

Задачи и упражнения по программированию



МАШИНОСТРОЕНИЕ,
ПРИБОРОСТРОЕНИЕ
И СВЯЗЬ



Задачи и упражнения по программированию

1 КНИГА

МАШИНОСТРОЕНИЕ,
ПРИБОРОСТРОЕНИЕ
И СВЯЗЬ

Издание 2-е, дополненное

Под редакцией
доктора технических наук,
профессора
А. Я. САВЕЛЬЕВА



Москва «Высшая школа» 1989

ББК 22.18
3-15
УДК 681.3.06

Рекомендовано к изданию
Государственным комитетом СССР
по народному образованию

Э.Н. Самохвалов, Ю.Н. Филиппович, Г.И. Ревунков

Рецензенты: канд. техн. наук А.Б. Николаев (МАДИ),
И.Н. Дмитриевская (ПТУ № 148, Москва)

Задачи и упражнения по программированию: В 5-ти кн.: Практик.
3-15 пособие / Под ред. А.Я. Савельева. Кн. 1. Машиностроение, приборостроение и связь / Э.Н. Самохвалов, Ю.Н. Филиппович, Г.И. Ревунков. — 2-е изд., доп. — М.: Высш. шк., 1989. — 96 с.: ил.

ISBN 5-06-000309-4

Книга содержит материалы для практического изучения приемов разработки алгоритмов типовых задач по программированию на языке БЕЙСИК алгоритмов различных структур.

Второе издание (1-е — в 1986 г. "Сборник задач и упражнений по программированию") дополнено задачами на применение машинной графики.

Для учащихся профтехучилищ. Может быть использовано широким кругом читателей, желающих овладеть приемами программирования.

З 2404010000 (4307000000) — 107 65-89
052 (01) — 89

ББК 22.18
517.8

Учебное издание

**Самохвалов Эдуард Николаевич
Филиппович Юрий Николаевич
Ревунков Георгий Иванович**

ЗАДАЧИ И УПРАЖНЕНИЯ ПО ПРОГРАММИРОВАНИЮ

К н и г а 1. Машиностроение, приборостроение и связь

Заведующая редакцией *С.В. Никитина*

Редактор *В.И. Мучкина*

Художники *Ю.М. Аратовский* и *С.Ю. Вериченко*

Художественные редакторы *В.И. Мешалкин* и *В.Г. Пасичник*

Технический редактор *И.А. Балелина*

Корректор *Р.К. Косинова*

Операторы *О.М. Кузьмина, Г.А. Шестакова*

ИБ № 7894

Изд. № ЭГ — 230. Сдано в набор 16.11.88. Подп. в печать 14.12.88. Формат 60×90¹/₁₆.
Бум. офс. № 2. Гарнитура Пресс-Роман. Печать офсетная. Объем 6 усл. печ. л. 6,37 усл.
кр.-отт. 5,33 уч.-изд. л. Тираж 120 000 экз. Зак. № 6 Цена 25 коп.
Издательство "Высшая школа", 101430, Москва, ГСП-4, Неглинная ул., д. 29/14.

Набрано на наборно-пишущих машинах издательства.

Отпечатано на Ярославском полиграфкомбинате "Союзполиграфпрома" при Государственном комитете СССР по делам издательств, полиграфии и книжной торговли.
150014, Ярославль, ул. Свободы, 97.

ISBN 5-06-000309-4

© Издательство "Высшая школа", 1986
© Издательство "Высшая школа", 1989,
с изменениями

ПРЕДИСЛОВИЕ

С 1985/86 учебного года в общеобразовательных школах и профессионально-технических училищах введено изучение нового учебного курса "Основы информатики и вычислительной техники", с помощью которого решается одна из важнейших задач реформы средней школы: вооружение учащихся знаниями и навыками по использованию современной вычислительной техники. Обеспечение всеобщей компьютерной грамотности учащейся молодежи — проблема огромного государственного, политического и социального масштаба, имеющая последующее влияние на совершенствование среднего школьного и профессионально-технического образования.

Современная цивилизация немыслима без электронно-вычислительной техники. Массовое овладение вычислительной техникой становится важным фактором ускорения научно-технического прогресса. Какую бы область мы ни взяли — промышленность, транспорт, сельское хозяйство, сферу быта, — с каждым годом роль ЭВМ становится все более явной, ощутимой. Процесс компьютеризации будет происходить и дальше, причем все нарастающими темпами.

Потенциал вычислительной техники в нашей стране сейчас необычайно высок. Отечественная промышленность освоила широкий выпуск универсальных электронных вычислительных машин Единой системы (ЕС ЭВМ), представляющих собой ряд цифровых ЭВМ различной производительности, достигающей в модели ЕС-1066 12,5 млн. операций/с, имеющих разнообразный набор периферийных устройств, полную техническую и программную совместимость. Большое распространение приобрел класс мини-ЭВМ (СМ ЭВМ), ориентированный на применение в системах управления различного назначения, на автоматизацию научных исследований и проектирования. Значительно возросло производство микропроцессоров и микроЭВМ, предназначенных для массового использования в самых различных областях.

Решениями XXVII съезда КПСС предусмотрено дальнейшее развитие и внедрение в народное хозяйство современной электронно-вычислительной техники. В Основных направлениях экономического и социального развития СССР на 1986—1990 годы и на период до 2000 года говорится: "Организовать массовый выпуск персональных компьютеров. Обеспечить рост объема производства вычислительной техники в 2—2,3 раза, повысить ее надежность. Высокими темпами наращивать масштабы применения со-

временных высокопроизводительных электронно-вычислительных машин всех классов”*

В условиях научно-технической революции информатика и ЭВМ глубоко проникают во все сферы человеческой деятельности, становятся мощной производительной силой. Велика роль ЭВМ в процессе построения материально-технической базы коммунизма, интенсификации современного общественного производства, технического и технологического перевооружения и автоматизации его, повышения производительности труда. Поэтому от уровня и качества компьютерной грамотности молодого поколения существенно зависит научный, технический и производственный потенциал страны.

Настоящее учебное пособие не заменяет соответствующего учебника по курсу ”Основы информатики и вычислительной техники”, а ставит цель — закрепить на практике основные разделы курса, научить учащихся ставить и решать задачи на ЭВМ по своей непосредственной специальности.

Автоматическая обработка информации на ЭВМ включает в себя три **основных этапа** от постановки задачи до получения окончательного результата: алгоритмизация, программирование и решение задачи на ЭВМ.

Алгоритмизация — это разработка алгоритма решения задачи, в которую входит изучение процесса или объекта, установление основных связей и зависимостей между его характеристиками, математическое описание алгоритма задачи.

Программирование — это перевод полученного алгоритма на язык, который понимает ЭВМ (БЕЙСИК, АЛГОЛ, ФОРТРАН, ПАСКАЛЬ и др.), т.е. построение программы на одном из языков программирования.

Решение задачи на ЭВМ — это реализация полученной программы на ЭВМ для выдачи результатов в удобной для пользователя форме.

В результате изучения курса ”Основы информатики и вычислительной техники” и решения задач данного сборника учащиеся ПТУ должны освоить основы алгоритмизации задач своей специальности, приемы их программирования на простейшем алгоритмическом языке БЕЙСИК, научиться отлаживать программы и решать задачи на современных микроЭВМ типа диалоговых вычислительных комплексов (ДВК).

В гл. 1 рассматриваются вопросы первого этапа обработки информации на ЭВМ — *алгоритмизации*. Приводится математическая формулировка типовых задач по профессиям. Каждая задача доводится до схемы алгоритма, который в дальнейшем должен быть запрограммирован на алгоритмическом языке.

В гл. 2 изучается второй этап — *программирование* на алгоритмическом языке БЕЙСИК. Приводятся краткие сведения по конструкциям языка, сопровождающиеся подробными примерами с решениями. После них дается набор упражнений для самостоятельного решения. Изучив приемы

*Материалы XXVII съезда Коммунистической партии Советского Союза. М., Политиздат. 1986. С. 281.

программирования на алгоритмическом языке, учащиеся должны написать программы типовых задач, алгоритмы которых были рассмотрены в гл. 1. Задачи на составление программ, как правило, имеют несколько решений. В этих случаях в качестве ответа приводится одно из возможных решений. К некоторым задачам приводятся численные ответы, получить которые можно, только составив программу задачи и выполнив ее на ЭВМ.

В Приложении 1 даны сведения по практическому использованию персональных компьютеров.

Пояснения к решению некоторых задач приводятся сразу же после их постановки с целью использования изложенных приемов при решении учащимися последующих задач самостоятельно. Задачи, отмеченные символом ●, имеют ответ.

Авторы выражают благодарность сотрудникам Всесоюзного научно-методического центра профтехобразования канд. пед. наук Э.Ю. Крассу, В.М. Оксману, рецензентам — преподавателю ПТУ-148 Москвы И.Н. Дмитриевской и канд. техн. наук, доц. А.Б. Николаеву за ценные замечания и советы, сделанные при подготовке данного сборника.

Авторы

ГЛАВА ПЕРВАЯ

АЛГОРИТМИЗАЦИЯ ТИПОВЫХ ЗАДАЧ ПО ПРОФЕССИЯМ

§ 1. АЛГОРИТМЫ И ПРОГРАММА. СПОСОБЫ ОПИСАНИЯ АЛГОРИТМОВ

Этап алгоритмизации включает в себя математическую формулировку задачи и разработку алгоритма ее решения.



Под алгоритмом понимают понятное и точное предписание (указание) исполнителю (человеку или ЭВМ) совершить последовательность действий, направленных на решение поставленной задачи.

Отметим основные свойства алгоритмов: *точность* — установление четкого порядка действий, чтобы, выполнив очередную команду, исполнитель точно знал, какую команду надо выполнять следующей; *массовость* — возможность с помощью одного алгоритма решать не только индивидуальную задачу, но и серии однотипных задач с различными исходными данными; *формальность* — возможность человека (программиста), даже не знающего физического смысла задачи, правильно составить программу по данному алгоритму.

Будем называть *программой* запись алгоритма на языке, воспринимаемом ЭВМ. Этот язык включает конечный набор инструкций, называемых иначе командами машины. Программу, по которой ЭВМ проводит вычисления согласно заданному алгоритму, можно представить в виде последовательности инструкций — команд, каждая из которых вызывает выполнение некоторого определенного действия ЭВМ. Таким образом в ЭВМ физически реализуется алгоритм решения задачи.

Прежде чем составить программу на языке машины или алгоритмическом языке, необходимо описать алгоритм задачи. Основное назначение этой записи — облегчить последующий процесс программирования. Для описания алгоритмов можно использовать различные способы. *Рассмотрим пример*: вычислить значение функции $y = x^2$. Можно предложить такую последовательность действий, выполнение которых обеспечивает получение искомого результата.

Шаг 1. Взять (задать) конкретное значение аргумента x .

Шаг 2. Умножить x на x , вычислив x^2 .

Шаг 3. Напечатать полученный результат $y = x^2$.

В приведенном примере дано простое *словесное* описание алгоритма.

Для наглядного изображения алгоритмов используют также *графические описания в виде структурных схем (блок-схем)*. Структурная схема алгоритма примера, рассмотренного выше, имеет вид, показанный на рис. 1, и представляет собой последовательность графических контуров символов, соединенных между собой линиями со стрелками. Направления переходов от блока к блоку сверху вниз и слева направо считаются стандартными и могут стрелками не отмечаться. Внутри символов словами или с помощью формул указывается выполняемая ими функция (вычислить, напечатать и т.п.). В разрыве верхней левой линии контура символа ставится его порядковый номер в схеме.

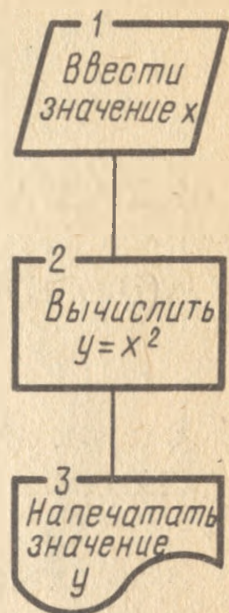


Рис. 1

Разработка машинного алгоритма реальных задач предполагает увязывание последовательности вычислительных действий, управляющих связей и проверок логических условий, операций по вводу и определению начальных значений действующих в программе переменных, а также по выводу получаемых результатов решения задачи.

Для наглядного графического изображения *схем алгоритмов* используются символы ГОСТ 19003—80; наиболее часто употребляемые символы действий приведены в табл. 1.

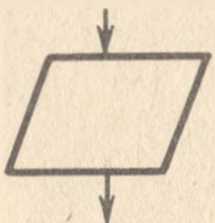
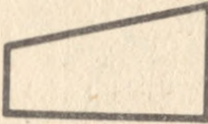
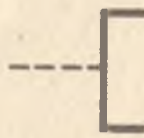


Алгоритмы задач составляют из типовых структурных элементов, которые имеют линейную, разветвленную или циклическую структуру.

В качестве типовых примеров и задач по профессиям в настоящем сборнике приведены задачи проведения прочностных расчетов заклепочных, сварных, клиновых, шпоночных и резьбовых соединений, расчеты деталей механизмов приборов и устройств, расчеты электрических цепей постоянного тока, катушек индуктивности, трансформаторов и радиотехнических фильтров.

Т а б л и ц а 1. Основные блоки схем алгоритмов

Наименование	Обозначение	Функции
Пуск – останов		Начало, конец и останов, вход и выход в подпрограммах
Процесс		Выполнение одной или группы операций

Наименование	Обозначение	Функции
Решение		Проверка условия. Выбор направления выполнения алгоритма
Модификация		Начало цикла
Ввод – вывод		Ввод данных в ЭВМ или вывод из ЭВМ результатов обработки
Ручной ввод		Ввод данных с клавиатуры
Документ		Вывод, печать результатов на бумаге
Соединитель		Разрыв линий схемы алгоритма
Комментарий		Пояснения к схеме алгоритма, формулы

§ 2. РЕШЕНИЕ ЗАДАЧ С ИСПОЛЬЗОВАНИЕМ ТИПОВЫХ СТРУКТУР АЛГОРИТМОВ

Алгоритм линейной структуры

1. Составить схему алгоритмов определения количества заклепок однорядного заклепочного шва (рис. 2) из условия обеспечения их прочности на срез.

Математическая формулировка задачи

Количество заклепок определить по выражению

$$i = \frac{4P}{\pi d^2 [\tau]_{\text{ср}}}, \quad (1)$$

где i — количество заклепок; P — нагрузка, Н; $[\tau]_{\text{ср}}$ — допустимое напряжение на срез полос и заклепок; d — диаметр заклепок.

Округление результата для достижения необходимой прочности следует провести в сторону большего целого числа.

Решением задачи 1 является алгоритм линейной структуры, схема которого представлена на рис. 3. Алгоритм линейной структуры — простейший. Его символы изображены на схеме в той последовательности, в какой должны быть выполнены описываемые ими действия. Такой порядок выполнения действий называют естественным.

2. Составить схему алгоритма определения напряжения на смятие и напряжения на растяжение заклепок и полос одностороннего заклепочного шва (см. рис. 2).

Математическая формулировка задачи

Напряжение на смятие определить по формуле

$$\sigma_{\text{см}} = \frac{P}{id\delta}. \quad (2)$$

Напряжение на растяжение определить по выражению

$$\sigma_{\text{р}} = \frac{P}{(b-id)\delta}, \quad (3)$$

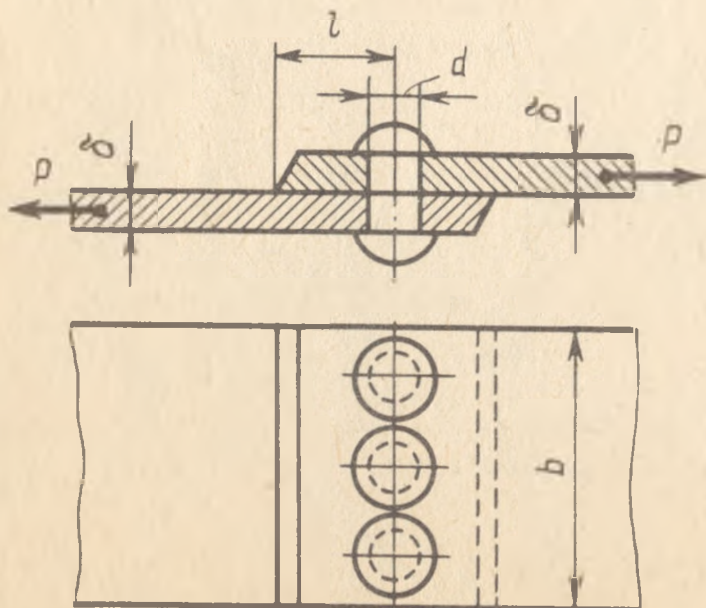


Рис. 2

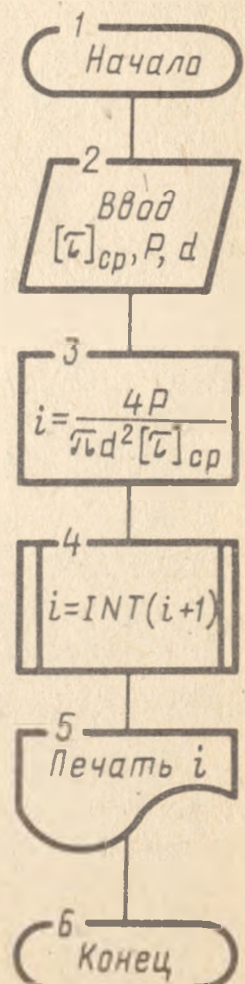


Рис. 3

где δ — толщина листа; b — ширина склепываемых полос. Остальные обозначения приведены в формуле (1).

3. Составить схему алгоритма определения расстояния заклепок l от поперечной кромки полос из условия прочности на срез.

Математическая формулировка задачи

Расстояние l определить по выражению

$$l \geq \frac{P}{id[\tau]_{\text{ср}}}. \quad (4)$$

Обозначения в формуле (4) те же, что и в формуле (1).

Вычисленное значение l округлить в большую сторону для обеспечения необходимой прочности шва.

4. Составить схему алгоритма расчета сопротивления проводов.

Математическая формулировка задачи

Сопротивление провода определить по формуле

$$R = \rho l / S, \quad (5)$$

где ρ — удельное сопротивление; l — длина провода; S — площадь сечения провода.

$$S = \pi d^2 / 4, \quad (6)$$

где d — диаметр провода.

Исходные данные: l , d , ρ .

Решением задачи 4 является алгоритм линейной структуры, схема которого представлена на рис. 4.

Алгоритм разветвленной структуры

5. Составить схему алгоритма поверочного расчета параметров однорядного заклепочного шва (см. рис. 2). Исходные данные: количество заклепок i , нагрузка P , толщина полос δ , ширина полос b .

Математическая формулировка задачи

$$\tau_{\text{ср}} = \frac{4P}{\pi id^2} \leq [\tau]_{\text{ср}}, \quad (7)$$

$$\sigma_{\text{см}} = \frac{P}{id\delta} \leq [\sigma]_{\text{см}}, \quad (8)$$

$$\sigma_{\text{р}} = \frac{P}{(b - id)\delta} \leq [\sigma]_{\text{р}}, \quad (9)$$

$$d = 2\delta \quad (10)$$

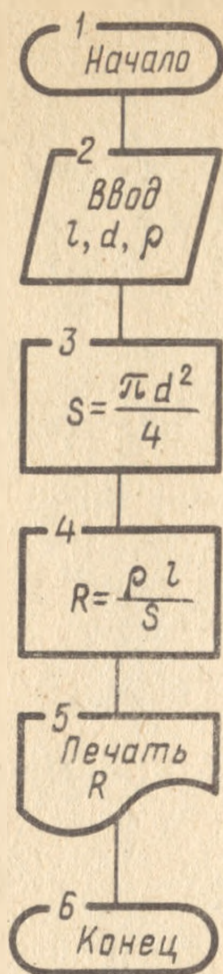


Рис. 4

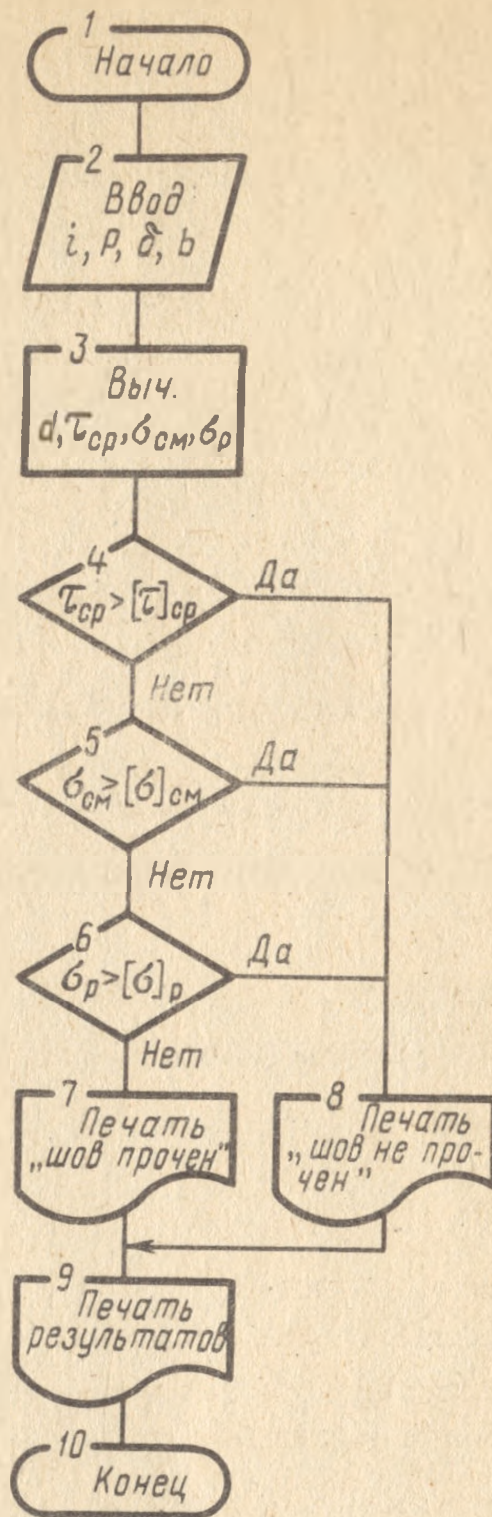


Рис. 5

Если обеспечивается прочность соединения на срез, смятие и растяжение, т.е. выдерживаются все соотношения (7)–(9), то выдается сообщение "ШОВ ПРОЧЕН". Если хотя бы одно из соотношений не соблюдается, то выдается сообщение "ШОВ НЕ ПРОЧЕН".

Решением этой задачи является алгоритм разветвленной структуры, схема которого представлена на рис. 5. Основная часть алгоритма разветвленной структуры – блоки проверки условия, и в зависимости от результата этих проверок в дальнейшем выполняются действия либо одной ветви, либо другой.

6. Для однорядного заклепочного шва, в котором заданы количество заклепок i , толщина δ и ширина b полос, рассчитать допустимые напряжения среза, смятия и растяжения, используя формулы (7)–(10). Определить, какая из нагрузок окажется максимальной. В выходном сообщении указать значение и тип этой нагрузки.

Математическая формулировка задачи

Преобразуя формулы (7)–(9), получим выражения для определения нагрузки при значениях:

допустимого напряжения среза

$$P_1 = \frac{1}{4} \pi i d^2 [\tau]_{\text{ср}}, \quad (11)$$

допустимого напряжения смятия

$$P_2 = i d \delta [\sigma]_{\text{см}}, \quad (12)$$

допустимого напряжения растяжения

$$P_3 = (b - i d) \delta [\sigma]_{\text{р}}. \quad (13)$$

Схема алгоритма разветвленной структуры, реализующего условия поставленной задачи, представлена на рис. 6.

7. Составить схему алгоритма проверки катушки провода, который предполагается использовать для изготовления реостата, имеющего сопротивление $R = 100$ Ом.

Математическая формулировка задачи

Определить требуемую длину провода l , используя формулы (5), (6).

Исходные данные: удельное сопротивление материала ρ , диаметр провода d . Рассчитав требуемую длину провода l , сравнить ее с длиной провода, имеющегося в наличии, $l_{\text{мах}}$:

$$l \leq l_{\text{мах}}. \quad (14)$$

Если условие (14) выполняется, выдается сообщение "ПРОВОД ПОДХОДИТ", если не выполняется, то выдается сообщение "ПРОВОД НЕ ПОДХОДИТ".

Решением этой задачи является алгоритм разветвленной структуры, схема которого приведена на рис. 7.

Алгоритмы циклической структуры

8. Составить схему алгоритма расчета допустимых значений нагрузки P в зависимости от числа заклепок i в однорядном заклепочном шве.

Исходные данные: допустимое напряжение на срез $[\tau]_{\text{ср}}$, диаметр заклепок d , пределы изменения числа заклепок i от 1 до 20.

Математическая формулировка задачи

Преобразуя формулу (1), получим расчетную формулу в виде

$$P = \frac{1}{4} \pi d^2 [\tau]_{\text{ср}} i. \quad (15)$$

Решением этой задачи является алгоритм циклической структуры,

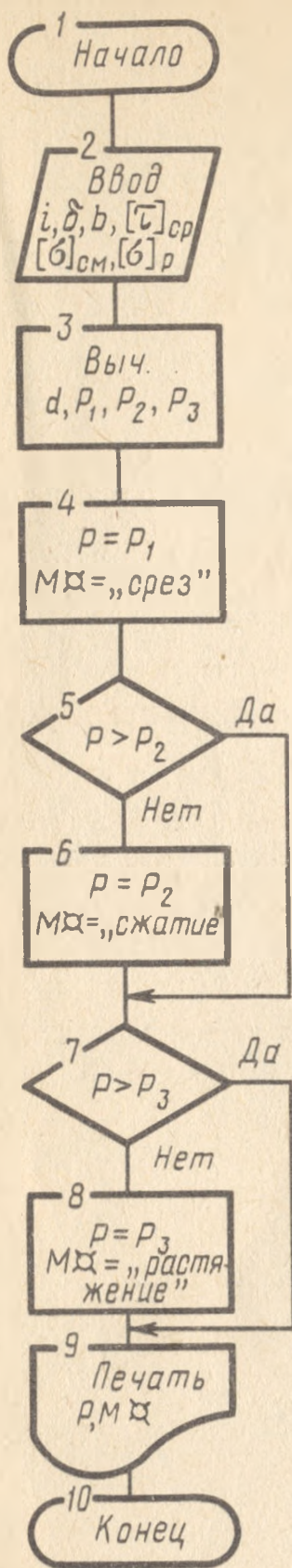


Рис. 6

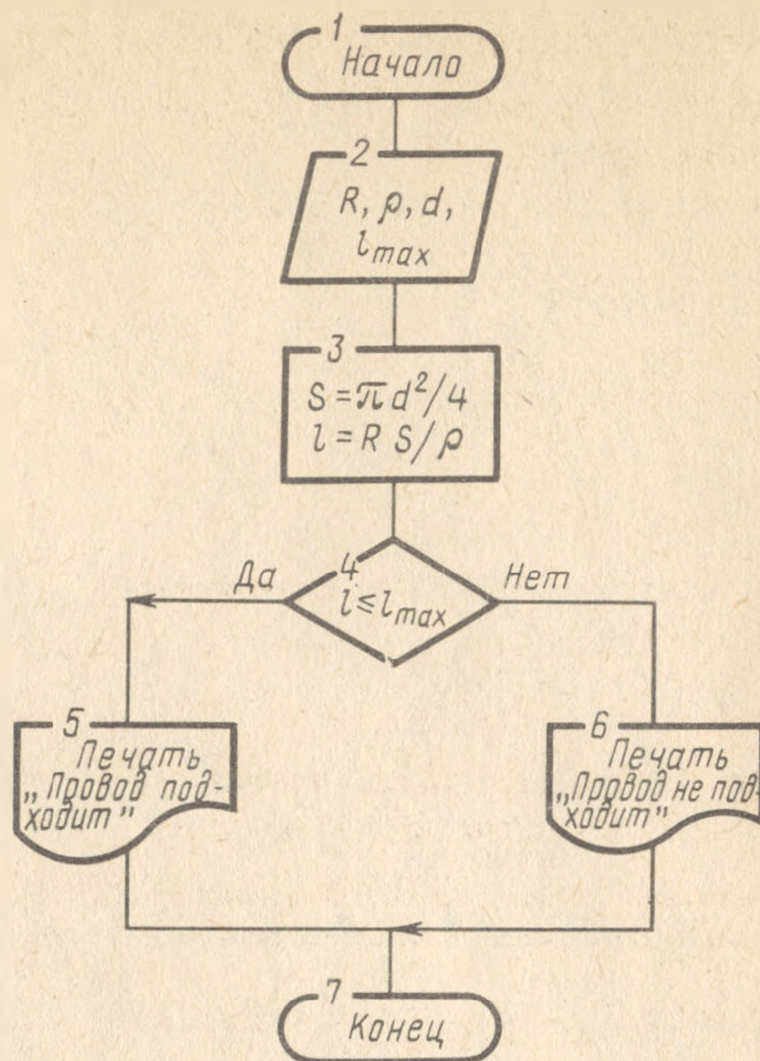


Рис. 7

схема которого представлена на рис. 8. Алгоритм циклической структуры позволяет описывать повторяющиеся действия.

9. Составить схему алгоритма определения допустимых значений исходных данных при расчете напряжения на смятие заклепок и полос однорядного заклепочного шва.

Математическая формулировка задачи

Напряжение на смятие $\sigma_{см}$ определяется по формуле (8).

