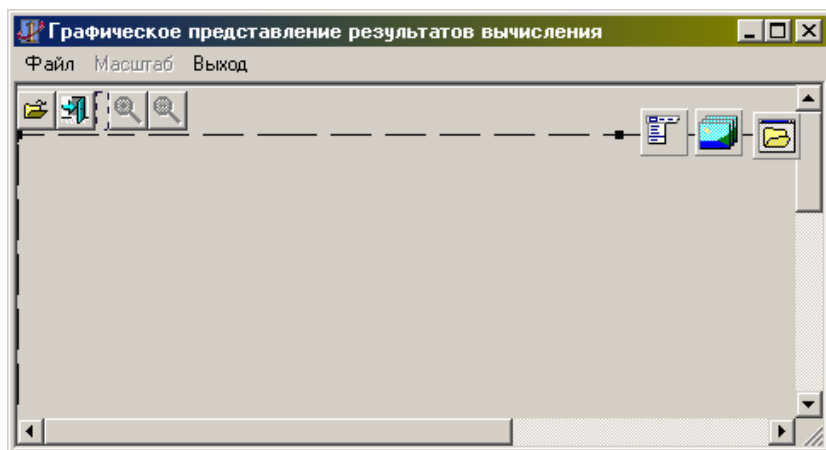


Рис.11 Форма для графического представления результатов

Вы можете увидеть, что на этой форме также присутствует главное меню и панель инструментов, дублирующих его команды. На форму был помещен объект *Image1* типа *tImage*.

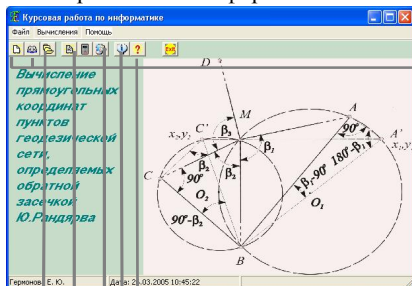
В модуле к этой форме в разделе инициализации переменных был создан объект *kImage* типа *tBitmap*. Фрагмент кода программы на событие *onClick* для пункта меню *Файл* приведен в приложении Е.

## СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

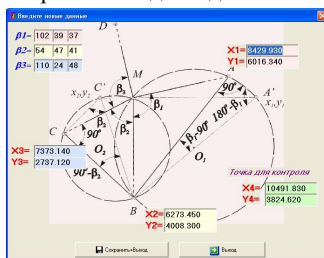
1. Фаронов В.В. Delphi 5. Учебный курс. - М.: Нолидж., 2000, 614 с.
2. Фаронов В.В., Шумаков П.В. Delphi 5. Руководство разработчика баз данных. - М.: Нолидж., 2000, 636 с.
3. Методические указания к лабораторным работам по курсу «Информатика и основы программирования», раздел «основы программирования на языке Turbo Pascal 7.0» в двух частях, часть 1 (для студентов первого курса специальностей: «инженерная геодезия», «геоинформационные системы и технологии», «землеустройство и кадастр»)/ Сост. Гермонова Е.А.- Донецк: ДонНТУ, 2001.-68 с.
4. Методические указания к курсовой работе по курсу «Информатика и основы программирования», для студентов первого курса специальностей: «инженерная геодезия», «геоинформационные системы и технологии», «землеустройство и кадастр»/ Сост. Гермонова Е.А.- Донецк: ДонНТУ, 2002.-68 с.
5. ГОСТ 19.402-78\* (СТ СЭВ 2092-80) Единая система программной документации

## ПРИЛОЖЕНИЕ А- ОРГАНИЗАЦИЯ ИНТЕРФЕЙСА

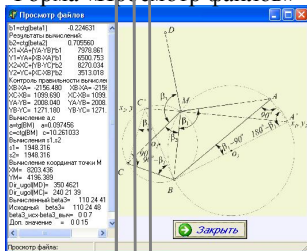
Форма «Главная форма»



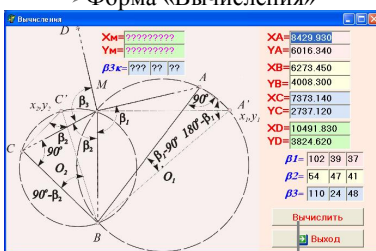
Форма «Исходные данные»



