



КАЗАНСКИЙ (ПРИВОЛЖСКИЙ)  
ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

**Е.Д. КОНДРАТЬЕВА, В.С. МЕНЖЕВИЦКИЙ, Г.Д. МУЛЬКАМАНОВ**

**ИНФОРМАТИКА**  
**МЕТОДИЧЕСКИЕ УКАЗАНИЯ К**  
**ЛАБОРАТОРНЫМ РАБОТАМ**

Казань  
2010

**Е.Д. КОНДРАТЬЕВА, В.С. МЕНЖЕВИЦКИЙ, Г.Д. МУЛЬКАМАНОВ**

**ИНФОРМАТИКА  
МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ЛАБОРАТОРНЫМ  
РАБОТАМ**



**Казанский (Приволжский) федеральный университет  
2010**

УДК 52

ББК 22.6

К 64

*Печатается по рекомендации  
Редакционно-издательского совета  
физического факультета КФУ*

Научный редактор –  
кандидат физ.-мат. наук, доцент кафедры астрономии  
и космической геодезии **М.Г. Ишмухаметова**

Рецензент:  
зав. кафедрой экономической информатики и математики ТГГПУ,  
кандидат физ.-мат. наук, доцент **И.Н. Голицына**

Кондратьева Е.Д.

**К 64 Информатика:** методические указания к лабораторным работам /  
Е.Д. Кондратьева, В.С. Менжевицкий, Г.Д. Мулькаманов. – Казань:  
Казан. фед. ун-т, 2010. – 36 с.

Пособие предназначено для студентов 1-го курса специальности «Астрономогеодезия».

Учебный план специальности «Астрономогеодезия» включает в себя четыре раздела по курсу информатика: 1 – 1 курс, 1 семестр – основы программирования на языке Basic; 2 – 2 курс, 1 семестр – MS Office (Word, Excel, PowerPoint), Origin, LaTex, Visual Basic; 3 – 2 курс, 2 семестр – статистическая обработка наблюдений; 4 – 2 курс, 2 семестр – Общие вопросы применения персональных компьютеров.

Данное пособие посвящено разделу 1.

УДК 52

ББК 22.6

© Казанский (Приволжский)  
федеральный университет, 2010

## 1. ВВЕДЕНИЕ

Трудно представить себе область знаний или практической деятельности человека, в которой не использовались бы компьютеры. Они применяются в наше время не только при научных и инженерных расчетах, но и для хранения и обработки информации.

Все это требует от специалистов в разных областях овладения навыками использования вычислительной техники.

Студенты кафедры астрономии приобщаются к использованию компьютеров и простейших численных методов с 1-го курса. Одной их основных дисциплин при этом становится вычислительная математика, изучающая численные методы решения задач, которые моделируют различные процессы, в том числе и относящиеся к геодезии.

### Этапы решения задач на компьютере.

При решении любой задачи на компьютере основная роль до сих пор принадлежит человеку. Вычислительная машина только выполняет задание, разработанное программистом. Роль человека и компьютера особенно ярко предстает перед нами, если разбить весь цикл работ на основные части.

**Постановка задачи.** Этот этап заключается в физической постановке задачи и определения конечных целей решения.

**Построение математической модели.** Здесь главное – верная математическая формулировка задачи. Модель должна правильно (адекватно) описывать основные законы изучаемого процесса. Поэтому построение или выбор математической модели требует не только глубоких знаний в данной области (например, в геодезии), но и ориентировки во множестве существующих математических методов.

**Разработка численных методов.** Компьютер может выполнять любые математические операции, но не может воспринять постановку задачи. Для ее решения необходимо создать или выбрать из уже имеющихся наилучший численный метод. Разработкой численных методов занимаются, в основном, специали-

сты в области вычислительной математики. Специалисту-прикладнику, как правило, надо из огромного арсенала уже имеющихся методов выбрать тот, который решит поставленную задачу с нужной точностью и кратчайшим путем.

**Разработка алгоритма.** Порядок вычислений при решении задачи, особенно если она изобилует логическими переходами между отдельными частями, записывается в виде элементарных арифметических и логических операций, называемых в целом алгоритмом. Часто он изображается в виде блок-схемы. Однако надо отметить, что опытный вычислитель может обойтись и без такого наглядного изображения алгоритма.

**Программирование.** Это запись решения задачи на одном из языков программирования.

**Отладка программы.** Очень важный этап, заключающийся по существу в проверке самой программы. Ведь она может содержать описки, неточности и прямые ошибки. Обычно отладка производится по примеру, решенному независимо. Желательно, чтобы после этого по ней были вычислены несколько тестовых задач.

**Проведение расчетов.** На этом этапе подготавливаются необходимые исходные данные, подбираются соответствующие константы и проводится сам расчет.

**Анализ результатов.** Один из самых важных этапов решения. Ведь мало просто вычислить какие-то величины. Главное – понять их скрытый смысл, их взаимные связи и сделать верные выводы.

**Математические модели.** К математическим моделям можно предъявлять различные требования (например – простота исполнения, скорость решения и т.д.). Однако главное из них – это адекватность модели и рассматриваемого процесса или явления. Модель должна достаточно точно отражать характерные черты изучаемого материала, быть尽可能 простой и соответствовать точности исходных данных.

Для решения задач, уже преобразованных в математическую модель, применяются три основные группы методов: аналитические, графические и численные.

**Аналитические методы** заключаются в том, чтобы найти решение в виде формул. Они широко используются в таких областях, как небесная механика, теория фигуры Земли и других планет, а также в любых случаях, когда надо найти решение алгебраических, трансцендентных или дифференциальных уравнений. Однако, как правило, это трудный путь. Например, над аналитическим решением задачи трех тел астрономы работают столетия.

**Графические методы** часто позволяют найти только приближенное значение искомой величины. Они состоят в том, чтобы найти решение путем геометрических построений. При этом всегда необходимо верно оценить точность такого решения.

**Численные методы** – это наибольшая по объему группа, особенно теперь, во время широкого применения компьютеров. В математике выделился специальный раздел – численные методы в приложении почти ко всем областям жизни человечества, от запусков космических аппаратов до медицины. Как правило, численные методы позволяют найти решение в определенных рамках постановки задачи, с некоторыми конкретными значениями исходных данных. Несмотря на это, численные методы незаменимы в сложных задачах, которые не допускают аналитических решений.