Исходные данные: нагрузка P , толщина полос δ , количество заклепок i , изменяющиеся в пределах:

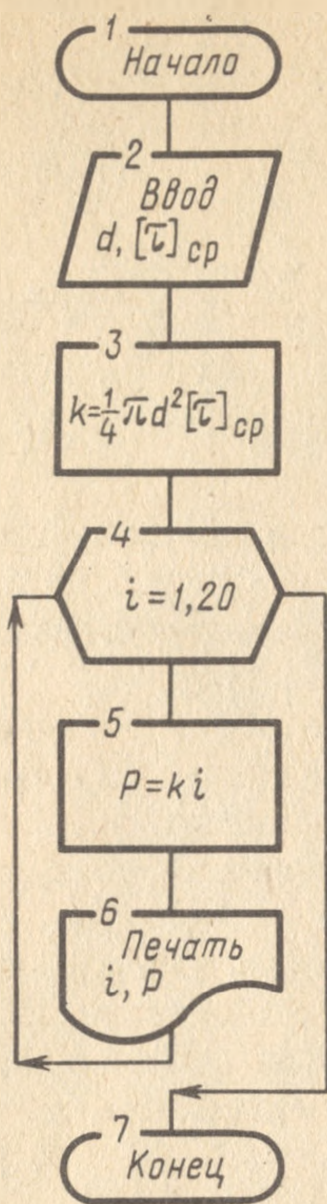


Рис. 8

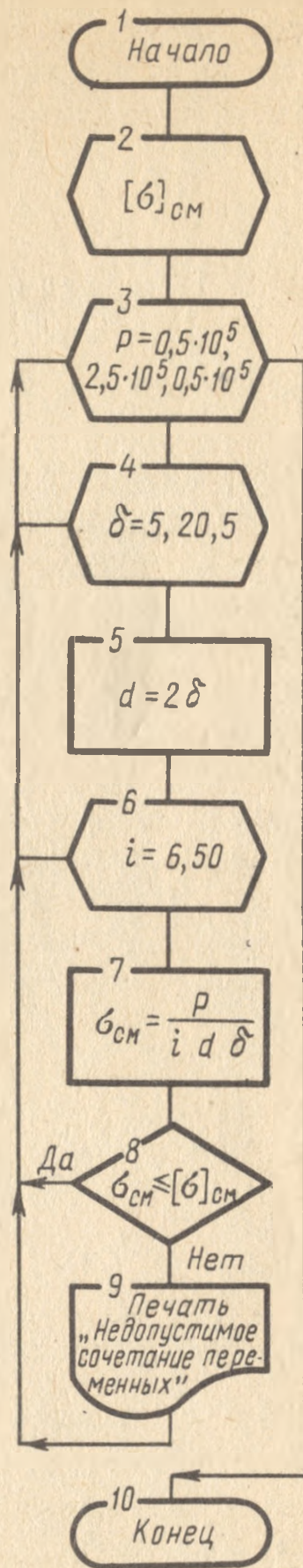


Рис. 9

P от $0,5 \cdot 10^5$ до $2,5 \cdot 10^5$ Н с шагом $0,5 \cdot 10^5$ Н;
 δ от 5 до 20 мм с шагом 5 мм;
 i от 6 до 50.

Допустимое напряжение на смятие $[\sigma]_{\text{см}} = 280$ МПа.

По результатам расчетов выдать сведения о недопустимых сочетаниях значений параметров исходных данных P , δ , i .

При расчетах использовать также формулу (10).

Решением этой задачи является алгоритм циклической структуры, схема которого представлена на рис. 9. Особенностью алгоритма является трехкратное использование циклов: для переменных P , δ и i .



ЗАДАЧИ ДЛЯ САМОСТОЯТЕЛЬНОГО СОСТАВЛЕНИЯ АЛГОРИТМОВ

10. Составить схему алгоритма определения допустимых значений исходных данных при расчете напряжения на растяжение [формулы (9, 10)] материала полос в опасном сечении заклепочного шва, ослабленного отверстиями под заклепки.

Исходные данные: ширина полос b , допустимое напряжение на растяжение $[\sigma]_p$. Пределы изменения значений исходных данных — величины нагрузки P , толщины полос δ и количества заклепок i — взять из задачи 9. В выходном сообщении отметить строки с недопустимыми сочетаниями значений параметров исходных данных.

11. Рассчитать параметры сварного углового лобового шва, примененного для соединения двух стальных полос (марка стали СТЗ) толщиной δ при статической растягивающей нагрузке P (рис. 10). Сварка произведена электродами Э34 с ионизирующим меловым покрытием, что допускает

$$[\tau]_{\text{ср}}' = 0,5 [\sigma]_p. \quad (16)$$

Для расчета ширины листа b использовать выражение

$$b = \frac{P}{1,4\delta [\tau]_{\text{ср}}'}, \quad (17)$$

где $[\tau]_{\text{ср}}'$ — допустимое напряжение среза при указанных условиях проведения сварки.

Исходные данные: нагрузка P , толщина листов δ , допускаемое напряжение на растяжение $[\sigma]_p$.

12. Рассчитать параметры сварного стыкового шва (рис. 11) двух полос из стали марки СТЗ толщиной δ при статической растягивающей нагрузке P .

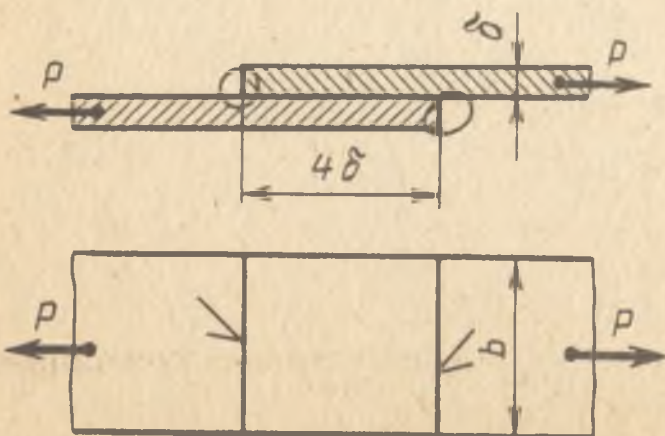


Рис. 10

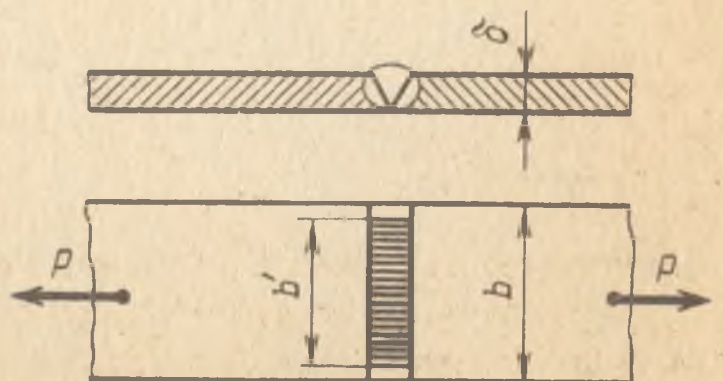


Рис. 11

Сварка произведена электродами Э34 с тонким покрытием, в связи с чем допускаемое напряжение для сварного шва равно

$$[\sigma]_p' = 0,6 [\sigma]_p. \quad (18)$$

Ширину листа с учетом условий сварки определить по выражению

$$b = \frac{P}{\delta [\sigma]_p'}. \quad (19)$$

Исходные данные: растягивающая нагрузка P , толщина полосы δ , допускаемое напряжение на растяжение $[\sigma]_p$.

Увеличить полученное значение b с учетом непровара в начале и конце шва. Введя это новое значение b_H в ЭВМ, проверить выполнение соотношения

$$\sigma_p = \frac{P}{b_H \delta} \leq [\sigma]_p. \quad (20)$$

Если отношение (20) выполняется, выдать сообщение "ШОВ ПРОЧЕН"; если не выполняется, вновь ввести в ЭВМ скорректированное значение b_H и повторить вычисления.

13. Провести расчет зависимости величины предельной растягивающей нагрузки P для сварного стыкового шва от его длины b' (рис. 11) для двух стальных полос (материал СТ3) толщиной δ по выражению

$$P = [\sigma]_p \delta b'. \quad (21)$$

Исходные данные: $[\sigma]_p$, δ и граничные значения длины сварного шва b' от 150 до 500 мм с шагом 50 мм.

14. Составить схему алгоритма, выбирающую тип сварного шва (угловой или стыковой) в зависимости от исходных данных: величины статической растягивающей нагрузки P , толщины δ и ширины b стальных полос. При сварке полос из стали СТ3 используются электроды Э34. Из формул (16), (17) получаем выражения для расчета параметров углового шва:

$$\begin{aligned} P_{\text{угл}} &= 1,4b\delta [\tau]_{\text{ср}}', \\ [\tau]_{\text{ср}}' &= 0,5 [\sigma]_p. \end{aligned} \quad (22)$$

Формулы (18), (19) используем для расчета параметров стыкового шва:

$$P_{\text{стык}} = b\delta [\sigma]_p', \quad (23)$$

$$[\sigma]_p' = 0,6 [\sigma]_p.$$

Принять $[\sigma]_p = 160$ МПа.

Решением задачи является алгоритм разветвленной структуры, основным элементом которого является блок, реализующий условия выбора (рис. 12):

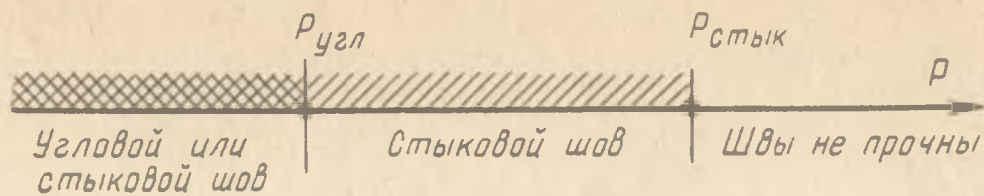


Рис. 12

если $P \leq P_{\text{угл}}$, то возможен угловой или стыковой шов;

если $P > P_{\text{угл}}$, то угловой шов не подходит, возможен только стыковой шов;

если $P > P_{\text{стык}}$, то и стыковой шов не подходит, швы не прочны.

По результатам проверок выдаются указанные сообщения.

15. Составить схему алгоритма определения диаметра d стержня в неослабленном отверстии под клин сечении для клинового соединения двух стержней втулкой при статической растягивающей нагрузке P , заданных материалах стержня, втулки и клина (рис. 13). Для вычисления диаметра использовать выражение

$$d = \sqrt{4P / (\pi [\sigma]_p)}. \quad (24)$$

В качестве исходных данных принять значение допустимого напряжения $[\sigma]_p = 100$ МПа и пять различных значений нагрузки P . Результаты расчетов округлить в большую сторону.

16. Составить схему алгоритма определения диаметра d_0 утолщенной части стержня в месте расположения клина (рис. 13). Ширина отверстия под клин

$$b = \frac{1}{4} d_0. \quad (25)$$

Для вычисления диаметра d_0 использовать выражение

$$d_0 = \sqrt{\frac{4P}{(\pi - 1) [\sigma]_p}}. \quad (26)$$

Исходные данные: нагрузка P , допустимое напряжение на растяжение $[\sigma]_p$, допустимое напряжение на смятие $[\sigma]_{\text{см}}$.

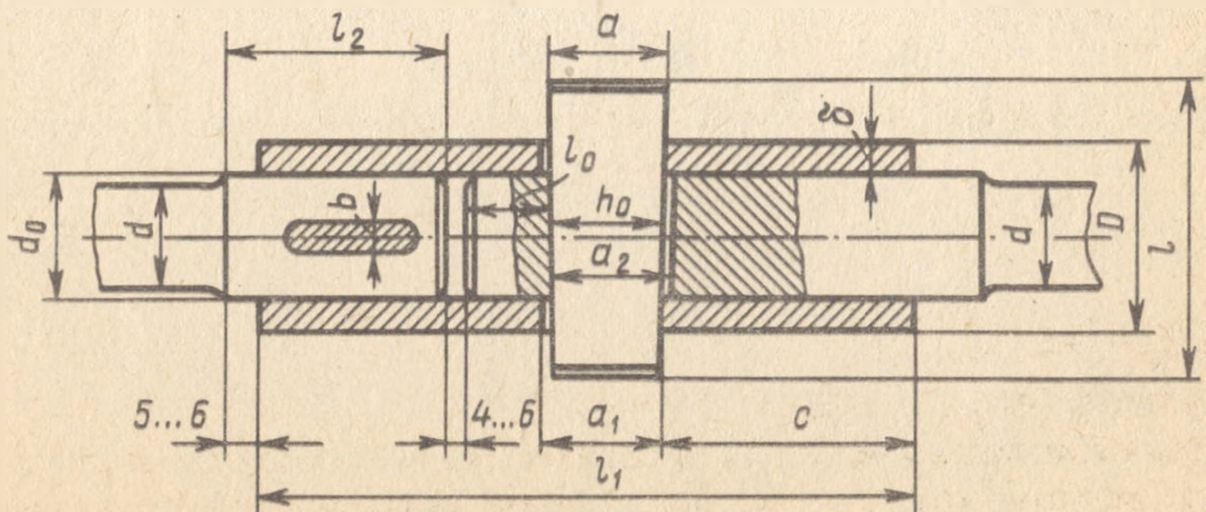


Рис. 13

Рассчитав по формуле (26) значение d_0 , округлить его до ближайшего целого числа. Введя его в ЭВМ, провести проверку на смятие стержня клином:

$$\sigma_{\text{см}} = \frac{P}{d_{0\text{окр}}b} \leq [\sigma]_{\text{см}} \quad (27)$$

Если отношение (27) выполняется, выдать сообщение "СОЕДИНЕНИЕ ПРОЧНО"; если не выполняется, вновь ввести в ЭВМ скорректированное значение $d_{0\text{окр}}$ и повторить вычисления.

17. Используя диалоговый режим работы проектировщика с ЭВМ, произвести расчет параметров клинового соединения (см. рис. 13).

Исходные данные: значения допустимых напряжений $[\tau]_{\text{ср}}$, $[\sigma]_{\text{см}}$, нагрузки P и диаметра d_0 .

Определить расстояние между клином и краем стержня по формуле

$$l_0 \geq \frac{P}{2d_0[\tau]_{\text{ср}}} \quad (28)$$

Получив расчетное значение l_0 , оценить его, скорректировать и новое принятое значение l_0 ввести в ЭВМ.

Далее из условия обеспечения прочности втулки на смятие клином определить толщину втулки, используя выражение

$$\delta \geq \frac{P}{2b[\sigma]_{\text{см}}} \quad (29)$$

Для расчета значения b использовать формулу (25). Получив из ЭВМ расчетное значение δ , округлить его до ближайшего большего четного числа и новое принятое значение δ ввести в ЭВМ. Затем рассчитать наружный диаметр втулки

$$D = d_0 + 2\delta \quad (30)$$

и расстояние между клином и краем втулки — в соответствии с выражением

$$C \geq \frac{P}{4\delta[\tau]_{\text{ср}}} \quad (31)$$

Получив расчетное значение C , оценить его, скорректировать и новое принятое значение C ввести в ЭВМ.

Выдать из ЭВМ все полученные параметры клинового соединения: b , l_0 , D , C .

18. Продолжить расчет параметров клинового соединения. В качестве исходных данных d_0 , b , δ , D , C ввести значения, полученные в задаче 17. Использовать значения нагрузки P и допустимого напряжения на изгиб $[\sigma]_{\text{и}}$.

Длина клина

$$l = d_0 + 4\delta \quad (32)$$

Средняя высота клина

$$h_0 = \sqrt{3PD / (4b [\sigma]_{\text{н}})}. \quad (33)$$

Наибольшая высота клина

$$a = h_0 + jl/2, \quad (34)$$

где $j = 1/30$ — уклон.

Затем рассчитать наименьшую ширину отверстия под клин во втулке a_1 и в стержне a_2 :

$$a_1 = h_0 - jD/2, \quad (35)$$

$$a_2 = h_0 - jd_0/2. \quad (36)$$

Следует скорректировать расчетные значения параметров в диалоговом режиме.

Определить длину втулки l_1 и утолщенной части стержня l_2 :

$$l_1 = 2(l_0 + h_0 + C) + 5; \quad (37)$$

$$l_2 = (l_0 + h_0 + C) + 5. \quad (38)$$

19. Произвести проверку прочности штифта из условия его прочности на смятие и срез в шпоночном соединении вала и ступицы при передаче вращающего момента M (рис. 14).

Исходными данными для решения задачи являются конструктивные параметры соединения: диаметр вала $d_{\text{в}}$, диаметр штифта $d_{\text{ш}}$, длина штифта $l_{\text{ш}}$ и момент нагрузки M , а также значения допустимых напряжений $[\tau]_{\text{ср}}$ и $[\sigma]_{\text{см}}$. Для расчета использовать формулу

$$\sigma_{\text{см}} = \frac{4M}{d_{\text{в}} l_{\text{ш}} d_{\text{ш}}} \leq [\sigma]_{\text{см}}. \quad (39)$$

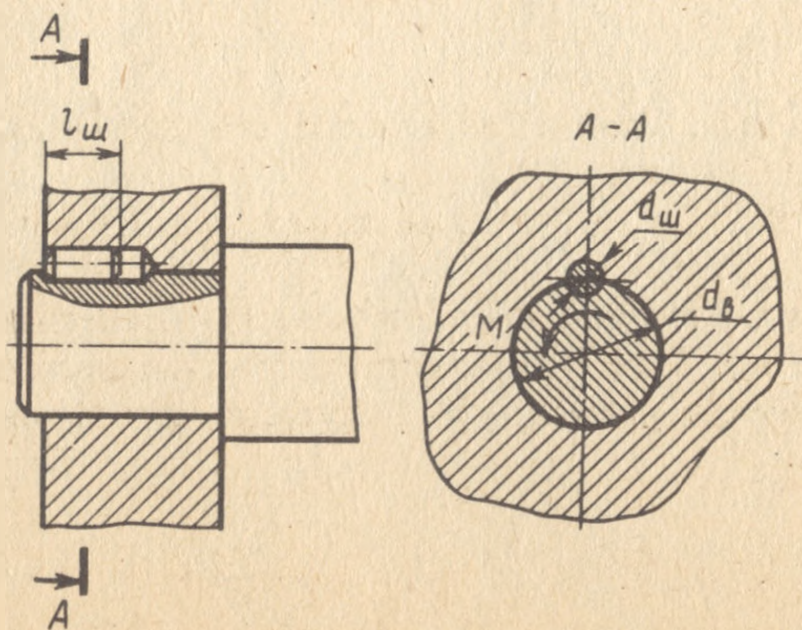


Рис. 14

Длина штифта $l_{ш}$ определяется по формуле

$$l_{ш} = 1,2d_{в}. \quad (40)$$

Далее рассчитать прочность штифта на срез, используя выражение

$$\tau_{ср} = \frac{2M}{d_{в}d_{ш}l_{ш}} \leq [\tau]_{ср}. \quad (41)$$

20. Произвести расчет на прочность соединения призматической шпонкой для передачи крутящего момента M (рис. 15), используя выражение

$$\sigma_{см} = \frac{2M}{dl_{п}k} \leq [\sigma]_{см}. \quad (42)$$

Исходные данные: диаметр вала d , конструктивные размеры шпонки b и k , допускаемое напряжение смятия $[\sigma]_{см}$.

Длина шпонки

$$l = 1,2d. \quad (43)$$

Расчетная длина шпонки

$$l_{п} = l - b. \quad (44)$$

Расчеты провести для пяти различных значений крутящего момента M . По результатам расчетов выдать сообщения "СОЕДИНЕНИЕ ПРОЧНО" или "СОЕДИНЕНИЕ НЕ ПРОЧНО", а также соответствующие значения моментов, расчетных и допустимых напряжений.

21. Произвести прочностной расчет резьбы хвостовика грузоподъемного крюка (рис. 16), изготовленного из стали СТЗ при заданной нагрузке P и допускаемом напряжении на растяжение в металле $[\sigma]_{р}$.

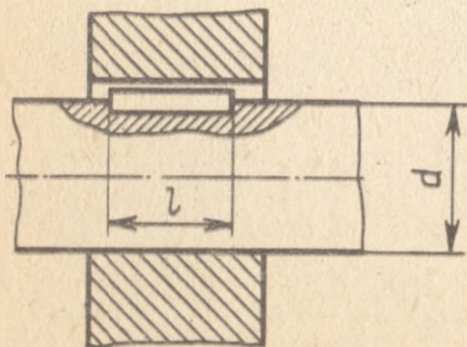


Рис. 15

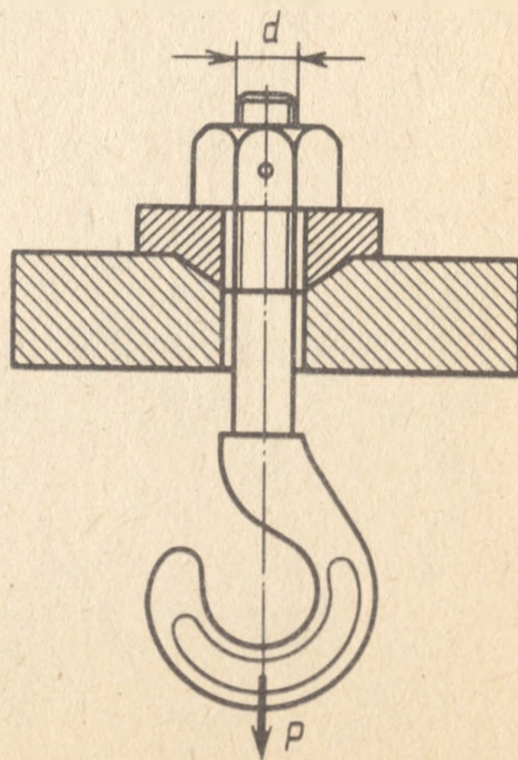


Рис. 16

Расчетный диаметр резьбы хвостовика определить по выражению

$$d_0 \geq 1,13 \sqrt{\frac{P}{[\sigma]_p}}. \quad (45)$$

Получив расчетное значение d_0 , необходимо по таблице стандартных размеров метрических резьб выбрать ближайшее к расчетному большее значение, которое будет искомым значением внутреннего диаметра резьбы:

$$d > d_0. \quad (46)$$

Так, если расчетное значение $d_0 = 24,914$ мм, а ближайшее табличное значение $d = 26,918$ мм (метрическая резьба М28), то ответом решения задачи будет сообщение "РЕЗЬБА М28, ВНУТРЕННИЙ ДИАМЕТР $D = 26,918$ ММ".

22. Произвести прочностной расчет резьбы хвостовика грузоподъемного крюка (см. рис. 16) с использованием формул (45), (46) для различных значений нагрузки P от 10 до 100 кН с шагом 10 кН.

Допускаемое напряжение на растяжение в металле $[\sigma]_p = 72$ МПа. Результатами расчета являются табличные значения внутреннего диаметра резьбы d и тип резьбы.

23. Составить схему алгоритма расчета результирующего изгибающего момента на вал (рис. 17), который определяется из соотношения

$$M_{\text{и}} = \sqrt{M_x^2 + M_y^2}, \quad (47)$$

где изгибающий момент по оси x

$$M_x = \frac{F_t}{2} \cdot \frac{l}{2}; \quad (48)$$

изгибающий момент по оси y

$$M_y = \frac{F_r}{2} \cdot \frac{l}{2}. \quad (49)$$

Исходные данные: нагрузки F_t и F_r , длина l .

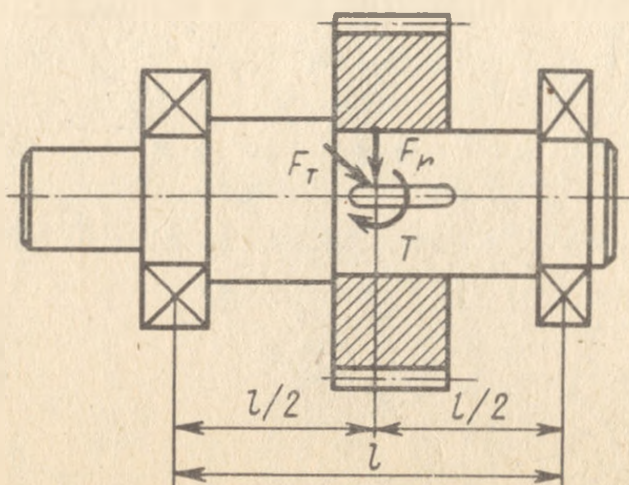


Рис. 17

24. Используя формулы (47)–(49), произвести расчет зависимости изгибающего момента на вал от нагрузок, которые изменяются в пределах:

F_t от 3000 до 8000 Н с шагом 500 Н;

F_r от 1500 до 3000 Н с шагом 500 Н;

$l = 170$ мм.

Результаты расчетов представить в виде таблицы.

25. Произвести расчет эквивалентного момента в опасном сечении вала:

$$M_{\text{экв}} = \sqrt{M_x^2 + M_y^2 + 0,75T^2}. \quad (50)$$

Исходные данные: момент T , пределы изменения моментов M_x и M_y ; формулы для их расчета, а также значение l взять из задачи 24.

Результаты расчетов представить в виде таблицы.

26. Составить схему алгоритма определения длины никелинового провода диаметра d , который используется для изготовления реостата, имеющего сопротивление R .

Из формул (5), (6) получим

$$l = \frac{\pi d^2 R}{4\rho}. \quad (51)$$

27. Составить схему алгоритма определения диаметра провода для обеспечения нужного сопротивления при заданном материале и длине. При расчете использовать формулу (51).

Получив расчетное значение диаметра d , выбрать ближайшее стандартное значение.

28. Составить схему алгоритма расчета сопротивления R проводов из 10 различных материалов, используя формулу (51). Исходные данные: длина провода l , диаметр провода d , значения удельных сопротивлений ρ .

29. Используя формулу (51), произвести расчет сопротивления 1 м медных проводов при различных значениях диаметра проводов d .

30. Произвести расчет диаметра провода d по значениям тока I и допустимой плотности тока δ . При расчете использовать формулу

$$d = 1,13 \sqrt{\frac{I}{\delta}}. \quad (52)$$

Значения тока I меняются в пределах от 100 мА до 1 А с шагом 100 мА.

•31. Произвести расчет делителя напряжения цепи постоянного тока (рис. 18). Исходные данные: $U_{\text{и}}$ — напряжение источника, подводимое к делителю; U — напряжение, снимаемое с делителя, изменяется в пределах от 20 до 100 В с шагом 10 В; I — ток потребителя; I_2 — ток через сопротивление R_2 делителя.

Результатом решения задачи являются значения сопротивления делителя R_1 и R_2 для различных значений напряжения U .

Расчетные соотношения вывести из рис. 18.

32. Произвести расчет индуктивности однослойной катушки (рис. 19). Индуктивность определяется по формуле

$$L = \frac{0,01 D w^2}{0,44 + (a/D)}, \quad (53)$$

где D — диаметр катушки; a — длина намотки; w — число витков.

●33. Используя формулу (53), рассчитать число витков катушки индуктивности. Исходные данные: D , a , L .

Затем определить значение диаметра провода с изоляцией:

$$d_{из} = a/w. \quad (54)$$

34. Составить схему алгоритма определения числа витков и диаметра провода с изоляцией для катушек индуктивности, обеспечивающих наименьшее активное сопротивление, которое достигается при отношении

$$a/D = 0,4. \quad (55)$$

Для расчета воспользуйтесь формулами (53) — (55).

Исходные данные: индуктивность L , значения диаметра катушки D изменяются в пределах от 1 до 10 см с шагом 1 см.

35. Провести расчет индуктивности многослойной катушки, имеющей размеры D_{min} , D_{max} , b (рис. 20) и число витков w . Индуктивность многослойных катушек определяется по формуле

$$L = \frac{0,08 D^2 w^2}{3D + 9b + 10C}, \quad (56)$$

где

$$D = 0,5 (D_{max} + D_{min}); \quad (57)$$

$$C = 0,5 (D_{max} - D_{min}). \quad (58)$$

36. Провести расчет количества витков многослойной катушки индуктивности, если заданы L , D_{min} , $d_{из}$, b . Воспользоваться выражениями (56) — (58), а также формулой

$$C = n \cdot d_{из}, \quad (59)$$

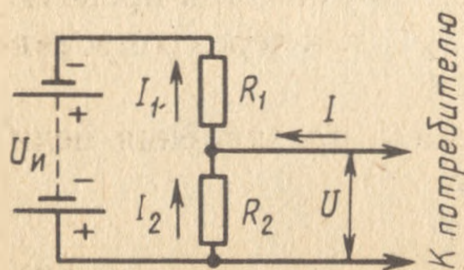


Рис. 18

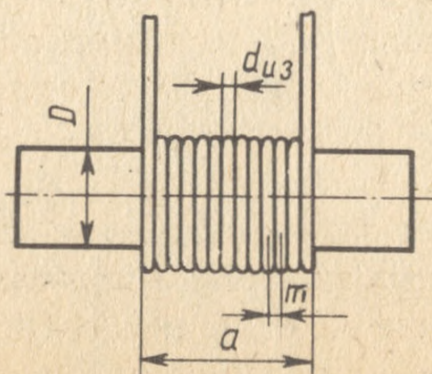


Рис. 19

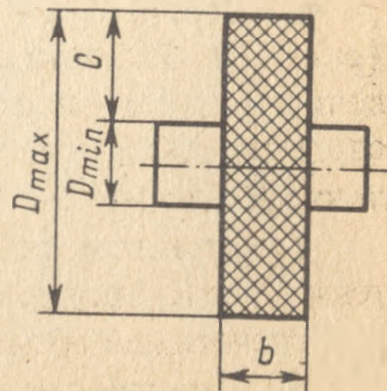


Рис. 20

где число слоев в намотке катушки

$$n = w/m; \quad (60)$$

число витков в слое

$$m = b/d_{\text{из}}. \quad (61)$$

Преобразование формул (56) – (61) приводит к выражению

$$L = \frac{0,08 \cdot (d_{\text{из}}^2 \cdot \frac{w}{b} + D_{\text{min}})^2 \cdot w^2}{3 \cdot (d_{\text{из}}^2 \cdot \frac{w}{b} + D_{\text{min}}) + 10 \cdot d_{\text{из}}^2 \cdot \frac{w}{b} + 9 \cdot b}, \quad (62)$$

откуда для определения искомой величины w приходим к сложному уравнению 4-й степени.

Решим задачу иначе. Проведем расчет числа витков многослойной катушки w , используя диалоговый режим работы проектировщика с ЭВМ.

В формуле (62) обозначим числитель $P1$, знаменатель $P2$, т.е.

$$L = P1/P2. \quad (63)$$

Тогда разность $LP2 - P1$, которую обозначим через P , при точном искомом значении w должна быть близка к нулю:

$$P = LP2 - P1 \approx 0. \quad (64)$$

Это отношение и используем для диалогового расчета w . Сначала найдем приближенное значение w_0 из выражения

$$w_0 = 10\sqrt{L/D_{\text{min}}}. \quad (65)$$

Подставив это значение w_0 вместе со всеми другими исходными данными в выражение (62) и конечную формулу (64), найдем значение P . Если оно больше нуля, необходимо ввести в ЭВМ большее значение w и т.д. Введенное в ЭВМ значение w , при котором величина P близка к нулю ($P \approx 0$), и будет решением задачи.