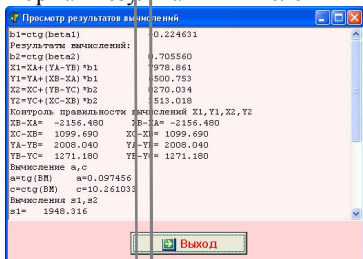
Форма «Просмотр файлов»



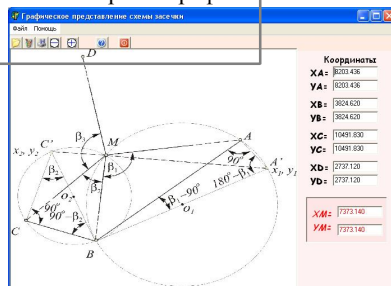
Форма «Вычисления»



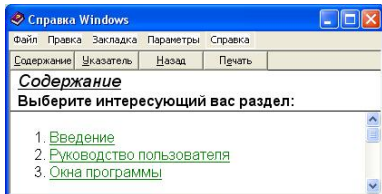
Форма «Результаты вычислений»



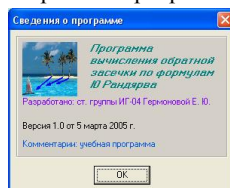
Форма «Графика»



Форма «Помощь»



Форма «О программе»



**ПРИЛОЖЕНИЕ Б- ВАРИАНТ ОФОРМЛЕНИЯ ТИТУЛЬНОГО  
ЛИСТА**

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ УКРАИНЫ  
ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

Кафедра «Геоинформатики  
и геодезии»

**КУРСОВАЯ РАБОТА**

**по курсу «Информатика и основы программирования»  
тема: «Применения Delphi5 для создания программного продукта,  
включающего решение обратной геодезической засечки по формулам  
Ю. Рандярва , работу с текстовыми и типизированными файлами,  
работу в графическом режиме»**

**Выполнил студент  
гр. ИГ-04**

\_\_\_\_\_

(подпись, дата)

\_\_\_\_\_

(ФИО)

**Проверил**

\_\_\_\_\_

(подпись, дата)

\_\_\_\_\_

(ФИО)

**г. Донецк – 2005 г.**

## ПРИЛОЖЕНИЕ В- ТЕКСТ ЗАДАНИЯ НА КУРСОВУЮ РАБОТУ

### МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Факультет Горно-геологический факультет  
Кафедра геоинформатики и геодезии  
Специальность: 7.070901 «Геодезия»

Утверждаю:  
Зав. кафедрой

«  »    2005 г.

#### ЗАДАНИЕ на курсовую работу

студента: Иванова Ивана Ивановича  
(фамилия, имя отчество)

1. Тема курсовой работы: «Применения Delphi5 для создания программного продукта, включающего решение обратной геодезической засечки по формулам Ю.Рандярова, работу с текстовыми и типизированными файлами, работу в графическом режиме»

2. Срок сдачи студентом законченной работы: 5.05.2005 г.

3. Исходные данные к работе: 1. Формулы Ю.Рандярова  
2. Инструктивно-нормативные документы по оформлению программного продукта;  
3. Методические указания к выполнению курсовой работы и лабораторных работ.

4. Содержание расчетно-пояснительной записки (перечень вопросов, которые необходимо рассмотреть): 1. Введение; 2. Алгоритм решение геодезической задачи; 3. Организация интерфейса создаваемой программы; 4. Структура программы, описание модулей; 5. Результаты апробации программного продукта; 6. Заключение; 7. Приложения.

5. Перечень графических материалов (с точным определением обязательных чертежей) 1. Схема организации интерфейса; 2. Структура программы; 3. Блок-схемы отдельных процедур и функций; 4. Результаты апробации программного продукта.

6. Дата выдачи задания: 18.02.2005 г.

