

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УО «БОРИСОВСКИЙ ГОСУДАРСТВЕННЫЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ»

Специальность Т 03 01
Технология, оборудование и
автоматизация машиностроения

ТЕХТРАН — СИСТЕМА ПРОГРАММИРОВАНИЯ ОБОРУДОВАНИЯ С ЧПУ

Методические указания

предмет: «Программирование обработки на станках с ЧПУ»

Разработал:

Е.К. Дыкуша

Утверждено на заседании
цикловой комиссии

Председатель цикловой комиссии

В.М. Адаменко

ПРЕДИСЛОВИЕ

Внедрение оборудования с числовым программным управлением (ЧПУ), средств вычислительной техники, а также создаваемых на их основе гибких производственных систем (ГПС) определяется современным развитием промышленного производства, которое характеризуют частая смена продукции, короткий жизненный цикл изделий, а также возрастающие сложность и точность обработки. В этих условиях одной из важнейших задач технологической подготовки производства является разработка управляющих программ (УП) для станков с ЧПУ.

Разработка УП – сложный и трудоемкий процесс, во многом определяющий эффективность использования оборудования с ЧПУ и качество обрабатываемых деталей, поэтому сейчас большое внимание обращается на разработку и совершенствование систем и методов автоматизированной подготовки УП. Особое внимание проблеме автоматизированной подготовки управляющих программ уделяется в составе комплексной системы САПР–АСТПП–ГПС. Управляющая программа при этом является результатом сквозного цикла обработки информации от чертежа детали до программы ее изготовления на станках с ЧПУ. Система автоматизированного программирования (САП) оборудования с ЧПУ – составная часть такой комплексной системы. Несмотря на большое число работ в области автоматизации программирования оборудования с ЧПУ в нашей стране и за рубежом, они не получили полного и систематизированного изложения в литературе. Предлагаемая книга содержит изложение основных вопросов, связанных с автоматизированной подготовкой УП для станков с ЧПУ на базе САП Техтран.

В гл. 1 дана классификация САП по различным критериям, позволяющая получить представление о направлениях развития автоматизации программирования, и приведена функциональная структура САП Техтран. Гл. 2 посвящена описанию проблемно-ориентированного языка программирования оборудования с ЧПУ Техтран. Описание приведено в полном объеме и может быть использовано на практике при разработке УП. Гл. 3 посвящена операторам постпроцессора, задающим необходимые технологические данные для выполнения УП на станке. В гл. 4 изложены особенности технологической подготовки производства в условиях использования САП, приведены примеры различных видов обработки и описание способов контроля УП.

В качестве основной задачи авторы ставили перед собой полное и систематизированное описание языка программирования оборудования с ЧПУ Техтран, изложенного в гл. 2. Гл. 1, 3, 4, включены для более глубокого восприятия основного материала.

Все замечания и пожелания авторы просят направлять по адресу: 191065, Ленин-град, ул. Дзержинского, 10, ЛО издательства «Машиностроение».

Глава 1 АВТОМАТИЗАЦИЯ ПРОГРАММИРОВАНИЯ ОБОРУДОВАНИЯ С ЧПУ

1.1 Системы автоматизации программирования

Роль систем автоматизации программирования в процессе подготовки управляющих программ для станков с ЧПУ можно сейчас сравнить с той ролью, которую сыграли компиляторы языков высокого уровня в процессе разработки программ на ЭВМ. Появление САП является логическим следствием развития вычислительной техники, программирования, станкостроения, а также обусловлено возрастающей сложностью деталей, и, следовательно, высокой трудоемкостью «ручного» программирования их обработки. За более чем двадцатилетний период, прошедший с тех пор, как начали эксплуатироваться САП, сменилось четыре поколения ЭВМ, появились микропроцессоры, разработаны и широко используются многооперационные станки типа «обрабатывающий центр» и роботы. Это естественным образом повлияло на методы подготовки управляющих программ и эволюцию развития САП.

Программирование оборудования с ЧПУ появилось совсем недавно. Результатом программирования является управляющая программа, которая представляет собой совокупность команд на языке программирования, соответствующая заданному алгоритму функционирования станка по обработке конкретной заготовки (см. ГОСТ 20523–80).

В книге будет идти речь об автоматизированном программировании, хотя последние достижения вычислительной техники и станкостроения в некоторой степени стирают грань между «ручным» и автоматизированным программированием. Рассмотрим программирование как составную часть процесса технологической подготовки производства.

«Ручное» программирование в традиционном понимании состоит в следующем. По заданному операционному технологическому процессу технолог рассчитывает траекторию движения инструмента, назначает необходимые технологические команды в соответствии с режимом обработки (например, подачу, охлаждение, смену инструмента и т. д.) и преобразует эти данные во внутреннее представление системы ЧПУ [4, 15, 17].

При автоматизированном программировании в идеальном случае все задачи ручного программирования должны решаться на ЭВМ, включая расчет режимов обработки. САП являются эффективным и надежным инструментом для их решения.

САП – это комплекс технических, программных, языковых, информационных средств, осуществляющих преобразование данных чертежа детали в коды устройства управления оборудованием с ЧПУ. Как правило, САП организованы по классической структуре: входной язык, процессор, промежуточный язык, постпроцессор. В зависимости от реализации возможны отклонения от этой структуры.

Входной язык САП – проблемно-ориентированный язык, предназначенный для описания исходных данных о детали и технологическом процессе ее обработки на оборудовании с ЧПУ.

Процессор САП – программное изделие, предназначенное для решения общих геометрических и технологических задач, а также задач управления процессом обработки данных на ЭВМ, реализация которого возможна в виде специализированного программного устройства.

Постпроцессор САП – программное изделие, предназначенное для адаптации управляющей программы к конкретному оборудованию с ЧПУ, реализация которого возможна в виде специализированного программного устройства.

Промежуточный язык процессор – постпроцессор – внутренний проблемно-ориентированный язык системы САП, служащий для представления данных, передаваемых от процессора к постпроцессору. (Все определения даны по ГОСТ 23501.603–84).

В литературе промежуточный язык процессор-постпроцессор встречается под названием CLDATA (*Cutter Location Data* – данные о перемещении инструмента).

Структура САП приведена на рис. 1.1. Обычно в состав САП входит библиотека постпроцессоров, каждый из которой предназначен для определенной модели системы ЧПУ или модели станка. Разрабатываются также универсальные постпроцессоры, способные формировать УП для различных систем ЧПУ. Графические постпроцессоры предназначены для вывода траектории движения Инструмента на графические отображающие устройства.