## 2. ОСНОВЫ ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ BASIC

В настоящее время разработано немало алгоритмических языков, ориентированных на особенности решаемых задач и независящих от наличия конкретной вычислительной машины. Одним из первых и наиболее удачных языков, применяемых в научных и технических задачах, был и является Fortran. Он был разработан в 1954 году фирмой IBM. Название языка происходит от сокращения слов FORmulae TRANslation – преобразование формул. За прошедшие десятилетия

Fortran дополнялся, изменялся, и поэтому сейчас в работе находятся несколько его версий.

Basic был создан на основе языка Fortran. Его первоначальная цель – язык для обучения навыкам программирования. Кроме того, Basic с самого начала содержал графику, которой нет в языке Fortran. Он был разработан в 1965 году сотрудниками Дортмутского колледжа. Название – первые буквы слов Beginners All-purpose Symbolic Instruction Code – многоцелевой язык символьических инструкций для начинающих. Одна из новых версий языка, наиболее удобная для выполнения геодезических задач, где наглядность получаемой конструкции имеет большое значение, это Visual Basic.

Алфавит Basic состоит из следующих символов:

1) 26 латинских букв: A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z.

Отметим сразу, запись программы надо всегда вести в строчных, маленьких буквах. При вводе строки РС переводит их в заглавные, если в операторе нет ошибок. Исключение – пояснения, стоящие в кавычках.

2) 10 десятичных цифр: 0,1,2,3,4,5,6,7,8,9.

3) 5 знаков препинания:

точка – .

запятая – ,

апостроф – ‘

двоеточие – :

точка с запятой – ;

4) 5 знаков арифметических действий:

сложения – +

вычитания – –

умножения – \*

деления – /

возведения в степень – ^.

5) 3 знака отношения (сравнения):

больше.– >

меньше – <

равно – =

6) скобки ( )

7) пробел (в программе не записывается)

Кроме того, есть специальные символы: # ! \ @ \$.

Все величины, необходимые нам в ходе решения задачи, обозначаются латинскими буквами, если надо – с добавлением цифр.

Длина имени каждого числа зависит от применяемой версии языка Basic. У нас это 4 символа, обязательно начинающегося с буквы. Компьютер сам записывает содержание этих имен в ячейки памяти, и сам делает выборку из неё в ходе решения задачи.

Можно напомнить, что при решении специальных задач удобнее использовать буквы соответствующих русских слов.

Например: ячейка для дисперсии – d,

ячейка для среднего – sr, и т.д.

Числа могут быть представлены в памяти компьютеров различными способами. Для хранения числа в современных компьютерах отводится 4 байта памяти, что соответствует 6-8 десятичным разрядам. Однако в большинстве современных языков программирования предусмотрена процедура, позволяющая работать с двойной точностью. 8 байтов – это 12–16 десятичных знаков.

Обычная, общепринятая запись числа в виде 137.8 называется формой записи с фиксированной точкой. Она используется на этапе ввода и вывода чисел. Но при выводе компьютер сам, чаще всего при выводе малых или, наоборот, больших чисел использует нормализованную форму. В ней число 137.8 может быть представлено в виде:  $13.78 \cdot 10^1$ ,  $1.378 \cdot 10^2$ ,  $0.1378 \cdot 10^3$ . На экране дисплея это выглядит так:

число 0.0038, вид выдачи 38E-4,

число 0.018, вид выдачи 18E-3

Надо отметить, что при десятичной записи чисел целую и дробную часть в России часто отделяют запятой. Но в англоязычных странах принято писать точку, т.е. 2.38, а не 2,38. Такая запись (с точкой) используется и в коде программ.

### Основные операторы языка Basic

1. Очистка экрана – cls происходит от сокращения слов clear screen-очистить экран

2. Ввод исходных данных с клавиатуры

input a

input a,b

input "a,b=", a,b

3. Оператор присвоения – знак равенства

p=a-g

s=d+k; I=a+j

4. В одной строке может быть несколько операторов, разделенных двоеточием.

5. Вывод на экран

print s

print "s=",s

print a,b,c

6. Конец программы

end – ставится в конце исполнения программы, может использоваться несколько раз, завершая отдельные ветви программы

7. Наиболее употребляемые функции:

a=sqr(x)       $\sqrt{x}$       c = exp(x)       $e^x$

b=abs(a)      |a|      d = log(x)      log(x)  
f = ln (x)      ln(x)

Все угловые величины переводятся в программе в радианную меру.

Перевод из градусов в радианы:

$$n = a \cdot 3.141593 / 180 \quad a - \text{угол в градусах}$$

Функция	Вид набора
---------	------------

синус	sin(a)
-------	--------

косинус	cos(a)
---------	--------

тангенс	tan(a)
---------	--------

арктангенс	atn(x)
------------	--------

Могут быть вычислены:

секанс	1/cos(x)
--------	----------

косеканс	1/sin(x)
----------	----------

котангенс	1/tan(x)
-----------	----------

арксинус	atn(x/sqr(1-x^2))
----------	-------------------

арккосинус	1.570796 – atn(x/sqr(1-x^2))
------------	------------------------------

Преобразование в целое:

с отбрасыванием дробной части      fix(x)

Наименьшее целое      int(x)

Если число <0, то int(x) делает следующее:

int(14.8)      ответ 14

int(-14.8)      ответ -15

int – сокращение от слова INTeger – целый.

8. Образование цикла

for i=1 to n	начало цикла	} тело цикла повторяется <i>n</i> раз
...		
next i	конец цикла	

9. Индексированные переменные

a(i)	b(k)	одномерные
------	------	------------

a(i,j)	b(k,n)	двумерные
--------	--------	-----------

10. Описание массива

`dim x(n)`      *x – числа массива*

*n – их количество*

`dim` – сокращение слова `dimension` – размер

## 11. Условный переход

`if ... goto ...`

`if ... then ...`

`if a<0 goto 3`      переход на строку с меткой 3

при невыполнении условия управление передается на следующую команду

`if s>d then n=m+1`

если условие не выполнено, оператор `n = m+1` тоже не выполняется, в ячейке `n` остается старое содержание, а РС переходит к выполнению следующей строки.

### *Для тех, кто не работал на компьютере*

После набора программы и прежде, чем идти на решение, необходимо записать ее в оперативную память РС. Обычно преподаватель заранее организует директорию для записи учебных программ. Вам надо уметь ее использовать. Для этого и существует верхнее меню. Вам будут говорить об этом подробно, а сейчас остановим внимание на первом пункте, меню `File`.