© Гермонова Е.А., Митрофанова Е.И.. Методические указания по информатике

## КАЛЕНДАРНЫЙ ПЛАН

№ пп	Название этапов курсовой работы	Срок выполнения этапов работы	Примечание
1	<i>Изучение математических формул для решения поставленной задачи. Разработка интерфейса. Определение входных и выходных данных и их типов.</i>	<i>4 марта 2005 г.</i>	
2	<i>Создание программы в среде Delphi5. Создание: главной формы, формы графики, формы сведений о программе, формы помощи, заставки, чтения и корректирования исходных данных.</i>	<i>18 марта 2005 г.</i>	
3	<i>Обработка событий создания формы, закрытие форм, нажатие на помещенные на форму кнопки. Подсоединение процедур и функций, необходимых для решения поставленной геодезической задачи.</i>	<i>1 апреля 2005 г.</i>	
4	<i>Работа с текстовыми и типизированными файлами. Запись результатов и всех промежуточных значений при вычислении в текстовый файл.</i>	<i>15 апреля</i>	
5	<i>Подготовка в процессе вычислений файла данных для графического представления задачи. Аprobация разработанного программного продукта. Устранение всех недостатков и выявленных в процессе апробации ошибок -</i>	<i>29 апреля 2005 г.</i>	
5	<i>Оформление пояснительной записки. Подготовка графических приложений для представления работы на защите. Сдача работы на проверку.</i>	<i>15 мая 2005 г.</i>	
6	<i>Устранение ошибок, выявленных преподавателем. Защита курсовой работы</i>	<i>22 мая 2005 г.</i>	

Студент \_\_\_\_\_  
(подпись)

Руководитель \_\_\_\_\_  
(подпись)

*Е.А.Гермонова*  
(Фамилия, Имя, Отчество)

“ 18 ” февраля 2005 г.

## ПРИЛОЖЕНИЕ Г- СТРУКТУРА ПРОГРАММЫ

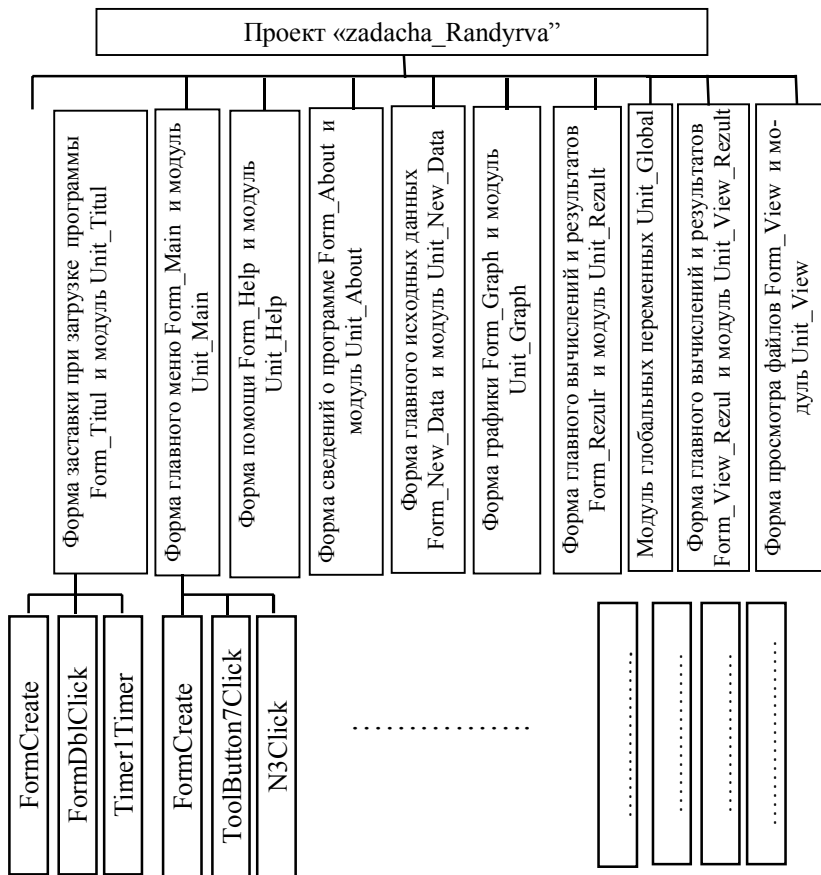
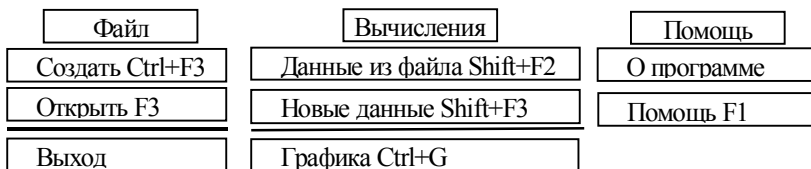