Рассмотрим классификацию САП по различным критериям (рис. 1.2) [9]. Первый критерий классификации САП – по числу управляемых координат – определяет

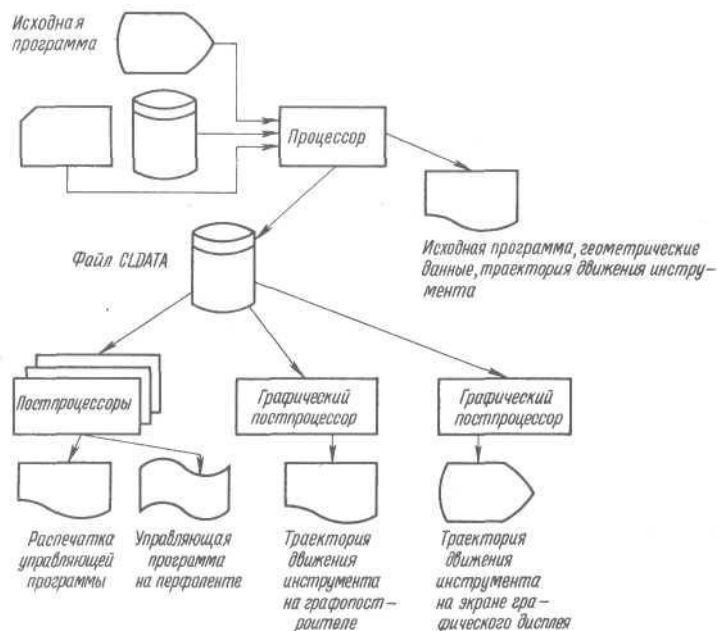


Рис. 1.1

функциональные возможности системы и класс обслуживаемого оборудования. Двух-координатные САП обеспечивают автоматизированную подготовку УП для двух-координатных станков (токарных, электроэрозионных, газорезательных и др.). Движение инструмента происходит в одной из координатных плоскостей. Движения по третьей координатной оси нет. 2,5-координатные САП обеспечивают подготовку УП для обработки произвольного плоского контура при условии, что поверхность детали является плоскостью, произвольно ориентированной в пространстве. Такие САП относятся к самому распространенному типу систем по двум причинам: во-первых, они охватывают подготовку УП для широкого класса станков (токарных, фрезерных, сверлильных и др.), во-вторых, трудоемкость разработки, сопровождения и эксплуатации такой САП не требует больших инженерных и технических ресурсов. К числу таких САП относятся ЕСПС-ТАУ, САП-СМ, Техтран, ADAPT (США), NELAPT (Великобритания), Autotech (ГДР), FAPT (Япония), IFAPT (Франция). В трех координатных САП в отличие от 2,5-координатных поверхность обрабатываемой детали может быть произвольной поверхностью второго порядка, что позволяет расширить область их применения. Типичными представителями таких САП являются MODAPT (Италия), PROMO (Франция). Многокоординатные САП в дополнение к обычным трех координатным перемещениям обеспечивают угловые перемещения вокруг одной из координатных осей. Реализация угловых перемещений производится постпроцессором и зависит от того, каким органом станка производится поворот – шпиндельной головкой или столом. Наиболее значительной из семейства многокоординатных САП является разработанная в 1961 г. в США система APT (*Automatically Programmed Tools*).

Следующий критерий классификации – *по уровню принятия решений* – определяет уровень автоматизации решаемых в САП технологических задач. К технологическим задачам относятся реализация типовых технологических циклов точения, сверления, нарезания резьбы, фрезерования кругового и прямоугольного (внутреннего и наружного), фрезерования пазов и карманов, разбиения припуска на проходы, расчет режимов резания и др.

В настоящее время трудно назвать САП, в которой не решалась бы одна из названных задач. Поэтому основной принцип классификации по этому критерию

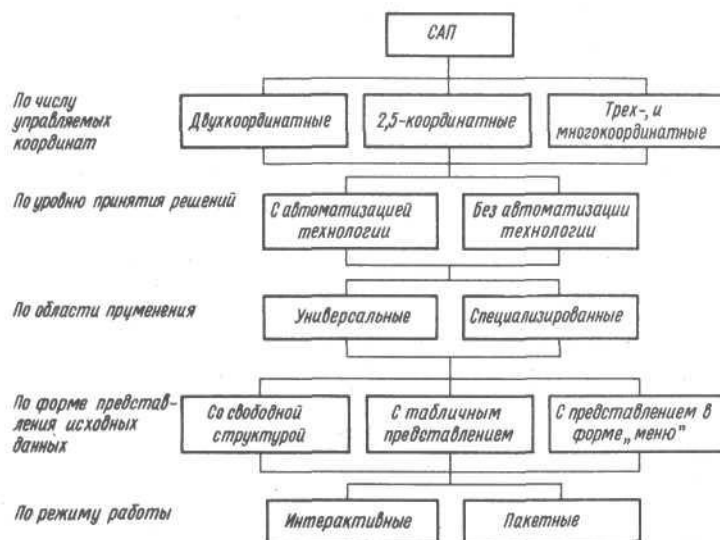


Рис. 1.2

состоит в том, является автоматизация технологических задач свойством, заложенным в концепцию САП при ее разработке, или такая автоматизация реализована на основе библиотеки типовых технологических процедур. В большинстве САП (например, САП-СМ, АРТ, FAPT, IFAPT, MODAPT) автоматизация технологических задач решается при помощи технологических процедур. Среди САП со встроенной автоматизацией технологических задач можно назвать ЕСПС-ТАУ, Техтран-Т, Autotech, EXAPT (ФРГ), FANUC-3Т и MAZATROL (Япония).

В зависимости от области применения САП можно разделить на универсальные и специализированные. Универсальные САП – это системы широкого назначения, с развитым входным языком, позволяющим разрабатывать программы, имеющие сложную логическую структуру, описывать обработку произвольных контуров, производить математические вычисления, а также создавать типовые технологические процедуры для различных классов оборудования. АРТ является типичной САП этого класса. Она представляет собой обобщенную универсальную САП, предназначенную для программирования обработки сложных поверхностей, с управлением по трем и пяти осям [5]. АРТ является прототипом семейства САП, разработанных в различных странах, в том числе в СССР. В частности, за основу языка Техтран было взято подмножество языка АРТ. АРТ позволяет программировать обработку поверхностей типа конусов, цилиндров, поверхностей второго порядка (эллипсоид, эллиптический конус, гиперболический параболоид и др.), линейчатых и каркасных поверхностей, сфер. АРТ является многофункциональной системой, но основными областями ее эффективного применения являются авиа-, судо- и автомобилестроение, где наиболее часто встречается обработка сложных поверхностей (например, формообразующих элементов штампов, гребных винтов, элементов конструкций самолета и др.). Несмотря на высокую производительность и большие возможности, система АРТ не нашла широкого применения вследствие сложности в изучении и эксплуатации [5].

Специализированные САП классифицируются по видам обработки (токарной, фрезерной, сверлильно-расточной и др.). Последние годы развитие САП идет по пути создания специализированных систем с высоким уровнем автоматизации решения технологических задач.

В ФРГ разработаны семейства САП EXAPT, состоящие из технологически ориентированных подсистем, в основу которых были положены концепции АРТ [18].