В меню `File` раскрываются следующие подпункты:

`New` – начать создание новой программы, т.е. очистить экран

`Open` – открыть рабочее окно для вызова на экран ранее созданной программы

`Save` – сохранить отредактированную программу под старым именем

`Save As` – открыть окно для записи новой программы под новым именем

`Print` – печать

`Load file` – загрузить файл

`Unload file` – убрать файл

`Dos Shell` – выход в систему DOS

`Exit` – выйти из работы с языком Basic.

При отладке и редактировании программы необходимо использовать управляющие клавиши в правой части клавиатуры:

`Home` – перенос курсора в первую левую позицию

`End` – перенос курсора в конец строки

`Del` – удаление символа, под которым стоит курсор

`Insert` – при нажатии этой клавиши курсор начинает мигать и не подчеркивает букву, а выглядит как прямоугольник. В этом режиме вы пишете новые буквы поверх старых знаков. Еще раз нажав `Insert` вы приводите курсор в обычное состояние.

## ЛАБОРАТОРНЫЕ РАБОТЫ

### ЦЕЛЬ ЗАДАНИЯ

- Приобретение навыков работы на компьютере в режиме диалога.  
Изучение клавиатуры.
- Составление программы с использованием операторов языка Бейсик для проведения вычислений с небольшим количеством информации.
- Ввод программы, овладение простейшими методами ее отладки.

### СОДЕРЖАНИЕ ЗАДАНИЯ

- Изучение необходимых операторов языка.
- Составление программы для данного варианта задания.
- Решение задачи. Вывод и обсуждение результатов.
- Представление отчета.

### СОДЕРЖАНИЕ ОТЧЕТА

- Описание отчета и исходных данных.
- Алгоритм решения или блок-схема.
- Текст программы.
- Анализ полученных результатов (если это реальная астрономо-геодезическая задача).

## ЛАБОРАТОРНАЯ РАБОТА 1

В заданиях 1–1...1–9 необходимо составить программу и вычислить искомое значение при заданных параметрах.

- 1–1. Перевести момент наблюдений, заданный датой (число) и временем (часы, минуты, секунды) в сутки и доли суток.

Дано: август 2009 года

№	Дата	Время	№	Дата	Время
1	18	17 <sup>h</sup> 29 <sup>m</sup> 30 <sup>s</sup> .4	4	23	21 <sup>h</sup> 48 <sup>m</sup> 27 <sup>s</sup> .5
2	19	20 <sup>h</sup> 21 <sup>m</sup> 43 <sup>s</sup> .6	5	24	01 <sup>h</sup> 18 <sup>m</sup> 37 <sup>s</sup> .8
3	20	22 <sup>h</sup> 18 <sup>m</sup> 28 <sup>s</sup> .5	6	25	02 <sup>h</sup> 30 <sup>m</sup> 57 <sup>s</sup> .3

- 1–2. Перевести угол в градусах, минутах секундах дуги в градусы и радианы.

Дано:

№	Угол	№	Угол
1	18°46'30".7	4	46°13'42".5
2	129°37'18".6	5	328°19'15".6
3	248°15'50".1	6	17°20'18".9

1–3. Дан магнитный азимут линии и восточное склонение магнитной стрелки (оба в градусах, минутах и долях минут). Найти истинный, т.е. географический азимут в градусах и радианах.

№	Азимут	Склонение
1	30°40'.3	10°03'
2	50°16'.3	10°04'
3	130°30'.8	10°12'
4	200°18'.9	10°30'
5	300°17'.8	10°15'
6	320°29'.3	10°20'

1–4. Вычислить координаты точки С, если известны координаты точки А( $x_1, y_1$ ), расстояние между точками  $d$  и дирекционный угол линии АС в градусах, минутах и долях минут.

№	$x_1$	$y_1$	$d$	$\alpha$
1	26.38	18.92	137.52	45 18.6
2	137.45	22.16	145.62	98 13.6
3	63.15	102.18	120.33	103 16.3
4	125.16	92.13	130.47	134 18.6
5	38.45	16.32	92.13	200 18.6
6	92.35	107.83	140.37	230 20.7

$$x_2 = x_1 + d \cdot \cos \alpha$$

$$y_2 = y_1 + d \cdot \sin \alpha$$

1–5. Вычислить значение

$$N = \sqrt{\left(\frac{ac^3}{b}\right)^2 + c^2 \cdot d^2} - \left(\frac{ac^3}{b}\right)^2$$

$$a = 0.7 \quad b = 7.7 \quad c = 1.8 \cdot 10^4 \quad d = 5.1$$

1–6. Вычислить значение

$$N = 3a^2 + x(b^3 + x(c^4 + x(a^b + e^x)))$$

$$a = 3.4 \quad b = 2 \quad c = 1.3 \quad x = 3.6$$

1–7. Вычислить значение

$$A = |\sin b| \cdot \cos \frac{b}{2} + \sqrt[3]{x^2 + y^2} + \ln b$$

$$b = 4.2 \quad x = 3.1 \quad y = 0.8$$

1–8. Вычислить значение

$$m = \frac{a \cdot \sqrt{a^2 + b^2}}{2} - \frac{b}{2} \cdot \ln |a + \sqrt{a^2 + b^2}|$$

$$a = \frac{1-x+|y|}{x+y} \quad x = 3.75 \quad y = -2.3 \quad b = 41.2 \cdot 10^{-2}$$

1–9. Вычислить значение

$$x = y \cdot \sqrt{a^2 + \left(1 - \left(\frac{a^2}{b^2}\right)\right) \cdot b^2} \\ \cdot \sqrt{1 - \left(\frac{a^2}{b^2}\right)} \cdot \sqrt{1 - \left(\frac{b^2}{c^2}\right)}$$

$$y = 5.1 \quad a = 6.27 \cdot 10^4 \quad c = 7 \cdot 10^5 \quad b = 1.5$$

## ЛАБОРАТОРНАЯ РАБОТА 2

В заданиях 2–1...2–9 необходимо составить разветвляющуюся программу при заранее неизвестных значениях параметров.

Для организации ветвления программы, т.е. переходов, связанных с выполнением некоторых условий, применяется оператор IF - если.

Общий вид оператора:

```
if a > b goto n    goto – идти к...
if a > b then c   then – тогда
```

Здесь a,b - произвольные арифметические выражения, > - символ операции сравнения, n - метка строки программы, к которой делается переход при выполнении условия. В Бейсике метка - это число.