37. Произвести расчет параметров трансформатора (рис. 21), где число витков первичной обмотки w_1 , число витков вторичной обмотки

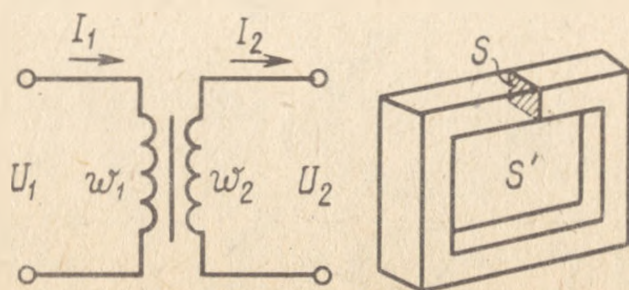


Рис. 21

w_2 , площадь сечения сердечника S , площадь сечения окна сердечника S' . Исходные данные: напряжение первичной обмотки U_1 , ток и напряжение вторичной обмотки I_2 и U_2 , КПД трансформатора η . При расчете использовать формулы

$$\eta = U_2 I_2 / (U_1 I_1), \quad (66)$$

$$S = \sqrt{U_1 I_1}, \quad (67)$$

откуда

$$S = \sqrt{U_2 I_2 / \eta}, \quad (68)$$

$$w_1 = 50 U_1 / S, \quad (69)$$

$$w_2 = 50 U_2 / S. \quad (70)$$

Значение диаметра проводов без изоляции:

$$d_{1,2} = 0,8 \sqrt{I_{1,2}}. \quad (71)$$

По вычисленным значениям d_1 и d_2 выбрать ближайшие стандартные значения диаметров проводов с изоляцией — $d_{из1}$ и $d_{из2}$. Площадь сечения окна сердечника S' определить по формуле

$$S' \geq 0,8 (w_1 d_{из1}^2 + w_2 d_{из2}^2). \quad (72)$$

38. Произвести расчет значения пульсации напряжения на выходе LC-фильтра (рис. 22), имеющего емкости $C_1 = C_2$, индуктивность катушки L , напряжение на выходе выпрямителя U_0 , значение нагрузки R , частоту пульсации f_{Π} . Напряжение пульсации для первой гармоники на входе фильтра

$$U_{m1 \text{ ВХ}} \approx \frac{U_0}{\pi f_{\Pi} C_1 R}. \quad (73)$$

Коэффициент фильтрации индуктивно-емкостного Г-образного фильтра

$$k_{\Phi} \approx 4\pi^2 f_{\Pi}^2 L C_2. \quad (74)$$

Напряжение пульсации для первой гармоники на выходе фильтра

$$U_{m1 \text{ ВЫХ}} = U_{m1 \text{ ВХ}} / k_{\Phi}. \quad (75)$$

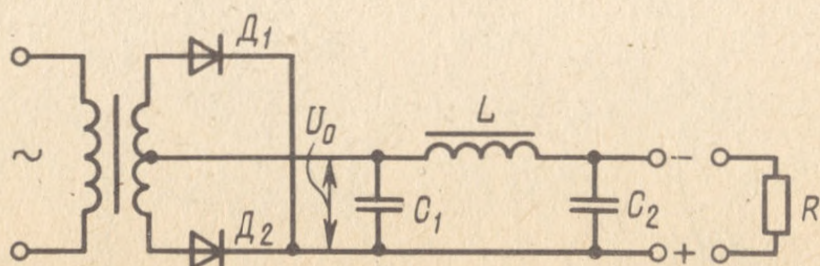


Рис. 22

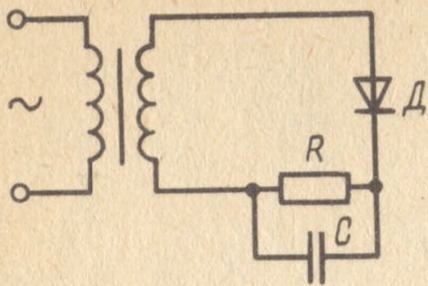


Рис. 23

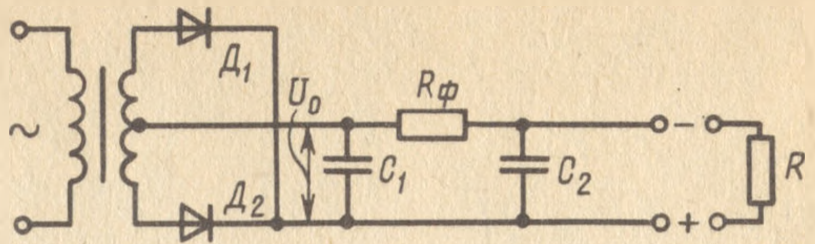


Рис. 24

39. Составить схему алгоритма расчета конденсатора C , сглаживающего пульсации выпрямленного тока в условиях изменяющейся нагрузки R (рис. 23), используя формулу

$$C = 1 / (\pi P f_{\text{п}} R), \quad (76)$$

где $f_{\text{п}}$ — частота пульсации; P — коэффициент пульсации.

40. Произвести расчет значения пульсации напряжения на выходе реостатно-емкостного фильтра (рис. 24), имеющего следующие исходные параметры; C_1 , C_2 , R_{ϕ} , U_0 , R , $f_{\text{п}}$. Значение коэффициента фильтрации k_{ϕ} для реостатно-емкостного фильтра определить по выражению

$$k_{\phi} = 2\pi f_{\text{п}} R_{\phi} C_2. \quad (77)$$

Вычислив k_{ϕ} , найти значение $U_{m_1 \text{ вых}}$ по формуле (73), $U_{m_1 \text{ вых}}$ по формуле (75). Результатом расчета является отношение напряжения пульсации для первой гармоники на выходе фильтра к величине напряжения U_0 в процентах, т.е.

$$\delta = \frac{U_{m_1 \text{ вых}}}{U_0} 100\%. \quad (78)$$

ГЛАВА ВТОРАЯ

ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ БЕЙСИК

§ 3. АЛФАВИТ ЯЗЫКА И ЭЛЕМЕНТАРНЫЕ КОНСТРУКЦИИ

БЕЙСИК — простой алгоритмический язык, используется практически во всех существующих ЭВМ и вычислительных системах и за последние годы (благодаря прежде всего своей простоте) выдвинулся в число ведущих языков программирования. Он один из основных языков в персональных компьютерах. Название языка BASIC возникло от сокращения английских слов Beginner's All-purpose Symbolic Instruction Code, в переводе на русский язык означающих "многоцелевой язык символических команд для начинающих".

Наибольшее применение БЕЙСИК нашел при программировании научно-технических задач вычислительного характера с небольшим объемом исходной информации в режиме диалога человека с машиной.

Основными элементами любого языка программирования являются: алфавит, синтаксис и семантика.

Алфавит — это набор разрешенных символов языка.

Синтаксис — это формальные правила образования отдельных конструкций (команд, операторов) языка из символов алфавита.

Семантика — это интерпретация, смысловое содержание тех или иных синтаксических конструкций языка.

На символику БЕЙСИКа немалое влияние оказала ориентация этого языка на диалоговые вычислительные системы. Выносной пульт, с помощью которого программист-пользователь общается с ЭВМ, это, как правило, электрифицированная пишущая машинка, телетайп или видеотерминал (дисплей). Поэтому алфавит БЕЙСИКа определяется возможностями клавиатуры соответствующих устройств. В его состав входят следующие символы:

1) 26 заглавных латинских букв от А до Z (для вывода надписей на экран дисплея, а также для записи комментариев к БЕЙСИК-программе можно использовать также заглавные буквы русского алфавита от А до Я)

2) 10 десятичных цифр от 0 до 9 (в системах программирования ЭВМ нуль иногда перечеркивают, чтобы отличить цифру нуль от буквы О),

3) специальные символы (их 19):

знаки арифметических операций: "+" (плюс), "-" (минус), "*" (звездочка — знак операции умножения), "/" (знак операции деления), "↑" (стрелка вверх — знак операции возведения в степень);

знаки операций отношения: "=" (равно), ">" (больше), "<" (меньше); а также символы: " " (пробел), "." (точка), "," (запятая), ";" (точка с запятой), "(" , ")" (круглые скобки), " " " (кавычки), " ' " (одинарные кавычки — апостроф), "\" (косая черта с левым наклоном), "#" (знак номера), "Q" (признак символьных данных).

БЕЙСИК-программа может быть записана только с использованием указанных символов.

В операционных системах некоторых микроЭВМ, использующих расширенные версии алгоритмического языка БЕЙСИК ("БЕЙСИК плюс"), в состав алфавита языка включены также символы: ":" (двоеточие), "?" (вопросительный знак), "!" (восклицательный знак), "%" (знак процента — признак данных целого типа).

К элементарным конструкциям языка БЕЙСИК относятся константы, переменные, массивы, стандартные функции.

Константами называются величины, которые имеют постоянное значение, не изменяющееся во время выполнения программы. В языке БЕЙСИК используются константы двух типов: числовые и строковые (текстовые). Числовые константы обозначают целые и вещественные десятичные числа.

Целые числа представляют собой обычную последовательность десятичных цифр со знаком (знак "+" может быть опущен), например -51, +123, 123.

Вещественные числа записываются в языке БЕЙСИК в виде десятичной дроби в обычной форме, принятой в математике; однако при отделении целой части числа от дробной вместо запятой используется десятичная точка, например 0.25, -36.85, 3.0.

При записи очень больших или очень малых чисел может быть использована показательная форма записи с десятичным множителем. Для этого применяется символ E, за которым следует целое число со знаком или без знака. Например, величину нагрузки $R_H = 15 \text{ кОм}$ ($15 \cdot 10^3 \text{ Ом}$) можно представить в показательной форме как 15E3, значение емкости $C = 20 \text{ мкФ}$ ($20 \cdot 10^{-6} \text{ Ф}$) можно записать 20E-6, величину тока $I = 5 \text{ мА}$ ($5 \cdot 10^{-3} \text{ А}$) можно представить как 5E-3. Формальные правила записи чисел в показательной форме: перед символом E должно быть число (любое), а показатель степени после символа E может быть только целым числом.

Диапазон допустимых чисел достаточно велик и зависит от конструкции ЭВМ. Так, для микроЭВМ типа ДВК диапазон допустимых чисел от 10^{-38} до 10^{+38} , для больших ЭВМ Единой системы (ЕС ЭВМ) — от 10^{-75} до 10^{+75} .

Закрепим данный материал, проделав упражнения.

•41. Какие из числовых констант в БЕЙСИК-программе записаны неверно? Укажите причины ошибок.

- | | | |
|-----------|-----------------|---------------|
| 1) 5 | 6) 1E3 | - 11) 0.5E2.1 |
| 2) +5 | ✓ 7) 2/3 | - 12) 12,5 |
| 3) 1000 | 8) -0.68E-3 | 13) 1.E2 |
| ✓ 4) E2 | ✓ 9) $\sqrt{9}$ | 14) 0125. |
| ✓ 5) E-02 | 10) +0.0 | 15) 1.1E5 |

Строковые константы представляют собой набор любых символов ЭВМ (а не только алфавита языка БЕЙСИК), ограниченный с двух сторон одиночными или двойными кавычками. Например,

“BASIC” ”АЛЬФА”, ”МАТРИЦА-3”, ‘LIST’
”ПТУ № 38 (ВЫСОКОЙ КУЛЬТУРЫ)”

Возможна комбинация двойных и одиночных кавычек, однако начальный и конечный ограничители должны быть одного типа. Изменение типа кавычек рекомендуется делать, когда кавычки необходимо употребить внутри самой строковой константы, например ‘ГАЗЕТА ”ТРУД”’.

При печати символьной строки начальные и конечные кавычки опускаются. Так, результатом работы оператора PRINT ‘ПРОФЕССИЯ – ”РЕГУЛИРОВЩИК РАДИОАППАРАТУРЫ”’ будет появление на экране дисплея текста ПРОФЕССИЯ – ”РЕГУЛИРОВЩИК РАДИОАППАРАТУРЫ”

Строковые константы называют также текстовыми константами (ТК) и широко используют при выдаче из ЭВМ различных поясняющих текстов. Максимальная длина одной ТК – 255 символов.

Переменными называются величины, значения которых в процессе выполнения программы могут меняться. Каждая переменная имеет свое собственное имя – идентификатор.

Идентификатор – символическое имя, это символ или комбинация символов, используемых для обозначения различных объектов программы – переменных, массивов и т.п. Каждой величине, обозначенной идентификатором, ставится в соответствие область памяти, в которую записывается текущее значение этой величины. Термин *переменная* обозначает величину, к которой обращаются, используя ее имя – идентификатор. В процессе решения задачи переменные могут принимать различные значения. В зависимости от типа принимаемых значений переменные делятся на числовые и символьные (последние используются чаще всего для вывода их значений на экран дисплея).

Существуют формальные грамматические правила задания имен – идентификаторов объектов программы. В алгоритмическом языке БЕЙСИК эти правила очень просты. Так, идентификатором числовой переменной могут быть любая буква латинского алфавита либо любая буква латинского алфавита, за которой следует одна цифра, например А, А1, В0, С7.

Идентификатором символьной переменной, могут быть любая буква латинского алфавита и за ней символ “Q” (он является признаком символьной переменной) или любая буква латинского алфавита и за ней одна любая цифра, за которой следует символ “Q”, например А Q, В1 Q, М Q.

Выдерживая указанные формальные правила, при составлении программы следует стремиться к наиболее естественным обозначениям переменных. Так, если в математической формулировке задачи некоторая переменная, например скорость, обозначена буквой v, то и в БЕЙСИК-программе за этой величиной целесообразно сохранить наименование V.

•42. Укажите правильно написанные идентификаторы (имена) в БЕЙСИК-программе. Найдите ошибки в неверно записанных идентификаторах.

1) E3
2) A

3) 8E
4) Ю1

5) M5
6) B5 Q

7) AB
8) 05
9) PO

10) p0
11) A15
12) A 1

13) A*B
14) A(2)
15) 05

Термин массив используется для обозначения совокупностей однородных данных с одним и тем же идентификатором (именем) и различными индексами. Так, например, массив А состоит из пяти элементов: a_1, a_2, a_3, a_4, a_5 .

В ЭВМ для каждого массива выделяется область памяти, в каждой ячейке которой хранится один из элементов массива. К элементу массива можно обратиться, задавая адрес начала массива и номер необходимого элемента. Адрес начала массива задается идентификатором массива, а номер — индексом элемента массива. Правила образования идентификаторов массивов такие же, как и у идентификаторов переменных. Значения индексов задаются в скобках сразу же за идентификатором массива, например А(3) — означает 3-й элемент a_3 массива А.

На языке БЕЙСИК могут быть описаны одномерные массивы (с одним индексом) и двумерные массивы (с двумя индексами — таблицы, матрицы). Так, например, таблицу коэффициентов

$m_{11} \ m_{12} \ m_{13}$

$m_{21} \ m_{22} \ m_{23}$

можно представить в виде массива с идентификатором М. Массив М состоит из элементов:

М(1,1), М(1,2), М(1,3), М(2,1), М(2,2), М(2,3).

Первый индекс двумерного массива задает номер строки соответствующей таблицы — матрицы, а второй индекс — номер столбца.

Массивы, так же как и переменные, могут быть *числовые* и *символьные*.

Объединение отдельных переменных в массивы позволяет упорядочить элементы массива в памяти ЭВМ и тем самым облегчить их массовую обработку, а также упрощает идентификацию элементов массивов, так как для ссылки на нужный элемент массива достаточно указать его индексы.

При программировании вычислительных задач часто приходится обращаться к однотипным вычислениям — величины тригонометрических функций, логарифма, квадратного корня и т.п.

Для таких наиболее распространенных функций программы их вычисления отлажены заранее и имеются на каждой ЭВМ. Такие функции принято называть *стандартными (или библиотечными)*.

Т а б л и ц а 2. Стандартные функции в языке БЕЙСИК

Математическая функция	Идентификатор стандартной функции	Пояснение
sin x	sin(X)	Синус числа X, выраженного в радианах Косинус числа X, выраженного в радианах
cos x	cos(X)	

Математическая функция	Идентификатор стандартной функции	Пояснение
$\operatorname{tg} x$	TAN(X)	Тангенс числа X, выраженный в радианах
$\operatorname{arctg} x$	ATN(X)	Арктангенс числа X, результат представлен в радианах
e^x	EXP(X)	Степень числа $e = 2.7182$ с показателем X
$\ln x$	LOG(X)	Натуральный логарифм числа X
$\lg x$	LOG10(X)	Десятичный логарифм числа X
\sqrt{x}	SQR(X)	Квадратный корень из числа X
$ x $	ABS(X)	Абсолютная величина – модуль числа
$\operatorname{entier} x$	INT(X)	Целая часть числа X
$\operatorname{sign} x$	SGN(X)	Знак числа X; при: $X = 0 \quad \operatorname{SGN}(X) = 0,$ $X > 0 \quad \operatorname{SGN}(X) = +1,$ $X < 0 \quad \operatorname{SGN}(X) = -1$

Чтобы обратиться к стандартной функции, надо указать ее имя (идентификатор из табл. 2) и записать за ним в круглых скобках аргумент – фактический параметр, для которого должна быть вычислена необходимая стандартная функция. Например, чтобы вычислить $\sin x$, нужно записать $\operatorname{SIN}(X)$; $\sin w_0 - \operatorname{SIN}(W0)$; $\sqrt{x} - \operatorname{SGR}(X)$; $\lg t - \operatorname{LOG10}(T)$.

Существует и несколько стандартных функций без аргументов. Например, функция с именем PI выдает значение константы – числа $\pi = 3,14159$.

Арифметическое выражение (формула, по которой надо произвести вычисления) представляет собой совокупность констант, переменных, стандартных функций, соединенных по определенным правилам знаками арифметических операций (сложение "+", вычитание "-", умножение "*", деление "/", возведение в степень "^") и скобками.

Примеры записи:

Математическая	На БЕЙСИКе
$a^2 + b^2$	$A \uparrow 2 + B \uparrow 2$
$c(a + b)^2$	$C * (A + B) \uparrow 2$
$\frac{-b + \sqrt{b^2 - 4ac}}{2a}$	$(-B + \operatorname{SQR}(B \uparrow 2 - 4 * A * C)) / (2 * A)$

В некоторых системах программирования вместо знака возведения в степень символа "^" используется символ "↑" или записанные подряд два символа **, т.е. математическая формула $a^2 + b^2$ в такой БЕЙСИК-программе запишется как

$$A \uparrow 2 + B \uparrow 2 \quad \text{или} \quad A ** 2 + B ** 2.$$

Как видно из примеров, запись арифметических выражений на БЕЙСИКе очень близка к обычной математической записи и отличается

от нее только тем, что формула пишется в одну строку с целью удобства последующего ее ввода в ЭВМ.

При записи формул на алгоритмическом языке необходимо учитывать следующие правила.

▷ 1. Порядок действий при вычислении арифметических выражений определяется правилом старшинства операций:

вычисляются выражения внутри скобок;

вычисляются стандартные функции;

производится операция возведения в степень;

производятся операции типа умножения и деления;

выполняются самые младшие по старшинству действия операции типа сложения и вычитания.

2. Операции одного ранга (равные по старшинству), например операции умножения и деления, производятся по формуле слева направо.

3. Нельзя опускать знаки операций, например символ операции умножения "*" ; нельзя ставить два знака операций подряд.

4. Количество открывающих скобок должно быть равно количеству закрывающих скобок.

•43. Указать последовательность операций при вычислении арифметического выражения и определить его значение

$$A/B * C + D \uparrow (E/2 - F * G)$$

при $A = 4$; $B = 2$; $C = -1$; $D = 10$; $E = 8$; $F = 3$; $G = 1$.

•44. Используя элементарные конструкции языка БЕЙСИК (константы, переменные, стандартные функции), записать следующие формулы:

1) $ax + by + cz$,

6) $\sin^2 x + \cos^2 x$,

2) $(x + y)^3$,

7) $\sqrt{a^3} + \sqrt{a - b}$,

3) $\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2}$,

8) $(x - y) \cdot \sin x + (x^3 + y^3) \cdot \operatorname{tg} x$,

4) $y + \frac{x^4}{2b} - 1,5$,

9) $\left(\frac{e^x + e^{-x}}{2}\right)^2 + \left(\frac{e^x - e^{-x}}{2}\right)^2$,

5) $\frac{r}{r_1} + \frac{cd}{r \cdot r_1 \cdot h}$,

10) $xy \cdot \sqrt{x^2 + y^2} - \ln|x - \sqrt{1 + x^2}|$.

§ 4. ОСНОВНЫЕ ОПЕРАТОРЫ ЯЗЫКА БЕЙСИК

Оператор — это приказ машине выполнить какое-либо действие, например: изменить значение переменной, изменить порядок выполнения операторов в программе, вывести на экран текущее значение. Начинается каждый оператор с ключевого слова. Например, INPUT (ввести исходные данные), PRINT (напечатать результат).

Программа, составленная на языке программирования, представляет собой последовательность операторов-команд, выполнение которой приводит к искомому результату.

Оператор является наименьшей самостоятельной синтаксической единицей программы. Каждый оператор БЕЙСИК-программы имеет свой

порядковый номер — число от 1 до 9999, которое определяет последовательность выполнения операторов в программе. Обычно нумерацию операторов осуществляют с шагом 10. Это правило вводится для того, чтобы можно было при необходимости вставлять в программу новые операторы, не изменяя всю нумерацию программы. В каждой строке записывается один оператор. Рассмотрим БЕЙСИК-программу:

```
10 A = 5
20 B = 10
30 C = A + B
40 PRINT A, B, C
50 END
```

В этой программе операторы с номерами 10 и 20 присваивают значения переменным A и B (5 и 10 соответственно). Оператор 30 вычисляет их сумму. Оператор с номером 40 печатает значения переменных A и B и вычисленной суммы C. Оператор 50 указывает на конец БЕЙСИК-программы.

Номера строк необходимы для:

указания порядка выполнения программы;
изменения обычного (естественного) порядка выполнения операторов, для обеспечения ссылок при условной и безусловной передаче управления;

модификации (изменения) отдельных строк и отладки программ.

Максимальная длина строки равна 132 символам. Однако в целях удобства чтения и редактирования программ рекомендуется выбирать длину строки, не превышающей строки дисплея, т.е. 72—80 символов.

В некоторых системах БЕЙСИК-программа может быть записана и более компактно, когда одна строка может содержать несколько операторов, отделяемых друг от друга разделителем — символом ”\” (в ряде систем разделителем операторов является символ ”: ” или ”;”). Программа из данного примера может быть записана короче так:

```
10 A = 5 \ B = 10 \ C = A + B
20 PRINT A, B, C
30 END
```

Операторы для программирования алгоритмов линейной структуры

О п е р а т о р п р и с в а и в а н и я

Оператор присваивания является самым распространенным оператором в любой программе вычислительного характера; он предназначен для вычисления значения некоторого выражения и присвоения этого значения переменной или элементу массива.

С и н т а к с и с

```
m LET a = b
```

где m — порядковый номер оператора; LET — ключевое слово оператора присваивания, оно означает — положить, присвоить значение; в большинстве систем ключевое слово LET может быть и опущено; a — идентификатор переменной или элемента массива; b — арифметическое выражение; "=" — знак операции присваивания.

Примеры записи операторов:

$$30 \text{ LET } A = B \uparrow 2 + C$$

$$50 \text{ LET } X(3) = X(2) - 5$$

Действие оператора (семантика)

присваивания

1. Действие оператора присваивания заключается в вычислении арифметического выражения, стоящего в правой части, и присваивания полученного значения переменной (или элементу массива), идентификатор которой стоит в левой части оператора.

2. К моменту выполнения оператора все величины, входящие в арифметическое выражение, должны быть определены, т.е. получить конкретные значения.

Итак, можно сказать, что оператор присваивания предназначен для программирования вычислений по формулам.

Например, для вычисления значения y по формуле $y = (a + b)/(c - d)$ используется оператор:

$$60 Y = (A + B) / (C - D).$$

Оператор присваивания может быть использован и для строковых (символьных) данных, например

$$30 C \text{ } \text{ } = \text{"ИВАНОВ"}$$

по которому в поле с именем $C \text{ } \text{ }$ помещается символьное данное — фамилия ИВАНОВ.

•45. Напишите операторы присваивания для вычисления по формулам:

$$1) S = 2\pi R,$$

$$5) y = \sqrt{x + 1} + \frac{1}{x - 1},$$

$$2) v = \frac{1}{3} \pi R^2 H,$$

$$6) y = \frac{\sqrt{x}}{x + e^x} \sin x,$$

$$3) r_2 = 2S \frac{\cos t}{2\pi x},$$

$$7) I = \frac{v}{\sqrt{x^2 + (CR - xC)^2}},$$

$$4) d = \sqrt{x^2 + y^2 + z^2},$$

•46. Даны фрагменты БЕЙСИК-программ, состоящие из операторов присваивания (некоторые из них для компактности записи занесены в одну строку). Определите значения переменных Y1 и Y2 после выполнения программ.

$$1) 10 A = 2 \setminus B = -3 \setminus C = -2 \setminus X = 2$$

$$20 Y1 = A * X \uparrow 2 + B * X + C$$

$$2) 10 A = -3 \setminus B = 2 \setminus C = 3.7$$

$$20 Y2 = A \uparrow B + C \uparrow B + A * B$$

Операторы ввода — вывода

Операторы ввода—вывода данных позволяют обмениваться информацией во время выполнения программы между программистом-пользователем и ЭВМ. В языке БЕЙСИК имеется оператор ввода данных INPUT и оператор вывода данных PRINT.

Оператор ввода данных INPUT позволяет вводить данные в ЭВМ во время выполнения программы и присваивать их значения переменным и элементам массивов.

Синтаксис

m INPUT $a_1, a_2, \dots,$

где m — порядковый номер оператора; INPUT — ключевое слово, означает — ввести; a_1, a_2, \dots — список переменных, элементов массива, разделенных запятой (это список ввода).

Пример:

50 INPUT A, B, C ☒

Семантика

1. При выполнении оператора INPUT вычисления по программе прерываются, и на экране дисплея появляется вопросительный знак "?".

2. Программист набирает на клавиатуре дисплея соответствующее списку ввода данное (значение данного) и нажимает клавишу **ВК** (возврат каретки, ввод).

3. Если в списке ввода оператора INPUT несколько элементов, то действия 1 и 2 повторяются до тех пор, пока весь список не будет исчерпан.

Так, если для приведенного примера необходимо ввести значения $A = 1,5$; $B = -2,7$ и значение текстовой переменной C ☒ — КОД, то это можно выполнить так:

? 1.5 **ВК**
? -2.7 **ВК**
? КОД **ВК**

Можно ввести соответствующие значения данных и одной строкой:

? 1.5, -2.7, КОД **ВК**

используя в качестве разделителя вводимых значений данных символ "," (запятую). Заметим, что текстовая константа КОД при использовании в операторе INPUT набирается без кавычек.

При проведении вычислений в БЕЙСИК-программе по какой-либо формуле необходимо, чтобы все переменные, участвующие в вычислениях по этой формуле, были определены (получили конкретные значения). Это можно сделать с помощью соответствующих операторов присваивания. Но тогда программа выполнит вычисления только для этих фиксированных значений переменных. Используя же оператор ввода INPUT, можно вводить различные значения исходных данных, т.е. сделать программу более гибкой.

Печать на терминале указателя оператора ввода вопросительного знака "?" при наличии в программе нескольких операторов INPUT не дает возможности определить, каким именно переменным требуется ввод. Поэтому некоторые системы программирования допускают включение в оператор INPUT сообщения (текстовой константы), заключенного в одиночные или двойные кавычки. Эти сообщения будут печататься на терминале и указывать, значения каких именно переменных необходимо ввести.

Например,

```
20 INPUT "МАССА" M.
```

При выполнении этого оператора на терминале печатается

```
МАССА?
```

В ответ программист должен набрать значение переменной M, например 125,5 г, и нажать клавишу **[BK]**, т.е.

```
МАССА? 125 5. [BK]
```

Оператор вывода данных PRINT позволяет выводить в процессе вычислений на экран дисплея сообщения, результаты счета программы в виде цифровой и текстовой информации.

С и н т а к с и с

```
m PRINT a1, a2, ...,
```

где m — порядковый номер оператора; PRINT — ключевое слово, означает — напечатать, вывести значение; a_1, a_2, \dots — список объектов вывода, разделенных символами "," или ";".

Пример.

```
100 PRINT A, "РЕЗУЛЬТАТ ="; X
```

Для вывода на печать или дисплей каждая строка устройства печати (или дисплея) делится на пять зон. Длина каждой зоны составляет 14 позиций (символов). Номера позиций печати, соответствующие началу и концу каждой зоны, следующие (рис. 25):

1-я зона : символы 1—14;

2-я зона : символы 15—28;

- 3-я зона : символы 29—42;
- 4-я зона : символы 43—56;
- 5-я зона : символы 57—70

(в некоторых системах длина зоны составляет 15 символов).

При печати элементов выводимого списка действует правило: разделитель — запятая в списке является сигналом печати значения очередного объекта вывода в следующей зоне.

Пример.

```
10 A = 1\B = -3\C = 4\D = -5\E = A * B + 1
20 PRINT A, B, C, D, E
30 END
```

Напечатанная строка будет выглядеть так, как показано на рис. 26.

Числа в кружочках обозначают позиции печати. Обратим внимание на то, что при печати числа располагаются с первого символа каждой зоны только в том случае, если они отрицательны (число -3, позиция 15; число -5, позиция 43; число -2, позиция 57). Положительные же числа располагаются со второго символа каждой зоны; на месте первого символа выдается пробел, так как знак "+" у числа не печатается (число 1, позиция 2; число 4, позиция 30).

Символьные (текстовые) константы при использовании разделителя “,” (запятая) печатаются с первого же символа каждой зоны.

Пример. 40 PRINT “A”, “B”, “C”, “D”, “E”.

Напечатанная строка также изображена на рис. 26.

Уплотнение печати может быть осуществлено за счет использования в качестве разделителя объектов вывода другого символа — “.” (точка с запятой). В этом случае соседние элементы выводимого списка отделяются при печати одним пробелом.

Пример. Если значения данных $A = 4$; $B = 352$; $C = 851$, то оператором

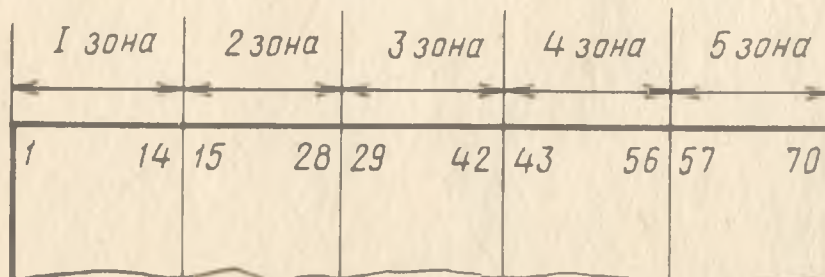


Рис. 25

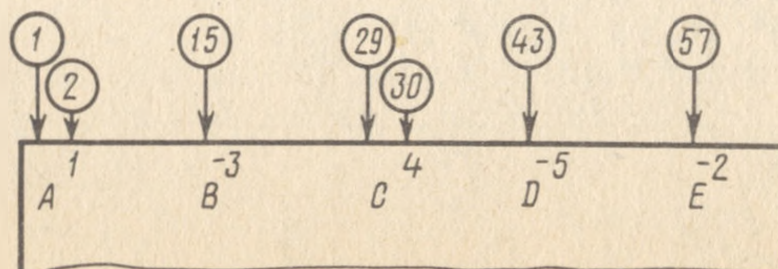


Рис. 26

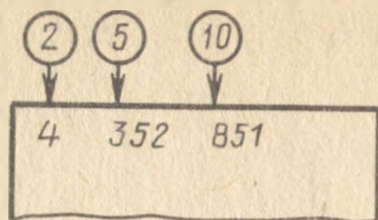


Рис. 27

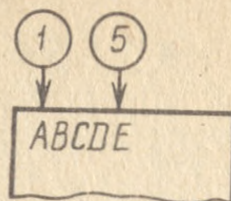


Рис. 28

80 PRINT A; B; C

данные выводятся в виде, показанном на рис. 27.