Типичным представителем специализированных САП является EXAPT (ФРГ). EXAPT состоит из трех подсистем. EXAPT-M01 (Basic - EXAPT) - универсальная САП – ядро системы; EXAPT-M02 – специализированная, технологически ориентированная САП для токарных станков, EXAPT-M03 – для сверлильных, фрезерных и станков типа обрабатывающий центр. Программирование в EXAPT-M02 состоит из следующих этапов: описания геометрии детали, описания технологических переходов, описания процесса обработки. При этом автоматизировано решение следующих задач: разбиение на проходы, расчет режимов резания, построение траектории движения инструмента при черновой и чистовой обработке, контроль на наличие столкновений.

В EXAPT-M03 автоматизированы задачи выбора инструмента, расчета ходов инструмента при позиционной и контурной обработке, расчет режимов резания, контроль на наличие столкновений.

Система EXAPT является наиболее типичным представителем семейства специализированных САП. Среди множества других САП можно назвать ориентированную на токарную обработку TAUPROG (Венгрия) и отечественные ЕСПС-ТАУ для сверлильных, токарных и фрезерных работ и Техтран-Т для токарных работ, в которых также решены перечисленные технологические задачи.

В настоящее время выпускаются системы ЧПУ типа CNC со встроенными САП – программаторы. Программаторы – это специализированные САП, предназначенные для оперативной подготовки УП непосредственно на станке. Это позволяет сократить простой станка за счет совмещения обработки детали с разработкой, отладкой и редактированием другой УП. Наибольший интерес представляют системы BOSCH-PEG (ФРГ) и FANUC-3T (Япония) – для токарных станков, FANUC 6M-MODEL E и MAZATROL-M (Япония) – для обрабатывающих центров [14, 16].

По *форме представления исходных данных* можно выделить САП со свободной структурой представления, с табличной формой и представлением в форме «меню».

Большинство известных САП имеют свободную форму представления исходных данных. Программа на языке таких САП представляет собой множество операторов, состоящих из элементов языка (служебных слов, идентификаторов, специальных символов и т. д.) и строящихся по заданному синтаксису (например, входные языки многих из названных САП соответствуют синтаксису языка АРТ). В дальнейшем для удобства будем отождествлять названия САП и ее входного языка.

При табличной форме представления технолог заполняет специальные бланки в форме таблиц. Каждая графа таблицы соответствует определенному виду исходных данных, например, геометрии детали, технологическим командам, управляющей информации и т. д. Имеются также САП, сочетающие обе формы представления (САДКО, АППР-АРМ).

Представления в форме «меню» – это свойство интерактивных САП. С экрана дисплея в определенном формате запрашивается требуемая информация. Технолог остается только заполнить соответствующие графы, высвечиваемые на экране дисплея, или ответить на задаваемые системой вопросы. Наиболее перспективными являются САП, сочетающие представление исходных данных со свободной структурой и в форме «меню», например, в системе ТС-АРТ [20].

Последний критерий классификации САП – по *режиму работы*. В настоящее время существуют два основных режима работы САП: пакетный и интерактивный, которые находятся в прямой зависимости от типа ЭВМ и операционной системы, для которых реализована данная САП. Первые САП работали в пакетном режиме. Программа, подготовленная технологом на входном языке САП и записанная на магнитный носитель, перфоленту или перфокарты, полностью вводилась в ЭВМ и преобразовывалась в УП для станка. В случае ошибок в исходной программе эта процедура повторялась.

Разработка интерактивных САП является следствием развития вычислительной техники – появлением алфавитных и графических дисплеев, а также мини- и микро-ЭВМ. Исходные данные в интерактивных САП задаются в форме «меню» или в режиме «вопрос – ответ». Существуют интерактивные САП и со свободной формой представления (например, FAPT, где последовательно после ввода каждого оператора можно получить на экране дисплеев или печатающем устройстве результат его выполнения). Значительное преимущество таких САП перед пакетными состоит в сокращении сроков обучения технологов-программистов и технологов-операторов методам программирования за счет диалога и возможности повторения расчета УП с любой исходной позиции. Недостатком их является отсутствие универсальности в подходе к разработке УП вследствие того, что диалог ограничивает разработчика в выборе средств для решения задачи. Интерактивные САП эффективны при эксплуатации в производствах с невысокой сложностью и высоким уровнем унификации изготавливаемых деталей или их элементов.

Представленная классификация САП является попыткой систематизировать накопленный опыт и направления развития данной области инженерной деятельности за последние годы и носит условный характер. В последующие годы параллельно с развитием диалоговых, оперативных систем подготовки УП непосредственно у станка дальнейшее развитие получают системы программирования, работающие в комплексных системах автоматизированного проектирования и производства (CAD/CAM – *Computer Aided Design/Computer Aided Manufacturing*). Комплексная система состоит из ряда подсистем, с помощью которых производится формирование цифровой модели детали, вычерчивание рабочих чертежей и эскизов на графопостроителях, проектирование технологического процесса и

формирование управляющей программы [19]. Объединение этих процессов в единый непрерывный процесс переработки информации и управления технологическими процессами и оборудованием дает ряд преимуществ перед описанными ранее системами и методами. Преимущества эти следующие: сокращение затрат и времени на подготовку УП; повышение качества УП; повышение эффективности использования оборудования с ЧПУ за счет быстрой загрузки программ в память системы ЧПУ и отсутствие брака в УП; высвобождение технологов-программистов, занимающихся разработкой УП; сокращение сроков подготовки производства новых изделий.

В частности, имеется опыт создания комплексной системы проектирования технологических процессов и подготовки УП для номенклатуры деталей тел вращения, функционирующей в условиях гибкой производственной системы «Групппроцесс-Техтран». Подробное описание перечисленных САП приведено в работах [1, 3, 5, 7, 10, 12, 13, 14, 16, 20, 22, 23].

1.2 Структура и функциональные возможности САП Техтран

В основу организации САП Техтран был положен традиционный подход. Система состоит из входного языка, процессора, промежуточного языка и постпроцессоров. При разработке системы преследовалось несколько целей. Во-первых, САП должна быть простой в обслуживании, т. е. вмешательство оператора должно быть сведено к минимуму. Во-вторых, САП должна быть мобильной (это позволяет минимизировать затраты при адаптации САП для различных классов ЭВМ и операционных систем). В-третьих, САП должна допускать расширение выполняемых функций.

Необходимую гибкость САП Техтран приобрела благодаря модульной структуре. Для уменьшения объема оперативной памяти процессор и постпроцессор выполняют обработку данных за несколько проходов: процессор построен по трехпроходной схеме, постпроцессоры – по двухпроходной схеме. Модульная структура дает возможность вводить новые функции без существенной корректировки ранее написанных программ. Кроме того, такая структура позволяет организовать перекрытия и уменьшить тем самым объем оперативной памяти.

