

Федеральное агентство по образованию

Томский государственный
архитектурно-строительный университет

РЕШЕНИЕ ИНЖЕНЕРНЫХ ЗАДАЧ БЖД С ПРИМЕНЕНИЕМ TURBO BASIC

Методические указания к лабораторным занятиям

Составитель М.В. Анисимов



Томск – 2008

Решение инженерных задач БЖД с применением Turbo Basic: методические указания / Сост. М.В. Анисимов. – Томск.: Изд-во Том. гос. архит.-строит. ун-та, 2008. – 19 с.

Рецензент д.т.н., С.А. Карауш

Редактор Е.Ю. Глотова

Методические указания к лабораторным занятиям по дисциплинам ЕН.В.2 «Численные методы решения задач БЖД» для студентов специальности 280102 «Безопасность технологических процессов и производств» очной формы обучения.

Печатаются по решению методического семинара кафедры охраны труда и окружающей среды, протокол № 10 от 05.10.2008 г.

Утверждены и введены в действие проректором по учебной работе В.В. Дзюбо

с 01.11.2008
до 01.11.2013

Оригинал-макет подготовлен автором

Подписано в печать 01.11.2008.

Формат 60х90/16. Бумага офсет. Гарнитура Таймс,
печать офсет. Уч.-изд. л. 1,0. Тираж 100 экз. Заказ №

Изд-во ТГАСУ, 634003, г. Томск, пл. Соляная, 2.

Отпечатано с оригинал-макета в ООП ТГАСУ.

634003, г. Томск, ул. Партизанская, 15.

ОГЛАВЛЕНИЕ

Введение	4
1. Рекомендуемый порядок работы на ПЭВМ при создании и выполнении программ на языке Турбо-Бейсик.....	4
2. Основные программные операторы Турбо-Бейсик	6
3. Пример решения задачи на ПЭВМ с выполнением программы на языке Турбо-Бейсик	15
4. Задание для лабораторной работы.....	17
Контрольные вопросы.....	19
Список рекомендуемой литературы.....	19

ВВЕДЕНИЕ

Использование различных языков программирования для численного решения различных задач, в том числе задач БЖД с использованием ПЭВМ, в настоящее время весьма широко распространено.

На сегодняшний день существует большое количество языков программирования, таких как Pascal, C++, Visual Basic и т. д. Все они основаны на использовании различных операторов, написание которых может различаться.

Целью данного методического указания является знакомство обучающихся с программной оболочкой Turbo Basic (Турбо-Бейсик) программный язык которой достаточно прост в изучении, но вместе с тем позволяет реализовывать на ПЭВМ задачи практически любой сложности.

Современный специалист в области охраны труда и безопасности жизнедеятельности должен уметь решать различные задачи и, кроме того, использовать возможности вычислительной техники.

1. РЕКОМЕНДУЕМЫЙ ПОРЯДОК РАБОТЫ НА ПЭВМ ПРИ СОЗДАНИИ И ВЫПОЛНЕНИИ ПРОГРАММ НА ЯЗЫКЕ ТУРБО-БЕЙСИК

Как любой другой программный продукт, Турбо-Бейсик имеет свои правила пользования. Ниже перечислен порядок работы на ПЭВМ при создании и выполнении программ на языке Турбо-Бейсик:

1. Войти в Турбо-Бейсик. Для этого набрать `tb` в командной строке и нажать клавишу «Ввод». Еще раз нажать «Ввод».

В верхней строке экрана высвечивается главное меню.

2. Для первоначального ввода или исправления программы войти в редактор (пункт меню EDIT).

3. После того, как программа составлена (или внесены исправления), ее необходимо записать в память на диск или дискету (пункт главного меню FILE, подпункт SAVE или клавиша F2).

Примечание: Рекомендуется после набора каждой 10 строк записывать программу на диск (простым нажатием клавиши F2), чтобы в случае сбоя ПЭВМ программа не пропала.

4. Выполнить программу (пункт RUN главного меню), переход на большой экран для анализа результатов – команда ALT/F5.

5. Выйти из Турбо-Бейсика (пункт FILE, подпункт EXIT или команда ALT/X).