В данном примере все числа положительные и при печати их позиции, отводимые под знак числа, заменяются пробелами.

Отметим, что одним пробелом отделяются при использовании разделителя ";" только числовые данные. Символьные (текстовые) данные печатаются без каких-либо пропусков (пробелов), например

90 PRINT "A"; "B"; "C"; "D"; "E"

печатают строку вида, показанного на рис. 28.

Если в данной строке не хватает места для печати очередного элемента списка вывода, то ЭВМ продолжит печать с начала следующей строки. Обобщим изложенные правила выполнения оператора вывода PRINT.

Семантика

1. Действие оператора состоит в последовательном выводе на экран дисплея (или устройство печати) значений объектов вывода.

2. Если объект списка — числовая величина (константа или переменная), то выводится ее числовое значение.

3. Если объект списка — символьная величина (константа или переменная), то на экран выводится ее значение в виде последовательности символов без кавычек.

4. Если список объектов вывода в операторе PRINT отсутствует, то это равносильно пропуску строки при печати.

5. Если в качестве разделителя объектов вывода применяется символ ";", то вывод производится в виде таблицы в пять колонок (зон) по 14 символов в каждой зоне. Выводимые данные размещаются, начиная от левой границы каждой зоны. Незаполненные позиции выводятся в виде пробелов.

6. В случае использования разделителя ";" между объектами вывода на экране сохраняется минимальное расстояние, равное одному пробелу.

Рассмотренные операторы присваивания, ввода и вывода позволяют писать законченные БЕЙСИК-программы простейшей линейной структуры.

Операторы для программирования алгоритмов разветвляющейся структуры

Операторы управления

Нумерация строк в программе на языке БЕЙСИК определяет естественный порядок выполнения операторов: сначала выполняется оператор с номером 10, затем 20, 30 и т.д. Однако большинство задач редко укладывается в такую линейную схему. Поэтому в языке БЕЙСИК введены операторы управления, которые дают возможность перехода из одного места программы в другое. К ним относятся:

- оператор безусловного перехода GO TO;
- оператор условного перехода IF;
- оператор перехода к подпрограмме GOSUB – RETURN.

Оператор безусловного перехода GO TO

Синтаксис

m GO TO *n*

где *m* – порядковый номер оператора; GO TO – ключевое слово, означает – перейти к; *n* – номер оператора, к выполнению которого следует перейти.

Пример: 50 GO TO 100

Семантика

Осуществляется переход к выполнению оператора, имеющего порядковый номер, указанный справа от ключевого слова GO TO.

Оператор безусловного перехода GO TO изменяет естественный порядок выполнения операторов в программе, но передает управление только по одному направлению.

Оператор условного перехода

Синтаксис

m IF условие THEN *n*

где *m* – порядковый номер оператора; IF, THEN – ключевые слова, означают – если, то; *n* – номер оператора, которому передается управление.

Условие является основной частью условного оператора. Оно представляет собой два выражения, соединенных знаком отношения. В БЕЙСИКЕ используются следующие знаки отношений:

"="	равно,	"<>"	не равно,
">"	больше,	"<"	меньше,
">="	больше или равно,	"<="	меньше или равно.

Проверка условия заключается в том, что вычисляются оба выражения, стоящие слева и справа от знака отношения, и выясняется, соблюдается ли указанный знак отношения для вычисленных выражений или нет.

Семантика

Действие условного оператора заключается в следующем:

1. Вычисляются выражения, соединенные знаком отношения, и осуществляется проверка условия.

2. Если условие выполняется, то управление передается оператору, номер которого указан за ключевым словом THEN.

3. Если условие не выполняется, то управление передается оператору, непосредственно следующему за данным оператором условного перехода.

47. Составить программу вычисления функции $y = x^2$ для аргумента x , изменяющегося от 1 до 10 с шагом 0,1.

```
10 X = 1
20 Y = X ^ 2
30 PRINT X, Y
40 X = X + 0.1
50 IF X <= 10 THEN 20
60 продолжение программы
```

В данном примере оператор 40 изменяет значение аргумента 1,1; 1,2 и т.д., а условный оператор 50 проверяет, не превышена ли правая граница изменения аргумента — значение 10. Таким образом, условный оператор IF разветвляет вычислительный процесс по двум направлениям.

Оператор перехода к подпрограмме

Если в нескольких местах БЕЙСИК-программы приходится выполнять однотипные вычисления, то для того, чтобы не увеличивать объем программы за счет повторения идентичной последовательности операторов, прибегают к специальному приему программирования — использованию подпрограмм. Подпрограмма — это специальным образом оформленная группа операторов, к которым можно обращаться из разных точек программы.

Обращение к подпрограмме на языке БЕЙСИК реализуется с помощью операторов GOSUB — RETURN.

Синтаксис

m GOSUB n

.....

n — начало подпрограммы

.....

p RETURN

где GOSUB и RETURN — ключевые слова, означают — перейти к подпрограмме и вернуться из подпрограммы; m , n , p — порядковые номера операторов.

Семантика

1. Оператор GOSUB осуществляет передачу управления оператору с порядковым номером n (началу подпрограммы).

2. Далее выполняется последовательность операторов с номерами от n до p (тело подпрограммы, которой передано управление).

3. Под действием оператора RETURN выполнение подпрограммы заканчивается и управление передается оператору, непосредственно следующему за оператором GOSUB.

Операторы для программирования алгоритмов циклической структуры

Операторы цикла

Основное достоинство ЭВМ заключается в ее способности выполнять повторяющиеся вычисления. Для того чтобы запрограммировать такие повторяющиеся вычисления, в языке БЕЙСИК предусмотрены два оператора: оператор начала цикла FOR и оператор конца цикла NEXT, между которыми должны быть заключены операторы, которые необходимо неоднократно повторить, — так называемое "тело цикла". Для того чтобы контролировать число повторений, в состав операторов цикла вводится управляющая переменная — параметр цикла, значение которой меняется при каждом повторении цикла.

Синтаксис

```
 $m$  FOR  $a = b_1$  TO  $b_2$  STEP  $b_3$   
.....  
 $n$  NEXT  $a$ 
```

где FOR, TO, STEP, NEXT — ключевые слова оператора цикла; a — управляющая переменная, параметр цикла; b_1 — арифметическое выражение, задающее начальное значение параметра цикла; b_2 — арифметическое выражение, задающее конечное значение параметра цикла; b_3 — шаг, приращение параметра цикла; m и n — порядковые номера операторов начала и конца цикла.

Семантика

1. При выполнении оператора FOR вычисляются начальное и конечное значения управляющей переменной и значение шага цикла.

2. Выполняются операторы, заключенные между операторами FOR и NEXT (тело цикла).

3. При выполнении оператора NEXT значение переменной цикла увеличивается на шаг, а полученное новое значение управляющей переменной сравнивается с конечным значением переменной цикла.

4. Если в результате сравнения окажется, что значение управляющей

переменной цикла превысило конечное значение, то цикл на этом заканчивается и управление передается оператору, непосредственно следующему за оператором конца цикла NEXT. В противном случае выполнение цикла повторяется (следует возврат к п. 2).

Если шаг цикла равен 1, то ключевое слово STEP (шаг) и арифметическое выражение b_3 , задающее шаг цикла, можно опустить.

48. Составить программу вычисления значений функции $y = x^2$ с использованием операторов цикла. Значения аргумента x изменяются так же, как в примере 47.

```
10 FOR X=1 TO 10 STEP 0.1
20 Y=X↑2
30 PRINT X, Y
40 NEXT X
50 продолжение программы
```

С обработкой больших массивов данных (чаще всего в цикле) связано использование оператора описания массива.

Использованию в программе массивов должно предшествовать их описание. В нем указывается размер массива, т.е. определяется объем памяти ЭВМ, который необходимо отвести под тот или иной массив.

С и н т а к с и с

m DIM $U_1(a_1), U_2(a_2), \dots, U_n(a_n)$

где m — порядковый номер оператора; DIM — ключевое слово алгоритмического языка БЕЙСИК: сокращение от слова DIMENSION — размер; U_1, \dots, U_n — имена, идентификаторы описываемых массивов; a_1, \dots, a_n — одно или два десятичных числа, разделенных запятой и указывающих максимальное количество строк и столбцов массива.

Например,

```
10 DIM A(6), B(12), C(2,3)
```

Оператор с номером 10 описывает три массива: массив A, состоящий из 6 элементов, символьный массив B(12), состоящий из 12 элементов, и массив C, состоящий из двух строк и трех столбцов, т.е. состоящий из 6 элементов.

С е м а н т и к а

1. Описание одномерного массива состоит в указании количества его элементов.

2. Описание двумерного массива состоит в задании количества строк соответствующей матрицы (первая десятичная цифра в описании массива) и количества столбцов (вторая десятичная цифра). Элементы двумерного массива располагаются в памяти ЭВМ по строкам.

3. Оператор описания массива должен предшествовать использова-

нию элементов данного массива в других операторах программы.

В большинстве систем, реализующих алгоритмический язык БЕЙСИК, индексы элементов массивов принимают значения целых чисел, начиная с \emptyset , т.е. оператор описания массива А:

10 DIM A (6)

резервирует место в памяти не для 6, а для 7 элементов с индексами:

A (\emptyset), A (1), A (2), A (3), A (4), A (5), A (6).

•49. Описать приведенные ниже последовательности числовых величин как массивы:

$f(f_0, f_1, f_2, f_3, f_4, f_5); g(g_0, g_1, \dots, g_{15})$.

Напишите оператор описания массива символьных величин, содержащего значения 10 различных материалов: серебро, медь и т.д.

§ 5. ДОПОЛНИТЕЛЬНЫЕ СРЕДСТВА ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ БЕЙСИК

О п е р а т о р д а н н ы х DATA

Задание констант в программе может быть осуществлено тремя способами:

непосредственная запись константы в арифметическом выражении;
ввод значения константы с помощью оператора ввода INPUT;
задание блока данных.

Последний способ применяется в тех случаях, когда объем данных достаточно велик, а вводить их во время работы программы неудобно. *Блок данных* — это список констант, созданный специальным оператором — DATA. Константы можно извлекать последовательно одну за другой с помощью оператора READ.

С и н т а к с и с

m DATA a_1, a_2, a_3, \dots ,

где m — порядковый номер оператора; DATA — ключевое слово, означает — данные; a_1, a_2, a_3, \dots — список констант, составляющий блок данных.

Пример: 20 DATA 1, 3, -5

С е м а н т и к а

1. Блок данных формируется перед началом счета программы и допускает только последовательную выборку.

2. Оператор DATA может находиться в любом месте программы.

3. Если в программе используется несколько операторов DATA, то

формируется один общий блок данных, включающий в себя все наборы констант в порядке очередности их записи.

Оператор чтения данных из блока DATA

Синтаксис

m READ $a_1, a_2, \dots,$

где m — порядковый номер оператора; READ — ключевое слово, означает — читать, ввести; a_1, a_2, \dots — список переменных, которым присваиваются соответствующие значения констант из блока.

Пример: 30 READ A, B, C

Семантика

1. Переменным, перечисленным в списке оператора READ, присваиваются последовательные значения констант из блока DATA.

2. Если в программе имеются несколько блоков DATA, то они воспринимаются оператором READ как один суммарный блок.

3. Если при выполнении оператора READ блок DATA исчерпан, то выполнение программы заканчивается и на экран выводится сообщение о конце блока данных.

50. Вычислить и отпечатать значения функции $y = \sqrt{x}$ для различных значений аргумента $x = 3, 4, 7, 35, 51, 59, 71$.

```
10 DATA 3, 4, 7, 35, 51, 59, 71
20 READ X
30 Y = SQR(X)
40 PRINT X, Y
50 GO TO 20
60 END
```

Здесь оператор DATA содержит целый набор — 7 значений. Когда программа начинает работу, она прежде всего читает число из оператора DATA и присваивает его переменной X. Затем по оператору 30 с помощью стандартной программы вычисляется квадратный корень из X. После оператора 50 происходит переход на оператор 20 — READ и повторяются те же действия. В первый раз из блока данных — оператора DATA будет прочитано число 3, во второй — 4, в третий — 7 и т.д. Так как оператор DATA содержит семь чисел, то программа по оператору 30 вычислит семь значений y . После того как из оператора DATA будут прочитаны все числа, программа напечатает

? OUT OF DATA AT LINE 20

(конец блока данных в строке 20) и остановится.

О п е р а т о р PRINT TAB

Так как вывод результатов вычислений в удобной форме имеет для программиста большое значение, в языке БЕЙСИК предусмотрены различные возможности использования оператора вывода PRINT. Формат

```
PRINT TAB (X); A
```

позиционирует положение символа строки при печати. Значение переменной A выводится, начиная с (X + 1) позиции строки.

Так, операторы

```
10 A = -5  
20 PRINT TAB (10); A
```

приведут к печати (рис. 29), а оператор

```
30 PRINT "ВРЕМЯ"; TAB(20); "ТЕМПЕРАТУРА"
```

печатает текст (рис. 30).

Функцию TAB(X); "*" (символ звездочка) удобно использовать при выводе результатов вычислений в виде графиков.

О п е р а т о р комментария REM

Часто возникает необходимость для лучшего документирования программы ввести в нее различные пояснения (комментарии). Для этого служит оператор REMARK (комментарий) или короче REM.

С и н т а к с и с

m REM текст

Текст в операторе REM — это сообщение, которое может содержать любой символ клавиатуры ЭВМ.

С е м а н т и к а

Оператор REM применяется для включения в программу всевозможных пояснений в целях ее наглядности. На выполнение программы никакого действия не отказывает.

Пример:

```
10 REM ПРОГРАММА ВЫЧИСЛЕНИЯ ФУНКЦИИ
```

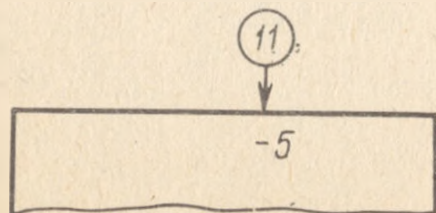


Рис. 29

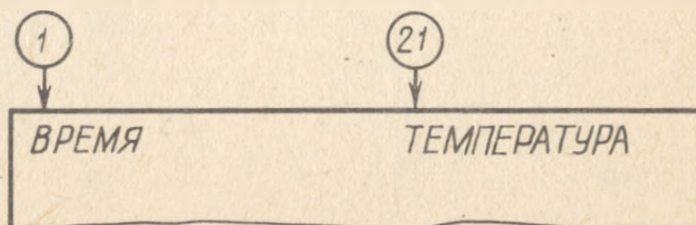


Рис. 30

Оператор прерывания PAUSE

Синтаксис

m PAUSE

где m – порядковый номер оператора; PAUSE – ключевое слово, означает – пауза, приостанов.

Семантика

Оператор PAUSE вызывает временную остановку (приостанов) программы. Программист может приостановить вычисления оператором PAUSE, чтобы проанализировать полученные данные, а затем, нажав клавишу **ВК**, продолжить выполнение программы.

Операторы окончания выполнения программы STOP, END

Синтаксис

m STOP

n END

где m, n – порядковые номера операторов; STOP, END – ключевые слова, означающие – останов и конец.

Семантика

1. Оператор END указывает на конец текста программы. Он является единственным и обязательным последним оператором в БЕЙСИК-программе.

2. Под действием оператора STOP выполнение программы заканчивается. В отличие от оператора END оператор STOP (полный останов) может быть поставлен в любом месте программы и несколько раз. Оператор 100 STOP останавливает программу и печатает сообщение:

```
STOP AT LINE 100
```

(останов на 100-й строке)



ПРОГРАММИРОВАНИЕ ТИПОВЫХ ЗАДАЧ ПО ПРОФЕССИЯМ

51. Составить программу решения задачи 1. Допускаемое напряжение $[\tau]_{\text{ср}} = 100$ МПа задать в операторе присваивания. Исходные данные $P = 1,1 \cdot 10^5$ Н и $d = 24$ мм ввести с клавиатуры дисплея.

Последовательность действий при составлении и выполнении программы на ЭВМ сводится к следующему.

Необходимо выбрать имена (идентификаторы) для объектов программы в соответствии с математическим описанием — формулами каждой конкретной задачи.

Схему алгоритма задачи, составленную в гл. 1, записать с помощью операторов алгоритмического языка, рассмотренных в гл. 2.

Выполнить составленную программу на ЭВМ для указанных значений исходных данных.

Выберем идентификаторы программы в соответствии с обозначениями расчетной формулы (1). Переменную P целесообразно назвать той же буквой P , переменную i буквой I , переменную d именем D . Будем придерживаться этого же правила и в дальнейшем, не забывая, однако, малые буквы в формуле обозначить соответствующими большими буквами в программе. Для получения значения π используем стандартную функцию с именем PI , которая дает соответствующее значение 3.14159. И только для величины $[\tau]_{\text{ср}}$ придется выбрать какой-либо, желательно близкий по обозначению, идентификатор, например $T\Phi$. Такие нестандартные идентификаторы будем в дальнейшем предлагать к целой группе задач в виде таблицы. Текст исходной программы на алгоритмическом языке БЕЙСИК может иметь вид:

```
05 REM ПРОГРАММА ОПРЕДЕЛЕНИЯ КОЛИЧЕСТВА ЗАКЛЕПОК ШВА
10 TO=100
20 PRINT "ВВОД ИСХОДНЫХ ДАННЫХ: P,D"
30 INPUT P,D
40 I=4*P/(PI*D*2*TO)
45 REM ВЫДЕЛЕНИЕ ЦЕЛОЙ ЧАСТИ БОЛЬШЕГО ЧИСЛА
50 I=INT (I+1)
60 PRINT "КОЛИЧЕСТВО ЗАКЛЕПОК ОДНОРЯДНОГО ШВА =" ; I
70 END
```

Оператор 20 выдаст на экран дисплея текст:

ВВОД ИСХОДНЫХ ДАННЫХ: P, D

т.е. укажет, какие величины нужно будет вводить в ЭВМ и в каком порядке. На оператор 30 ввода значений исходных данных учащийся должен ответить:

1.1E5, 24 BK

т.е. ввести значение нагрузки $P = 1,1 \cdot 10^5$ Н и величину диаметра заклепок $d = 24$ мм.

Нажатие клавиши BK вводит набранные значения исходных данных в ЭВМ. Оператор 40 производит вычисления в соответствии с формулой (1). Оператор 50 использует стандартную функцию INT для выделения целого числа (заклепок), а ее аргумент $(I + 1)$ осуществляет переход к большему числу. Оператор 60 выдаст результат работы программы в виде сообщения:

КОЛИЧЕСТВО ЗАКЛЕПОК ОДНОРЯДНОГО ШВА = 3.

Отметим также, что в тексте программы использован оператор комментария REM; он поясняет отдельные участки программы. Этот оператор поставлен на нестандартных позициях строк программы (05, 45), с тем, чтобы учащиеся могли и не вводить эти строки в ЭВМ, особенно при первоначальной отладке программы.

Перед следующей группой программ укажем возможные идентификаторы объектов программ.

Идентификаторы к программам 51—70:

Математическое обозначение . . .	$[\tau]_{\text{ср}}$	$\tau_{\text{ср}}$	$\sigma_{\text{см}}$	$[\sigma]_{\text{см}}$	$\sigma_{\text{р}}$	$[\sigma]_{\text{р}}$	δ
Идентификатор	TØ	T	S	SØ	R	RØ	G

- 52. Составить программу решения задачи 2. Исходные данные $P = 1,1 \cdot 10^5 \text{ Н}$, $i = 3$; $d = 24 \text{ мм}$, $\delta = 12 \text{ мм}$, $b = 240 \text{ мм}$ ввести с клавиатуры дисплея.
- 53. Составить программу решения задачи 3. Допустимое напряжение $[\tau]_{\text{ср}} = 100 \text{ МПа}$ задать в операторе присваивания. Исходные данные $P = 1,1 \cdot 10^5 \text{ Н}$, $i = 3$, $d = 24 \text{ мм}$ ввести с клавиатуры дисплея.

Для округления вычисленного значения l в большую сторону для обеспечения необходимой прочности шва использовать стандартную программу INT аналогично программе задачи 51. Результат выдать в виде сообщения:

РАССТОЯНИЕ ЗАКЛЕПОК ОТ ПОПЕРЕЧНОЙ КРОМКИ ПОЛОС =

- 54. Составить программу решения задачи 4. Значения длины провода $l = 30 \text{ м}$ и диаметра провода $d = 0,1 \text{ мм}$ задать в операторе присваивания, значение удельного сопротивления $\rho = 0,017 \text{ Ом} \cdot \text{мм}^2/\text{м}$ медного провода ввести с клавиатуры дисплея.

Текст исходной программы на алгоритмическом языке БЕЙСИК может иметь вид:

```

05 REM ПРОГРАММА РАСЧЕТА СОПРОТИВЛЕНИЯ ПРОВОДОВ
10 L=30\ D=0.1
20 PRINT "ВВОД ИСХОДНЫХ ДАННЫХ: МАТЕРИАЛ, УД. СОПРОТИВЛЕНИЕ"
30 INPUT M%, RO
40 S=PI*D^2/4
50 R=RO*L/S
55 REM ПЕЧАТЬ РЕЗУЛЬТАТА
60 PRINT "МАТЕРИАЛ - "; M%, "R="; R; "ОМ"
70 END

```

Операторы строки 10 программы присваивают начальное значение переменным l и d .

Оператор 20 выдаст на экран дисплея текст:

ВВОД ИСХОДНЫХ ДАННЫХ: МАТЕРИАЛ, УД. СОПРОТИВЛЕНИЕ

т.е. укажет, какие величины нужно будет вводить в ЭВМ и в каком порядке.

Особенностью данной программы является то, что кроме привычных в вычислительных задачах цифровых величин используются также символьные (текстовые) данные.

Так, на оператор 30 ввода значений исходных данных учащийся должен ответить:

МЕДЬ, 0.017 ВК

Операторы 40 и 50 производят вычисления в соответствии с формулами (5), (6). Оператор 60 выдаст результат работы программы в виде сообщения:

МАТЕРИАЛ – МЕДЬ $R = 64.93506 \text{ Ом}$

Отметим, что для компактности записи программы в некоторых строках используется сразу несколько операторов. Однако их можно вводить в ЭВМ и отдельно. Так, например, два оператора присваивания строки 10 можно ввести и двумя строками с номерами 10, 11.

● 55. Составить схему программы решения задачи 5. Допустимые напряжения среза, смятия и растяжения $[\tau]_{\text{ср}} = 100 \text{ МПа}$, $[\sigma]_{\text{см}} = 280 \text{ МПа}$, $[\sigma]_{\text{р}} = 160 \text{ МПа}$ задать в операторах присваивания. Исходные данные $P = 1,1 \cdot 10^5 \text{ Н}$, $i = 3$, $\delta = 12 \text{ мм}$, $b = 240 \text{ мм}$ ввести с клавиатуры дисплея.

Программа реализует схему алгоритма разветвленной структуры (см. рис. 5).

Хотя данная задача может быть решена и на микрокалькуляторах, это требует многих промежуточных вычислений, записей, сравнений.

Результат работы с использованием ЭВМ получается сразу в виде необходимых сообщений.

56. Составить программу решения задачи 6. Значения допустимых напряжений задать в операторах присваивания, как в программе задачи 55. Исходные данные $i = 5$, $\delta = 10 \text{ мм}$, $b = 100 \text{ мм}$ ввести с клавиатуры дисплея.

Программа реализует схему алгоритма разветвленной структуры (см. рис. 6).

```
75 REM ПРОГРАММА ОПРЕДЕЛЕНИЯ МАКСИМАЛЬНОЙ НАГРУЗКИ
10 TO=100 \ SO=280 \ RO=160
20 PRINT "ВВОД ИСХОДНЫХ ДАННЫХ: I,G,B"
30 INPUT I,G,B"
40 D=2*G
45 REM РАСЧЕТ ДОПУСТИМЫХ НАГРУЗОК
50 P1=PI*I*DI2*TO/4
60 P2=I*D*G*SO
70 P3=(B-I*D)*G*RO
75 REM ОПРЕДЕЛЕНИЕ МАКСИМАЛЬНОЙ НАГРУЗКИ
80 P=P1\MX="СРЕЗ"
90 IF P>P2 THEN 110
100 P=P2 \ MX="СМЯТИЕ"
110 IF P>P3 THEN 130
120 P=P3 \ MX="РАСТЯЖЕНИЕ"
125 REM ПЕЧАТЬ ВЫХОДНОГО СООБЩЕНИЯ
130 PRINT MX
140 PRINT "МАКС.ЗНАЧЕНИЕ НАГРУЗКИ P=" ; P ; "Н"
150 END
```

Основной частью программы являются операторы 80–120, определяющие наибольшую из нагрузок. Если максимальной окажется нагрузка P_1 , то значением символьной переменной $M \text{ } \text{\textcircled{X}}$ определяем слово "СРЕЗ"; если наибольшей окажется нагрузка P_2 , то $M \text{ } \text{\textcircled{X}} = \text{"СМЯТИЕ"}$; если P_3 , то $M \text{ } \text{\textcircled{X}} = \text{"РАСТЯЖЕНИЕ"}$.

57. Составить программу решения задачи 7. Значение $R = 100$ Ом задать в операторе присваивания. Исходные данные материал для провода, его удельное сопротивление ρ , диаметр провода d и максимальную длину катушки провода l_{\max} ввести с клавиатуры дисплея. Дополнительный идентификатор программы $l_{\max} \rightarrow L1$.

Программа реализует схему алгоритма разветвленной структуры (см. рис. 7) и может иметь вид:

```

05 REM ПРОГРАММА ПРОВЕРКИ КАТУШКИ ПРОВОДА
10 R=100
20 PRINT "ВВОД ИСХОДНЫХ ДАННЫХ: МАТЕРИАЛ, R0, D, L MAX"
30 INPUT M\X, R0, D, L1
40 L=PI*D12*R/(4*R0)
50 IF L<=L1 THEN 80
60 PRINT "ПРОВОД - МАТЕРИАЛ -"; M\X; " - НЕ ПОДХОДИТ"
70 GO TO 90
80 PRINT "ПРОВОД - МАТЕРИАЛ -"; M\X; " - ПОДХОДИТ"
90 STOP
100 END

```

Основная часть этой программы: операторы 50–80, проверяющие выполнение условия (14) и выдающие необходимые сообщения.

•58. Составить программу решения задачи 8. Значения допустимого напряжения на срез $[\tau]_{\text{ср}} = 100$ МПа и диаметр заклепок $d = 30$ мм задать в операторах присваивания. Изменяющиеся значения количества заклепок i от 1 до 20 задать в операторе цикла.

Результаты расчетов оформить в виде таблицы, заголовков которой имеет вид:

КОЛ. ЗАКЛЕПОК – I	ДОП. НАГРУЗКА – P
1-я зона	2-я зона

Таблица результатов представляет собой две колонки данных, расположенных в двух зонах. Программа реализует схему алгоритма циклической структуры (см. рис. 8).

•59. Составить программу решения задачи 9. Значение допустимого напряжения $[\sigma]_{\text{ср}} = 280$ МПа задать в операторе присваивания. Исходные данные – изменяющиеся значения P , δ , i задать операторами цикла.

Программа реализует схему алгоритма циклической структуры (см. рис. 9). Результаты расчетов выдать в виде сообщения:

НЕДОПУСТИМЫЕ ЗНАЧЕНИЯ ПАРАМЕТРОВ		
НАГРУЗКА	КОЛ. ЗАКЛЕПОК	ТОЛЩИНА ЛИСТА
1-я зона	2-я зона	3-я зона

60. Составить программу решения задачи 10. Значения $[\sigma]_p = 160$ МПа

и $b = 200$ мм задать в операторах присваивания; изменяющиеся значения P, δ, i задать операторами цикла.

Результаты расчетов оформить в виде таблицы, заголовков которой имеет вид:

НАГРУЗКА	ТОЛЩИНА ПОЛ.	КОЛ. ЗАКЛЕПОК
1-я зона	2-я зона	3-я зона
НАПР. РАСТЯЖ.	ПРИМЕЧАНИЕ	
4-я зона	5-я зона	

Таблица представляет собой пять колонок данных, расположенных в пяти зонах. Для строк с недопустимыми сочетаниями значений параметров исходных данных в 5-й колонке таблицы "ПРИМЕЧАНИЕ" поставить сообщение "ШОВ НЕ ПРОЧЕН".

Программа реализует схему алгоритма циклической структуры и имеет вид:

```
05 REM ПРОГРАММА РАСЧЕТА ЗАКЛЕПОЧНОГО ШВА
10 DIM M%(2)
20 M%(1)="ШОВ НЕ ПРОЧЕН" \ M%(2)=" "
30 R0=160 \ B=200
35 REM ПЕЧАТЬ ЗАГОЛОВКА ТАБЛИЦЫ
40 PRINT "НАГРУЗКА;","ТОЛЩИНА ПОЛ.", "КОЛ. ЗАКЛЕПОК",
      "НАПР. РАСТЯЖ.", "ПРИМЕЧАНИЕ"
50 FOR P=0.5E5 TO 3.5E5 STEP 0.5E5
60 FOR G=5 TO 20 STEP 5
70 FOR I=5 TO 50
80 D=2*G
90 R=P/(B-I*D)/G
100 IF R<=R0 THEN 120
110 I=1 ! ШОВ НЕ ПРОЧЕН \ GO TO 130
120 I=2 ! ШОВ ПРОЧЕН
130 PRINT P,G,I,R,M%(I)
140 NEXT I \NEXT G\ NEXT P
150 END
```

Оператор 10 объявляет две строки символьного массива с идентификатором $M\%$; в первую строку этого массива оператор 20 заносит текст "ШОВ НЕ ПРОЧЕН", вторая строка – пустая. Операторы 100–130 проверяют условие прочности шва для конкретных значений параметров исходных данных и в графе "ПРИМЕЧАНИЕ" выдают соответствующие сообщения. Тексты внутри строк 110 и 120 программы – "ШОВ НЕ ПРОЧЕН" и "ШОВ ПРОЧЕН" являются поясняющими комментариями.

•61. Составить программу решения задачи 11. Значение допускаемого напряжения $[\sigma]_p = 160$ МПа задать в операторе присваивания; исходные данные $P = 4,1 \cdot 10^5$ Н и $\delta = 12$ мм ввести с клавиатуры; полученное значение ширины листа в миллиметрах вывести на печать. Дополнительный идентификатор программы $[\tau]_{cp} \rightarrow T1$.

62. Составить программу решения задачи 12. Значение $[\sigma]_p = 160$ МПа задать в операторе присваивания. Исходные данные $P = 3,5 \cdot 10^5$ Н и $\delta = 8$ мм ввести с клавиатуры дисплея. Дополнительный идентификатор программы $[\sigma]_p' \rightarrow R1$.