Символы операций отношения:

```
> больше   <= или = < меньше или равно
< меньше   >= или = > больше или равно
= равно
then c – с любой оператор Бейсика, после него – безусловный переход
go to ...
```

При невыполнении условия управление передается следующей строке программы.

Например:

Вычислить 1)  $y = a - 3b$ , если  $ab < 0$   
2)  $y = (ab)^{1/2}$ , если  $ab > 0$

Алгоритм решения:

1. Ввести значения переменных a,b;
2. проверить знак произведения  $a*b$ , если  $a*b < 0$ , то вычислить у по формуле 1 и перейти к пункту 3, если  $a*b > 0$  - то по формуле 2 и также перейти к пункту 3 алгоритма;

3. Выдать на экран значение функции у;
4. Завершить работу оператором end.

```
cls
input a,b
if a*b > 0 goto 40
y1 = a - 3*b
print "y1 = ",y1
end
40 y2 = sqr(a*b)
print "y2 = ",y2
end
```

Данная работа может быть расширена. Можно возвращаться к пункту 1 для задания новых исходных данных, можно использовать в программе только один оператор print. Мы рассмотрим эти варианты позднее.

2–1. При обработке замкнутого теодолитного хода прежде всего вычисляется сумма внутренних углов полигона. Известно, что для выпуклых многоугольников она теоретически равняется  $S = 180*(n - 2)$ , где n - число углов. Получение практической суммы может быть связано с необходимостью перевода минут дуги (если они  $> 60$ ) в градусы.

Составьте программу для примера:

Дано:

1	u1 =	40°18'	u2 =	100°50'
2		60°30'		200°40'
3		120°57'		45°18'
4		200°45'		123°56'
5		46°34'		87°43'
6		112°56'		56°34'

Найти  $s1=u1 + u2$

**2-2.** Дирекционные углы полигона вычисляются по формуле

$$\alpha_2 = \alpha_1 + 180^\circ - \beta_{\text{пп}},$$

где  $\beta_{\text{пп}}$  — правый по ходу угол полигона. При этом  $\alpha_2$  может получиться больше  $360^\circ$ , которые из его окончательного значения надо вычесть.

Составьте программу для примера:

1.	$\alpha_1 =$	$290^\circ$	$\beta_{\text{пп}} =$	$60^\circ$
2.		$270^\circ$		$90^\circ$
3.		$200^\circ$		$10^\circ$
4.		$285^\circ$		$70^\circ$
5.		$300^\circ$		$100^\circ$
6.		$320^\circ$		$150^\circ$

**2-3.** Вычислить значение

$$u = \begin{cases} ax^2 + b \ln x & x > \sqrt{a+b} \\ \sqrt{a+\sin 2x} - e^{3x} & x < \sqrt{a+b} \end{cases} \quad a = 1.35 \quad b = 2.8 \quad x = 4.8$$

**2-4.** Вычислить значение

$$(x^2 - a) - c\sqrt{2x} \quad a + x^2 > c \quad c = 30.6$$

$$N = c^{-2} - \sqrt{\lg(2x-1)} \quad a + x^2 < c \quad a = 15.8 \quad x = 1.5$$

**2-5.** Вычислить значение

$$a^x - e^x + b^3 \cdot \cos(4x - 0.2) \quad |a^2 - b^2| < 10x \quad b = 2123.78 \cdot 10^{-2}$$

$$M = \frac{\tg(4.5x) + \frac{x}{\sin 0.5x}}{\sin 0.5x} \quad |a^2 - b^2| > 10x \quad a = 234.56 \quad x = \frac{\sqrt{2}}{10}$$

**2-6.** Вычислить значение

$$y = \begin{cases} \frac{x \cdot e^3}{|a+x|} + \sin^2 a & a < 2x \quad x = 0.45 \cdot 10 \\ (ax^3 + 6x^2 + 105) \cdot \sqrt[3]{\sin a} & a > 2x \quad a = 2 \end{cases}$$

**2-7.** Вычислить значение

$$C = \begin{cases} \frac{2(2a+b)^{3/2}}{(b-a)^2 + a+1} & a > b \quad a = 2^{-3.981} \cdot \sqrt{3.981 + \sqrt[4]{1.625}} \\ \frac{3(2b+a)^{3/2}}{(b-a)^2 + a+1} & a < b \quad b = \sqrt[3]{e^{3.981 - \sqrt{\sin 0.512}}} \end{cases}$$

**2-8.** Вычислить значение

$$H = \begin{cases} \frac{(a+b)^2 + 3(b-a)^2}{\sqrt{(3a-2)^2 + 4b^2}} & |ab| > 1 \\ \frac{(a-b)^2 - ab}{2 + ab} & |ab| < 1 \end{cases} \quad a = 2 \cos \left( \frac{1.426 - 3.14159}{6} \right) \quad b = 1 + \frac{3.5^2}{3 + \frac{3.5^2}{5}}$$

**2-9.** Вычислить значение

$$K = \begin{cases} 4 \operatorname{arctg}(x) + \frac{ax+7}{\sqrt{b^2+x^2}} & a+b < \sin^2 x \quad b = \cos a \\ \sqrt{b^2 - \tg(x)} - \sin x & a+b > \sin^2 x \quad a = 0.1 \quad x = 0.3 \end{cases}$$

### ЛАБОРАТОРНАЯ РАБОТА 3

В заданиях 3–1...3–10 необходимо составить программу, используя организацию цикла через `for...next`.

В общепринятом смысле цикл – это повторение некоторого действия или набора действий. Например, вычислим угол в радианах, если дано 10 исходных данных.

Это можно сделать посредством уже знакомого Вам оператора `if...goto`

```
cls
i=10
5 input g,m
u=(g+m/60)*3.141592/180
print "u=",u
i=i+1
if i<=10 goto 5
end
```

Однако в Бейсике того же можно добиться, используя другую конструкцию:

```
cls
n=10
for i=1 to n
input g,m
u=(g+m/60)*3.141592/180
print "u=",u
next i
end
```

Величина `i` называется переменной цикла, оператор `next i` возвращает вычисления на начало цикла и повторяет их `n` раз. Шаг, заданный здесь по умолчанию равным 1, может быть любым в модификации

`for i=1 to n step m`

Величины `n` и `m` задаются до цикла.

Кроме того, для запоминания массива используется оператор `dim`.