Требование мобильности программного обеспечения предопределило выбор языка программирования. Поскольку ко времени начала разработки (1975 г.) и вплоть

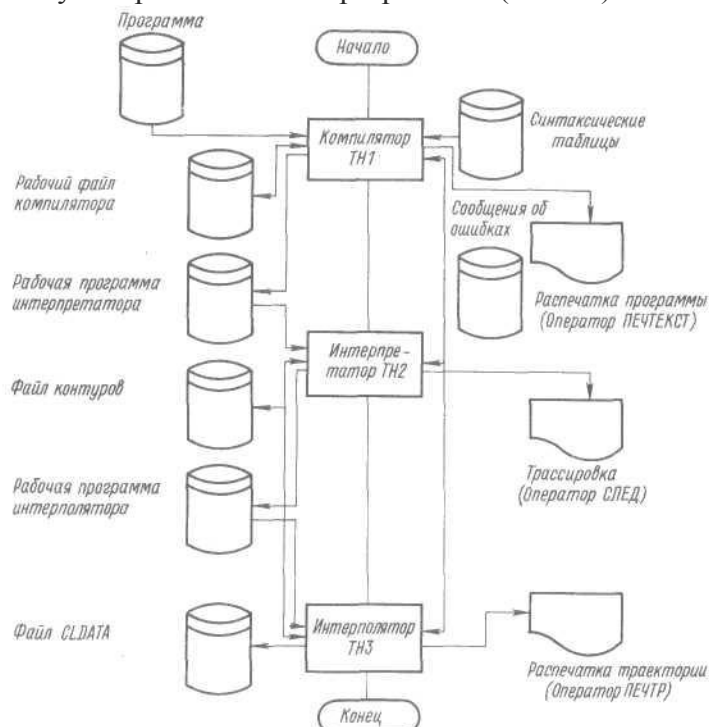


Рис. 1.3

до настоящего времени единственным языком программирования, компиляторы которого входят в состав всех операционных систем и обеспечивают необходимое качество объектного кода, был и остается Фортран, то именно ему и было отдано предпочтение. Около 95 % программ, составляющих САП Техтран, написано на Фортране, остальные – на Ассемблере. В настоящее время синхронные версии системы эксплуатируются на ЭВМ типа ЕС с операционной системой ОС ЕС и СМ с операционной системой ОС РВ.

Процессор осуществляет ввод и трансляцию исходной программы на входном языке системы, выполняет заданные геометрические построения и рассчитывает траекторию движения центра инструмента. Результаты расчета выводятся в файл CLDATA. Структура процессора системы Техтран приведена на рис. 1.3. Процессор организован в виде трех последовательно выполняемых модулей: компилятора, интерпретатора и интерполятора. Каждый модуль может рассматриваться как законченная программа. Обмен данными между модулями производится только через промежуточные файлы на диске.

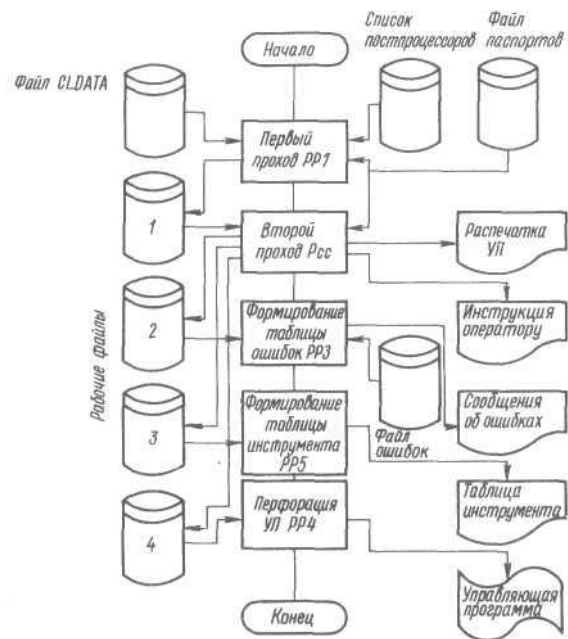


Рис. 1.4

Компилятор вводит программу на Техтране, предварительно записанную на диск, и выполняет ее синтаксический контроль. На этапе компиляции проверяется, правильно ли заданы имена и служебные слова в программе, соблюдены ли требуемые форматы. Если часть текста оформлена в виде отдельного файла, то компилятор вставляет этот текст в исходную программу. Макросредства дают программисту возможность описывать типовые элементы в параметрическом виде и хранить их на магнитном диске. В процессе макрогенерации компилятор выполняет вставку тела макроса в исходную программу и заменяет формальные параметры фактическими значениями. Если обнаружена ошибка, то компилятор отмечает ее место в строке и печатает диагностическое сообщение. Затем он переходит к следующему оператору, чтобы за один прогон выявить максимальное число ошибок. Файл синтаксических

таблиц содержит описание форматов операторов Техтрана и служит для управления работой компилятора. Тексты сообщений об ошибках также хранятся на магнитном диске. Компилятор формирует на диске рабочую программу интерпретатора, в которой арифметические выражения преобразованы в польскую инверсную запись, геометрические определения преобразованы во внутренний формат и заданные в них модификаторы упорядочены. В процессе макрогенерации используется рабочий файл на диске.

Интерпретатор производит расчет канонических форм геометрических объектов. Другой функцией интерпретатора является вычисление значений арифметических и логических выражений и стандартных функций. Если в программе были использованы операторы цикла и операторы передачи управления, то интерпретатор выполняет эти операторы и тем самым приводит исходную программу к линейной. Рабочая программа интерполятора, формируемая интерпретатором, будет содержать только таблицу геометрических объектов, операторы движения и операторы постпроцессора. Интерпретатор может выполнять трассировку, т. е. распечатывать значения переменных и переходов в программе. Если в исходной программе были определены контуры, то их описания во внутреннем формате выводятся в специальный файл. При обнаружении ошибки в программе интерпретатор печатает диагностическое сообщение и продолжает обработку до заданного числа ошибок.

Интерполятор обрабатывает операторы движения и рассчитывает эквидистантную траекторию с учетом диаметра рабочего инструмента. Если интерполятор обнаруживает ошибку, то он выдает диагностическое сообщение и продолжает обработку от следующего оператора поточечного движения до заданного числа ошибок. В процессе вычислений интерполятор использует рабочую программу и файл контуров, подготовленные интерпретатором, а также файл сообщений об ошибках. Операторы постпроцессора не обрабатываются и передаются в выходной файл. Выходные данные интерполятора формируются в стандартном формате CLDATA.

Процессор рассчитан на обработку группы исходных программ за один проход. Каждый модуль процессора за один вызов обрабатывает всю группу. Программы, в которых на стадии компиляции или интерпретации обнаружены ошибки, удаляются из соответствующего выходного файла и не поступают на дальнейшую обработку. Программы с ошибками в операторах движения выводятся в файл CLDATA, но место ошибки отмечается специальной записью.

Постпроцессоры выполняют следующие функции: чтение файла CLDATA, подготовленного процессором, и синтаксический контроль операторов постпроцессора; перевод геометрических

данных в систему координат станка; проверку длины перемещений, допустимости подачи и оборотов; расчет кодов подач, кодов оборотов шпинделя, коррекции; учет динамических характеристик оборудования; формирование и вывод на печать и перфоленту управляющей программы и другие функции. - САП Техтран оснащена рядом постпроцессоров, каждый из которых формирует управляющие программы для определенной системы управления. Настройка постпроцессора на конкретную модель станка производится с помощью паспорта. Структура постпроцессора приведена на рис К 4.