Примечание: В каждом режиме в нижней строке экрана указано назначение функциональных клавиш. Например, если курсор находится в главном меню, то нажатие клавиши F1 (Help) вызывает появление на экране сведений по использованию соответствующего режима, переход к следующей странице – «Ввод». Нажатие F5 (Zoom) вызывает раскрытие текущего окна на весь экран, повторное нажатие F5 – возврат в исходное положение; F6 (Next) – переход к следующему окну; F7 (Goto) – переход в режим редактирования; ALT/X (Exit) – выход из Турбо-Бейсика.

В режиме редактирования: F1 – Help (см. выше), F2 – Save (запись на диск или дискету);

F3 – New (загрузка новой программы, имя вводится в появившееся окно) и т. д.

2. ОСНОВНЫЕ ПРОГРАММНЫЕ ОПЕРАТОРЫ ТУРБО-БЕЙСИК

Кроме основных программных операторов Турбо-Бейсик включает в себя и различные общеизвестные математические действия, без которых невозможно решение никаких серьезных математических задач.

Ниже приведен список и показано написание математических действий в Турбо-Бейсик:

1. Возведение в степень (^).
2. Умножение, деление (*, /).
3. Деление нацело (\), например, $5 \setminus 2$ равно 2.
4. Сложение, вычитание (+, -)

Стандартные функции, используемые в Турбо-Бейсике, приведены в табл. 1.

Таблица 1

Стандартные функции Турбо-Бейсика

Запись в Турбо-Бейсике	Математическая запись
ABS(x)	$ x $
SIN(x)	$\sin x$
COS(x)	$\cos x$
TAN(x)	$\operatorname{tg} x$

Продолжение табл. 1

ATN(x)	$\arctg x$
EXP(x)	e^x
LOG(x)	$\ln x$
LOG2(x)	$\log_2 x$
LOG10(x)	$\lg x$
INT(x)	целая часть x
SGN(x)	знак x (+1 при $x > 0$, 0 при $x = 0$, -1 при $x < 0$)
SQR(x)	квадратный корень из x
LEN(a)	количество символов a
LEFT(a,n)	выбирает из a , n символов, начиная с первого
MID(a,m,n)	выбирает из a , n символов, начиная с m -го
RIGHT(a,n)	выбирает n символов a , начиная с последнего
STR(x)	преобразует число к символьному виду
VAL(a)	определяет числовое значение a

ASC(a)	определяет код первого символа <i>a</i>
TAB(n)	указывает номер позиции для вывода следующего элемента в списке вывода оператора PRINT
SPC(n)	вывод <i>n</i> пробелов, используется в списке вывода оператора PRINT
RND[(x)]	выдает случайное число из интервала (0,1)

Примечания: [] – обозначают необязательный параметр. Если $x = 0$, то повторяется последнее сгенерированное число; если $x < 0$, то для каждого x генерируется новая последовательность случайных чисел; если $x > 0$ или отсутствует, то генерируется очередное случайное число, но при повторном запуске последовательность повторяется.

Можно использовать оператор RANDOMIZE (перед первым использованием функции RND), чтобы при новом запуске программы получать новую последовательность случайных чисел.

Операции отношения: <, <=, >, >=, <>.

Левая и правая части отношения – это числовое или символьное выражение (в последнем случае сравниваются числовые коды символов). Если отношение удовлетворяется (является истинным), то его значение равно 1, если не удовлетворяется (является ложным), то его значение 0. Отношения могут быть использованы также в числовых выражениях.

Оператор очистки экрана **CLS**. Обычно располагается в начале программы для очистки экрана перед выводом результатов программы.

Оператор ввода **INPUT** имеет вид:

INPUT [;] [поясняющий текст], список переменных. Поясняющий текст заключается в кавычки. Переменные в списке отделяются друг от друга запятыми. «;» после **INPUT** означает, что курсор останется на той же строке, когда будет нажат «Ввод».

Оператор вывода **PRINT** имеет вид:

PRINT [список вывода].

Элементы списка разделяются «,» или «;» или пробелом. В первом случае вывод следующего элемента начинается в новую зону (строка экрана условно разделена на 5 зон по 14 позиций каждая), во втором – через пробел (для чисел) или непосредственно вслед (для символьных строк) за предыдущим элементом.