**3–1.** Даны углы замкнутого теодолитного хода. Определить невязку в углах, распределить ее поровну между исходными значениями, получить исправленные углы и проверить их сумму.

Теоретическая сумма внутренних углов выпуклого многоугольника

$$S_t = 180 * (n - 2)$$

Исходные данные на форме "ВЕДОМОСТЬ КООРДИНАТ – 1"

№ точек	Углы изме- ренные	Углы исправ- ленные	Азимуты или дирекц. углы	Румбы	Горизонт. проложе- ния, м	Вычисления				Исправленные				Координаты	
						+	$\Delta x$	+	$\Delta y$	+	$\Delta x$	+	$\Delta y$	$x$	$y$
1															
2	81°44'0														
3	95°15'3														
4	95°50'1														
1	87°21'0														

**3–2.** Вычислить приращения координат между вершинами замкнутого теодолитного хода и невязки по осям  $fx, fy$ .

Дано:  $d(i)$  расстояние между соседними вершинами хода

$al(i)$  дирекционные углы, выраженные в радианах.

Рабочие формулы:  $dx(i)=d(i) \cdot \cos(al(i))$

$$dy(i)=d(i) \cdot \sin(al(i))$$

$$fx=\sum dx(i) \quad fy=\sum dy(i)$$

Исходные данные на форме "ВЕДОМОСТЬ КООРДИНАТ – 2".

ВЕДОМОСТЬ КООРДИНАТ															
Но мер точек	Углы изме- ренные	Углы исправ- ленные	Азимуты или дирекц. углы	Румбы	Горизонт. протяж- ния, м	Вычисленные				Исправленные			Координаты		
						+	$\Delta x$	+	$\Delta y$	+	$\Delta x$	+	$\Delta y$	$x$	$y$
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1			38°27'9"		32.749										
2			136°46'9"		49.910										
3			221°34'0"		23.924										
4			305°46'9"		44.967										
1			38°27'9"												

3–3. При прокладке замкнутого нивелирного хода определяются превышения между опорными точками. Обработка хода начинается с определения их суммы. Отличие от нуля – это невязка, распределяемая между измеренными величинами. Исправленные превышения замкнутого хода должны в сумме давать ноль.

Задание: вычислить невязку в превышениях замкнутого нивелирного хода.

(Превышения записаны в миллиметрах)

1)	1 – 2	+ 1000
	2 – 3	+ 0922
	3 – 4	+ 1832
	4 – 5	- 1004
	5 – 1	- 2748

2)	1 – 2	- 0200
	2 – 3	+ 1306
	3 – 4	+ 1052
	4 – 5	+ 1022
	5 – 1	- 0564

3–4. При работе на стационарных станциях мы получаем не 1–2 измерения, а ряд гораздо больший. В таком случае после вычисления среднего и его ошибки интересно выявить величины, превышающие общепринятый порог (тройная средняя квадратичная ошибка).

Задание: ввести массив и выдать на экран величины, превышающие заданное значение  $W$ .

3–5. Подсчитать количество нулевых значений в заданном ряду.

3–6. Выдать на экран все отрицательные значения заданного ряда.

3–7. Вычислить значение

$$y = \frac{(\cos x + \sin 2x)^2 - a \cdot \sin x}{3.141592 + \sin x}$$

при  $a = 2.5$   $0^\circ < x < 45^\circ$  шаг  $\Delta x = 5^\circ$

3–8. Вычислить значение

$$y = \frac{\sqrt{a} \cdot \sin x}{x + \cos^2 x}$$

при  $a = 8.45$   $5^\circ < x < 65^\circ$  шаг  $\Delta x = 10^\circ$

3–9.

$$\text{Вычислить } x_{n+2} = 2x_{n+1} - 0.7x_n$$

при  $x_0 = 1$   $x_1 = 1.5$  Найти  $x_5$

3–10.

$$\text{Вычислить } z_{n+2} = \sqrt{x_{n+1} + z_n^2}$$

при  $z_0 = 0.5$   $z_1 = 0.8$  Найти  $z_5$

## ЛАБОРАТОРНАЯ РАБОТА 4

Двумерный массив, работа с матрицами.

В практике астрономо-геодезических работ мы приходим к уравниванию измеренных величин. С этой целью чаще всего используется метод наименьших квадратов. Теорию его Вам будут давать на втором курсе, а сейчас мы покажем, как легко программируются части этого метода, если использовать понятие двумерного массива.

Будем считать, что двумерный массив – это матрица, в которой  $n$  строк и  $m$  столбцов.

Программа для ввода матрицы по строкам и вывода её на экран.

```
cls
input n, m
dim a(n,m)
for i=1 to n
for j=1 to m
input a(i,j)
next i
next j
for i=1 to n
for j=1 to m
print a(i,j)
next j
print
next i
end
```

**4–1.** Вычислить сумму всех членов матрицы.

**4–2.** Умножить матрицу на скаляр.

**4–3.** Транспонировать квадратную матрицу.

**4–4.** Умножить матрицу построчно на одностолбцовую.

**4–5.** Разделить матрицу построчно на их первые члены.

**4–6.** Привести условные неравноточные уравнения к равноточным, если вес каждого уравнения известен.

Приведение к равной точности – это умножение всей строчки на квадратный корень из ее веса.

**4–7.** Составить из  $n$  условных равноточных уравнений с  $m$  неизвестными и свободным членом новую систему, которая называется нормальной. В ней число уравнений равно числу неизвестных.

### Контрольная работа.

Обработка замкнутого теодолитного хода.

Исходные данные на форме "ВЕДОМОСТЬ КООРДИНАТ – 3"

№ точки	Углы изме- ренные	Углы исправ- ленные	Азимуты или дирекц. углы	Рубли	Горизонт. проложе- ния, м	Вычисленные				Исправленные				Координаты	
						+	$\Delta x$	+	$\Delta y$	+	$\Delta x$	+	$\Delta y$	$x$	$y$
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0					48°43'										
1	167°41'1									22.29					
2	82°45'1									24.73					
3	58°09'3									37.20					
4	102°37'6									35.10					
1	116°23'2									16.32					

## ЛАБОРАТОРНАЯ РАБОТА 5

В практике астрономо-геодезических задач мы часто имеем дело с выборкой данных из таблиц. Совершаемые при этом действия называются интерполяцией. Рассмотрим три основных вопроса этой темы.