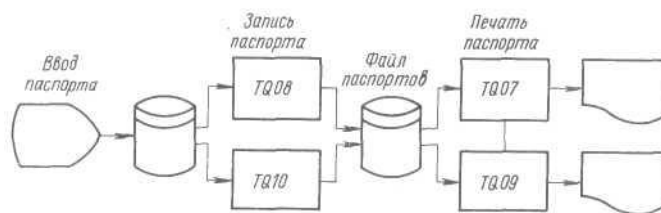


Рис. 1.5

Постпроцессоры построены по двухпроходной схеме. Программа первого прохода *PP1* читает записи *CLDATA*, сглаживает траекторию, проверяет, принадлежат ли перемещения рабочей зоне станка, рассчитывает вектор коррекции и формирует в рабочем файле 1 исходные данные для второго прохода. Программа второго прохода *Рсс*¹ выполняет синтаксический контроль операторов постпроцессора, учитывает динамические особенности оборудования, формирует на диске образ управляющей программы и другие рабочие файлы, распечатывает управляющую программу и инструкцию оператору станка. Постпроцессор формирует три рабочих файла, каждый из которых обрабатывается специальной программой:

рабочий файл 2 содержит коды сообщений об ошибках; он обрабатывается программой *PP3*, которая печатает таблицу сообщений об ошибках и таблицу недопустимых для данного станка операторов;

рабочий файл 3 содержит таблицу инструментов; он обрабатывается программой *PP5*, которая печатает таблицу инструмента и таблицу времени работы каждого инструмента;

рабочий файл 4 содержит управляющую программу во внутреннем формате (во внутреннем коде ЭВМ); он обрабатывается программой *PP4*, которая выполняет перекодировку в код системы ЧПУ и перфорирует ее.

Пользователям САП Техтран предоставлена возможность заполнять паспорта для имеющегося у них оборудования с ЧПУ при условии, что в составе библиотеки постпроцессоров имеется модуль для требуемой системы управления.

Для работы с паспортами предназначена подсистема Вопросник, состоящая из четырех программ. Структура Вопросника показана на рис. 1.5. Паспорт состоит из двух частей – характеристик оборудования и таблиц. Каждая часть паспорта обрабатывается двумя программами: одна предназначена для записи паспорта (*TQ08* и *TQ10*), другая – для его распечатки (*TQ07* и *TQ09*). Вопросник читает исходный текст паспорта и проверяет форматы и значения данных. После перекодировки паспорт в двоичном формате записывается в файл паспортов, который используется при прогоне постпроцессоров.

Графический контроль в САП Техтран решен с помощью постпроцессоров, которые либо формируют перфоленту для графопостроителей, работающих в автономном режиме, либо непосредственно выводят на экран или бумагу траекторию перемещения центра режущего инструмента.

Технологический процессор является расширением геометрического процессора Техтран в части программирования технологических переходов, а входной язык технологического процессора – расширением базового языка, который дополняется технологическими операторами. Выделение геометрического процессора в автономный модуль позволило определить пути такого расширения системы Техтран, при котором ядро остается неизменным. Расширение функций системы осуществляется включением технологических процессоров специального назначения. Положение технологического процессора в системе Техтран показано на рис. 1.6.

Процессор формирует файл *CLDATA1*, в котором содержатся геометрические и технологические данные. Технологические операторы обрабатываются процессором так же, как и операторы постпроцессора: процессор заменяет служебные слова их внутренними кодами и

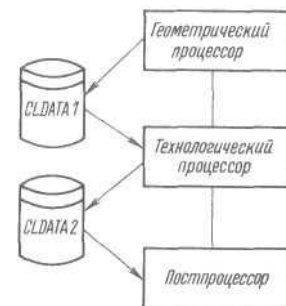


Рис. 1.6

вычисляет арифметические выражения. Расширенный файл *CLDATA1* обрабатывается технологическим процессором, на выходе которого формируется файл *CLDATA2*, содержащий траекторию движения инструмента и операторы постпроцессора.

Принцип расширения функций системы Техтран реализован в системе Техтран-Т – технологическом процессоре для программирования токарных переходов.

¹ Идентификация постпроцессоров осуществляется с помощью двухсимвольного кода постпроцессора ее. Например, постпроцессор для системы управления Н-33 называется М1, а для системы Н-22 – L1.

2.1 Основные понятия

Программа. Структура программы. Программа на Техтране представляет собой последовательность операторов и комментариев, описывающих алгоритм обработки деталей. Программа имеет следующую структуру:

ДЕТАЛЬ *литерал*
текст программы
КОНЕЦ

Литерал (см. п. 2.2) задает имя программы. В программу могут быть вставлены фрагменты исходного текста и тексты макросов. Программе на входном языке Техтран может соответствовать одна или несколько управляющих программ. Исходный текст программы состоит из последовательности строк. Максимальное число символов в строке равно 72. В одной строке может быть записан один или несколько операторов, между которыми должен стоять символ «точка с запятой» (;). Все символы текущего оператора должны размещаться до 72-й позиции. В противном случае часть оператора может быть перенесена на следующую строку. Для этого, после прерванной части оператора (до 72-й позиции) необходимо записать символ продолжения «процент» (%). Для совместимости с предыдущими версиями языка разрешается в качестве символа продолжения использовать знак «коммерческое И» с пробелом (&_). Число строк продолжения не ограничивается.

Текст оператора или комментария может начинаться с любой позиции. Число пробелов между операндами не ограничено. В качестве разделителя операндов вместо запятой можно использовать пробел (_). Для удобства чтения программ и ускорения их отладки рекомендуется располагать операторы и комментарии таким образом, чтобы текст программы являлся документом, по которому можно понять структуру и назначение программы. Желательно, чтобы имена меток и переменных, используемых в программе, отражали их назначение.

Для выделения функциональных частей программы рекомендуется использовать строки, состоящие из пробелов, знаков комментария, а также отступов.

Символические имена. Для идентификации в программе переменных величин (геометрических, вещественных, логических, меток операторов, имен макросов) используют символические имена. Символическое имя – это последовательность букв и цифр, начинающаяся с буквы. Длина его не должна превышать восьми символов. Пробелы внутри имени не разрешены. Символическое имя должно быть уникальным в программе и описывать один объект или класс объектов, например:

АБ – верно
КР15 – верно
K152АБ – верно
К 1 – неверно
K123455678910 – неверно
123K22 – неверно

Операторы, метки, комментарии. Оператор – основная функциональная единица языка. Он предназначен как для описания операций, которые предстоит выполнить, так и для установления свойств данных, идентифицируемых с помощью символических имен. Любой оператор может быть помечен меткой, для того чтобы на него можно было сослаться в других операторах данной программы.