Рис.Г.1 Фрагмент схематичного представления структуры программы



ис.Г.2 Структура главного меню

## ПРИЛОЖЕНИЕ Д- ФРАГМЕНТ КОДА МОДУЛЯ С ПОЛЬЗОВАТЕЛЬСКИМИ ПРОЦЕДУРАМИ И ФУНКЦИЯМИ

```
unit Unit_global;
interface
uses Math;
var
file_name, directories:shortstring;
kod_open_data,kod_open_result:byte;
stroki:array[1..30] of shortstring;
function DIR_UG(XA,YA,XB,YB:double):real;
function deg_rad(deg,min,sec:integer):real;
Procedure rad_deg(UG:real;var deg,min,sec:integer);
Function ctg(deg,min,sec:integer):real;
Procedure X_Y(x1,y1,x2,y2:double;deg,min,sec:integer;var x,y:double);
Procedure X_Y2(x1,y1,x2,y2:double;deg,min,sec:integer;var x,y:double);
Procedure kontrol(deg1,min1,sec1,deg2,min2,sec2:integer;
x1,y1,x2,y2,xa,ya,xb,yb,xc,yc:double;
var xb_xa,xc_xb,ya_yb,yb_yc:double);
Function tg_bm(x1,y1,x2,y2:double):real;
Function ctg_bm(x1,y1,x2,y2:double):real;
Function s(x,y,xb,yb:double;a:real):real;
Procedure X_Y_m(s1:real;xb,yb:double;a:real;var x,y:double);
implementation
Procedure X_Y2(x1,y1,x2,y2:double;deg,min,sec:integer;var x,y:double);
begin
x:=x2+(y1-y2)*ctg(deg,min,sec);
y:=y2+(x2-x1)*ctg(deg,min,sec);
end; {конец процедуры X_Y2}
Procedure X_Y_m(s1:real;xb,yb:double;a:real;var x,y:double);
begin
x:=xb+s1/(1+sqr(a));
y:=yb+a*(x-xb);
end; {конец процедуры X_Y_m}
Function s(x,y,xb,yb:double;a:real):real;
begin
s:=a*(y-yb)+(x-xb);
end; {конец функции s}
Function tg_bm(x1,y1,x2,y2:double):real;
begin
tg_bm:=(x2-x1)/(y1-y2);
```



```
end; {конец функции tg_bm}
Function ctg_bm(x1,y1,x2,y2:double):real;
begin
ctg_bm:=(y1-y2)/(x2-x1);
end; {конец функции ctg_bm}
Procedure kontrol(deg1,min1,sec1,deg2,min2,sec2:integer;
x1,y1,x2,y2,xa,ya,xb,yb,xc,yc:double;
var xb_xa,xc_xb,ya_yb,yb_yc:double);
var
zn1,zn2:real;
begin
zn1:=1+sqr(ctg(deg1,min1,sec1));
zn2:=1+sqr(ctg(deg2,min2,sec2));
xb_xa:=(ctg(deg1,min1,sec1)*(y1-yb)-(x1-xb))/zn1;
xc_xb:=(ctg(deg2,min2,sec2)*(y2-yb)+(x2-xb))/zn2;
ya_yb:=(ctg(deg1,min1,sec1)*(x1-xb)+(y1-yb))/zn1;
yb_yc:=(ctg(deg2,min2,sec2)*(x2-xb)-(y2-yb))/zn2;
end;{конец процедуры kontrol}
Procedure X_Y(x1,y1,x2,y2:double;deg,min,sec:integer;var x,y:double);
begin
x:=x1+(y1-y2)*ctg(deg,min,sec);
y:=y1+(x2-x1)*ctg(deg,min,sec);
end;{конец процедуры X_Y}
Function ctg(deg,min,sec:integer):real;
var
vr:real;
begin
vr:=deg_rad(deg,min,sec);
vr:=tan(vr);
ctg:=1/vr;
end;{конец функции ctg}
Procedure rad_deg(UG:real;var deg,min,sec:integer);
begin
UG:=UG*180/Pi;
deg:=trunc(ug);
min:=trunc((UG-deg)*60);
sec:=round(((UG-deg)*60-min)*60);
end; {конец процедуры rad_deg}
function deg_rad(deg,min,sec:integer):real;
begin
deg_rad:=(deg+min/60+sec/3600)*Pi/180;
```

```
end; {конец функции deg_rad }
function DIR_UG(XA,YA,XB,YB:double):real;
var
dx,dy:real;
UG:real;
begin
dx:=XB-XA; dy:=YB-YA;
if dx=0 then begin
    if dy=0 then begin
        DIR_UG:=0;
    end else begin
        if dy>0 then begin
            DIR_UG:=3*Pi/2;
        end else begin
            DIR_UG:=Pi/2;
        end; end; end else begin
UG:=arctan(dy/dx);
if dx>0 then begin
    if dy=0 then begin
        DIR_UG:=0;
    end else begin
        if dy>0 then begin
            DIR_UG:=UG;
        end else begin
            DIR_UG:=2*Pi-abs(UG);
        end;
    end;
end else begin
if dy=0 then begin
    DIR_UG:=Pi;
end else begin
    if dy>0 then begin
        DIR_UG:=Pi-abs(UG);
    end else begin
        DIR_UG:=Pi+abs(UG);
    end;
end;
end; end;
end; {конец функции DIR_UG}
end.
```