Программа имеет вид:

```

05 REM ПРОГРАММА РАСЧЕТА СВАРНОГО УГЛОВОГО ШВА
10 R0=160 \ R1=0.6*R0
20 PRINT "ВВОД ИСХОДНЫХ ДАННЫХ: P,G"
30 INPUT P,G
40 B=P/(G*R1)
50 PRINT "B=" ; B
60 PRINT "ВВЕДИТЕ СКОРРЕКТИРОВАННОЕ ЗНАЧЕНИЕ B"
70 INPUT B
80 R=P/(B*G)
90 IF R<R0 THEN 110
100 PRINT "ШОВ НЕ ПРОЧЕН" \ GO TO 60
110 PRINT "ШОВ ПРОЧЕН "
120 PRINT "НАПР.РАСТЯЖЕНИЯ=" ; R
130 END

```

Особенностью программы является то, что она позволяет проектировщику работать с ЭВМ в *диалоговом режиме*. Сначала для введенных исходных данных рассчитывается величина B и оператором 50 выдается на печать:

$$B = 455.7$$

Оператор 60 выдает текст:

ВВЕДИТЕ СКОРРЕКТИРОВАННОЕ ЗНАЧЕНИЕ B

Введем, например, 480 мм. Тогда операторы 110–120 выдают сообщения:

ШОВ ПРОЧЕН

НАПР. РАСТЯЖЕНИЯ = 91.14

Если бы результат проверки шва на прочность оператором 90 оказался отрицательным, операторы строки 100 выдали бы сообщение "ШОВ НЕ ПРОЧЕН" и передали бы управление оператору 60 для ввода нового скорректированного значения ширины листа B .

●63. Составить программу для решения задачи 13. Значения допускаемого напряжения $[\sigma]_p = 160$ МПа и толщины листа $\delta = 16$ мм задать в операторах присваивания. Исходные данные — изменяющиеся величины длины шва b задать в операторе цикла.

Результаты расчетов оформить в виде таблицы, заголовков которой имеет вид:

ДЛИНА ШВА, ММ	ПРЕД. РАСТ. НАГРУЗКА, Н
1-я зона	2-я зона

Таблица представляет собой две колонки данных, расположенных в двух зонах. Дополнительный идентификатор программы $b^o \rightarrow B1$.

64. Составить программу решения задачи 14. Значение допускаемого напряжения $[\sigma]_p = 160$ МПа задать в операторе присваивания. Исходные данные $P = 8 \cdot 10^5$ Н, $\delta = 20$ мм, $b = 30$ мм ввести с клавиатуры дисплея.

Дополнительные идентификаторы программы:

Математическое обозначение , . . .	$[\tau]_{\text{ср}}$	$[\sigma]_{\text{р}}$	$P_{\text{угл}}$	$P_{\text{стык}}$
Идентификатор	T1	R1	P1	P2

Программа может иметь вид:

```

05 REM ПРОГРАММА ВЫБОРА ТИПА СВАРНОГО ШВА
10 RO=160 \ T1=0,5*RO \ R1=0,6*RO
20 PRINT "ВВОД ИСХОДНЫХ ДАННЫХ: P, G, B"
30 INPUT P,G,B
40 P1=1.4*B*G*T1
50 P2=B*G*R1
55 REM ПРОВЕРКА УСЛОВИЙ
60 IF P<=P1 THEN 90
70 IF P<=P2 THEN 100
80 PRINT "ШВЫ НЕ ПРОЧНЫ" \ GO TO 110
90 PRINT "УГЛОВОЙ ИЛИ СТЫКОВОЙ ШОВ" \ GO TO 110
100 PRINT "СТЫКОВОЙ ШОВ"
110 PRINT "P УГЛ.-"; P1, "P СТЫК.-"; P2, "НАГРУЗКА P -"; P
120 END

```

Операторы 60—110 проверяют условия нагрузок согласно рис. 12 и выдают соответствующие сообщения.

●65. Составить программу решения задачи 15. Значение допускаемого напряжения $[\sigma]_{\text{р}} = 100$ МПа задать в операторе присваивания. Для задания различных значений нагрузки P ($3 \cdot 10^4$, $5 \cdot 10^4$; $7 \cdot 10^4$; $1,1 \cdot 10^5$; $1,5 \times 10^5$ Н) использовать оператор DATA.

●66. Составить программу решения задачи 16. Значения допускаемых напряжений $[\sigma]_{\text{р}} = 100$ МПа и $[\sigma]_{\text{см}} = 200$ МПа задать в операторах присваивания. Величину нагрузки $P = 5 \cdot 10^4$ Н ввести с клавиатуры дисплея.

●67. Составить программу решения задачи 17. Значения допускаемых напряжений $[\tau]_{\text{ср}} = 60$ МПа и $[\sigma]_{\text{см}} = 200$ МПа задать в операторах присваивания. Исходные данные $P = 5,1 \cdot 10^4$ Н и значение d_0 , мм, полученное в задаче 66, ввести с клавиатуры. Дополнительный идентификатор программы: Д → D1.

Полученные параметры клинового соединения выдать на печать зонами — по 10 символов в каждой зоне.

Оператор выдачи значений выходных пяти параметров будет иметь вид:

```
PRINT B; TAB(10)LØ; TAB(20)G; TAB(30)D1; TAB(40)C
```

68. Составить программу решения задачи 18. Значения $P = 5,1 \cdot 10^4$ Н и $[\sigma]_{\text{и}} = 140$ МПа задать в операторе присваивания. Исходные данные ввести с клавиатуры. Дополнительный идентификатор программы: $[\sigma]_{\text{и}} \rightarrow S1$.

●69. Составить программу решения задачи 19. Допускаемые напряжения $[\tau]_{\text{ср}} = 110$ МПа и $[\sigma]_{\text{см}} = 200$ МПа задать в операторах присваивания. Исходные данные $d_{\text{в}} = 20$ мм, $d_{\text{ш}} = 1$ мм, $M = 24$ Н · м ввести с клавиатуры дисплея.

По результатам проверки выдать сообщения в виде текстов:

№ 1 — "УСЛОВИЕ ПРОЧНОСТИ ШТИФТА НА СМЯТИЕ — ВЫПОЛНЕНО"
 № 2 — "УСЛОВИЕ ПРОЧНОСТИ ШТИФТА НА СМЯТИЕ — НЕ ВЫПОЛНЕНО"

№ 3 – "УСЛОВИЕ ПРОЧНОСТИ ШТИФТА НА СРЕЗ – ВЫПОЛНЕНО"
 № 4 – "УСЛОВИЕ ПРОЧНОСТИ ШТИФТА НА СРЕЗ – НЕ ВЫПОЛНЕНО"

Напечатать также значения расчетных и допускаемых напряжений. Дополнительные идентификаторы программы:

$$l_{ш} \rightarrow L1, d_{ш} \rightarrow D1, d_{в} \rightarrow D2$$

●70. Составить программу решения задачи 20. Допускаемое напряжение $[\sigma]_{см} = 120$ МПа и конструктивные размеры шпонки $b = 5$ мм и $k = 2,3$ мм задать в операторах присваивания. Значение диаметра вала $d = 16$ мм ввести с клавиатуры дисплея. Для задания различных значений крутящего момента M (12; 20; 26; 30; 50 Н·м) использовать оператор DATA. Дополнительный идентификатор программы $l_p \rightarrow L1$.

71. Составить программу решения задачи 21. Допускаемое напряжение $[\sigma]_p = 72$ МПа задать в операторе присваивания. Значение нагрузки $P = 35$ кН ввести с клавиатуры.

Для работы программы ввести в ЭВМ табл. 3.

Т а б л и ц а 3. Стандартные размеры метрических резьб (ГОСТ 9150–81)

Тип резьбы – метрическая	M10	M12	M16	M20	M24	M28
Внутренний диаметр, мм	8,918	10,918	14,918	18,918	22,918	26,918
Тип резьбы – метрическая	M30	M36	M42	M48	M56	M64
Внутренний диаметр, мм	28,918	34,918	40,918	46,918	54,918	62,918

Программа может иметь вид:

```

05 REM ПРОГРАММА РАСЧЕТА РЕЗЬБОВЫХ СОЕДИНЕНИЙ
10 DIM M(12),D(12)
20 RO=72
30 PRINT "ВВОД ИСХОДНЫХ ДАННЫХ: НАГРУЗКА P"
40 INPUT P
50 PRINT "ВВОД ТАБЛИЦЫ"
60 FOR I=1 TO 12
70 PRINT "ВВОД СТРОКИ -"; I
80 INPUT M(I),D(I)
90 NEXT I
95 REM ВЫЧИСЛЕНИЕ РАСЧЕТНОГО ДИАМЕТРА РЕЗЬБЫ
100 DO=1,13*(SQR(P/RO))
105 REM ПОИСК В ТАБЛИЦЕ ЗНАЧЕНИЯ ВНУТР. ДИАМЕТРА РЕЗЬБЫ
110 FOR I=1 TO 12
120 IF D(I)<DO THEN 150
130 D=D(I)
140 GO TO 160 ! ВЫХОД ИЗ ЦИКЛА
150 NEXT I
155 REM ПЕЧАТЬ ВЫХОДНОГО СООБЩЕНИЯ
160 PRINT "РЕЗЬБА-"; M(I)
170 PRINT "ВНУТР. ДИАМЕТР РЕЗЬБЫ D="; D ; "ММ"
180 END

```

В операторе 10 DIM — объявляются два массива $M\emptyset$ и D , предназначенные для хранения данных табл. 3. Значениями символьного массива $M\emptyset$ будут значения первой строки таблицы — тип метрической резьбы: M10, M12, ..., M64. Элементами массива D будут соответствующие значения внутреннего диаметра резьбы d , мм: 8,918; 10,918; ..., 62,918. Ввод значений табл. 3 в ЭВМ осуществляется операторами 50—90.

●72. Составить программу решения задачи 22. Данная программа является развитием предыдущей задачи. Различные значения нагрузки P задать в операторе цикла.

Результаты расчетов оформить в виде таблицы, заголовков которой имеет вид:

НАГРУЗКА — P , КН РАСЧ. ДИАМЕТР — $D\emptyset$, ММ ВНУТР. ДИАМ. РЕЗЬБЫ — D , ММ
 1-я зона 2-я зона 3-я зона

Таблица представляет собой три колонки данных, расположенных в трех зонах.

●73. Составить программу решения задачи 24. Исходные данные: $l = 170$ мм задать в операторе присваивания; значения $F_t = 4500$ Н и $F_r = 1640$ Н ввести с клавиатуры дисплея.

Результатом работы программы является значение результирующего изгибающего момента на вал $M_{и}$.

Идентификаторы к программам 73—75:

Математическое обозначение	F_t	F_r	M_x	M_y	$M_{и}$	$M_{экв}$
Идентификатор	F1	F2	M1	M2	M3	M \emptyset

●74. Составить программу решения задачи 24. Значение $l = 170$ мм задать в операторе присваивания, изменяющиеся нагрузки F_t и F_r задать в операторах цикла.

Результаты расчетов оформить в виде таблицы, заголовков которой имеет вид:

НАГРУЗКА — F_t НАГРУЗКА — F_r ИЗГИБ. МОМЕНТ — $M_{и}$
 1-я зона 2-я зона 3-я зона

Таблица представляет собой три колонки данных, расположенных в трех зонах.

●75. Составить программу решения задачи 25. Значения $T = 180$ Н·мм и $l = 170$ мм задать в операторах присваивания; изменяющиеся нагрузки F_t и F_r задать в операторах цикла.

Результаты расчетов оформить в виде таблицы, заголовков которой имеет вид:

НАГРУЗКА — F_t НАГРУЗКА — F_r ЭКВИВ. МОМЕНТ — $M_{экв}$
 1-я зона 2-я зона 3-я зона

●76. Составить программу решения задачи 26. Значения $d = 0,5$ мм и $R = 40$ Ом задать в операторах присваивания; значение ρ для никелинового провода взять из табл. 4 и ввести с клавиатуры дисплея.

Результатом работы программы является сообщение:

ДЛИНА ПРОВОДА = М

●77. Составить программу решения задачи 27. Значение $l = 30$ м и $R = 100$ Ом задать в операторах присваивания, значение ρ для провода из серебра ввести с клавиатуры дисплея.

●78. Составить программу решения задачи 28. Значения $l = 30$ м и $d = 0,1$ мм задать в операторах присваивания. Исходные данные материал провода и его удельное сопротивление ρ ввести с клавиатуры дисплея.

Расчет провести для проводов из 10 различных материалов, используя табл. 4.

Т а б л и ц а 4

Материал	ρ , Ом · мм ² /м	Материал	ρ , Ом · мм ² /м
Серебро	0,016	Сталь	0,1
Медь	0,017	Свинец	0,21
Алюминий	0,026	Никелин	0,42
Вольфрам	0,055	Константан	0,5
Цинк	0,06	Нихром	1,05

Результаты расчетов оформить в виде таблицы, заголовок которой имеет вид:

МАТЕРИАЛ	СОПРОТИВЛЕНИЕ, ОМ
1-я зона	2-я зона

Таблица представляет собой две колонки данных, расположенных в двух зонах.

Программа может иметь вид:

```

05 REM ПРОГРАММА РАСЧЕТА СОПРОТИВЛЕНИЯ ПРОВОДОВ
10 L=30 \ D=0.1
20 PRINT "ВВОД ИСХОДНЫХ ДАННЫХ: МАТЕРИАЛ, УД.СОПРОТИВЛЕНИЕ
30 INPUT M$,RO
40 IF M$="КОНЕЦ" THEN 100
50 S=PI*D^2/4
60 R=RO*L/S
65 REM ПЕЧАТЬ РЕЗУЛЬТАТОВ
70 PRINT "МАТЕРИАЛ", "СОПРОТИВЛЕНИЕ, ОМ"
80 PRINT M$,R
90 GO TO 20
100 PRINT "КОНЕЦ СЕРИИ"
110 END

```

Проведя расчеты для всех значений табл. 4, необходимо вместо типа очередного материала ввести слово "КОНЕЦ", тогда условный оператор 40 передает управление оператору 100, по которому данная программа печатает сообщение "КОНЕЦ СЕРИИ" и заканчивает работу.

●79. Составить программу решения задачи 29. Значение $l = 1$ м задать в операторе присваивания; материал провода и его удельное сопротивление

ввести с клавиатуры дисплея. Для задания различных значений диаметра проводов d (0,05; 0,08; 0,10; 0,12; 0,15; 0,20; 0,25; 0,30; 0,40; 0,50; 0,60; 0,70; 0,80; 1,0 мм) использовать оператор DATA.

●80. Составить программу решения задачи 30. Значение $\delta = 2 \text{ А/мм}^2$ задать в операторе присваивания; изменяющиеся значения тока T, A , задать в операторе цикла. Дополнительный идентификатор программы: $\delta \rightarrow G$.

●81. Составить программу решения задачи 31. Исходные данные $U_{\text{н}} = 200 \text{ В}$, $I = 20 \text{ мА}$, $I_2 = 10 \text{ мА}$ задать в операторе присваивания; изменяющиеся значения напряжения U задать в операторе цикла. Дополнительный идентификатор программы $U_{\text{н}} \rightarrow U1$.

●82. Составить программу решения задачи 32.

Исходные данные $D = 2 \text{ см}$, $a = 4 \text{ см}$, $w = 50$ ввести с клавиатуры дисплея.

●83. Составить программу решения задачи 33. Исходные данные $D = 2 \text{ см}$, $a = 3 \text{ см}$, $L = 10 \text{ мкГ}$ ввести с клавиатуры дисплея. Дополнительный идентификатор программы $d_{\text{из}} \rightarrow D\emptyset$.

●84. Составить программу решения задачи 34. Значение $L = 10 \text{ мкГ}$ задать в операторе присваивания; изменяющиеся значения диаметра катушки индуктивности D задать в операторе цикла.

●85. Составить программу решения задачи 35. Исходные данные $D_{\text{min}} = 1,4 \text{ см}$, $D_{\text{max}} = 2,6 \text{ см}$, $b = 1,1 \text{ см}$, $w = 200$ ввести с клавиатуры дисплея.

Идентификаторы к программам 85–86:

Математическое обозначение	D_{min}	D_{max}	$d_{\text{из}}$
Идентификатор	D1	D2	D \emptyset

●86. Составить программу решения задачи 36. Исходные данные $L = 600 \text{ мкГ}$, $D_{\text{min}} = 1,5 \text{ см}$, $d_{\text{из}} = 0,5 \text{ мм}$, $b = 1 \text{ см}$ ввести с клавиатуры дисплея.

Эта программа особенно наглядно показывает достоинства вычислительной техники. Аналитическое решение уравнения 4-й степени (62) для нахождения искомой величины w очень сложно. Составим программу, которая позволит учащемуся решить эту задачу, работая с ЭВМ в диалоговом режиме.

```

05 REM ПРОГРАММА РАСЧЕТА КОЛИЧЕСТВА ВИТКОВ МНОГОСЛОЙНОЙ
06 REM КАТУШКИ ИНДУКТИВНОСТИ
10 PRINT "ВВОД ИСХОДНЫХ ДАННЫХ L,D1,DO,B"
20 INPUT L,D1,DO,B
30 REM РАСЧЕТ НАЧАЛЬНОГО ЗНАЧЕНИЯ W0
40 W0=10*SQR(L/D1)
50 PRINT W0
60 PRINT "ВВЕДИТЕ СКОРРЕКТИРОВАННОЕ ЗНАЧЕНИЕ W"
70 INPUT W
80 P1=0.08*(DO^2*W/B+D1)^2*W^2
90 P2=3*(DO^2*W/B+D1)+10*DO^2*W/B+9*B
100 P=L*P2-P1
110 PRINT "ОТКЛОНЕНИЕ =" ;P,"W =" ;W
120 GOTO 60
130 END

```

Работа с программой в диалоговом режиме заключается в том, что операторами 60, 70 учащийся последовательно вводит новые скорректированные значения w до тех пор, пока отклонение не сделается минимальным — близким к нулю. Значение w , при котором получено это минимальное отклонение, и будет искомым решением задачи.

●87. Составить программу решения задачи 37. Исходные данные $U_1 = 127$ В, $U_2 = 220$ В, $I_2 = 0,5$ А, $\eta = 0,8$.

Идентификаторы программы:

Математическое обозначение	η	d_1	d_2	$d_{из1}$	$d_{из2}$	S'
Идентификатор	E	D1	D2	D3	D4	S

●88. Составить программу решения задачи 38. Исходные данные: $C_1 = C_2 = 20$ мкФ = $20 \cdot 10^{-6}$ Ф, $L = 40$ Г, $U_0 = 25$ В, $R = 5$ кОм = $5 \cdot 10^3$ Ом, $f_{\Pi} = 100$ Гц.

Идентификаторы к программам 88—90:

Математическое обозначение	$U_{m1вх}$	$U_{m1вых}$	U_0	K_{ϕ}	f_{Π}	R_{ϕ}	δ
Идентификатор	U1	U2	U0	K	F	R0	G

●89. Составить программу решения задачи 39. Исходные данные: $f_{\Pi} = 200$ Гц, $P = 5\% = 0,05$. Значения сопротивления нагрузки R изменяются в пределах от 1 до 20 кОм с шагом 1 кОм.

Результаты расчетов оформить в виде таблицы, заголовок которой имеет вид:

R — КОМ	C — МКФ
1-я зона	2-я зона

●90. Составить программу решения задачи 40. Исходные данные: $C_1 = 20$ мкФ, $C_2 = 30$ мкФ, $R_{\phi} = 2$ кОм, $U_0 = 30$ В, $R = 6$ кОм, $f_{\Pi} = 100$ Гц.



ОТВЕТЫ

31. Расчетные соотношения:

сопротивления делителя

$$R_1 = (U_{\text{н}} - U)/(I + I_2),$$

$$R_2 = U/I_2;$$

рассеиваемые мощности

$$P_1 = (U_{\text{н}} - U)/(I + I_2),$$

$$P_2 = UI_2.$$

33. Формула для расчета:

$$W = 10\sqrt{L(0,44 + a/D)/D}.$$

41. Ошибки в записи числовых констант: 4), 5) – число (константа) в показательной форме не может начинаться с буквы, перед показателем степени должно стоять число; 7), 9) – в записи числа знак операций не допускается; 11) – порядок числа в показательной форме должен быть только целым; 12) – в записи числа на алгоритмическом языке целая часть числа отделяется от дробной не запятой, а десятичной точкой.

42. Ошибки в записи идентификаторов, имен объектов БЕЙСИК-программ: 3), 5) – второй символ не цифра; 6), 15) – начинается с цифры; 7) – в имени используется буква русского алфавита; 9) – слишком длинное имя, допускается только два символа; 10) – знак "⌀" должен быть последним символом идентификатора символьной переменной; 13), 14) – никакие символы, кроме латинских букв, цифр и знака "⌀", в записи идентификаторов БЕЙСИК-программ не допускаются.

43. Значение арифметического выражения равно 8.

44.

- 1) $A \cdot X + B \cdot Y + C \cdot Z$
- 2) $(X + Y) \uparrow 3$
- 3) $X \uparrow 2 / A \uparrow 2 + Y \uparrow 2 / B \uparrow 2 + Z \uparrow 2 / C \uparrow 2$
- 4) $Y + X \uparrow 4 / (2 \cdot B) - 1.5$
- 5) $R / R \uparrow 1 + C \cdot D / (R \cdot R \uparrow 1 \cdot H)$
- 6) $\text{SIN}(X) \uparrow 2 + \text{COS}(X) \uparrow 2$
- 7) $A \uparrow (3/2) + \text{SQR}(A - B)$
- 8) $(X - Y) \cdot \text{SIN}(X) + (X \uparrow 3 + Y \uparrow 3) \cdot \text{TAN}(X)$
- 9) $((\text{EXP}(X) + \text{EXP}(-X)) / 2) \uparrow 2 + ((\text{EXP}(X) - \text{EXP}(-X)) / 2) \uparrow 2$
- 10) $X \cdot Y \cdot \text{SQR}(X \uparrow 2 + Y \uparrow 2) - \text{LOG}(\text{ABS}(X - \text{SQR}(1 + X \uparrow 2)))$

45

- 1) $S=2 \cdot \pi \cdot R$
- 2) $V=1/3 \cdot \pi \cdot R^2 \cdot H$
- 3) $R^2=2 \cdot S \cdot \cos(T)/(2 \cdot \pi \cdot X)$
- 4) $D=\sqrt{X^2+Y^2+Z^2}$
- 5) $Y=\sqrt{X+1}+1/(X-1)$
- 6) $Y=\sqrt{X} \cdot \sin(X)/(X+\exp(X))$
- 7) $I=V/\sqrt{X^2+(C \cdot P-X \cdot C)^2}$

46.

ЗНАЧЕНИЯ ПЕРЕМЕННЫХ ПОСЛЕ ВЫПОЛНЕНИЯ ПРОГРАММ:

- 1) $Y1=0$
- 2) $Y2=16.69$

49.

ОПЕРАТОР ОПИСАНИЯ МАССИВОВ:
DIM F(5), G(15), M(10)

52.

```
05 REM ПРОГРАММА РАСЧЕТА НА ПРОЧНОСТЬ ЗАКЛЕПОЧНОГО ШВА
10 PRINT "ВВОД ИСХОДНЫХ ДАННЫХ: P,I,D,G,B"
20 INPUT P,I,D,G,B
30 S=P/(I*D*G)
40 R=P/(B-I*D)/G
50 PRINT "НАПРЯЖЕНИЕ НА СМЯТИЕ ="; S; "МПА"
60 PRINT "НАПРЯЖЕНИЕ НА РАСТЯЖЕНИЕ ="; R; "МПА"
70 END
```

ОТВЕТ:
НАПРЯЖЕНИЕ НА СМЯТИЕ = 127.315 МПА
НАПРЯЖЕНИЕ НА РАСТЯЖЕНИЕ = 54.5635 МПА

53.

```
05 REM ПРОГРАММА РАСЧЕТА ПАРАМЕТРОВ ЗАКЛЕПОЧНОГО ШВА
10 TO=100
20 PRINT "ВВОД ИСХОДНЫХ ДАННЫХ: P,I,D"
30 INPUT P,I,D
40 L=P/(I*D*TO)
45 REM ОКРУГЛЕНИЕ "L" В БОЛЬШУЮ СТОРОНУ
46 REM И ВЫДЕЛЕНИЕ ЦЕЛОЙ ЧАСТИ
50 L=INT(L+1)
60 PRINT "РАССТ.ЗАКЛЕПОК ОТ ПОПЕРЕЧНОЙ КРОМКИ ПОЛОС =";L; "ММ"
70 END
```

ОТВЕТ:

РАССТ.ЗАКЛЕПОК ОТ ПОПЕРЕЧНОЙ КРОМКИ ПОЛОС = 16 ММ

55.

```
05 REM ПРОВЕРКА ЗАКЛЕПОЧНОГО ШВА НА ПРОЧНОСТЬ
10 TO=100 \ SO=280 \ RO=160
20 PRINT "ВВОД ИСХОДНЫХ ДАННЫХ: P,I,G,B"
30 INPUT P,I,G,B
40 D=2*G
50 T=4*P/(PI*I*D^2)
60 S=P/(I*D*G)
70 R=P/(B-I*D)/G
80 IF T>TO THEN 120
90 IF S>SO THEN 120
100 IF R>RO THEN 120
110 PRINT "ШОВ ПРОЧЕН" \ GO TO 130
120 PRINT "ШОВ НЕ ПРОЧЕН"
130 PRINT "РАСЧЕТНЫЕ НАПРЯЖЕНИЯ"
140 PRINT "НАПР.СРЕЗА=";T, "НАПР.СМЯТИЯ=";S, "НАПР.РАСТ.=";P
150 END
```

58.

```
05 REM ПРОГРАММА РАСЧЕТА НАПРЯЖЕНИЙ НАГРУЗКИ
10 TO=100 \ D=30
15 REM ПЕЧАТЬ ЗАГОЛОВКА ТАБЛИЦЫ
20 PRINT "КОЛ.ЗАКЛЕПОК - I", "ДОП.НАГРУЗКА - P"
30 K=PI*D^2*TO/4
40 FOR I=1 TO 20
50 P=K*I
60 PRINT I,P
70 NEXT I
80 END
```

59.

```
05 REM ПРОГРАММА ОПРЕДЕЛЕНИЯ НЕДОПУСТИМЫХ
06 REM СОЧЕТАНИЙ ПАРАМЕТРОВ
10 SO=280
15 REM ПЕЧАТЬ ЗАГОЛОВКА ТАБЛИЦЫ
20 PRINT "НЕДОПУСТИМЫЕ ЗНАЧЕНИЯ ПАРАМЕТРОВ"
30 PRINT "НАГРУЗКА", "КОЛ.ЗАКЛЕПОК", "ТОЛЩИНА ЛИСТА"
40 FOR P=0.5E5 TO 3.5E5 STEP 0.5E5
50 FOR G=5 TO 20 STEP 5
60 D=2*G
70 FOR I=6 TO 50
80 S=P/(I*D*G)
90 IF S<=SO THEN 110
100 PRINT P,I,G
110 NEXT I \ NEXT G \ NEXT P
120 END
```

61.

```
05 REM ПРОГРАММА РАСЧЕТА СВАРНОГО УГЛОВОГО ЛОБОВОГО ШВА
10 RO=160 \ T1=0.5*RO
20 PRINT "ВВОД ИСХ.ДАННЫХ: P,G"
30 INPUT P,G
```

```

40 B=P/(1.4*G*T1)
50 PRINT "B="; INT(B); "MM"
60 END

```

ОТВЕТ:

B= 305 MM

63.

ДЛИНА ШВА, MM : ПРЕД.РАСТ.НАГРУЗКА, Н

150	:	384000
200	:	512000
250	:	640000
300	:	768000
350	:	896000
400	:	1.024E+06
450	:	1.152E+06
500	:	1.280E+06

65.

НАГРУЗКА - P : ДИАМЕТР D

30000	:	20
50000	:	26
70000	:	30
110000	:	38
150000	:	44

66.

```

05 REM ПРОГРАММА РАСЧЕТА УТОЛЩЕННОЙ ЧАСТИ СТЕРЖНЯ
10 SO=200 \ RO=100
20 PRINT "ВВЕСТИ ЗНАЧ. НАГРУЗКИ - P"
30 INPUT P
40 DO=SQR(4*P/(PI-1)/RO)
50 PRINT "DO="; DO; "MM"
60 PRINT "ВВЕСТИ СКОРРЕКТ ЗНАЧЕНИЕ DO"
70 INPUT DO
80 B=DO/4
90 S=P/(DO*B)
100 IF S<=SO THEN 130
110 PRINT "СОЕДИНЕНИЕ НЕ ПРОЧНО, СНОВА СКОРРЕКТИРУЙТЕ DO"
120 GOTO 70
130 PRINT " СОЕДИНЕНИЕ ПРОЧНО", " ДИАМЕТР DO="; DO; "MM"
140 END

```

67.

```

05 REM ПРОГРАММА РАСЧЕТА ПАРАМЕТРОВ КЛИНОВОГО СОЕДИНЕНИЯ
10 TO=60 \ SO=200

```

```

20 PRINT "ВВОД ИСХОДНЫХ ДАННЫХ : P, DO"
30 INPUT P,DO
40 B=DO/4
50 LO=P/(2*DO*TO)
60 PRINT "LO=";LO; "ММ"
70 PRINT "ВВЕДИТЕ СКОРЕКТ.ЗНАЧЕНИЕ LO "
80 INPUT LO
90 G=P/(2*B*SO)
100 PRINT "G="; G; "ММ"
110 PRINT "ВВЕДИТЕ СКОРЕКТ.ЗНАЧЕНИЕ G"
120 INPUT G
130 D1=DO+2*G
140 C=P/(4*G*TO)
150 PRINT "C=";C;"ММ"
160 PRINT "ВВЕДИТЕ СКОРЕКТ.ЗНАЧЕНИЕ C"
170 INPUT C
180 PRINT "ПАРАМЕТРЫ КЛИНОВОГО СОЕДИНЕНИЯ: "
190 PRINT "B"; TAB(10)"LO"; TAB(20)"G"; TAB(30)"D"; TAB(40)"C"
200 PRINT B ; TAB(10)LO; TAB(20)G; TAB(30)D1; TAB(40)C
210 END

```

69.

```

05 REM ПРОГРАММА РАСЧЕТА ПРОЧНОСТИ ШТИФТА
10 SO=200 \ TO=110
20 PRINT "ВВОД ИСХОДНЫХ ДАННЫХ : M,D1,D2"
30 INPUT M,D1,D2
40 L1=1.2*D2
50 S=4*M/(D2*D1*L1)
60 T=2*M/(D2*D1*L1)
65 REM ПРОВЕРКА УСЛОВИЙ И ФОРМИРОВАНИЕ ВЫХОДНЫХ СООБЩЕНИЙ
70 IF S<=SO THEN 100
80 PRINT "УСЛОВИЕ ПРОЧНОСТИ ШТИФТА НА СМЯТИЕ НЕ ВЫПОЛНЕНО"
90 GOTO 110
100 PRINT "УСЛОВИЕ ПРОЧНОСТИ ШТИФТА НА СМЯТИЕ - ВЫПОЛНЕНО"
110 IF T<=TO THEN 140
120 PRINT "УСЛОВИЕ ПРОЧНОСТИ ШТИФТА НА СРЕЗ НЕ ВЫПОЛНЕНО"
130 GOTO 150
140 PRINT "УСЛОВИЕ ПРОЧНОСТИ ШТИФТА НА СРЕЗ - ВЫПОЛНЕНО"
150 PRINT "НАПР.СМЯТИЯ = "; S;"НАПР.СРЕЗА";T
160 END

```

70.

```

05 REM ПРОГРАММА РАСЧЕТА ШПОНОЧНОГО СОЕДИНЕНИЯ
10 SO=120 \ B=5 \ K=2.3
20 PRINT "ВВОД ИСХОДНЫХ ДАННЫХ :D"
30 INPUT D
40 DATA 12,20,26,30,50
50 L=1.2*D \ L1=L-B
60 READ M
70 S=2*M*1E3/(D*L1*K)
80 IF S<=SO THEN 110
90 PRINT "M="; M, " СОЕДИНЕНИЕ НЕ ПРОЧНО"
100 GOTO 120
110 PRINT "M=";M; "СОЕДИНЕНИЕ ПРОЧНО"
120 END

```

72.