Элементом списка может быть имя переменной, константа (символьная заключается в кавычки) или выражение (перед выводом вычисляется).

Оператор вывода **PRINT USING** имеет вид:

PRINT USING формат; список вывода.

Формат задается в виде символьной строки (заключенной в кавычки), задающей форму вывода каждого элемента списка по следующим правилам для чисел: # – для каждой цифры, ^^^ – для вывода в показательной форме. Остальные символы обозначают сами себя.

Логические операции: **AND** (логическое и), **OR** (логическое или), **XOR** (исключающее или), **EQV** (эквивалентность), **IMP** (импликация).

Оператор безусловного перехода имеет вид:

GOTO метка.

Условный оператор имеет вид:

IF условие **THEN** оператор [**ELSE** операторы].

Например,

IF $a < b$ THEN $t = 15$: $V = 16$ ELSE $t = 17$

В качестве условия может использоваться целое выражение, которое интерпретируется как FALSE, если его значение равно 0, и TRUE, если не равно 0.

Если после **THEN** или после **ELSE** располагается целая группа операторов, то можно использовать **IF** блок, который имеет следующую структуру:

IF условие THEN

операторы

ELSE

оператор

END IF

При этом ELSE и операторы за ним могут отсутствовать, т. е. возможна конструкция:

IF условие THEN

операторы

END IF

Если после **ELSE** необходима проверка условия, то используется оператор ELSEIF:

IF условие THEN

оператор

ELSEIF условие THEN

операторы

ELSE

операторы

END IF

Оператор конца **END** используется для окончания выполнения программы. Может использоваться в программе несколько раз. Используется также с **IF, SUB, DEF, SELECT**.

Операторы цикла всегда идут в связке (**FOR** и **NEXT**).

Например,

FOR i=1 to 10

операторы

NEXT i

FOR i=1 to 10

FOR j=1 to 20

операторы

NEXT j: NEXT i

Подпрограмма – помеченная последовательность операторов, заканчивающаяся оператором **RETURN**, выполняется, когда достигнут **GOSUB**.

Пример:

операторы

GOSUB aa

операторы

END

aa:

операторы

RETURN

Оператор **RETURN** осуществляет возврат к оператору, непосредственно следующему за **GOSUB**.

Функция может быть определена одной строкой (оператор-функция) или несколькими строками (подпрограмма-функция).

В графическом режиме экран рассматривается как решетка, точки в узлах которой могут быть включены (белым или другим возможным цветом) или выключены. Точка идентифицируется координатами x, y ; x – номер позиции по горизонтали (от 0 до 319 или до 639 в зависимости от режима), y – номер позиции по вертикали (от 0 до 199 или до 349 в зависимости от режима). Точка с координатами 0,0 находится в верхнем левом углу экрана.

Графический режим устанавливается (или отменяется) оператором **SCREEN**.

SCREEN $i[j]$

$i = 0, j = 0$ (черно-белый текстовый режим).

$i = 0, j = 1$ (цветной текстовый режим).

$i = 1, j = 0$ (цветная графика при средней разрешающей способности (320 позиций по горизонтали, 200 по вертикали)).

$i = 1, j = 1$ (черно-белая графика при средней разрешающей способности).

$i = 2$ (черно-белая графика при высокой разрешающей способности (640 позиций по горизонтали, 350 по вертикали)).

После выполнения оператора **SCREEN** точка устанавливается в середину экрана.

Оператор **COLOR i** устанавливает цвет фона экрана и палитру (из трех цветов) для окрашивания точек (для текстового режима – еще и цвет рамки).

Существует две палитры цветов: 0 и 1, следующего состава:

Палитра. Палитра 1

1 – зеленый 1 – голубой

2 – красный 2 – сиреневый

3 – желтый 3 – белый

Для фона можно выбрать один из 16 цветов (0 – черный, 1 – синий, 2 – зеленый, 3 – бирюзовый, 4 – красный, 5 – лиловый, 6 – коричневый, 7 – белый, 8 – серый, 9 – голубой, 10 – ярко-зеленый, 11 – ярко-бирюзовый, 12 – ярко-красный, 13 – ярко-лиловый, 14 – желтый, 15 – ярко-белый).