## ПРИЛОЖЕНИЕ Е- ФРАГМЕНТ КОДА ПРОГРАММЫ ДЛЯ ГРАФИКИ

```
procedure TForm_Graph.N_FileClick(Sender: TObject);
var
  f      :textfile;
  x,y    :vect4_double;1
  xg,yg  :vect4_integer;2
  i,width_ris,height_ris :integer;
  str1   :shortstring;
  nm     :vect4_string;3
  mx,my,maxx,maxy,minx,miny,width_ish,height_ish :double;
  kod    :integer;
begin
  kod_gr:=0;
  if opendialog1.Execute=true then begin
    n1.Enabled:=true;
    ToolButton4.Enabled:=true;  ToolButton5.Enabled:=true;
    assignFile(f,opendialog1.FileName);
    reset(f);
    for i:=1 to 34 do
      begin
        readln(f,str1);
        if i>30 then begin
          preobr(str1,x[i-30],y[i-30],nm[i-30],kod);
          if kod<0 then begin
            showmessage('Ошибка в файле результатов, '+#10#13+'еще раз вы-
            полнить'+ ' вычисления, '+ #10#13+'
            'сохранить результаты и затем' + ' выполнить графику. ');
            exit;
          end; end; end;
          minx:=x[1];  maxx:=x[1];
          miny:=y[1];  maxy:=y[1];
          for i:=2 to 4 do begin
            if minx>x[i] then minx:=x[i];
            if miny>y[i] then miny:=y[i];
            if maxx<x[i] then maxx:=x[i];
```

---

<sup>1</sup> Созданный пользователем тип vect4\_double=array[1..4] of double;

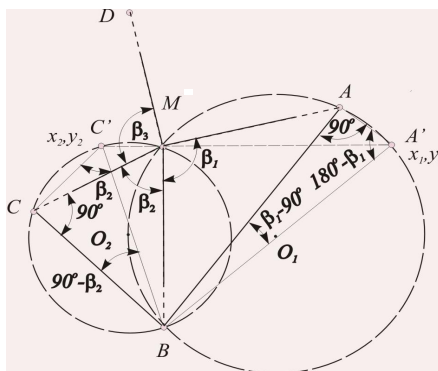
<sup>2</sup> Созданный пользователем тип vect4\_integer=array[1..4] of integer;