НАГРУЗКА - P, кН : РАСЧ. ДИАМЕТР, мм : ВНУТР. ДИАМЕТР РЕЗЬБЫ, мм

10	:	13.3172	:	M16	14.918
20	:	18.8333	:	M20	18.918
30	:	23.0660	:	M28	26.918
40	:	26.6344	:	M28	26.918
50	:	29.7781	:	M36	34.918
60	:	32.6203	:	M36	34.918
70	:	35.2339	:	M42	40.918
80	:	37.6667	:	M42	40.918
90	:	39.9515	:	M42	40.918
100	:	42.1126	:	M48	46.918

73.

```

05 REM ПРОГРАММА РАСЧЕТА ИЗГИБАЮЩЕГО МОМЕНТА
10 L=170
20 PRINT "ВВОД ИСХОДНЫХ ДАННЫХ: F1,F2"
30 INPUT F1,F2
40 M1=F1*L/4
50 M2=F2*L/4
60 M3=SQR(M1^2+M2^2)
70 PRINT "ИЗГИБ.МОМЕНТ-"; M3;"Н.ММ"
80 END

```

74.

```

05 REM ПРОГРАММА РАСЧЕТА ИЗГИБАЮЩЕГО МОМЕНТА НА ВАЛ
10 L=170
15 REM ПЕЧАТЬ ЗАГОЛОВКА ТАБЛИЦЫ
20 PRINT "НАГРУЗКА - FT", "НАГРУЗКА - FR", "ИЗГИБ.МОМЕНТ -МИ"
30 FOR F1=3000 TO 8000 STEP 500
40 FOR F2=1500 TO 3000 STEP 500
50 M1=F1*L/4
60 M2=F2*L/4
70 M3=SQR(M1^2+M2^2)
80 PRINT F1,F2,M3
90 NEXT F2 \ NEXT F1
100 END

```

75.

```

05 REM ПРОГРАММА РАСЧЕТА ЭКВИВАЛЕНТНОГО МОМЕНТА
10 T=180 \ L=170
15 REM ПЕЧАТЬ ЗАГОЛОВКА ТАБЛИЦЫ
20 PRINT "НАГРУЗКА - FT", "НАГРУЗКА - FR", "ЭКВИВ.МОМЕНТ - МЭКВ"
30 FOR F1=3000 TO 8000 STEP 500
40 FOR F2=1500 TO 3000 STEP 500
50 M1=F1*L/4

```

```

60 M2=F2*L/4
70 MO=SQR(M1^2+M2^2+0.75*T^2)
75 REM ПЕЧАТЬ РЕЗУЛЬТАТОВ
80 PRINT F1,F2,MO
90 NEXT F2 \ NEXT F1
100 END

```

76.

```

05 REM ПРОГРАММА РАСЧЕТА ДЛИНЫ ПРОВОДА
10 D=0.5 \ R=40
20 PRINT "ВВОД ИСХ. ДАННЫХ: R0"
30 INPUT R0
40 L=PI*D^2*R/(4*R0)
50 PRINT "ДЛИНА ПРОВОДА = "; L; "М"
60 END

```

ОТВЕТ:
ДЛИНА ПРОВОДА - 18.9995 М

77.

D=0.0782 мм
БЛИЖАЙШЕЕ СТАНДАРТНОЕ ЗНАЧЕНИЕ: D=0.08 мм

78.

2 *- R0*

МАТЕРИАЛ	:	СОПРОТИВЛЕНИЕ , Ом
1 СЕРЕБРО	:	61.11534
2 МЕДЬ	:	64.93506
3 АЛЮМИНИЙ	:	99.31244
4 ВОЛЬФРАМ	:	210.0840
5 ЦИНК	:	229.1825
6 СТАЛЬ	:	381.9709
7 СВИНЕЦ	:	802.1389
8 НИКЕЛИН	:	1604.278
9 КОНСТАНТАН	:	1909.855
10 НИХРОМ	:	4010.695

X серебро

79.

```

05 REM ПРОГРАММА РАСЧЕТА СОПРОТИВЛЕНИЯ 1 М ПРОВОДА
10 L=1
20 DATA 0.05,0.08,0.10,0.12,0.15,0.20,0.25,0.30
21 DATA 0.40,0.50,0.60,0.70,0.80,1.0
30 PRINT "ВВОД ИСХОДНЫХ ДАННЫХ: МАТЕРИАЛ, УД. СОПРОТИВЛЕНИЕ"
40 INPUT M,R0
45 REM ПЕЧАТЬ ЗАГОЛОВКА ТАБЛИЦЫ

```

```

50 PRINT "МАТЕРИАЛ -"; M
60 PRINT "ДИАМЕТР ПРОВОДА", "СОПР. 1 М ДЛИНЫ"
70 READ D
80 S=PI*D^2/4
90 R=RO*L/S
95 REM ПЕЧАТЬ РЕЗУЛЬТАТОВ
100 PRINT D,R
110 GO TO 70
120 END

```

80.

ЗНАЧЕНИЕ ТОКА, А : ДИАМЕТР ПРОВОДА, ММ

0.1	:	0.253
0.2	:	0.357
0.3	:	0.437
0.4	:	0.505
0.5	:	0.565
0.6	:	0.619
0.7	:	0.668
0.8	:	0.714
0.9	:	0.758
1.0	:	0.799

81.

```

05 REM ПРОГРАММА РАСЧЕТА ДЕЛИТЕЛЯ ЦЕПИ ПОСТОЯННОГО ТОКА
10 U1=200 \ I=20 \ I2=10
15 REM ПЕЧАТЬ ЗАГОЛОВКА
20 PRINT "R1,КОМ", "R2,КОМ", "P1,ВТ", "P2,ВТ"
30 FOR U=20 TO 100 STEP 10
40 R1=(U1-U)/(I+I2)
50 R2=U/I2
60 P1=(I+I2)*(U1-U)/1E3
70 P2=I2*U/1E3
75 REM ПЕЧАТЬ РЕЗУЛЬТАТОВ
80 PRINT R1, R2, P1, P2
90 NEXT U
100 END

```

82.

L=20.49 МКГ

83.

```

05 REM ПРОГРАММА РАСЧЕТА ЧИСЛА ВИТКОВ КАТУШКИ ИНДУКТИВНОСТИ
10 PRINT "ВВОД ИСХОДНЫХ ДАННЫХ: D,A,L"
20 INPUT D,A,L
30 W=10*SQR(L*(0.44+A/D)/D)
40 DO=A/W
45 REM ПЕЧАТЬ РЕЗУЛЬТАТОВ
50 PRINT "W=", W, "D ИЗОЛ.=", DO
60 END

```

84.

```
05 REM ПРОГРАММА РАСЧЕТА КАТУШКИ ИНДУКТИВНОСТИ
06 REM С НАИМЕНЬШИМ АКТИВНЫМ СОПРОТИВЛЕНИЕМ
10 L=10
20 PRINT "D", "A", "W", "D ИЗОЛ."
30 FOR D=1 TO 10
40 W=10*SQR(0.84*L/D)
50 A=0.4*D
60 DO=A/W
70 PRINT D,A,W,DO
80 NEXT D
90 END
```

85.

L=584.47 МКГ

86.

W=228 ВИТКОВ

87.

```
05 REM ПРОГРАММА РАСЧЕТА ТРАНСФОРМАТОРА
10 PRINT "ВВОД ИСХОДНЫХ ДАННЫХ : U1,U2,I2,E"
20 INPUT U1,U2,I2,E
30 S=SQR(U2*I2/E)
40 I1=U2*I2/(U1*E)
50 W1=50*U1/S
60 W2=50*U2/S
70 D1=0.8*SQR(I1)
80 D2=0.8*SQR(I2)
90 PRINT "D1=";D1,"D2";D2
100 PRINT "ВВОД СКОРРЕКТ.ЗНАЧЕНИЙ ДИАМЕТРА ПРОВОДА"
110 INPUT D3,D4
120 S1=0.8*(W1*D32+W2*D42)
130 PRINT "S="; S,"S1=",S1
140 END
```

88.

```
05 REM ПРОГРАММА РАСЧЕТА ИНДУКТИВНО-ЕМКОСТНОГО ФИЛЬТРА
10 PRINT "ВВОД ИСХОДНЫХ ДАННЫХ : C1,L,U0,R,F"
20 INPUT C1, L,U0,R,F
30 C2=C1
40 U1=U0/(PI*F*C1*R)
50 K=4*PI2*F*2*L*C2
60 U2=U1/F
70 PRINT U1,U2,K
80 END
```

89.

```
05 REM ПРОГРАММА РАСЧЕТА КОНДЕНСАТОРА ФИЛЬТРА
10 PRINT "ВВОД ИСХОДНЫХ ДАННЫХ : F,P"
20 INPUT F,P
25 REM ПЕЧАТЬ ЗАГОЛОВКА ТАБЛИЦЫ
30 PRINT "R-КОМ", "С-МКФ"
40 FOR R=1E3 TO 20E3 STEP 1E3
50 C=1/(PI*P*F*R)
60 PRINT R/1E3,C/1E6
70 NEXT R
80 END
```

90.

```
05 REM ПРОГРАММА РАСЧЕТА РЕОСТАТНО-ЕМКОСТНОГО ФИЛЬТРА
10 PRINT "ВВОД ИСХОДНЫХ ДАННЫХ : C1,C2,RO,UO,R,F"
20 INPUT C1,C2,RO,UO,R,F
30 K=2*PI*F*RO*C2
40 U1=UO/(PI*F*C1*R)
50 U2=U1/K
60 G=U2/UO*100
70 PRINT "G=";G;"%"
80 END
```

ОТВЕТ:

G=0.07%

Приложение 1. ОРГАНИЗАЦИЯ РАБОТЫ НА ПЕРСОНАЛЬНОМ КОМПЬЮТЕРЕ

Характеристики и структура микроЭВМ

Одно из наиболее перспективных направлений электронной вычислительной техники — разработка и практическое применение так называемых *персональных компьютеров (ПК)*.

Персональный компьютер — это микроЭВМ, центральная часть которого (микропроцессор) размещен на кристалле площадью в несколько квадратных сантиметров.

Все узлы ПК свободно размещаются на небольшом столике: сама вычислительная машина с микропроцессором, осуществляющая сотни тысяч операций в секунду, блок внешней памяти, куда можно вносить и свой архив и данные для повседневной работы. Здесь же небольшой телевизор (дисплей) и печатающее устройство, которые позволяют получать на экране и на бумаге необходимые расчетные таблицы, тексты, рисунки.

Персональным компьютер называют потому, что он предназначен для индивидуального пользования подобно магнитофону, телевизору, пишущей машинке.

Кроме того, ПК также называют профессиональной ЭВМ, так как она предназначена для решения широкого круга профессиональных задач.

Для школ и ПТУ разработан целый ряд простых малогабаритных персональных ЭВМ — "Корвет", "Агат", "Микроша", "Школьница" и др.

Нашей промышленностью широко выпускаются персональные компьютеры — диалоговые вычислительные комплексы (ДВК). По способу взаимодействия с пользователем ДВК можно отнести к диалоговым, по предоставлению вычислительных ресурсов — к персональным, по ориентации на область применения — к профессиональным ЭВМ. Различные модели ДВК (ДВК-1, ДВК-2, ДВК-3, ДВК-4) отличаются, в основном, составом оборудования.

На рис. 31 представлена структура микроЭВМ ДВК-2, в состав которой входят:

1) микропроцессор с быстродействием 500 тыс. операций/с с блоком оперативной памяти объемом 64 К байт;

2) внешнее запоминающее устройство (ВЗУ) в виде накопителя на гибких магнитных дисках (НГМД), которое имеет четыре дисководов, в каждый из которых может быть установлен сменный гибкий магнитный диск — дискета с емкостью памяти на одной стороне диска 128 К байт; оперативная память ДВК может обмениваться с ВЗУ файлами программ и файлами данных;

3) дисплей с буфером и клавиатурой, позволяющий вводить, редактировать и запускать программы, вводить данные, выводить листинг программы и результаты на экран дисплея; на экране дисплея размещаются 24 строки по 80 символов в каждой;

4) устройство печати, позволяющее создавать документальную копию текста программы, обрабатываемых данных и полученных результатов.

Клавиатура дисплея микроЭВМ (рис. 32) представляет собой набор клавиш, предназначенных для выполнения основных функций по вводу информации в ЭВМ:

1) клавиши для набора алфавитно-цифровых и специальных символов

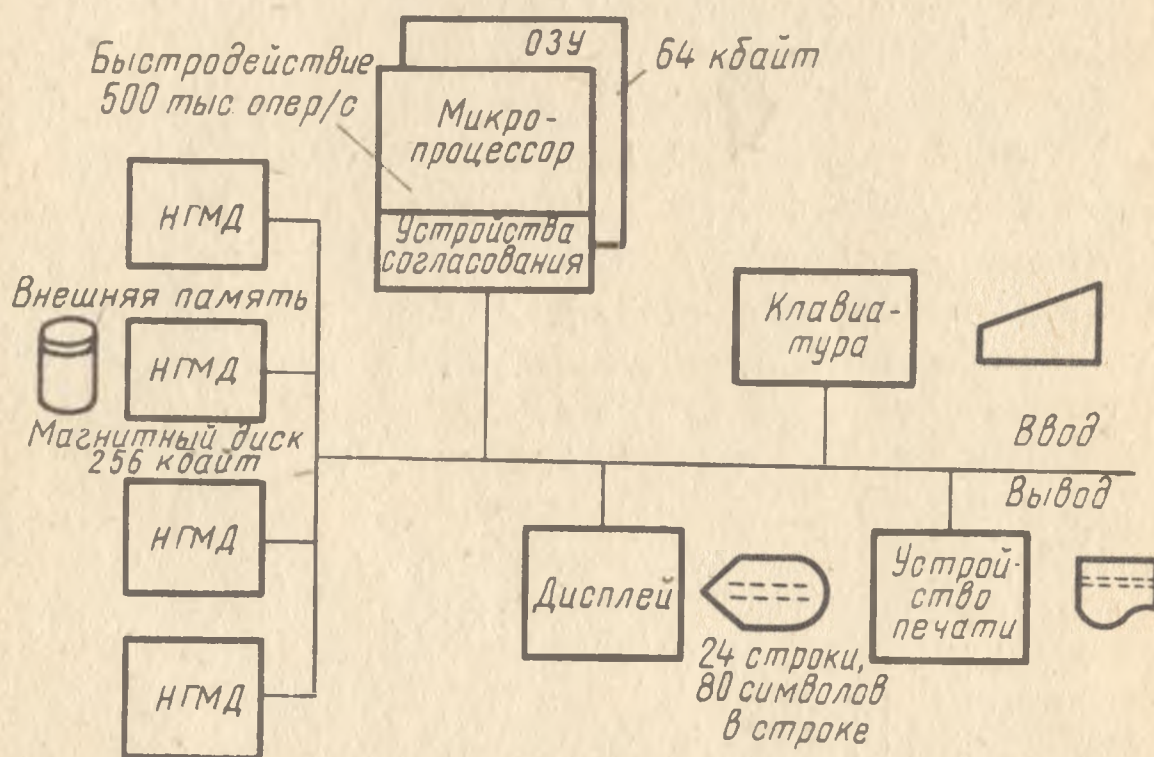


Рис. 31

;	1	2	3	4	5	6	7	8	9	∅	=	ТАВ	ГТ	7	8	9
+	!	”	#	ж	%	&	•	()		=	ТАВ	ГТ	4	5	6
Й	Ц	У	К	Е	Н	Г	Ш	Щ	З	Х	:	ПС	ВК	1	2	3
Э	С	И	К	Е	Н	Г	С	Ш	Щ	З	*	ПС	ВК	∅	,	
СУ	Ф	Ы	В	А	П	Р	О	Л	Д	Ж	Э	.	ЗБ			
ВР	Я	Ч	С	М	И	Т	Ь	Б	Ю	,	/	>	НР			
	Q	—	S	M	I	T	X	B	@	<	?	-				

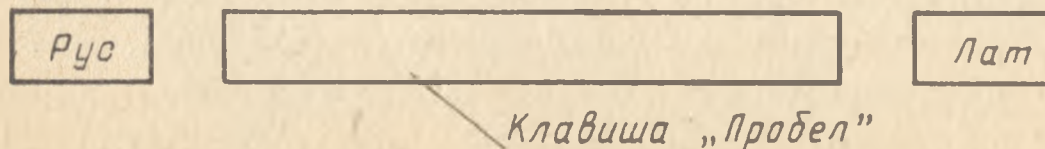


Рис. 32

(белого цвета), позволяющие ввести все основные символы — русские и латинские буквы, знаки арифметических действий "+", "-" и т. п.;

2) функциональные клавиши (серого цвета), управляющие режимами работы клавиатуры, процессора, дисплея; обозначения этих клавиш заключают в рамку, например, **ВК**, **ЗБ**, **ЛАТ**, **РУС**;

3) клавиши выбора символов и управления курсором.

Набор информации на дисплее осуществляется с помощью рассмотренной алфавитно-цифровой клавиатуры. Место на экране, в которое попадает очередной набираемый символ, подсвечивается с помощью специального указателя — *курсора*. На экране курсор выглядит как светящаяся черточка, расположенная под набираемым символом.

При включении дисплея курсор устанавливается в начальное положение (левый верхний угол экрана). Нажатие каждого нового символа перемещает курсор вправо на одну позицию; после заполнения последнего символа строки курсор автоматически смещается в начало следующей строки.

На большинстве алфавитно-цифровых клавиш нанесены два символа: один символ — на верхнем регистре (клавиша **ВР**), другой символ — на нижнем регистре (клавиша **НР**). Для переключения регистров (выбора необходимых символов) и перемещения курсора на экране дисплея используются эти и некоторые другие специальные функциональные клавиши.

При нажатии клавиши **ЛАТ** (при этом загорается сигнальная лампочка "ЛАТ") вводятся буквы латинского алфавита, указанные внизу клавиш. При нажатии клавиши **РУС** (при этом лампочка "ЛАТ" гаснет) вводятся буквы русского алфавита, указанные вверху клавиш.

Буквы русского и латинского алфавита, имеющие одинаковые начертания, но расположенные на разных клавишах, воспринимаются ЭВМ по-разному.

При нажатии клавиши **НР** (при этом загорается лампочка "НР") вводятся специальные символы, указанные внизу клавиш. Буквы в этом режиме вводятся прописным шрифтом. При нажатии клавиши **ВР** (при этом лампочка "НР" гаснет) вводятся специальные символы и цифры, указанные вверху клавиш. Буквы в этом режиме вводятся печатным шрифтом.

При вводе числовой информации для уменьшения числа регистровых переключений цифры рекомендуется набирать на особой цифровой группе клавиш, расположенной в правой части клавиатуры, где нажатие клавиши ",," (запятая) воспринимается при вводе как десятичная точка, отделяющая целую часть числа от дробной.

Выполнение программ на ЭВМ

Итогом второго этапа решения задачи на ЭВМ (этапа программирования) является программа на алгоритмическом языке. Она называется *исходной программой* (ИП). Затем эта исходная программа переводится (транслируется) на машинный язык двоичных кодов. Перевод осуществляется ЭВМ с помощью *специальной программы* — *транслятора* (транслятор с алгоритмического языка БЕЙСИК называется также интерпретатором). Полученная программа на машинном языке называется *рабочей про-*

граммой (РП). После ее выполнения на ЭВМ с введением необходимых исходных данных получают окончательные результаты. Процесс указанных преобразований программы показан на рис. 33 и реализуется в микро-ЭВМ с помощью специального языка системных команд.

Язык системных команд. В процессе преобразования и выполнения программы на ЭВМ БЕЙСИК-система (БЕЙСИК-интерпретатор является ее основной частью) предоставляет пользователю следующие возможности:

- ввод и редактирование ИП;
- вывод на печать или дисплей текста ИП;
- трансляция и выполнение РП;
- запоминание ИП и файлов данных в архиве.

Последовательность работ, относящихся к данной задаче, задается с помощью *управляющих операторов – директив*. В отличие от операторов ИП текст директивы всегда начинается сразу с управляющего слова – имени директивы (NEW, OLD, RUN и др.). Вслед за именем могут указываться некоторые параметры директивы, если они нужны, например название запоминаемой или вызываемой программы.

Основные директивы БЕЙСИК-системы

Директива BASIC вводит с магнитного диска программу БЕЙСИК-интерпретатор.

Директива NEW употребляется при вводе новой ИП. Вслед за директивой NEW должен следовать текст ИП. Команда очищает содержимое памяти и присваивает имя создаваемой программе.

Директива SAVE (сохранить) используется для записи ИП в архив пользователя. Параметром директивы является имя программы, которое ей дал программист, например SAVE TEST (сохранить в архиве программу с именем "TEST").

Директива UNSAVE (стереть, уничтожить) употребляется для уничтожения программы с указанным именем в архиве.

Директива OLD употребляется при вызове ИП из архива пользователя

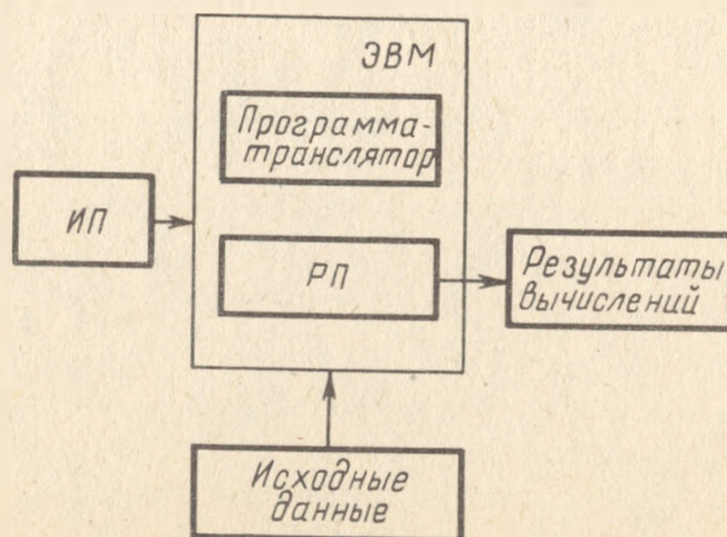


Рис. 33

(эта программа была ранее занесена в архив директивой SAVE). Параметром директивы OLD является имя вызываемой программы, например OLD TEST. Если в архиве ИП с указанным именем нет, то система выдает соответствующее сообщение.

Директива RUN (выполнять) служит указанием для начала трансляции ИП. Счет по изготовленной транслятором РП начинается сразу же автоматически, если при трансляции ИП не было обнаружено ошибок.

Директива LIST употребляется для печати или выдачи на дисплей текста ИП. Особенно эффективно ее применение во время редактирования или отладки БЕЙСИК-программы.

Директива LENGTH (длина) позволяет пользователю определить длину программы, находящейся в оперативной памяти; в сообщении ЭВМ указывается также длина свободного участка оперативной памяти в байтах.

Директива CATALOG (в некоторых системах директива DIR) распечатывает имена программ, файлов, текстов, хранимых в архиве пользователя.

Директива DELETE (стереть) или короче DEL, употребляется для уничтожения создаваемой (текущей) программы или ее части.

Директива BYE (конец сеанса, связи, работы на ЭВМ) является признаком конца всех работ, относящихся к данной задаче.

Рассмотрим теперь подробно действия учащегося за пультом ДВК при вводе, редактировании и выполнении программы с использованием перечисленных системных директив.

Н а ч а л о р а б о т ы с с и с т е м о й . В в о д Б Е Й С И К - и н т е р п р е т а т о р а

1. Включите клавишу **СЕТЬ**.
2. Включите клавиши **ПИТ**, **ПУСК** (на некоторых моделях ДВК вместо клавиши **ПУСК** используется клавиша **ЭВМ**), затем нажмите клавишу **ПР**. Проверяйте включение этих клавиш по красным сигнальным лампочкам.

В случае программных сбоев, когда надо провести повторную загрузку интерпретатора, произвести эту процедуру выключением и включением клавиши **ПР**.

3. Откройте дисковод. Вставьте в него гибкий магнитный диск, на котором записан БЕЙСИК-интерпретатор. Закройте дисковод. Далее, когда это понадобится, используйте номер дисковода, указанный на его корпусе, минус 1. Например, третий дисковод имеет программный номер 2.

4. После появления на экране символа \mathcal{D} наберите на клавиатуре команду связи с требуемым магнитным диском:

X <номер дисковода - 1>, например для третьего дисковода - команду X2.

Мигание сигнальной лампочки на соответствующем дисководе показывает, что команда установления связи с нужным дисководом воспринята.

5. После появления на экране символа-точки наберите и введите директиву BASIC (или в некоторых системах RUN BASIC).

6. Через некоторое время на экране появится запрос о требуемых фун-

кциях интерпретатора. Возможны несколько вариантов ответов пользователя, из которых чаще используются такие:

А **ВК** — требуются все дополнительные функции (ALL — все);

Н **ВК** — не требуется дополнительных функций (NOT — нет).

7. Ввод БЕЙСИК-интерпретатора завершается сообщением, выдаваемым из ЭВМ: READY (готов), указывающим, что интерпретатор введен и находится в режиме приема, редактирования ИП.

Набор (ввод) и редактирование исходной БЕЙСИК-программы

Сначала зафиксируем имя новой создаваемой программы директивой NEW <имя программы>.

Напомним, что директива отличается от оператора БЕЙСИК-программы тем, что вводится без номера строки и выполняется немедленно после нажатия клавиши ВК. Признаком окончания выполнения директивы является сообщение READY.

Имя программы должно содержать не более шести символов и может использоваться, если программа будет запоминаться в архиве на внешнем запоминающем устройстве.

Если программа не именуется, то после ввода директивы NEW интерпретатор делает повторный запрос юб имени программы, на который надо вводить либо имя, либо пустую строку. Неименованной программе интерпретатор присваивает имя NONAME (без имени).

Набор программы

При вводе программы с клавиатуры терминала каждую строку начинают с номера строки, а полностью набранный оператор заканчивают нажатием клавиши **ВК** ("Возврат каретки"), что вызывает запись набранной строки в оперативную память машины, при этом маркер (курсор) на экране дисплея переходит в начало следующей строки. После этого можно набирать и вводить следующую строку программы.

Каждый раз перед нажатием клавиши **ВК** следует проконтролировать набранную информацию на экране дисплея.

Ошибки, выявленные в процессе набора программы, устраняются следующим образом.

1. Последний набранный символ строки удаляется с экрана клавишей **ЗБ** (забой, стирание символа). Повторное нажатие клавиши **ЗБ** стирает еще один символ.

2. Набранная, но еще не введенная строка удаляется одновременным нажатием клавиш СУ и U.

3. Если при вводе ошибочно был пропущен какой-то оператор, то его можно набрать и ввести позже, так как операторы программы окончательно располагаются в памяти в порядке возрастания номеров строк.

4. Чтобы вычеркнуть ранее введенный оператор, надо набрать его номер и нажать клавишу ВК.

Удаление части программы производится директивой DELETE (или,

короче, DEL). Например, директива DEL 10—60 стирает операторы БЕЙСИК-программы с 10-го до 60-й включительно.

5. Чтобы исправить ранее введенный оператор, он вводится в исправленном виде вновь с тем же номером строки.

6. При необходимости можно вывести на экран для просмотра всю программу или ее части директивой LIST. Например,

LIST — выводится вся программа;

LIST 30—120 выводит часть программы: операторы с 30-го по 120-й.

Выполнение программ на ЭВМ

Программа запускается на выполнение директивой RUN.

Если в процессе выполнения программы интерпретатор обнаружит ошибку, на экран выводится диагностическое сообщение в виде фразы на английском языке, например

? SYNTAX ERROR AT LINE 20

(синтаксическая ошибка в строке 20).

Ошибки, обнаруживаемые интерпретатором в БЕЙСИК-программе, могут быть двух типов:

грубые ошибки, при которых программа останавливается и выдается сообщение об ошибке;

негрубые, их поправляет интерпретатор; программа при этом не останавливается, но результат ее выполнения может оказаться неверным.

Программист может прервать выполнение программы, нажав одновременно клавиши **СУ** и **С**.

ЭВМ печатает сообщение: STOP AT LINE N1.

Интерпретатор переходит в режим редактирования. Выполнение программы можно возобновить директивой GO TO N2.

При нормальном завершении программы печатается сообщение READY.

Запоминание созданных программ в архиве

Отлаженная, законченная программа в ДВК может храниться длительное время только на единственном носителе данных — гибком магнитном диске. Запись программы из оперативной памяти на диск производится по директиве

SAVE <спецификация файла>

Если имя программы в спецификации файла не упоминается, то программа при записи на диск сохраняет имя, под которым она создавалась в оперативной памяти (после записи на диск директивой SAVE программа остается также в оперативной памяти).

Если программисту при работе с ВЗУ необходимо ознакомиться с именами файлов, размещенных на диске, то он должен сначала использовать

директиву выхода из БЕЙСИК-системы и передачи управления монитору ВУЕ.

Затем после появления на экране символа . (точки) ввести директиву чтения оглавления

DIR <имя устройства >

В режиме умолчания (когда имя устройства не указано) опрашивается дисковод, с которым последний раз была установлена связь.

На экран выводится оглавление файлов диска, их размеры и размеры занятой и свободной областей памяти диска.

Для продолжения работы с БЕЙСИК-системой программист вновь вызывает интерпретатор директивой BASIC.

Если необходимо стереть уже ненужную программу на диске, используется директива UNSAVE <спецификация файла >

**СОБЛЮДАЙТЕ ОСТОРОЖНОСТЬ!
НЕ СОТРИТЕ ЧУЖИЕ ПРОГРАММЫ!**

Считывание программы из архива (НГМД) в оперативную память осуществляется директивой OLD <спецификация файла >.

Если в спецификации не упомянуто имя файла (программы или массива данных), то последует сообщение ЭВМ

OLD FILE NAME ?

Необходимо либо указать имя вызываемого файла (программы), либо нажать клавишу BK, после чего будет считываться программа с именем NONAME (если она есть на диске).