Оператор **COLOR** имеет вид:

COLOR i, j,

где i – цвет фона, j – номер палитры, k – цвет рамки.

Например, операторы

SCREEN 1,0

COLOR 9,0

устанавливают цветной графический режим при средней разрешающей способности; цвет фона – голубой; последующие операторы должны использовать цвета из 0-й палитры.

Оператор **LINE 1, 2** позволяет начертить отрезок прямой, указав его начало (x_1, y_1), конец (x_2, y_2) и цвет k:

LINE [(x₁, y₁)] (x₂, y₂), k

Если начальная точка отсутствует, то линия будет прочерчена, начиная с точки, в которой остановился предыдущий оператор.

С помощью оператора LINE можно также чертить и закрашивать прямоугольники:

LINE (x₁, y₁) - (x₂, y₂), [k], B[F]

Точки (x₁, y₁) и (x₂, y₂) трактуются при этом как левая нижняя и правая верхняя вершины прямоугольника. Если цвет k отсутствует, то прямоугольник будет начерчен 3-м цветом палитры. При наличии параметра F площадь внутри прямоугольника будет закрашена. Например, оператор LINE (80, 175) - (100, 25), 2, BF закрашивает красным цветом прямоугольник, расположенный между 80-й и 100-й позициями по горизонтали и 175-й и 25-й позициями по вертикали.

Оператор **CIRCLE** позволяет строить окружности, дуги, секторы и эллипсы.

Оператор CIRCLE (x,y), r, k чертит окружность радиуса r с центром в точке (x, y) цветом k.

После вычерчивания окружности последней использованной точкой считается ее центр.

Оператор **BEEP** вызывает звуковой сигнал определенной частоты и длительности 0,25 секунды. Используется, когда нужно привлечь внимание при выполнении какой-либо части программы.

Оператор **SOUND** вызывает звуковой сигнал частоты x (от 37 до 32767 Гц) и длительности в «тиках» (1/18 секунды).

SOUND x, y

Оператор **SOUND** с нулевой длительностью прерывает работу предыдущего оператора **SOUND**, даже если тот не отзвучал до конца. Используется для специальных звуковых эффектов (сирена и пр.).

Это, конечно, далеко не полный перечень всех операторов Турбо-Бейсика, но для выполнения инженерных расчетов, как правило, его бывает достаточно.

3. ПРИМЕР РЕШЕНИЯ ЗАДАЧИ НА ПЭВМ С ВЫПОЛНЕНИЕМ ПРОГРАММЫ НА ЯЗЫКЕ ТУРБО-БЕЙСИК

В качестве примера рассмотрим решение инженерной задачи обеспечения нормативной освещенности в рабочем помещении.

Для определения количества светильников в помещении определяем световой поток, падающий на поверхность, по формуле

$$F = \frac{E \cdot K \cdot S \cdot Z}{n}, \quad (1)$$

где F – рассчитываемый световой поток, лм;

E – нормированная минимальная освещенность, лк (работа программиста относится к разряду точных работ, поэтому минимальная освещенность $E = 300$ лк, при газоразрядных лампах);

S – площадь освещаемого помещения, м²;

Z – отношение средней освещенности к минимальной (обычно принимается равным 1.1 – 1.2);

K – коэффициент запаса, учитывающий уменьшение светового потока лампы в результате загрязнения светильников в процессе эксплуатации (обычно принимается $K = 1.5$);

n – коэффициент использования (выражается отношением светового потока, падающего на расчетную поверхность, к суммарному потоку всех ламп и исчисляется в долях единицы; зависит от характеристик светильника, размеров помещения, окраски стен и потолка, характеризуемых коэффициентами отражения от стен (P_c) и потолка (P_n)). Обычно $P_c = 30\%$, $P_n = 50\%$, при этих значениях $n = 0,28$.

Необходимое количество ламп:

$$N = \frac{F}{F_{\text{л}}}, \quad (2)$$

где N – определяемое число ламп, шт.; F – световой поток, лм.; $F_{\text{л}}$ – световой поток лампы, лм., (стандартная люминесцентная лампа ЛБ40-1 имеет $F_{\text{л}} = 4320$ лк).