<sup>3</sup> Созданный пользователем тип vect4\_string=array[1..4] of shortstring;

```
    if maxy<y[i] then maxy:=y[i];
end;
width_ish:=maxy-miny;      height_ish:=maxx-minx;
width_ris:=Image1.Width-160; height_ris:=Image1.Height-160;
mx:=width_ris/width_ish;   my:=height_ris/height_ish;
if mx>my then mx:=my;
{$R-}
KImage:=TBitmap.Create;
KImage.height:=height_ris+160;
KImage.width:=width_ris+160;
KImage.Canvas.Brush.Color:=clWhite;
KImage.Canvas.pen.Color:=clBlack;
KImage.Canvas.Pen.Style:=psSolid;
KImage.Canvas.font.color:=clBlack;
KImage.Canvas.font.Size:=36;
KImage.Canvas.font.Name:='Times new roman';
KImage.Canvas.font.Style:=[fsBold];//fsItalic
KImage.Canvas.pen.width:=2;
KImage.Canvas.font.Size:=15;
KImage.Canvas.font.Color:=clGray;
KImage.Canvas.TextOut(10,5,
    'Схема обратной однократной геодезической засечки');
for i:=1 to 4 do
begin
xg[i]:=40+Round((y[i]-miny)*mx);
yg[i]:=height_ris-Round((x[i]-minx)*my)+40;
end;
    KImage.Canvas.pen.Color:=clblue;
    KImage.Canvas.Polyline([Point(xg[1],yg[1]), Point(xg[2],yg[2]),
Point(xg[3],yg[3]),
    Point(xg[4],yg[4]), Point(xg[1],yg[1]),Point(xg[3],yg[3])]);
    KImage.Canvas.Moveto(xg[2],yg[2]);
    KImage.Canvas.Lineto(xg[4],yg[4]);
    KImage.Canvas.pen.Color:=clred;
for i:=1 to 4 do
begin
if i<3 then begin
    KImage.Canvas.pen.Color:=clblue;
    KImage.Canvas.Rectangle(xg[i]-5,yg[i]-5,xg[i]+5,yg[i]+5);
end else begin
    KImage.Canvas.pen.Color:=clred;
```

```
KImage.Canvas.Ellipse(Rect(xg[i]-5,yg[i]-5,xg[i]+5,yg[i]+5));
end; end;
KImage.Canvas.font.Color:=clblue;
KImage.Canvas.TextOut(xg[1]+7,yg[1]+7,'1');
KImage.Canvas.font.Size:=8; str(x[1]:10:3,str1);
KImage.Canvas.TextOut(xg[1]+7,yg[1]+27,'X1='+str1);
str(y[1]:10:3,str1);
KImage.Canvas.TextOut(xg[1]+7,yg[1]+39,'Y1='+str1);
KImage.Canvas.font.Size:=15;
KImage.Canvas.TextOut(xg[2]+7,yg[2]+7,'2');
KImage.Canvas.font.Size:=8; str(x[2]:10:3,str1);
KImage.Canvas.TextOut(xg[2]+7,yg[2]+27,'X2='+str1);
str(y[2]:10:3,str1);
KImage.Canvas.TextOut(xg[2]+7,yg[2]+39,'Y2='+str1);
KImage.Canvas.font.Size:=15;
KImage.Canvas.font.Color:=clred;
KImage.Canvas.TextOut(xg[3]+7,yg[3]+7,'P');
KImage.Canvas.font.Size:=8; str(x[3]:10:3,str1);
KImage.Canvas.TextOut(xg[3]+7,yg[3]+27,'XP='+str1);
str(y[3]:10:3,str1);
KImage.Canvas.TextOut(xg[3]+7,yg[3]+39,'YP='+str1);
KImage.Canvas.font.Size:=15;
KImage.Canvas.TextOut(xg[4]+7,yg[4]+7,'Q');
KImage.Canvas.font.Size:=8; str(x[4]:10:3,str1);
KImage.Canvas.TextOut(xg[4]+7,yg[4]+27,'XQ='+str1);
str(y[3]:10:3,str1);
KImage.Canvas.TextOut(xg[4]+7,yg[4]+39,'YQ='+str1);
{SR}
kod_gr:=1;
Image1.Picture.Assign(KImage);
end else begin
Showmessage('Файл с данными для графики не открыт! Работа прервана!');
close;
end;
end;
```

## ПРИЛОЖЕНИЕ Ж – ПОДГОТОВКА ТЕСТОВОГО ПРИМЕРА В MATHCAD \*



Тестовый пример 1 для апробации программного продукта  
 Задача: вычисление прямоугольных координат пунктов геодезической сети, определяемых обратной засечкой (Ю.Рандярев)

$$\beta_1 := (102 \ 39 \ 37) \quad \beta_2 := (54 \ 47 \ 41) \quad \beta_3 := (110 \ 24 \ 48)$$

$$XA := 8429.93 \quad YA := 6016.34$$

$$XB := 6273.45 \quad YB := 4008.30$$

$$XC := 7373.14 \quad YC := 2737.12$$

$$b_1 := \cot \left[ \left( \beta_{10,0} + \frac{\beta_{10,1}}{60} + \frac{\beta_{10,2}}{3600} \right) \cdot \frac{\pi}{180} \right] \quad b_1 = -0.224631$$

$$b_2 := \cot \left[ \left( \beta_{20,0} + \frac{\beta_{20,1}}{60} + \frac{\beta_{20,2}}{3600} \right) \cdot \frac{\pi}{180} \right] \quad b_2 = 0.705560$$