**Постановка задачи.** Пусть величина  $y$  является функцией  $t$ , которую мы будем называть аргументом. Явная связь этих величин может быть записана как  $y = f(t)$ . Однако вид этой функции может быть достаточно сложным, содержать большой объем дополнительных вычислений, сложные интегралы и т.д. В таком случае один раз просчитанные значения  $y_i = f(t_i)$  часто задаются таблицей, что значительно упрощает работу всех остальных пользователей. На практике это приводит к тому, что мы хотим получить значение  $f$  не только для заданных точек  $t$ , но и для любого значения аргумента из некоторой области, входящей в данные таблицы.

**Интерполирование по таблице с постоянным шагом.** Чаще всего мы встречаемся со случаем, когда значения аргумента разделены равными промежутками. Эта разность  $(t_i - t_{i-1}) = \text{const}$  обычно обозначается через  $\phi$  и называется шагом данной таблицы. Обратимся к основному справочнику, используемому при обработке и организации наблюдений в астрономогеодезии - Астрономическому Ежегоднику. Он дает богатый и разносторонний материал, необходимый как при небесно-механических расчетах, так и при астрометрических наблюдениях. Например, при вычислении азимута по Солнцу.

Введем наиболее широко используемый тип разностей – обыкновенные (в отличие от центральных и отрицательных, применяемых очень редко).

Пусть нам дана таблица:

Аргумент	Функция	Разности
$t_0$	$f_0$	$f^{1/2}$
$t_1$	$f_1$	$f^2$
$t_2$	$f_2$	$f^{3/2}$
$t_3$	$f_3$	$f^2$
		$f^{5/2}$

Построим первые разности:

$$f_1 - f_0 = f^{1/2}$$

Здесь цифра "1" наверху – это порядок разности, цифра "1/2" внизу – полусумма порядковых номеров функций, из которых она образована:  $(1+0)/2 = 1/2$

Также вычислим  $f_2 - f_1 = f^{3/2}$ ,  $f_3 - f_2 = f^{5/2}$

Теперь построим вторые разности:

$$f^{3/2} - f^{1/2} = f^2 \quad 2 - \text{порядок разности}$$

1 – индекс, 1 =  $(3/2 + 1/2)/2$

$$f^{5/2} - f^{3/2} = f^2 \quad 2 = (5/2 + 3/2)/2$$

Аналогично третья разность  $f^2 - f^2 = f^3$

При этом надо заметить, что специальные геодезические таблицы, в том числе и Астрономический Ежегодник, построены так, чтобы можно было не учитывать даже третью разность. В АЕ, например, интервал задания аргумента подобран в каждой таблице в соответствии с этим требованием.

Наиболее часто для интерполяции используется формула Ньютона.

$$f_t = f_0 + n f_{1/2}^1 + \frac{n(n-1)}{2} f_1^2 + \frac{n(n-1)(n-2)}{2 \cdot 3} f_{3/2}^3 + \dots \quad (5.2)$$

В ней  $n$  – коэффициент интерполяции

$$n = \frac{t - t_0}{\omega}$$

Здесь  $t$  – значение аргумента, для которого надо найти искомую величину,  $t_0$  – предыдущее по таблице значение аргумента, также как  $f_0$  – соответствующее ему предыдущее значение функции,  $\omega$  – табличный шаг, т.е.  $(t_1 - t_0)$ .

Формула (5.2) используется для интерполяции вперед, когда в таблице есть еще в запасе 1-2 члена. Она называется первой формулой Ньютона для интерполяции вперед. Существует, кроме этого, формула Ньютона для интерполяции назад, когда значение  $t_0$  находится в конце таблицы. Как мы уже отмечали, она используется чрезвычайно редко.

В АЕ 2006, например, все исходные данные приведены не только на 31 декабря текущего года, но и на 32 декабря, т.е. на 1 января следующего года.

**Интерполяция по таблице с переменным шагом.** Рассмотрим таблицу исходных данных, в которой значения аргумента разделены неравными промежутками, т.е.

$$t_1 - t_0 \neq t_2 - t_1 \neq t_3 - t_2 \quad \text{и т.д.}$$

Чаще всего это бывает тогда, когда мы имеем дело с рядом наблюдений, разделенных разными промежутками времени.

Для подобного случая, когда  $\omega \neq \text{const}$  существует специальный очень удобный алгоритм.

Возьмем обычную таблицу исходных данных

$t_0$	$f_0$
$t_1$	$f_1$
$t_2$	$f_2$
$t_3$	$f_3$

Составим новые, необычные разности такого вида:

$$\frac{f_1 - f_0}{t_1 - t_0}, \quad \frac{f_2 - f_1}{t_2 - t_1}, \quad \frac{f_3 - f_2}{t_3 - t_2}$$

Такие величины называются в соответствии со своей формой вычисления разделенными разностями первого порядка. Обычно они обозначаются так:

$$f(t_1 t_0) = (f_1 - f_0)/(t_1 - t_0) \quad f(t_2 t_1) = (f_2 - f_1)/(t_2 - t_1) \text{ и т.д.}$$

Разделенной разностью второго порядка называется отношение двух разделенных разностей первого порядка к разности крайних аргументов.

$$f(t_2 t_1 t_0) = (f(t_2 t_1) - f(t_1 t_0))/(t_2 - t_0)$$

$$f(t_3 t_2 t_1) = (f(t_3 t_2) - f(t_2 t_1))/(t_3 - t_1)$$

Аналогично вычисляется разделенная разность третьего порядка:

$$f(t_3 t_2 t_1 t_0) = (f(t_3 t_2 t_1) - f(t_2 t_1 t_0))/(t_3 - t_0)$$

Рабочая формула интерполяции по такой таблице для значения аргумента  $t$  записывается следующим образом:

$$f(t) = f_0 + (t - t_0) \cdot f(t_1 t_0) + (t - t_1) \cdot (t - t_0) \cdot f(t_2 t_1 t_0) + \dots + (t - t_2) \cdot (t - t_1) \cdot (t - t_0) \cdot f(t_3 t_2 t_1 t_0) + \dots \quad (5.3)$$

Здесь, как и прежде,  $t_0$  – предыдущее значение аргумента,  $f_0$  – предыдущее табличное значение функции.

**Обратная задача.** Определение аргумента по заданному значению функции называется обратным интерполярованием. Предположим, что функция  $f(t)$  монотонна и задана таблицей с постоянным шагом. Пусть дано значение функции  $f$ , для которого необходимо найти соответствующий аргумент  $t$ .