В соответствии с расчетными зависимостями (1 – 2) и информацией, представленной в данном методическом указании, выполняем программу расчета нормативной освещенности в заданном помещении.

Ниже представлен листинг программы, выполненной на языке программирования Турбо-Бэйсик, которая позволяет определить необходимое количество светильников типа ЛБ 40-1, руководствуясь существующими нормами, также с ее помощью можно рассчитать оптимальное количество компьютеров в помещении с заданными геометрическими размерами.

Необходимое количество светильников типа ЛБ 40-1, допустимое количество ПК в помещении

```
cls
input "Введите длину помещения, м",a
input " Введите ширину помещения, м ",b
input " Введите высоту помещения, м ",h
s=a*b*h
F=1767.8*s
N=F/4320
```



```

N1=int(N)
color 3
print "Необходимое количество светильников типа ЛБ
40-1 для данного помещения"N1",шт"
m=s/15
m1=int(m)
color 5
print " Допустимое количество компьютеров для дан-
ного помещения "m1" шт"

```

4. ЗАДАНИЕ ДЛЯ ЛАБОРАТОРНОЙ РАБОТЫ

Разработать, пользуясь данным методическим указанием и примером выполнения программы, программу решения задач, представленных в указанных методических указаниях, которые выдаются преподавателем.

В табл. 2 приведены названия методических указаний, в которых содержатся задачи, для их последующей реализации в виде программы, выполненной на языке Турбо-Бейсик. Выбор МУ производится студентом по его варианту.

Таблица 2

Данные для составления блок-схемы

№ вари- анта	Название МУ с приведенными в них задачами для программирования на языке Турбо-Бейсик
1	Обеспечение параметров микроклимата в зданиях за счет естественной вентиляции
2	Формирование микроклимата в зданиях различного назначения за счет применения систем отопления
3	Расчет мощности систем кондиционирования общест- венных зданий

№ варианта	Название МУ для составления блок-схемы указанных в ней задач
4	Теплообмен человека в условиях искусственного микроклимата
5	Обеспечение параметров микроклимата в зданиях за счет естественной вентиляции
6	Формирование микроклимата в зданиях различного назначения за счет применения систем отопления
7	Расчет мощности систем кондиционирования общественных зданий
8	Теплообмен человека в условиях искусственного микроклимата
9	Обеспечение параметров микроклимата в зданиях за счет естественной вентиляции
10	Формирование микроклимата в зданиях различного назначения за счет применения систем отопления
11	Расчет мощности систем кондиционирования общественных зданий
12	Теплообмен человека в условиях искусственного микроклимата
13	Обеспечение параметров микроклимата в зданиях за счет естественной вентиляции
14	Формирование микроклимата в зданиях различного назначения за счет применения систем отопления
15	Расчет мощности систем кондиционирования общественных зданий
16	Теплообмен человека в условиях искусственного микроклимата

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какой оператор Turbo Basic производит очистку экрана?
2. Какой оператор Turbo Basic служит для ввода в программу переменных?

3. Какой оператор Turbo Basic служит для вывода из программы результатов вычисления?
4. Какие операторы Turbo Basic являются операторами цикла?
5. Какой оператор Turbo Basic является оператором перехода?
6. Какой оператор Turbo Basic является оператором задания цвета текста?

СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1. Репин, С.В. Математические методы обработки статистической информации с помощью ЭВМ / С.В. Репин. – Минск: Университетское, 1990. – 127 с.
2. Вержбицкий, В.Н. Основы численных методов / В.Н. Вержбицкий. – М. : Высшая школа, 2005. – 840 с.
3. Введение в математическое моделирование: учебное пособие / В.Н. Ашихмин, М.Г. Бояршинов, М.Б. Гитман [и др.]. – М.: Интернет Инжиниринг, 2000. – 336 с.
4. Аляев, Ю. А./ Алгоритмизация и языки программирования Pascal, C++, Visual Basic: учебно-справочное пособие / Ю. А. Аляев, О. А. Козлов. – М.: Финансы и статистика, 2002. — 320 с.