Метка является символическим именем. Она должна предшествовать помечаемому оператору и может находиться либо в одной строке с оператором, либо на отдельной строке. После метки всегда должен быть служебный символ «двоеточие» (:). Метка используется в операторах НАМЕТКУ.

Для пояснения действий, выполняемых в программе, используют комментарии. Комментарий не обрабатывается процессором, а только выводится на печать при распечатке исходного текста. Комментарий может находиться как в одной строке с операторами исходной программы, так и занимать целую строку. Для выделения текста комментария нужно поставить перед ним служебный символ «номер» (#).

2.2 Структура языка

Алфавит языка. Для записи программы используют следующие символы: буквы русского алфавита А–Я; буквы латинского алфавита А–Z; цифры: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9; символы: «_» – пробел; «,» – запятая; «•» – точка; «=» – равно; «+» – плюс; «–» – минус; «*» – звездочка; «/» – наклонная черта; «&» – коммерческое «И»; «;» – точка с запятой; «(» – левая скобка; «)» – правая скобка; «:» – двоеточие; «'» – апостроф; «#» – номер; «%» – процент.

Синтаксис языка. В связи с тем, что в описании языка не приводится формального описания синтаксиса Техтрана, этот раздел содержит основные правила и обозначения, используемые для описания форматов его операторов. Каждый оператор входного языка, формируется из операндов в соответствии с синтаксическими правилами языка. В описании языка приведена структура каждого оператора (формат).

При описании формата каждого оператора использованы следующие правила и условные обозначения:

прописные буквы используют для обозначения служебных слов;

строчные буквы используют для обозначения опорных элементов, числовых значений параметров или арифметических выражений;

необязательные служебные слова заключают в квадратные скобки [];

возможные варианты записи модификаторов, являющиеся альтернативными, заключают в фигурные скобки { } и записывают в столбец или разделяют символом « | »;

параметры, назначающие линейные и угловые размеры, задаются целыми и десятичными долями миллиметров и градусов.

Элементы языка. В языке разницы между целыми и вещественными числами не существует. Число представляется в виде последовательности цифр, среди которых может встретиться точка или буква Е. Точка отделяет целую часть числа от дробной, а буква Е – мантиссу числа от его порядка. Перед самым числом и перед его порядком может стоять знак «минус» (–). Пробелы и другие символы внутри числа недопустимы, например: 1.23 – верно; 2.15Е5 – верно; –3Е–6 – верно; 3 Е2 – неверно; Е4 – неверно.

Служебные слова являются зарезервированными словами и составляют словарь языка Техтран. Они используются для написания операторов программы и состоят только из букв русского алфавита. Служебные слова делятся на главные, вспомогательные и модификаторы. Главное слово определяет тип оператора. Вспомогательное слово и модификатор служат для однозначного задания оператора. Служебное слово не может использоваться в качестве имени, переменной или метки. Список служебных слов приведен в Приложениях 1–4.

Переменная в Техтране – это символическое изображение величины, которая может принимать различные значения. Основными характеристиками переменной являются ее имя и тип.

В группу *служебных символов* входят следующие символы: «_», «+», «–», «*», «/», «=», «;», «.», «,», «:», «'», «#», «&», «()», «%». Употребление служебных символов в программе зависит от контекста. Описание их функций производится по мере надобности в соответствующих разделах.

Литералом является заключенный в апострофы произвольный текст, не содержащий символа *ф*. В литералах могут быть использованы любые символы (например, латинские буквы, !, >, < и т. п.). При необходимости использования апострофа его нужно записать два раза подряд, например, 'АВСДЕ' РОН' определяет литерал, состоящий из символов АВСДЕ' РОН. В основном литералы используются для задания текстовых параметров макросов, но кроме этого имеются и некоторые другие применения, например, в операторах ДЕТАЛЬ, СТАНОК и др.

2.3 Типы данных.

Возможности языка программирования в значительной степени определяются набором допустимых типов данных. Входной язык Техтран допускает применение следующих типов данных: вещественный, логический, вектор, прямая, точка, окружность, плоскость, матрица, контур. Тип каждой используемой в программе переменной фиксирован, и на протяжении всей программы каждая переменная обозначает данное только одного типа. Тип переменной описывается явно строкой вида:

тип имя₁, имя₂, имя₃, ...

где *тип* есть: ВЕЩ, ЛОГИЧ, ТОЧКА, ПРЯМАЯ, ВЕКТОР, ОКРУЖИ, ПЛОСК, МАТР, КОНТУР; *имя₁, имя₂, имя₃, ...* – имена переменных.

Переопределение типа переменной в программе не разрешено. Если в программе встречается переменная, не описанная ранее, то имя переменной, начинающееся с символов ТЧ, считается неявно описанным, как точка; имя, начинающееся с ПР, – как прямая, с КР, – как окружность. Все остальные имена переменных определяются неявным образом, как вещественные числа.

Можно совмещать описание типа переменной и оператор присваивания, например, оператор ТОЧКА Т1 = 0, 1, 2 равносителен двум операторам: ТОЧКА Т1; Т1 = 0, 1, 2.

2.4 Массивы

Данные одного типа могут объединяться в массивы. Массивы могут быть одномерными и двумерными. Каждый используемый в программе массив должен быть описан с указанием типа, числа индексов (размерности массива) и максимального значения каждого из индексов. Минимальным значением индекса является единица.

Оператор ВЕЩ А (16) описывает одномерный массив А, состоящий из 16 вещественных чисел. Оператор ВЕЩ В (5, 6) описывает двумерный массив, состоящий из 30 вещественных чисел. Для использования элементов массива в вычислениях следует указать имя массива, а за ним в круглых скобках – значение индексов. Индекс не может быть опущен. В частности, одно имя не представляет первый элемент массива.

В качестве индекса может использоваться любое арифметическое выражение. Число индексов и их значения должны соответствовать описанию массива. Число элементов в массиве не ограничено. Например, если значение переменной И = 2, то запись В (3, И + 3) именуется элемент массива В (3,5), а запись вида В (3) также, как и запись В (1,6 + И), вызывает ошибку трансляции.

В качестве элементов массива не могут использоваться объекты типов МАТР и КОНТУР.

2.5 Выражения

Арифметические выражения. Вместо чисел в программе могут использоваться арифметические выражения. Простейшими арифметическими выражениями могут быть число; вещественная переменная; элемент вещественного массива; обращение к стандартной функции; выражение, заключенное в скобки.

Более сложные арифметические выражения могут быть образованы из простейших при помощи следующих знаков арифметических операций: « + » – сложение; « - » – вычитание; « * » – умножение; « / » – деление; « ** » – возведение в степень.

Операции выполняются слева направо в соответствии с приведенными ниже приоритетами операций: вычисление функций – 1; возведение в степень – 2; умножение и деление – 3; сложение и вычитание – 4.

Приоритет используется для того, чтобы определить, какая из двух последовательных операций выполняется первой. Для изменения порядка вычислений следует применять круглые скобки.

Примечание. При возведении в степень следует помнить, что отрицательное число допускается возводить только в целую степень.