Как известно, интерполяционная формула Ньютона следующая:

$$f_t = f_0 + n f_{1/2}^1 + \frac{n(n-1)}{2} f_1^2 + \frac{n(n-1)(n-2)}{2 \cdot 3} f_{3/2}^3 + \dots$$

Искомый аргумент входит в коэффициент интерполяции

$$n = \frac{t - t_0}{\omega}$$

Следовательно, необходимо сначала найти  $n$  методом последовательных приближений, а затем неизвестное  $t$ .

В первом приближении

$$n_1 = \frac{f(t) - f_0}{f_{1/2}^1} \quad (5.4)$$

Второе приближение

$$n_2 = \frac{f(t) - f_0}{f_{1/2}^1} - \frac{n_1(n_1-1)}{2f_{1/2}^1} f_1^2 - \frac{n_1(n_1-1)(n_1-2)}{2 \cdot 3 \cdot f_{1/2}^1} f_{3/2}^3 - \dots \quad (5.5)$$

Обычно третья разность настолько мала, что последующие члены можно не учитывать. Кроме того, можно повторить вычисление по формуле (5.5), подставив в правую часть  $n_2$  вместо  $n_1$ , и сделать это несколько раз, пока разность двух последовательных значений  $n_2$  не станет меньше заданной точности.

Определив  $n$  с нужной точностью, находим аргумент

$$t = t_0 + n\omega \quad (5.6)$$

### Пример 1

В курсе высшей геодезии Вы будете заниматься приближенным определением широты и долготы по зенитным расстояниям Солнца. В Ваших руках будет Универсальный Высокоточный теодолит. Точность его отсчета – 0.2 секунды дуги. Следовательно, необходимые для вычисления прямое восхождение и склонение Солнца надо выбрать из АЕ с не меньшей точностью.

Найдем, для примера, видимое склонение Солнца не используя часовое изменение, а прямым интерполированием.

Дано:

2006, ноябрь, 0 часов земного времени

16	$-18^\circ 38' 39'' .80$	Найти склонение Солнца
17	$-18^\circ 53' 36'' .36$	на момент 16 октября $20^h 45^m .00$
18	$-19^\circ 08' 12'' .78$	

Составим таблицу разностей

$-18^\circ 38' 39'' .80$	
$-0^\circ 14' 56'' .56$	
$-18^\circ 53' 36'' .36$	$+0^\circ 00' 20'' .14$
	$-0^\circ 14' 36'' .42$
$-19^\circ 08' 12'' .78$	

Ввиду малости второй разности третий порядок можно не учитывать.

Вычислим коэффициент интерполяции  $n$ .

$$n = (10 + 20/60 + 45/3600)/24 = 0.431075$$

Далее, учитывая, что компьютер работает с углами в радианной мере, можно пойти по двум путям, 1 – перевести необходимые нам разности в радианную меру, вычислить второй и третий члены формулы 5.2, перевести их сумму в градусную меру и сложить с первым членом; 2 – перевести в радианную меру  $f_0$ ,  $f_1$ ,  $f_2$ , получить разности, вычислить  $f_1$  по формуле 5.2, и затем результат перевести в градусную меру.

Ответ:

$$\delta = -18^\circ 39' 08'' .75$$

## Пример 2

Даны наблюдения:

Дата		$\alpha$	$\delta$
июль	1.12918	UT 2 <sup>h</sup> 36 <sup>m</sup> 44 <sup>s</sup> .9	1° 00' 39"
	20.35892	2 00 36.7	-10 15 20``
	26.45903	1 45 54.4	-16 37 30``

Для сравнения с эфемеридой найти координаты тела на момент июль 10.92378 UT. Покажем, как это сделать, на примере прямого восхождения  $\alpha$ .

Для удобства вычислений переведем исходные данные в радианную меру, взяв один запасной знак. Ведь известно, что если склонение измерено с точностью до секунд дуги (а прямое восхождение до десятых долей секунд времени), то ответ будет содержать только 5 верныхзначащих цифр. Вычислим разделенные разности первого и второго порядков.

$\alpha$
0.683944
- 0.0081995
0.526268
- 0.0000915
- 0.0105183
0.462105

$$\begin{aligned}\alpha = & 0.683944 + (10.92378 - 1.12918) \cdot (-0.0081995) + \\ & + (10.92378 - 20.5892) \cdot (10.92378 - 1.12918) \cdot (0.0000915) = 0.612089\end{aligned}$$

Переводя  $\alpha$  в обычную временную меру, окончательно получаем:

$$\alpha = 2^h 20^m 16^s.8$$

Задания. Пользуясь астрономическим ежегодником 1990 г., выполнить:

5-1. Найти радиус-вектор Венеры на момент:

1 – январь	12 <sup>d</sup> .5 <sup>h</sup> 30 <sup>m</sup>	6 – март	25 <sup>d</sup> .5 <sup>h</sup> 40 <sup>m</sup>
2 –	22 .4 20	7 – апрель	14 .4 50
3 – февраль	13 .3 40	8 –	24 .6 50
4 –	23 .4 10	9 – май	14 .5 40
5 – март	15 .6 05	10 –	24 .6 20

5-2. Найти геоцентрическое расстояние Юпитера на заданный момент.

Внимательно изучите данную таблицу. Вы увидите, что она дана с готовой первой разностью. Учтите это в своей программе.

1 – июнь	1 <sup>d</sup> .22 <sup>h</sup> 30 <sup>m</sup>	6 –	13 <sup>d</sup> .21 <sup>h</sup> 50 <sup>m</sup>
2 –	3 .21 40	7 –	16 .23 15
3 –	6 .22 45	8 –	18 .24 20
4 –	8 .22 15	9 –	21 .21 45
5 –	11 .23 05	10 –	23 .21 30

5-3. Найти момент, которому соответствует радиус-вектор Марса

1.52010

1.50080

1.41200

1.39030

1.38300

Найти момент, которому соответствует радиус-вектор Юпитера

5.15756

5.17837

5.20108

5.22310

5.24318

## ВОПРОСЫ К ЗАЧЕТУ

(уровень минимальных знаний)

1. Методы введения исходных данных:

- а) единичные числа;
- б) одномерные массивы;
- в) двумерные массивы.

2. Работа с угловыми величинами:

- а) суммирование углов;
- б) вычисление тригонометрических функций.