$$X1 := XA + (YA - YB) \cdot b_1 \quad Y1 := YA + (XB - XA) \cdot b_1$$

$$X1 = 7978.861 \quad Y1 = 6500.753$$

$$X2 := XC + (YB - YC) \cdot b_2 \quad Y2 := YC + (XC - XB) \cdot b_2$$

$$X2 = 8270.034 \quad Y2 = 3513.018$$

Контроль правильности вычислений

$$XB\_XA := \frac{[b_1 \cdot (Y1 - YB) - (X1 - XB)]}{1 + b_1^2} \quad XB\_XA = -2156.480 \quad XB - XA = -2156.480$$

$$XC\_XB := \frac{[b_2 \cdot (Y2 - YB) + (X2 - XB)]}{1 + b_2^2} \quad XC\_XB = 1099.690 \quad XC - XB = 1099.690$$

\* Не обязательно тестовые примеры подготавливать в MathCad, возможно применить любой другой вариант расчета (на микрокалькуляторе, в Excel и т.п.)

$$Y_A\_YB = \frac{[b1 \cdot (X1 - XB) + (Y1 - YB)]}{1 + b1^2} \quad Y_A\_YB = 2008.040 \quad Y_A - YB = 2008.040$$

$$YB\_YC = \frac{[b2 \cdot (X2 - XB) - (Y2 - YB)]}{1 + b2^2} \quad YB\_YC = 1271.180 \quad YB - YC = 1271.180$$

Вычисляем дирекционный угол BM

$$tg\_BM := \frac{X2 - X1}{Y1 - Y2} \quad tg\_BM = 0.097456$$

$$ctg\_BM := \frac{Y1 - Y2}{X2 - X1} \quad ctg\_BM = 10.261033 \quad a := tg\_BM \quad c := ctg\_BM$$

Определяем координаты точки M

$$S1 := a \cdot (Y1 - YB) + (X1 - XB) \quad S1 = 1948.316$$

$$S2 := a \cdot (Y2 - YB) + (X2 - XB) \quad S2 = 1948.316$$

$$XM := XB + \frac{S1}{1 + a^2} \quad XM = 8203.436 \quad XM := XB + \frac{S2}{1 + a^2} \quad XM = 8203.436$$

$$YM := YB + a \cdot (XM - XB) \quad YM = 4196.389$$

Контроль

$$S3 := c \cdot (X1 - XB) + (Y1 - YB) \quad S3 = 19991.734$$

$$S4 := c \cdot (X2 - XB) + (Y2 - YB) \quad S4 = 19991.734$$

$$XM := XB + c \cdot (YM - YB) \quad XM = 8203.436$$

$$YM := YB + (YM - YB) \quad YM = 4196.389$$

Заключительный контроль

$$b\_MD := dir\_ug(XM, YM, XD, YD) \quad b\_MD = 6.122 \quad ugo(b\_MD) = (350 \ 46 \ 21)$$

$$b\_MC := dir\_ug(XM, YM, XC, YC) \quad b\_MC = 4.195 \quad ugo(b\_MC) = (240 \ 21 \ 40)$$

b3\_k := b\_MD - b\_MC

bk\_3 := ugol(b3\_k)      bk\_3 = (110 24 41)

<pre> ugol(ug) ≡    ug ← ug · <math>\frac{180}{\pi}</math>    gr ← trunc(ug)    min ← trunc[(ug - gr) · 60]    sec ← round([(ug - gr) · 60 - min] · 60, 1]    βk3<sub>0,0</sub> ← gr    βk3<sub>0,1</sub> ← min    βk3<sub>0,2</sub> ← sec    return βk3         </pre>	<pre> dir_ug(x1,y1,x2,y2) ≡    dx ← x2 - x1    dy ← y2 - y1    if dx = 0       ug ← <math>\frac{\pi}{2}</math> if dy &gt; 0       ug ← 0 if dy = 0       ug ← <math>\frac{\pi \cdot 3}{2}</math> if dy &lt; 0    if dx &gt; 0       ug1 ← atan(<math>\frac{dy}{dx}</math>)       ug ← ug1 if dy ≥ 0       ug ← 2 · π -  ug1  if dy &lt; 0    if dx &lt; 0       ug1 ← atan(<math>\frac{dy}{dx}</math>)       ug ← π -  ug1  if dy ≥ 0       ug ← π +  ug1  if dy &lt; 0    return ug         </pre>
---	---