Например, выражение $\frac{6a^2 + 2b(c^3 - d^3)}{(a - b)\sin\varphi}$ записывается на входном языке Техтран так:

$$(6*A**2 + 2*B*(C**3 - D**3)) / ((A - B) * \text{СИН}(\Phi)).$$

Логические выражения. Язык Техтран допускает использование логических выражений в условных операторах (ЕСЛИ), операторах цикла (ПОВТОР), операторах логического присваивания. Значением логического выражения является ЛОЖЬ или ИСТИНА.

Простейшее логическое выражение состоит из логического первичного. Логическим первичным может быть: логическая константа ЛОЖЬ и ИСТИНА; логическая переменная; элемент логического массива; оператор корректности; отношение; логическое выражение, заключенное в скобки:

Отношения образуются путем объединения двух арифметических выражений с помощью операций отношения. Шесть операций отношения изображаются следующим образом:

БЛШ или > – больше (>)

БРВ или $> =$ – больше или равно ($\geq$)
 МНШ или $<$ – меньше ($<$)
 МРВ или $< =$ – меньше или равно ($\leq$)
 РВН или $=$ – равно ($=$)
 НРВ или $< >$ или $< >$ – не равно ($\neq$)

Например, выражение $(K+1)$ РВН5 принимает значение ИСТИНА при $K=4$, в остальных случаях принимает значение ложь.

Логические выражения образуются из логических первичных путем использования логических операций. В языке Техтран существуют три логические операции: НЕ – отрицание; ЛИ – логическое умножение; ИЛИ – логическое сложение.

Вычисление логического выражения производится слева направо в соответствии с приведенными ниже приоритетами операций: вычисление функций – 1; возведение в степень – 2; умножение и деление – 3; сложение и вычитание – 4; операции отношения БЛШ, БРВ, МНШ, МРВ, РВН, НРВ – 5; НЕ – 6; ЛИ – 7; ИЛИ – 8.

Для изменения порядка вычислений можно использовать круглые скобки. Логическое выражение, к которому относится операция НЕ, должно заключаться в скобки, если оно содержит не менее двух операндов.

2.6 Стандартные функции

Язык Техтран содержит ряд стандартных математических функций, часто используемых при вычислениях. Вызов математической функции производится указанием в арифметическом выражении имени функции и ее аргументов (параметров). Результат, вычисленный функцией, замещает в выражении обращение к ней. Стандартные функции в языке могут быть вещественного (табл. 2.1) и логического (табл. 2.2) типов.

Таблица 2.1

Имя	Функция	Аргумент	Результат
СИН	Вычисление синуса	<i>число</i>	Синус
КОС	» косинуса	»	Косинус
ТАН	» тангенса	»	Тангенс
АСИН	» арксинуса	»	Арксинус
АКОС	» арккосинуса	»	Арккосинус
АТАН	» арктангенса	»	Арктангенс
КВКОР	Извлечение квадратного корня	»	Квадратный корень числа
EXP	Вычисление экспоненты e^*	<i>число (x)</i>	e^*
ЛОГ	Вычисление натурального логарифма	<i>число</i>	Натуральный логарифм числа
АБС	Вычисление абсолютного значения величины	»	Абсолютное значение величины
МОДВ	Вычисление модуля вектора	<i>вектор</i>	Модуль вектора
ЦЕЛ	Выделение целой части переменной	<i>число или имя</i>	Целая часть числа или переменной с заданным именем
УГОЛВ	Вычисление угла между двумя векторами	<i>вектор₁, вектор₂</i>	Угол между двумя векторами
РАСТ	Вычисление расстояния между двумя точками	<i>точка₁, точка₂</i>	Расстояние между двумя точками
КООРД	Извлечение составляющей канонической формы	<i>число (a), имя</i>	Составляющая под номером «a» канонической формы геометрического объекта с заданным именем
СКАЛПР	Вычисление скалярного произведения	<i>вектор₁, вектор₂</i>	Скалярное произведение векторов
ДИСТ	Вычисление расстояния от точки до прямой	<i>точка, прямая</i>	Расстояние между точкой и прямой
УГОЛПР	Вычисление угла между прямыми	<i>прямая₁, прямая₂</i>	Угол между прямыми
ЛОГ10	Вычисление десятичного логарифма числа	<i>число (a)</i>	$\lg a$

Примечания: 1. Аргументы заключаются в круглые скобки и перечисляются через запятую. 2. Вместо чисел и имен переменных в качестве аргументов могут использоваться арифметические выражения и вложенные определения. 3. Угол между векторами или прямыми отсчитывается от элемента, заданного в качестве первого аргумента. 4. Значения стандартных функций могут использоваться в качестве оператора.

Таблица 2.2

Имя	Функция	Аргумент	Результат
KORP	Корректность	Вложенное геометрическое определение	ИСТИНА, если геометрический объект может быть построен; ЛОЖЬ – в противном случае
НЕКОРР	Некорректность	Вложенное геометрическое определение	ИСТИНА, если геометрический объект может быть построен; ЛОЖЬ – в противном случае

Пример использования логической функции KORP:

ЛОГИЧ А = KORP ТОЧКА ТЧ1 (ПЕРЕСЕЧ, ПР1, КР1, УБ)

ЕСЛИ (НЕ А) ТЧ1=ТЧ0.

Переменной ТЧ1 присваивается значение одной из точек пересечения окружности и прямой, если они пересекаются; в противном случае переменной ТЧ1 присваивается значение переменной ТЧ0.

2.7 Оператор присваивания

Для присваивания значения переменным используется оператор присваивания. Он состоит из левой части, знака равенства и правой части. В результате выполнения оператора присваивания значение правой части присваивается переменной, находящейся в левой части (тип левой части должен допускать такое присваивание). Если в левой части стоит переменная вещественного (логического) типа, то и в правой части должно стоять арифметическое (логическое) выражение. Например: оператор $X = X + 1$ – правильный оператор присваивания, в результате выполнения которого значение переменной X будет увеличено на 1.

2.8 Система координат

В системе Тектран все объекты и траектория движения рассматриваются и описываются в правосторонней прямоугольной системе координат (см. рис. 2.1). На рис. 2.2, а – в приведены примеры отсчета координат на плоскости (рис. 2.2, а), в пространстве (рис. 2.2, б) и угловых размеров (рис. 2.2, в). Определяемые прямые и окружности всегда находятся в горизонтальной плоскости xOy , поэтому описания геометрии и движения по контуру и поверхности детали не должны зависеть от системы координат конкретного станка. Рассчитанные процессором системы перемещения (координаты) приводятся в соответствие с фактическими направлениями и адресами координат станка при работе постпроцессора, формирующего управляющую программу.