3. Циклические программы и их использование при обработке полевых наблюдений.

4. Определение угла (долгота Солнца) по таблице (Астрономический ежегодник): интерполяция.

## ПРИЛОЖЕНИЕ

### Командные интерпретаторы BASIC.

#### QBasic

С момента первой реализации языка Basic в 1963 году было создано множество диалектов и поддерживающих их компиляторов (интерпритаторов). Огромную популярность получила разработка от компании Microsoft - QBasic. Данная модификация поставлялась вместе со старыми версиями операционных систем MS DOS и Windows, но ее также можно скачать с сайта Microsoft по адресу: [http://download.microsoft.com/download/win95upg/tool\\_s/1.0/w95enus/olddos.exe](http://download.microsoft.com/download/win95upg/tool_s/1.0/w95enus/olddos.exe).

Основное отличие QBasic от QuickBasic, на базе которого он создан, состоит в том, что в первом отсутствует компилятор. Таким образом, программы пользователей могут быть запущены только на компьютерах с установленным экземпляром QBasic.

Возможны следующие варианты запуска интерпритатора:

qbasic.exe - запуск QBasic

qbasic.exe /b - запуск QBasic в черно-белом режиме

qbasic.exe /ed - QBasic для работы с текстами программ будет использовать редактор MS-DOS

qbasic.exe /g - специальный режим работы на мониторах CGA

qbasic.exe /h - отображение максимально возможного числа строк в режиме редактора

qbasic.exe /mbf - режим использования чисел в IEEE-формате как чисел в формате MS Binary

qbasic.exe /nohi - работа с мониторами, не поддерживающих режимы отображения символьной информации повышенной интенсивности

qbasic.exe /run {имя файла} - загрузка файла с программой на BASIC и немедленный ее запуск

## FreeBasic

Первоначально компилятор разрабатывался как свободная альтернатива Microsoft QuickBasic, но быстро оброс новыми возможностями и стал мощным средством разработки. В язык FreeBasic были добавлено множество расширений и возможностей для соответствия современным требованиям, стандартами и совместимости с библиотеками и API написанными на C/C++.

Компилятор FreeBasic Compiler доступен для скачивания на официальном сайте проекта <http://www.freebasic.net/>. Чтобы облегчить работу программиста, используются интегрированные среды разработки. Мы рекомендуем простую в освоении IDE Geany. Одним из достоинств которой является возможность использования разных компиляторов и языков программирования.

Некоторые варианты запуска FreeBasic из командной строки:

fbc -arch < type > - Устанавливает архитектуру (по умолчанию: 486)

fbc -d < name=val > - Добавить препроцессор define

fbc -dylib - Создать DLL, включая импортированную библиотеку

fbc -entry < name > - Основной файл без расширения, точка входа (по умолчанию является первым .bas файл в командной строке)

fbc -eсх - Добавить проверку ошибок и resume, проверять границы массивов и правильность указателей

fbc -i < name > - Добавить путь, для поиска подключаемых файлов

fbc -lang < name > - Выбор языка компиляции: fb, fblite, qb

fbc -lib - Создать статическую библиотеку

fbc -m < name > - Основной файл без расширения, точка входа (по умолчанию является первый .bas файл в командной строке)

fbc -p < name > - Добавить путь, для поиска подключаемых библиотек

fbc -s < name > - Установить подсистему (графический интерфейс пользователя (-s gui), консоль (-s console))

fbc -w < value > - Установите минимальный уровень предупреждений: all, pedantic

fbc -x < name > - Установите имя/путь выходного файла

## ЛИТЕРАТУРА

### Основная:

1. Киреев В.И., Пантелеев А.В. Численные методы в примерах и задачах. М. Высшая школа. 2006. 480 с.
2. Поляков А., Брусенцев В.М. Методы и алгоритмы компьютерной графики. СПб. БХВ-Петр. 2003. 547 с.
3. Тагиров Т.С. Введение в программирование на VS Quick BASIC. Учебное пособие. Казань, КГУ. 2000. 59 с.
4. Турчак Л.И., Плотников П.В. Основы численных методов. М. Физматлит. 2002. 304 с.
5. Симонович С.В. Информатика. Базовый курс. Питер. 2008. 640 с.

### Дополнительная:

1. Абалакин В.К., Краснорылов И.И., Плахов Ю.В. Геодезическая астрономия и астрометрия. М. Геодезиздат. 1996. 475 с.
2. Вострикова З.П., Вострикова О.Ю., Туева С.С. Программирование на языке бейсик для персональных ЭВМ. М. Машиностроение. 1993. 352 с.
3. Кергаль И. Методы программирования на Бейсике. М. Мир. 1991. 287 с.
4. Покровский Г.Б., Ананьев М.П. Программирование на языке Бейсик. Казань, КГУ. 1987. 197 с.

## **Содержание**

<b>1. ВВЕДЕНИЕ.....</b>	<b>3</b>
<b>2. ОСНОВЫ ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ BASIC .....</b>	<b>5</b>
<b>ЛАБОРАТОРНЫЕ РАБОТЫ.....</b>	<b>12</b>
ЛАБОРАТОРНАЯ РАБОТА 1 .....	13
ЛАБОРАТОРНАЯ РАБОТА 2 .....	16
ЛАБОРАТОРНАЯ РАБОТА 3 .....	20
ЛАБОРАТОРНАЯ РАБОТА 4 .....	24
ЛАБОРАТОРНАЯ РАБОТА 5 .....	26
<b>ВОПРОСЫ К ЗАЧЕТУ .....</b>	<b>34</b>
<b>ЛИТЕРАТУРА .....</b>	<b>35</b>

**Кондратьева Екатерина Дмитриевна  
Менжевицкий Владимир Сергеевич  
Мулькаманов Глеб Димович**

## **ИНФОРМАТИКА МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ЛАБОРАТОРНЫМ РАБОТАМ**

**Выходит в авторской редакции**

**Подписано в печать 20.07.2010.  
Бумага офсетная. Печать ризографическая.  
Формат 60x84 1/16. Гарнитура «Times New Roman». Усл.печ.л. 5,58  
Уч.-изд.л. 6,0. Тираж 100 экз. Заказ 65/7**

**Отпечатано с готового оригинала-макета в типографии  
Казанского (Приволжского) федерального университета**

**420008, Казань, ул. Профессора Нужина, 1/37  
Тел. (843) 231-53-59, 292-65-60**