Директива RUN <спецификация файла > считывает программу с диска и сразу же запускает ее на выполнение. В этом случае имя программы в спецификации файла обязательно.

Приложение 2. АЛГОРИТМЫ И ПРОГРАММЫ МАШИННОЙ ГРАФИКИ

Одним из новых важных направлений в области применения вычислительной техники для автоматизации проектно-конструкторских работ является машинная графика (МГ). Под МГ понимают процессы создания, преобразования, хранения и воспроизведения графической информации с помощью ЭВМ. Машинная графика позволяет получать различные изображения на экране дисплея, а затем при необходимости на печатающем устройстве; выполнять машиностроительные чертежи; изображать графики функций, гистограммы, электрические схемы, эюры сигналов, схемы алгоритмов и программ и т. д. Использование графических образов для представления результатов решения профессиональных задач на ПК повы-

шает наглядность решений и удобство их дальнейшего применения. Решение этих задач возможно благодаря способности МГ отражать такие качества объектов, как форма, размеры, положение в пространстве, цвет и т. п.

Машинная графика обеспечивается комплексом технических и программных средств.

Техническими средствами МГ являются: устройства ввода графической информации, кодировщики, графические планшеты, специальные клавиши перемещения по экрану дисплея курсора, различные манипуляторы и указатели — "световое перо", джойстик (рукоятка), устройства типа "мышь" и др.; устройства вывода графической информации (графопостроители, принтеры, графические дисплеи); специальная аппаратура, преобразователи, устройства связи, встраиваемые в ПК для приема, преобразования и выдачи графической информации.

Символьный режим работы дисплея

Рабочее поле экрана дисплея в символьном режиме — это 24 строки (в некоторых ЭВМ может быть 12, 16 или 25 строк), по 80 позиций, символов в каждой строке. То есть общий объем одного кадра экрана дисплея составляет $24 \times 80 = 1920$ знакомест, в каждом из которых может быть использован любой символ, имеющийся на клавиатуре дисплея. Все примеры, разбираемые в настоящем пособии, использовали пока только символьный режим работы дисплея; при выдаче результатов решения задач в виде таблиц, текстов, надписей использовался оператор PRINT, позволяющий объединять буквы, числа и другие символы, имеющиеся на клавиатуре ПК, в единый рисунок.

```

10 PRINT '      РАСЧЕТ НИЗКОЧАСТОТНОГО ФИЛЬТРА '
20 PRINT '
30 PRINT '
40 PRINT '          C2 ===          OY !
50 PRINT '          !      R3          !
60 PRINT ' < - !====! - - - !====! - - - !+ ! !
70 PRINT '      R11 !          !          ! - - - >
80 PRINT '          -          !          ! - - - !
90 PRINT '          ! !      C4 === ! - - - !
91 PRINT '      R12 ! !          !          !
92 PRINT '          -          !          ! - - - !
93 PRINT '          !          !          !
94 PRINT '          - - -          - - -
95 PRINT '          ///          ///

```

Рис. 34

Простые изображения можно нарисовать на экране дисплея, располагая некоторые символы в соответствии с требуемыми очертаниями. Символьный режим работы дисплея позволяет вычертить также и схему устройства и график функции.

На рис. 34, 35 показаны фрагменты программы, в которых операторами PRINT с символьными константами вычерчивается схема низкочастотного фильтра и его амплитудно-частотная и фазочастотная характеристики.

Для организации вывода на экран дисплея (а затем при необходимости и на печатающее устройство) подобного рисунка его сначала следует нарисовать на бумаге в клетку. С помощью этого рисунка (шаблона) можно определить, в каких позициях следует разместить необходимые символы, чтобы получилось изображение контура или силуэта предмета. Одной горизонтальной строке бумаги будет соответствовать одна строка изображения на экране дисплея.

```

10 PRINT ' A, ДВ (А-КОЭФФИЦИЕНТ УСИЛЕНИЯ)'
20 PRINT ' (QP=2)'
30 PRINT ' '
40 PRINT ' K !.....! *
50 PRINT ' !.....! *
60 PRINT ' !.....! *
70 PRINT ' !-----LGT W'
80 PRINT ' WM WP
90 PRINT ' F(W) (F-ФАЗА)
91 PRINT ' !
92 PRINT ' 0 !.....!
93 PRINT ' !
94 PRINT ' -90!-----LGT W'
95 PRINT ' !.....!
96 PRINT ' -180!-----LGT W'
97 PRINT ' WP

```

Рис. 35

```

----->
!*
!*
!*
!*
!*
PRINT "----->"
FOR X=1 TO 10
Y=X*2/2
PRINT "!"; TAB(Y); " "
NEXT X

```

Рис. 36

На рис. 36 оператор PRINT TAB изображает график функции $y = \frac{1}{2} x^2$.

В тех ПК, в которых дисплей может работать только в описанном здесь символьном режиме, такие рисунки создают часто, однако гораздо лучше изображения выполняются в другом режиме работы дисплея — графическом.

Управление экраном дисплея с использованием оператора PRINT вместе с функцией специальных шестнадцатеричных кодов — HEX-кодов (несколько таких кодов в качестве иллюстрации приведены в табл. 5) позволяет очищать экран или его часть (стирать на нем информа-

Т а б л и ц а 5. HEX-коды для символьного режима работы дисплея

HEX-код	Выполняемая функция в символьном режиме
01	Установка курсора в левый верхний угол экрана
03	Очистка экрана и установка курсора в левый верхний угол
07	Подача звукового сигнала
08	Сдвиг курсора на одну позицию влево ←
09	Сдвиг курсора на одну позицию вправо →
0A	Перемещение курсора на одну строку вниз ↓
0C	Перемещение курсора на одну строку вверх ↑
0D	Перемещение курсора на начало строки
85	Перемещение курсора в начало следующей строки

цию), перемещать курсор влево, вправо, вверх, вниз и т. п. Например, оператор

PRINT HEX (03)

очищает экран перед выдачей новой порции результатов, а оператор

PRINT HEX (07)

подает звуковой сигнал, который может использоваться, например, для сигнализации о наличии среди результатов некоторых критических или неверных значений.

Графический режим работы дисплея

В современных типах ПК (ДВК-3, ДВК-4, ЕС-1840 и др.) дисплей ЭВМ кроме рассмотренного символьного режима может работать и в другом — *графическом* режиме.

Экран дисплея, работающего в этом режиме, состоит из большого числа мелких фиксированных точек. Пишущим узлом является световой маркер — *графический курсор*. При программировании траектории движения пишущего узла рабочее поле представляет собой прямоугольную матрицу, состоящую из конечного количества фиксированных точек, в любой из которых допустима остановка пишущего узла после его перемещения. Например, в рабочем поле экрана дисплея ПК "Искра-226" с размерами 102×220 мм помещается 131072 точек: 256 строк, по 512 точек в каждой строке (рис. 37).

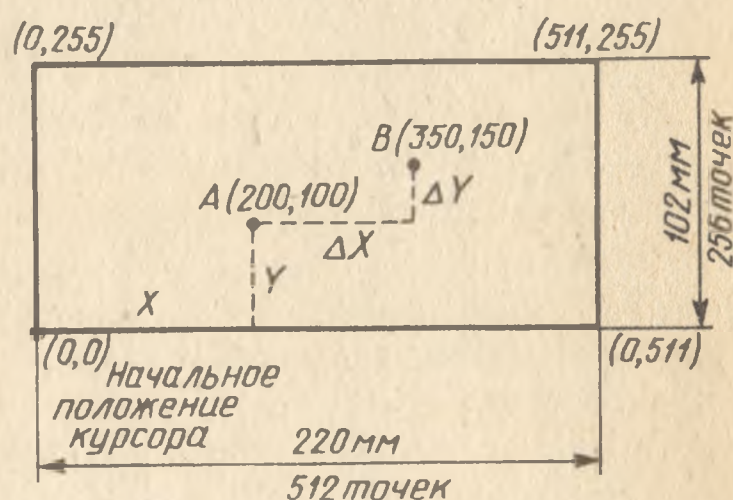


Рис. 37

Размер матрицы зависит от размера экрана дисплея (бывают экраны размером 400×287 точек, 640×200, 1024×500 и т. п.). Чем больше точек на экране дисплея, тем выше его разрешающая способность и качество изображений. Расстояние между соседними точками (по горизонтали или вертикали) 0,2–0,4 мм.

Вычерчивание изображений выполняется в хорошо известной прямоугольной декартовой системе координат с началом в левом нижнем углу рабочего поля (в некоторых системах начальное состояние графического курсора — левый верхний угол экрана). Пишущий узел перемещается дискретно (элементарное перемещение — один шаг или дискрет). Пишущий узел за один дискрет перемещается по прямой.

Чтобы выводить точки и строить линии, нужно иметь возможность задавать то место на экране, где они должны располагаться. Существуют два способа идентификации точки, отображаемой на экране графического дисплея.

При абсолютной координатной идентификации для точки $A (X, Y)$ указываются ее абсолютные координаты в дискретах по оси X и Y от начала координат системы. Например, точка $A (200, 100)$ расположена так (рис. 37), что по оси X отстоит от начала координат на 200, а по оси Y на 100 дискрет. Если X и Y имеют вещественные значения, то берется их целая часть.

Другим способом является относительная координатная идентификация, при которой координаты точки задаются относительно последней выведенной на экран точки (так называемой текущей точки). Так, если следом за только что выведенной в абсолютных координатах точкой A выводится точка $B (\Delta X, \Delta Y)$ с указанием относительных координат (смещений), например $B (150, 50)$, то текущей точкой является точка A , а абсолютные координаты точки B будут $B (350, 150)$. Параметры ΔX и ΔY при относительной координатной идентификации указывают величину и направление перемещений. Если $\Delta X > 0$, то перемещение осуществляется вправо, если $\Delta X < 0$, то влево. Аналогично значения ΔY задают перемещения либо вверх, либо вниз.

На рис. 37 указаны абсолютные координаты рассмотренных в примере точек A и B и граничных точек экрана дисплея. Отметим также, что хотя программирование изображений ведется в электронных дискретах, зная величину рабочего поля экрана в миллиметрах, можно перейти к построению графических объектов в обычных метрических единицах.

Чтобы перейти в графический режим, необходимо выполнить специальную команду подготовки—инициализации экрана дисплея GRAPHICS (установить графический режим) (или SCREEN 1, графический экран; или INIT, режим инициализации). В некоторых системах этот режим устанавливается автоматически при выполнении любого из графических операторов, например, оператора PLOT.

Программное управление графическим экраном можно реализовать на трех уровнях различными способами:

на 1 уровне — графическими микрооперациями, например, с использованием шестнадцатеричных HEX-кодов;

на 2 уровне — более крупными агрегированными средствами, операторами построения и преобразования графических объектов, входящими в состав различных алгоритмических языков, например, оператором БЕЙСИКа PLOT (чертить, строить);

на 3 уровне — специальными графическими пакетами прикладных программ.

Элемент информации, для создания которого в системе МГ существует специальный аппаратно-программный блок, называется *графическим примитивом*. Таким графическим примитивом в системах МГ является отрезок прямой линии. При этом различные графические элементы (точка, дуга, окружности, эллипс и т. д.) рассматриваются как сочетания, композиции примитивов и должны на нижних уровнях использования средств МГ синтезироваться самими пользователями.

У каждой из указанных выше групп графических средств есть свои

особенности. Так, например, оператор PLOT позволяет вывести изображение одним оператором.

Операторы работы с графическими объектами позволяют не только формировать графические объекты, но и объединять их друг с другом, преобразовывать — поворачивать, перемещать и т. п.

Использование при проектировании сложных графических изображений пакетов прикладных программ требует от пользователя только профессиональных знаний; здесь нет специфических терминов, относящихся к системам программирования, техническим средствам ЭВМ и т. д.

Независимо от выбора того или иного способа программного управления графическим экраном пользователь может:

- зафиксировать текущую точку экрана;
- соединить две точки экрана отрезком прямой;
- вывести на экран, начиная с указанной точки, символьную или числовую информацию;
- очистить графический экран.

Прежде чем перейти непосредственно к рассмотрению программных средств МГ опишем еще две команды, относящиеся к подготовительному этапу установления (инициализации) графического режима работы дисплея.

Команда выбора цвета экрана COLOR

Современные графические системы (например, в ПК ДВК-4, ЕС-1842, IBM PC) позволяют существенно улучшить качество изображений за счет использования цвета. Цвет позволяет сделать рисунки более удобными для восприятия, акцентировать внимание на самых важных элементах изображений. Для выбора цветов используется команда

COLOUR (цвет) или короче COLOR

Формат команды в графическом режиме
COLOR A, F

где параметр A устанавливает цвет для всех точек, высвечиваемых последующими командами, а параметр F устанавливает цвет для фона.

Коды цветов A и F:

0 — черный	3 — красный	6 — голубой
1 — белый	4 — зеленый	7 — оранжевый
2 — синий	5 — желтый	

Команда COLOR в программе может использоваться несколько раз. Например, следующие команды устанавливают голубой фон, а также последовательно красный, зеленый и желтый цвета для элементов рисунка:

COLOR 3, 6

.....
COLOR 4, 6

.....
COLOR 5, 6

Количество цветов и оттенков, используемых в различных ПК изменяется от 2 (черный и белый) до 20.

Подготовительная команда CLS (сокращение от слов CLEAR SCREEN – очистить экран) стирает содержимое экрана перед формированием нового графического изображения.

Программное управление графическим экраном на первом элементарном уровне можно осуществлять оператором PRINT с шестнадцатиричными HEX-кодами. Операторы PRINT HEX аналогично конструкциям, описанным выше, но уже в другом – графическом режиме, используя коды табл. 6, позволяют перемещать пишущий узел с вычерчиванием или без вычерчивания изображения на один дискрет вправо, влево, вверх, вниз, по диагонали, стирать любую часть уже выведенного изображения, блокировать вывод графической информации на определенное время и т. п.

Т а б л и ц а 6. HEX-коды для графического режима работы дисплея

HEX-код	Выполняемая функция в графическом режиме
01	Установка графического курсора в начало координат (левый нижний угол экрана)
02	Перемещение курсора по диагонали на шаг $(-\Delta X, -\Delta X)$ ↙
03	Перемещение курсора по диагонали на шаг $(+\Delta X, -\Delta Y)$ ↘
04	Перемещение курсора по диагонали на шаг $(-\Delta X, +\Delta Y)$ ↖
05	Перемещение по диагонали на шаг $(+\Delta X, +\Delta Y)$ ↗
06	Установка режима "ЧЕРТИТЬ", опускание пишущего узла
07	Установка режима "СТИРАТЬ", поднятие пишущего узла
08	Перемещение курсора по горизонтали на шаг влево $(-\Delta X)$ ←
09	Перемещение курсора по горизонтали на шаг вправо $(+\Delta X)$ →
0A	Перемещение курсора по вертикали на шаг вниз $(-\Delta Y)$ ↓
0C	Перемещение курсора по вертикали на шаг вверх $(+\Delta Y)$ ↑
0D	Очистка экрана, установка курсора в начало координат
0E	Блокировка отображения графики на экране
0F	Разблокировка отображения графики на экране

Графические операторы БЕЙСИКа

Эти средства, применяемые на втором уровне программирования графических образов, являются более удобными для пользователя. Однако следует отметить, что в отличие от основных операторов БЕЙСИКа, подробно рассмотренных во второй главе настоящего пособия, графические операторы пока не стандартизованы. В разных типах ПК используются различные команды для выполнения одинаковых действий. Рассмотрим наиболее употребительные конструкции.

О п е р а т о р п о с т а н о в к и т о ч к и PSET

Оператор PSET (сокращение слов – POINT SET, поставить точку) предназначен для отображения на экране дисплея, работающего в графиче-

ческом режиме, отдельной точки с заданными абсолютными координатами X и Y. Оператор имеет формат

PSET (X,Y)

например, PSET (50, 50) — точка A с координатами X = 50 Y = 50

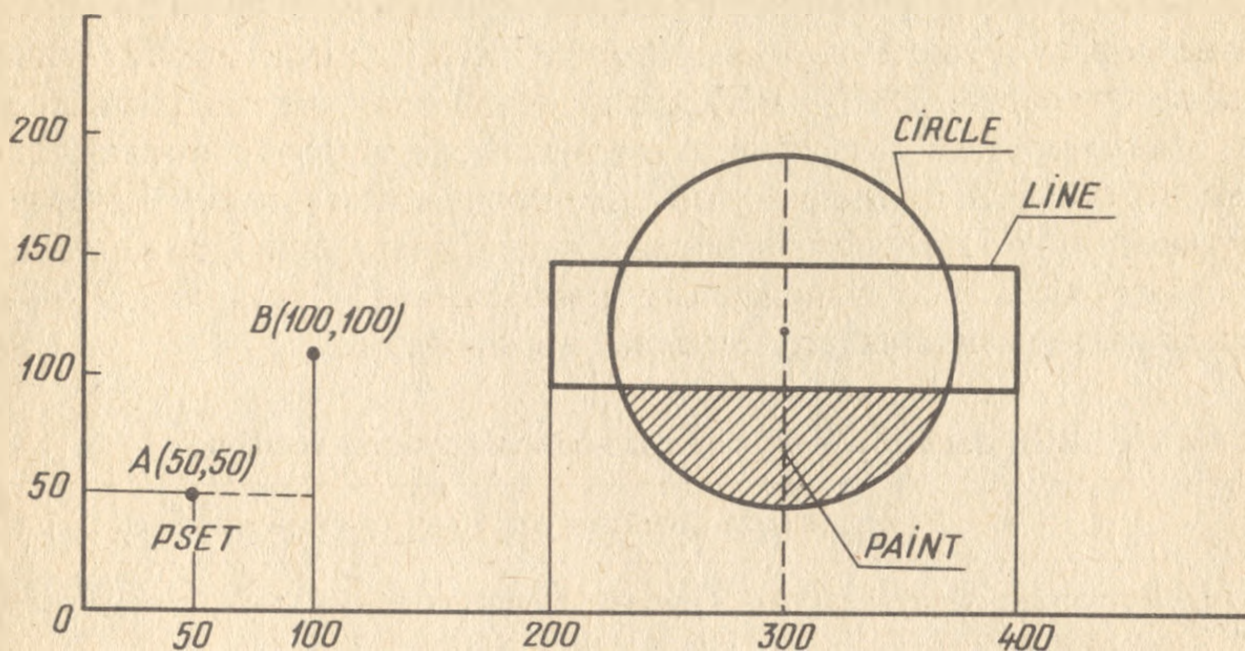


Рис. 38

Изображения объектов этого и ряда последующих примеров приводятся на рис. 38.

Если в операторе PSET координатам точки предшествует слово STEP (шаг), то эти координаты интерпретируются как смещения относительно координат последней выведенной на экран точки (текущей точки). Например,

PSET STEP (50,60)

вычерчивает точку с абсолютными координатами B (100, 110).

Использование относительных координат удобнее при вычерчивании рисунка, начиная от некоторой установленной базы; в этом случае нет необходимости в определении точных абсолютных координат каждой точки контура.

Оператор вычерчивания точки есть во всех реализациях графического БЕЙСИКа иногда с другим именем, например ДОТ — точка или PNT (сокращение слова POINT) — точка.

Вычерчивание линий

В БЕЙСИК входит также графическая конструкция — оператор LINE (линия). С его помощью можно чертить на экране отрезки прямых линий, контуры прямоугольников и т. д. Поскольку любая прямая линия определяется двумя точками, то в операторе LINE задаются абсолютные координаты двух точек — начальной (XН, YН) и конечной (XК, YК):

LINE (XН, YН) - (XК, YК)

например,

```
LINE (200,100) - (400,100)
```

С помощью нескольких операторов LINE можно вычертить на экране несколько отрезков прямой так, чтобы каждый последующий отрезок начинался в той точке, в которой закончился предыдущий. Например, четыре оператора LINE строят прямоугольник

```
LINE (200,100) - (400,100)
LINE (400,100) - (400,150)
LINE (400,150) - (200,150)
LINE (200,150) - (200,100)
```

Для выполнения подобных построений существует более удобный способ, который основан на том, что всякий раз при выводе на экран какой-либо линии в памяти ЭВМ запоминаются координаты последней выведенной точки, которыми можно воспользоваться при выполнении очередного оператора LINE. Если в операторе LINE опустить координаты начальной точки отрезка (поставив на их месте прочерк, дефис), то вычерчиваемый с помощью такого оператора отрезок будет начинаться в последней, участвующей в предыдущих построениях точке, и кончаться в точке, заданной в данном операторе.

Тогда предыдущий пример можно записать короче:

```
LINE (200,100) - (400,100)
LINE - (400,150)
LINE - (200,150)
LINE - (200,100)
```

Оператор вычерчивания линии есть во всех реализациях графического БЕЙСИКа, но иногда с другим именем, например, DRAW (чертить).

Вычерчивание окружности производится оператором CIRCLE (окружность):

```
CIRCLE (X0,Y0),R
```

где X0, Y0 — абсолютные координаты центра окружности, R — ее радиус в дискретах, например,

```
CIRCLE (300,125), 75
```

Оператор PAINT (закрасить) позволяет выделить заданную замкнутую область экрана, заполнив ее другим по сравнению с контуром цветом или штриховкой; для этого необходимо указать хотя бы одну точку внутри закрашиваемого контура, например,

```
PAINT (300,75)
```

Все разобранные выше примеры графических операторов БЕЙСИКа собраны в одной программе и дают изображение, представленное на рис. 38.

```

10 SCREEN 1
20 COLOR 1,0
30 CLS
40 P SET (50,50)
50 P SET STEP (50,60)
60 LINE (200,100) - (400,100)
70 LINE - (400,150)
80 LINE - (200,150)
90 LINE - (200,100)
100 CIRCLE (300,125),75
110 PAINT (300,75)
120 END

```

В системах МГ допустимо совместное использование графики и текста, что необходимо при изготовлении чертежей. С помощью оператора PRINT можно и в графическом режиме выводить любые допустимые для ПК символы. Необходимое местоположение символов на экране может быть задано оператором LOCATE

LOCATE T1, T2 (или PRINT AT T1, T2)

где T1 — номер строки (1 — 24), а T2 — номер столбца (1 — 80) экрана дисплея в символьном режиме. Операторы графического вывода — PSET, LINE и др. на размещение символов влияния не оказывают.

О п е р а т о р P L O T

В некоторых реализациях БЕЙСИКа существует удобный графический оператор PLOT, который позволяет объединять в одной конструкции возможности графического и символьного режимов работы дисплея. Рассмотрим реализацию этого оператора для ПК "Искра-226". Оператор PLOT имеет вид:

PLOT $\langle \Delta X, \Delta Y, p \rangle, \langle \Delta X, \Delta Y, p \rangle, \dots$

где $\Delta X, \Delta Y$ — перемещения пишущего узла вдоль осей X и Y в шагах, дискретах, а p — параметр, который может принимать значения U, D, R, C, S, символьная константа или символьная переменная.

Перемещения пишущего узла $\Delta X, \Delta Y$ и значения параметра p являются элементами списка оператора PLOT и заключаются в угловые скобки \langle, \rangle , которым на клавиатуре ПК соответствуют клавиши "меньше" и "больше".

Другой особенностью оператора PLOT является то, что он работает не с абсолютными, а только с относительными координатами, приращениями. Любые линии и символы вычерчиваются оператором PLOT как последовательность шагов, приращений.

Перемещение пишущего узла осуществляется в поднятом (нерабочем) положении, если параметр p принимает значение U (Up — вверх) или вообще отсутствует (не задан); и опущенном (рабочем, рисующем) положении, если принимает значение D (*Down* — вниз). Значение R (*Reset* — восстановить) параметра R оператора PLOT соответствует перемещению пишущего узла в начало координат (левый нижний угол рабочего поля). Указанные значения R, U и D действуют только в пределах одного элемента списка оператора PLOT. Если перед каким-либо элементом списка оператора

PLOT задано арифметическое выражение, то элемент списка выполняется повторно, а количество повторений задается целой частью указанного арифметического выражения.

Покажем, как вычерчивается тот же прямоугольник (рис. 38) вместо четырех операторов LINE одним оператором PLOT.

PLOT < 200, 100, U > , < 200, 0, D > , < 0, 50, D > , < -200, 0, D > ,
< 0, -50, D > , < , , R > .

В данном примере список оператора PLOT состоит из шести элементов. Сначала пишущий узел переходит из начального состояния в точку (200, 100) без вычерчивания (элемент 1), затем от этой точки вычерчивает отрезок прямой (элемент 2) до точки (400, 100) и т. д. Последний элемент списка оператора PLOT возвращает курсор в исходное состояние.

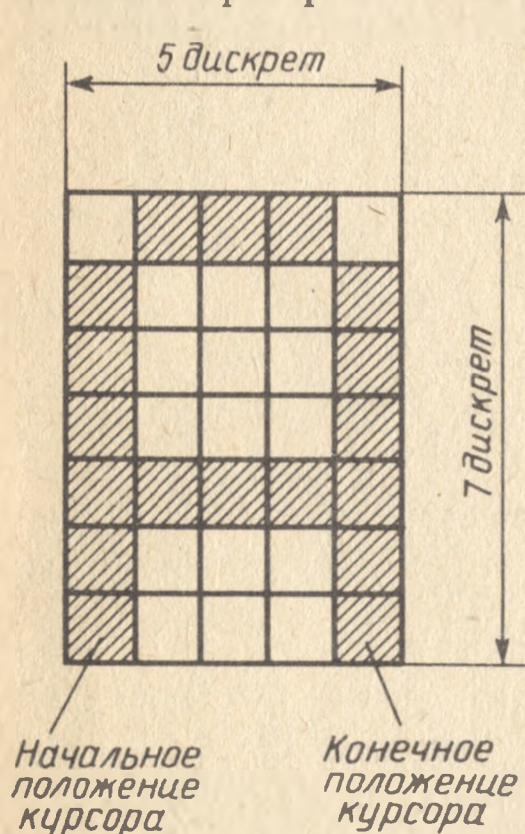


Рис. 39

Вычерчивание символов текста проводится оператором PLOT в том же графическом режиме, если в качестве значений параметра p используются символьная константа или символьная переменная. Значения ΔX и ΔY задают проекции перемещений пишущего узла по горизонтальной и вертикальной осям, которые осуществляются в поднятом положении узла. Затем вычерчивается строка текста, заданная символьной константой или символьной переменной.

Символы вычерчиваются (рис. 39) в формате матрицы 5×7 дискрет (шагов), начиная с левого нижнего угла этой матрицы, к которому был осуществлен подвод пишущего узла в соответствии с указанными значениями ΔX и ΔY (рис. 39). После вычерчивания каждого символа пишущий узел устанавливается в правом нижнем углу матрицы в поднятом (нерисующем) положении.

Значения параметра p , равные C и S, позволяют менять масштаб выводимого текста и расстояние между символами.

Если параметр p принимает значение C, то ΔX устанавливает масштаб символов (причем, $\Delta X \geq 1$), при ΔY может иметь любое значение. Базовые размеры символов равны 5×7 дискрет.

Если параметр p принимает значение S, то ΔX и ΔY задают расстояния между символами по горизонтали и вертикали соответственно в дискретах.

Значения C, S действуют только в пределах данного оператора PLOT. Если значение C не задано, то масштаб выводимых символов равен 1. Если значение S не задано, то промежуток между символами равен двум шагам по горизонтали.

Приведем программу вычерчивания оператором PLOT графика гиперболы по формуле $y = 480/x$ на участке $x \in [2, 350]$ (рис. 40).

Операторы 10 и 20 этой программы вычерчивают координатные оси

X и Y, их обозначения и возвращают курсор в начальное положение (левый нижний угол рабочего поля).

Операторы 30 и 40 устанавливают начальную точку графика $x = 2$; $y = 240$.

График, вычерчиваемый в цикле операторами 50 – 110, представляет собой линейную аппроксимацию оператором PLOT (строка 100) приращений функции, взятых с постоянным шагом $\Delta X = 1$ (строка 50). Так как в операторе PLOT с помощью стандартной функции INT автоматически используется только целая часть параметра Y, то необходимо накапливать малые приращения аргумента (строки 70 – 90). Пошаговое выполнение операторов 50 – 110 на начальном этапе вычерчивания функции показано в табл. 7.

Оператор 120 перемещает курсор в начальное положение, а оператор 130, начиная от точки с координатами (100, 100), вычерчивает строку $y = 480/x$ символами в удвоенном масштабе.

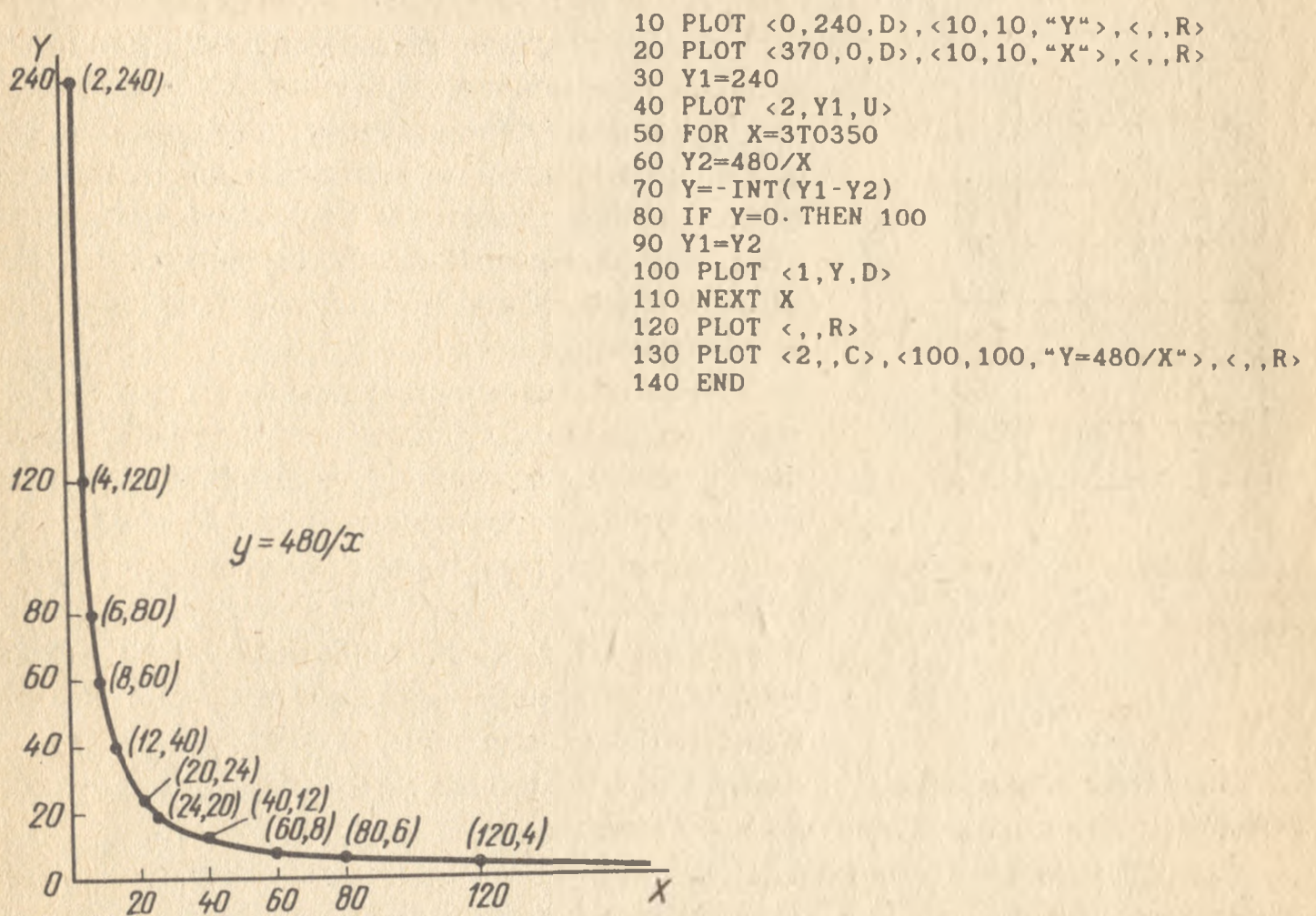


Рис. 40

При программировании изображений необходимо учитывать размеры рабочего поля экрана дисплея. Выход пишущего узла за его пределы приводит к неправильному выполнению графических операторов и искажению графических изображений.

Кроме рассмотренных выше отдельных графических операторов и интегрального графического оператора PLOT в различных расширениях БЕЙСИКа есть еще целая совокупность графических операторов. Например, для ПК "Искра-226" таких операторов 17: DOT, DRAW, LABEL,

Таблица 7. Значения для вычерчивания функции $y = 480/x$

y_1	x	$y_2=480/x$	y_1-y_2	Элемент графического оператора PLOT < 1, Y, D > где $Y = -INT(y_1-y_2)$
240	3	160	80	<1, -80, D>
160	4	120	40	<1, -40, D>
120	5	96	26	<1, -26, D>
96	6	80	16	<1, -16, D>
80	7	68,6	11,4	<1, -11, D>
68,6	8	60	8,6	<1, -8, D>
60	9	53,3	6,7	<1, -6, D>
53,3	10	48	5,3	<1, -5, D>
48	11	43,6	4,4	<1, -4, D>
43,6	12	40	3,6	<1, -3, D>

⊗ LET, ⊗ MOVE, TURN и др. Так, например, операторы DOT, DRAW задают точку или вектор в формируемом графическом объекте (ГО). Операторы ⊗ MOVE, TURN производят сдвиг или поворот ГО. Оператор ⊗ LET осуществляет объединение нескольких графических объектов в один ГО. Указанная совокупность операторов позволяет проводить эффективные преобразования разнообразных ГО, к числу которых следует отнести машиностроительные чертежи, электрические схемы, карты географические, синоптические и т. д. Однако, программы построения таких графических образов с помощью отдельных графических операторов очень сложны и трудоемки в составлении и отладке. Поэтому мы рассмотрим далее более мощные средства МГ, имеющиеся на большинстве современных ПК — пакеты прикладных программ машинной графики (ППП МГ).

Пакеты прикладных программ машинной графики

Эти средства, представляющие собой сложные программные системы, можно разделить на две большие группы: универсальные и пользовательские.

Универсальные пакеты программ ориентированы на широкую область применения, например, построение графиков для различного типа функций (алгебраических, тригонометрических и др.), формирование элементов чертежей и схем (вычерчивание многоугольников, окружностей, эллипсов, дуг, спиралей, штриховых, осевых и размерных линий, выполнение надписей, нанесение размеров и т. д.), построение диаграмм и пиктограмм. Использование универсальных ППП предполагает знания элементов программирования.

Пользовательские ППП МГ создаются в интересах некоторой определенной, как правило, значительной группы пользователей, многократно решающих более узкие специальные задачи. Первая особенность

таких ППП состоит в том, что в качестве структурных единиц чертежа выступают более сложные ГО. Для машиностроительных чертежей — это условно-графические изображения крепежных элементов (винтов, болтов, гаек, шайб и т. д.), типовых деталей и узлов; для электрических схем — это условно-графические изображения электронных элементов (резисторов, конденсаторов, транзисторов, микросхем и т. д.), типовых узлов (фильтров, трансформаторов, усилительных каскадов и т. п.). Второй особенностью пользовательских ППП является то, что они, в основном, ориентируются на конкретные типы технических средств МГ. Следствием указанных особенностей является то, что использование таких пакетов не гребует от конструктора, проектировщика специальных знаний в области программирования. Пакет прикладных программ машинной графики состоит из библиотеки программ, реализующих графические примитивы и элементы сложных ГО, и управляющей программы, обеспечивающей взаимосвязь пользователей пакета с библиотекой программ. Управляющая программа позволяет пользователю последовательно выполнять необходимые программы из библиотеки. Указания пользователь может передавать управляющей программе или в виде последовательности инструкций или в виде программы на некотором алгоритмическом языке. Как правило, в последнем случае применяется такой алгоритмический язык, который является основным для программирования библиотеки программ, например, БЕЙСИК или ФОРТРАН.

Если ППП МГ являются универсальными, то управляющая программа является частью транслятора некоторого языка программирования. Наиболее известными являются универсальные ППП МГ — ГРАФОР, ФОКАЛ.

В связи с тем, что ППП СГ в значительной степени отражают специфику используемых технических средств, конкретных алгоритмических языков и областей применения, их изучение целесообразно проводить непосредственно на конкретных технических и программных средствах, имеющихся в распоряжении учебного заведения.

Примеры использования графических операторов для изображения элементов машиностроительных чертежей и электрических схем

Л и н и и

1. Напишите программу построения отрезка сплошной прямой линии, ограниченного точками с координатами $A(X, Y)$, $B(X, Y)$. Значения координат точек введите с клавиатурного устройства, результат выведите на экран дисплея.

Текст программы на алгоритмическом языке БЕЙСИК с использованием оператора PLOT может иметь вид:

```
05 REM ПРОГРАММА ПОСТРОЕНИЯ ОТРЕЗКА СПЛОШНОЙ ЛИНИИ
10 PRINT "ВВОД ИСХОДНЫХ ДАННЫХ: A(X1,Y1),B(X2,Y2)"
20 INPUT X1,Y1,X2,Y2
30 PLOT <.,R>,<X1,Y1,U>,<X2-X1,Y2-Y1,D>
40 END
```

2. Напишите программу построения отрезка двойной сплошной линии, ограниченного точками с координатами $A(X, Y)$, $B(X, Y)$. Величину расстояния между линиями возьмите равной 5 дискретам экрана дисплея. Значения координат точек введите с клавиатуры дисплея.

Текст программы на алгоритмическом языке БЕЙСИК с использованием оператора PLOT может иметь вид:

```

05 REM ПРОГРАММА ПОСТРОЕНИЯ ДВОЙНОЙ СПЛОШНОЙ ЛИНИИ
10 PRINT "ВВОД ИСХОДНЫХ ДАННЫХ: A(X1,Y1), B(X2,Y2)"
20 INPUT X1,Y1,X2,Y2
30 IF X2=X1 THEN 100
40 REM ПОСТРОЕНИЕ ОТРЕЗКА НАКЛОННОЙ ИЛИ ГОРИЗОНТАЛЬНОЙ ЛИНИИ
50 D2=INT(5*COS(ARCTAN((Y2-Y1)/(X2-X1))))
60 D1=INT(5*SIN(ARCTAN((Y2-Y1)/(X2-X1))))
70 PLOT < , ,R> , <X1,Y1,U>
80 PLOT <X2-X1,Y2-Y1,D> , <-D1,D2,U> , <-(X2-X1), -(Y-Y1),D>
90 GOTO 120
100 REM ПОСТРОЕНИЕ ОТРЕЗКА ВЕРТИКАЛЬНОЙ ЛИНИИ
110 PLOT < , ,R> <X1,Y1,U> , < ,Y2-Y1,D> , <-5,,U> , < , -(Y2-Y1),D>
120 END

```

При выполнении оператора 30 выясняется расположение отрезка строящейся линии. Если отрезок расположен вертикально ($X1=X2$), то осуществляется переход к оператору 90. В случае если $X1 \neq X2$, т.е. расположение отрезка строящейся линии или наклонное под углом α к горизонтальной оси, или горизонтальное, то выполняются операторы 40 – 80.

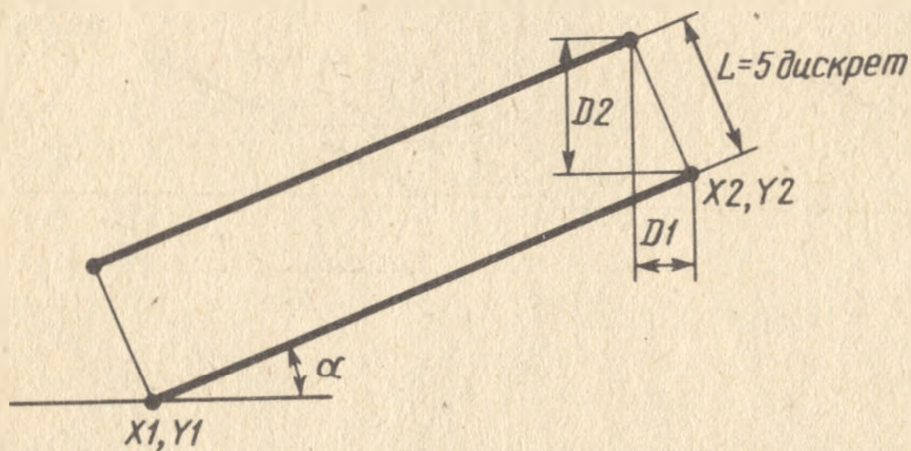


Рис. 41

На рис. 41 представлены геометрические построения, лежащие в основе операторов 50 и 60.

3. Напишите программу построения отрезка штриховой прямой линии, ограниченного точками с координатами $A(X, Y)$, $B(X, Y)$. Размеры штриха и интервала между штрихами возьмите равными. Исходные данные: значения координат точек и размер штриха введите с клавиатуры дисплея.

Текст программы построения отрезка штриховой линии может иметь вид:

```

05 REM ПРОГРАММА ПОСТРОЕНИЯ ШТРИХОВОЙ ЛИНИИ
10 PRINT "ВВОД ИСХОДНЫХ ДАННЫХ: A(X1,Y1), B(X2,Y2), ДЛИНА ШТРИХА"
20 INPUT X1,Y1,X2,Y2,L
30 IF X2=X1 THEN 120
40 REM ПОСТРОЕНИЕ ОТРЕЗКА НАКЛОННОЙ ИЛИ ГОРИЗОНТАЛЬНОЙ ЛИНИИ
50 D2=INT(L*COS(ARCTAN((Y2-Y1)/(X2-X1))))
60 D1=INT(L*SIN(ARCTAN((Y2-Y1)/(X2-X1))))
70 PLOT < , ,R> , <X1,Y1,U>

```

```

80 FOR I=X1TOX2STEP2*D2
90 PLOT <D2,D1,D>,<D2,D1,U>
100 NEXT I
110 GOTO 170
120 REM ПОСТРОЕНИЕ ВЕРТИКАЛЬНОЙ ЛИНИИ
130 PLOT <,,R>,<X1,Y1,U>
140 FOR I=1TOY2STEP2*L
150 PLOT <,,L,D>,<,,L,U>
160 NEXT I
170 END

```

При выполнении оператора 30 выясняется расположение отрезка строящейся линии. В случае наклонного или горизонтального расположения линии выполняются операторы 50 и 60, в основе которых лежат геометрические построения, выполненные на рис. 42. Построение штриховой линии осуществляется в цикле операторами 80 — 100.

В случае вертикального расположения линии выполняются операторы 120 — 160.

Построение вертикальной линии осуществляется в цикле операторами 130 — 160.

4. Напишите программу построения отрезка горизонтальной ломаной линии, начинающегося в точке $A(X, Y)$ и имеющего N изломов. Ширину

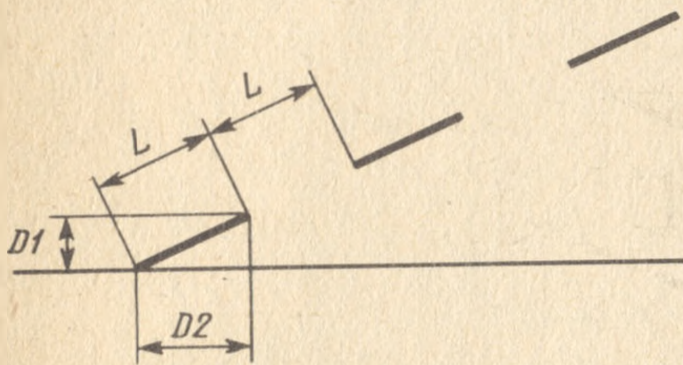


Рис. 42

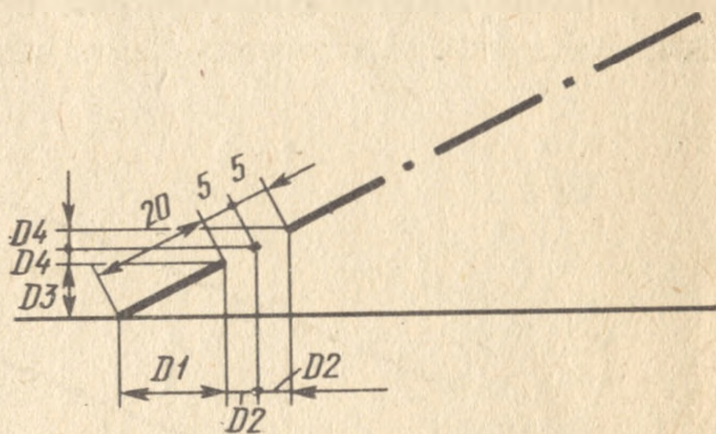


Рис. 43

линии возьмите равной 10 дискретам экрана дисплея, а угол излома равным $\pi/2$. Исходные данные: координаты точки начала линии и количество изломов введите с клавиатуры дисплея.

Программа построения отрезка ломаной линии может иметь вид:

```

05 REM ПРОГРАММА ПОСТРОЕНИЯ ЛОМАННОЙ ЛИНИИ
10 PRINT "ВВОД ИСХОДНЫХ ДАННЫХ: A(X1,Y1), КОЛИЧЕСТВО ИЗЛОМОВ"
20 INPUT X1,Y1,N
30 PLOT <,,R>,<X1,Y1,U>
40 FOR I=1TON/2
50 PLOT <10,10,D>,<10,-10,D>
60 NEXT I
70 END

```

5. Напишите программу построения отрезка штрих-пунктирной прямой линии, ограниченного точками с координатами $A(X, Y)$, $B(X, Y)$. Величину штриха возьмите равной 20 дискретам экрана дисплея, интервал между штрихами равным 5 дискретам, точку-разделитель разместите посередине интервала между штрихами. Исходные данные: координаты отрезка линии введите с клавиатуры дисплея.

Текст программы построения отрезка штрих-пунктирной прямой линии может иметь вид:

```
05 REM ПРОГРАММА ПОСТРОЕНИЯ ШТРИХ-ПУНКТИРНОЙ ЛИНИИ
10 PRINT "ВВОД ИСХОДНЫХ ДАННЫХ: A(X1,Y1),B(X2,Y2)"
20 INPUT X1,Y1,X2,Y2
30 IF X2=X1 THEN 140
40 REM ПОСТРОЕНИЕ НАКЛОННОЙ ИЛИ ГОРИЗОНТАЛЬНОЙ ЛИНИИ
50 D1=INT(20*COS(ARCTAN((Y2-Y1)/(X2-X1))))
60 D2=INT(5*COS(ARCTAN((Y2-Y1)/(X2-X1))))
70 D3=INT(20*SIN(ARCTAN((Y2-Y1)/(X2-X1))))
80 D4=INT(5*SIN(ARCTAN((Y2-Y1)/(X2-X1))))
90 PLOT <,,R>,<X1,Y1,U>
100 FOR I=X1TOX2STEPD1+2*D2+1
110 PLOT <D1,D3,D>,<D2,D4,U>,<1,1,D>,<D2,D4,U>
120 NEXT I
130 GOTO 190.
140 REM ПОСТРОЕНИЕ ВЕРТИКАЛЬНОЙ ЛИНИИ
150 PLOT <,,R>,<X1,Y1,U>
160 FOR I=1TOY2STEP20+5*2+1
170 PLOT <,20,D>,<,5,U>,<,1,D>,<,5,U>
180 NEXT I
190 END
```

При выполнении оператора 30 выясняется расположение отрезка строящейся линии. На рис. 43 представлены геометрические построения, в соответствии с которыми в операторах 50 — 80 осуществляется расчет координат строящейся наклонной или горизонтальной линий. Далее в цикле (операторы 100 — 120) осуществляется построение штрих-пунктирной линии. Построение вертикальной линии осуществляется в цикле операторами 160 — 180.

6. Напишите программу построения отрезка сплошной линии малой дуги радиуса R , ограниченного точками с координатами $A(X, Y)$, $B(X, Y)$. Величину радиуса R и координаты точек введите с клавиатуры дисплея.

7. Напишите программу построения отрезка сплошной линии большой дуги радиуса R , ограниченного точками с координатами $A(X, Y)$, $B(X, Y)$. Величину радиуса R и координаты точек введите с клавиатуры дисплея.

8. Напишите программу построения отрезка плавной кривой линии, ограниченного точками с координатами $A(X, Y)$, $B(X, Y)$. Кривую линию образуйте из двух дуг, построенных на частях отрезка прямой линии, соединяющего точки A и B . Отношение величин частей отрезка прямой линии между точками A и B задайте в виде констант в программе. Исходные данные координаты точек $A(X, Y)$ и $B(X, Y)$ введите с клавиатуры дисплея. Рассмотрите несколько вариантов: кривая образуется из двух малых дуг окружностей с радиусами R_1 и R_2 ; кривая образуется из малой и большой дуг окружностей с радиусом R_1 или R_1 и R_2 . Значение величин радиусов окружностей задайте в программе константами.

9. Решите задачи 6, 7, 8 для случая, когда строится штриховая линия. В соответствии с требованиями ГОСТ 2.303—68 длина штриха такой линии может изменяться в пределах 2—8 мм, а расстояние между штрихами 1—2 мм.

10. Напишите программу построения линии (отрезка сплошной прямой горизонтальной или вертикальной линии с изображением стрелок на концах) длиной L в соответствии с ГОСТ 2.307—68. Координаты точки начала (конца) линии и ее длину введите с клавиатуры дисплея.

11. Напишите программу построения отрезка сплошной волнистой линии, предназначенной для изображения линии обрыва, между точками с координатами $A(X, Y)$ и $B(X, Y)$ с помощью светового пера. Координаты точек $A(X, Y)$ и $B(X, Y)$ введите с клавиатуры дисплея.

Геометрические фигуры и стандартные графические символы

12. Напишите программу построения квадрата со стороной A . Величину стороны квадрата и координаты нахождения в рабочее поле экрана дисплея одной из его вершин введите с клавиатуры.

Текст программы построения квадрата может иметь вид:

```
05 REM ПРОГРАММА ПОСТРОЕНИЯ КВАДРАТА
10 PRINT "ВВОД ИСХОДНЫХ ДАННЫХ:"
20 PRINT "СТОРОНА КВАДРАТА - A, КООРДИНАТЫ ЛЕВОЙ НИЖНЕЙ ВЕРШИНЫ"
30 INPUT A, X1, Y1
40 PLOT <, ,R>, <X1,Y1,U>, <A,0,D>, <0,A,D>, <-A,0,D>, <0,-A,D>
50 END
```

13. Напишите программу построения треугольника с вершинами, имеющими координаты $A(X, Y)$, $B(X, Y)$, $C(X, Y)$. Координаты вершин треугольника введите с клавиатуры дисплея.

Текст программы может иметь вид:

```
05 REM ПРОГРАММА ПОСТРОЕНИЯ ТРЕУГОЛЬНИКА
10 PRINT "ВВОД ИСХОДНЫХ ДАННЫХ:"
20 PRINT "КООРДИНАТЫ ТРЕХ ВЕРШИН - A(X1,Y1), B(X2,Y2), C(X3,Y3)"
30 INPUT X1, Y1, X2, Y2, X3, Y3
40 PLOT <, ,R>, <X1,Y1,U>
50 PLOT <X2-X1, Y2-Y1, D>, <X3-X2, Y3-Y2, D>, <X1-X3, Y1-Y3, D>
60 END
```

14. Напишите программу построения окружности радиуса R с центром в точке с координатами $A(X, Y)$. Исходные данные введите с клавиатуры дисплея.

15. Напишите программу построения правильного шестиугольника, имеющего длину стороны L . Исходные данные: длину стороны L и координаты точки начала построения введите с клавиатуры дисплея.

Текст программы может иметь вид:

```
05 REM ПРОГРАММА ПОСТРОЕНИЯ ШЕСТИУГОЛЬНИКА
10 PRINT "ВВОД ИСХОДНЫХ ДАННЫХ:"
20 PRINT "СТОРОНА ШЕСТИУГОЛЬНИКА, КООРДИНАТЫ ТОЧКИ НАЧАЛА ПОСТРОЕНИЯ"
30 INPUT L, X1, Y1
40 SELECT D
50 PLOT <, ,R>, <X1, Y1, U>
60 FOR I=0 TO 6
70 X=L*COS(I*60)
80 Y=L*SIN(I*60)
90 PLOT <X, Y, D>
100 NEXT I
110 END
```

16. Напишите программу построения правильного N -угольника, имеющего длину стороны L . Исходные данные: длину стороны L и координаты точки начала построения введите с клавиатуры дисплея.

Программа построения правильного N -угольника может иметь вид:

```

05 REM ПРОГРАММА ПОСТРОЕНИЯ ПРАВИЛЬНОГО N-УГОЛЬНИКА
10 PRINT "ВВОД ИСХОДНЫХ ДАННЫХ : "
20 PRINT "СТОРОНА N-УГОЛЬНИКА, КОЛИЧЕСТВО УГЛОВ, "
30 PRINT "КООРДИНАТЫ ТОЧКИ НАЧАЛА ПОСТРОЕНИЯ"
40 INPUT L, N, X1, Y1
50 SELECT D
60 B=360/N
70 PLOT <, ,R>, <X1,Y1,U>
80 FOR I=0 TO N
90 X=L * COS(90-B * I)
100 Y=L * SIN(90-B * I)
110 PLOT<X,Y,D>
120 NEXT I
130 END

```

17. Напишите программу построения прямоугольника со сторонами A и B . Длины сторон задайте в дискретах. Исходные данные: длины сторон A и B , а также координаты точки одной из вершин прямоугольника введите с клавиатуры дисплея.

18. Напишите программу построения выпуклого многоугольника с вершинами заданных точек с помощью светового пера. Начальное положение графического курсора введите с клавиатуры дисплея.

19. Напишите программу построения многоугольника по заданным размерам. В качестве примера возьмите графические символы для отображения технических средств на структурных схемах (ГОСТ 24.303—80) или графические обозначения носителей данных (ГОСТ 19.003—80). В соответствии с требованиями приведенных ГОСТ величина A имеет минимальное значение 10 мм, дальнейшее ее увеличение возможно на число кратное 5, а размер $B = 1,5 A$. Исходные данные: величины A и B , координаты точки начала построения введите с клавиатуры дисплея.

20. Напишите программу построения фигуры, с одних двух сторон ограниченной прямыми линиями удвоенной толщины, а с двух других сторон — сплошными волнистыми линиями. Координаты точек пересечения линий введите с клавиатуры дисплея. В программе используйте световое перо.

21. Напишите программу построения окружности с осевыми линиями. Величину радиуса R и координаты точки центра $A(X, Y)$ введите с клавиатуры дисплея.

22. Напишите программу штриховки квадрата. Координаты вершин квадрата, угол наклона линий штриховки и шаг штриховки введите с клавиатуры дисплея.

Крепежные изделия

23. Напишите программу построения изображения чертежа шайбы по ГОСТ 2.315—68. Размеры шайбы и расположение в рабочем поле экрана дисплея введите с клавиатуры.

24. Напишите программу построения изображения чертежа шайбы пружинной по ГОСТ 2.315—68. Размеры шайбы и расположение в рабочем поле экрана дисплея введите с клавиатуры.

25. Напишите программу построения изображения чертежа шпонки призматической по ГОСТ 2.315—68. Размеры шпонки и ее расположение в рабочем поле экрана дисплея задайте с клавиатуры.

26. Напишите программу построения изображения чертежа штифта конического по ГОСТ 2.315–68. Размеры штифта и его расположение в рабочем поле экрана дисплея введите с клавиатуры.

27. Напишите программу построения изображения чертежа винта с потайной головкой по ГОСТ 2.315–68. Размеры винта и его расположение в рабочем поле экрана дисплея введите с клавиатуры.

28. Решите задачу 27 для изображения винта с полукруглой головкой по ГОСТ 2.315–68.

29. Напишите программу построения изображения чертежа гайки шестигранной по ГОСТ 2.315–68. Размеры и расположение гайки в рабочем поле экрана дисплея введите с клавиатуры.

Узлы машиностроительных чертежей

30. Напишите программу построения изображенного на рис. 10 чертежа сварного углового лобового шва двух стальных полос. Толщину полос δ задайте равной 12 мм в качестве константы в программе. Координаты точек начала построения и величину ширины полос введите с клавиатуры дисплея.

31. Напишите программу построения изображенного на рис. 11 чертежа сварного стыкового шва двух стальных полос. Толщину и ширину полос, ширину шва задайте с клавиатуры дисплея. Изображение чертежа поместите в центре рабочего поля экрана дисплея.

32. Напишите программу построения изображенного на рис. 2 чертежа однорядного заклепочного шва двух стальных полос. Исходные данные для построения изображения введите с клавиатуры дисплея: $d = 24$ мм, $\delta = 12$ мм, $b = 240$ мм, $l = 40$ мм.

33. Напишите программу построения изображенного на рис. 14 чертежа шпоночного соединения вала и ступицы. Исходные данные для построения изображения введите с клавиатуры дисплея.

34. Напишите программу построения изображенного на рис. 15 чертежа соединения призматической шпонкой. Размеры, указанные на чертеже, введите с клавиатуры дисплея.

35. Напишите программу построения изображенного на рис. 16 чертежа грузоподъемного крюка. Основные размеры задайте с клавиатуры дисплея, соблюдая при этом пропорциональность. Для построения изображения крюка воспользуйтесь световым пером, гайку представьте в упрощенном виде.

Элементы электрических схем

36. Напишите программу построения изображения резистора. Исходные данные для построения возьмите из ГОСТ 2.728–74 и задайте их в качестве констант в тексте программы.

37. Напишите программу построения изображения конденсатора в соответствии с требованиями ГОСТ 2.728–74. Исходные данные задайте в качестве констант в тексте программы, а координаты точки начала построения введите с клавиатуры дисплея.

38. Напишите программу построения изображения катушки индуктивности в соответствии с требованиями ГОСТ 2.723–68. Исходные данные

задайте в качестве констант в тексте программы, а координаты точки начала построения введите с клавиатуры дисплея.

39. Напишите программу построения изображения трансформатора (рис. 21). Исходные данные для построения изображения возьмите в соответствии с ГОСТ 2.723—68. Координаты точки начала построения введите с клавиатуры дисплея.

40. Напишите программу построения изображения транзистора в соответствии с требованиями ГОСТ 2.730—73. В качестве примера возьмите многоэмиттерный транзистор. Исходные данные возьмите из описания ГОСТ 2.730—73. Координаты точки начала построения введите с клавиатуры дисплея.

41. Напишите программу построения изображения диода в соответствии с требованиями ГОСТ 2.730—73. Размеры изображения задайте в качестве констант в тексте программы, а координаты точки начала построения введите с клавиатуры дисплея.

42. Напишите программу построения изображения микросхемы И—ИЛИ—НЕ, имеющей в своем составе три элемента И, каждый на четыре входа. Изображение постройте в соответствии с требованиями ГОСТ 2.743—82. Координаты точки начала построения введите с клавиатуры дисплея.

Узлы электрических схем

43. Напишите программу построения изображения электрической схемы делителя, представленного на рис. 18. Исходные данные для построения задайте в тексте программы в качестве констант.

44. Напишите программу построения изображения электрической схемы LC-фильтра, представленной на рис. 22. Исходные данные для построения задайте в качестве констант в тексте программы.

45. Напишите программы построения изображения электрической схемы реостатно-емкостного фильтра, изображенной на рис. 24. Исходные данные для построения задайте в качестве констант в тексте программы.

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

Григорьев М.М. Сборник задач и упражнений по регулировке и ремонту телевизоров цветного изображения. — М.: Высшая школа, 1983.

Дьяконов В.П. Справочник по алгоритмам и программам на языке БЕЙСИК для персональных ЭВМ. — М.: Наука, 1987.

Жеребцов И.П. Электрические и магнитные цепи. Основы электротехники. — Л.: Энергоатомиздат, 1982.

Мягков В.Д. Краткий справочник конструктора. — Л.: Машиностроение, 1975.

Пантелеева З.Т. Графика вычислительных процессов. — М.: Финансы и статистика, 1983.

Разработка и оформление конструкторской документации РЭА. Справочное пособие / Э.Т. Романычева, А.К. Иванова, А.С. Куликов, Т.А. Новикова. — М.: Радио и связь, 1984.

Справочник по машинной графике в проектировании / В.Е. Михайленко, В.А. Анпилогова, Л.А. Кириевский и др. — К.: Будивельник, 1984.

Соскин Г.Б., Оксман В.М. Применение микрокалькуляторов при подготовке чертежников-конструкторов в технических училищах. Методические рекомендации. — М.: ВНМЦентр профессионально-технического обучения молодежи, 1984.

ОГЛАВЛЕНИЕ

Предисловие	3
Глава первая. Алгоритмизация типовых задач по профессиям	6
§ 1. Алгоритм и программа. Способы описания алгоритмов	6
§ 2. Решение задач с использованием типовых структур алгоритмов . .	8
Задачи для самостоятельного составления алгоритмов	15
Глава вторая. Программирование на языке БЕЙСИК	27
§ 3. Алфавит языка и элементарные конструкции	27
§ 4. Основные операторы языка БЕЙСИК	32
§ 5. Дополнительные средства программирования на языке БЕЙСИК .	43
Программирование типовых задач по профессиям	46
Ответы	59
Приложение 1. Организация работы на персональном компьютере	69
Приложение 2. Алгоритмы и программы машинной графики	76
Рекомендуемая литература	96

25 коп.

МАШИНОСТРОЕНИЕ,
ПРИБОРОСТРОЕНИЕ
И СВЯЗЬ

1



ТЯЖЕЛАЯ
ПРОМЫШЛЕННОСТЬ
И ТРАНСПОРТ

2

ЛЕГКАЯ
ПРОМЫШЛЕННОСТЬ

3

СТРОИТЕЛЬСТВО

4

СЕЛЬСКОЕ ХОЗЯЙСТВО

5