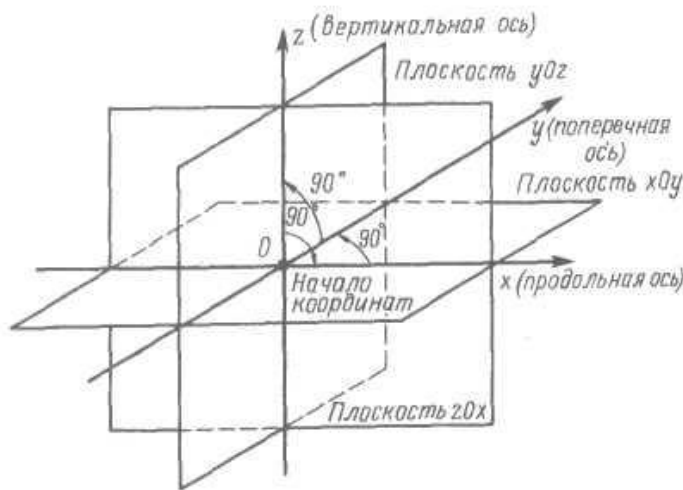


Рис. 2.1

На рис. 2.3, а – г приведены примеры перехода от системы координат детали к системе координат станка для токарных (рис. 2.3, а, б) и фрезерных (рис. 2.3, в, г) станков.

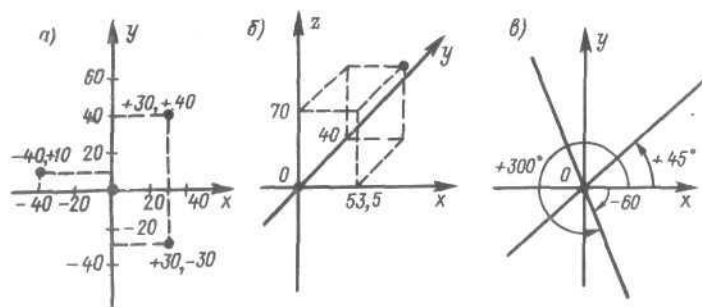


Рис. 2.2

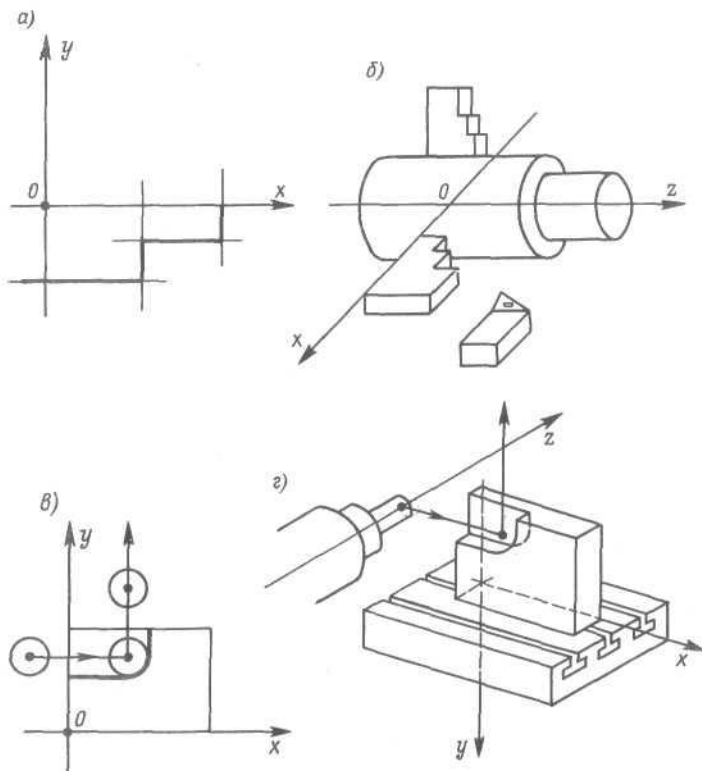


Рис. 2.3

2.9 Геометрические определения

Геометрическое определение является оператором присваивания, в левой части которого находится символическое имя определяемого геометрического объекта, а в правой – один из вариантов задания этого объекта в строгом соответствии с форматом, приведенным в данном разделе.

В общем виде геометрическое определение имеет следующий формат:

тип имя = список параметров

где *тип* – служебное слово, определяющее тип геометрического объекта; *имя* – символическое имя определяемого геометрического объекта.

Параметрами, определяющими геометрический объект, являются имена опорных элементов или вложенные геометрические определения, модификаторы, числа или заменяющие их арифметические выражения (опорным элементом называют ранее определенный геометрический объект).

Служебными словами, определяющими тип геометрического объекта, являются ТОЧКА, ПРЯМАЯ, ОКРУЖИ, ВЕКТОР, МАТР, ПЛОСК. Служебное слово, определяющее тип геометрического объекта, может отсутствовать, если тип объекта был указан ранее явно (отдельной строкой) или если используется символическое имя, определяющее тип геометрического объекта неявно.

2.9.1 Внутреннее представление геометрических объектов

Для внутреннего представления геометрических объектов в процессоре системы Техтран

используются канонические формы (табл. 2.3).

Таблица 2.3

Тип объекта	Внутреннее представление
Точка	x, y, z – координаты точки
Прямая	$\cos \alpha, \sin \alpha, p$ – коэффициенты нормального уравнения прямой ($x \cos \alpha + y \sin \alpha - p = 0$)
Окружность	x, y, r – координаты центра окружности и ее радиус
Вектор	$\Delta x, \Delta y, \Delta z$ – проекции вектора на координатные оси
Плоскость	$\cos \alpha, \cos \beta, \cos \gamma, p$ – коэффициенты нормального уравнения плоскости ($x \cos \alpha + y \cos \beta + z \cos \gamma - p = 0$)
Матрица	$a_{11}, a_{12}, \dots, a_{34}$ – коэффициенты нормальных уравнений координатных плоскостей системы координат, полученной при преобразовании исходной системы координат в соответствии с заданной матрицей

Внутреннее представление геометрических объектов применяют при выводе на печать результатов работы процессора в случае использования отладочных операторов СЛЕД и ПЧТ1 (см. п. 2.12.2). Для доступа к элементам канонической формы точек, прямых, окружностей и векторов следует использовать функцию КООРД. Например, требуется определить прямую, проходящую через центры двух окружностей, координаты которых неизвестны (рис. 2.4). Найдем эти координаты следующим образом:

$X1 = \text{КООРД}(1, \text{КР}1)$

$Y1 = \text{КООРД}(2, \text{КР}1)$

$X2 = \text{КООРД}(1, \text{КР}2)$

$Y2 = \text{КООРД}(2, \text{КР}2)$

Искомая прямая имеет вид

$\text{ПРА} = X1, Y1, X2, Y2.$

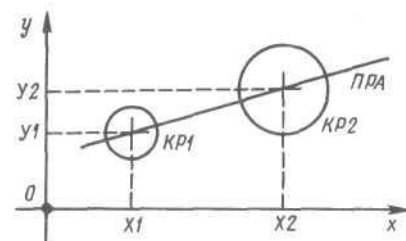


Рис. 2.4

2.9.2 Определение точки

В определениях точки используют следующие группы модификаторов:

Модификаторы выбора точки из двух возможных: ХБ – выбирается точка с большей координатой x , УБ – с большей координатой y , ХМ – с меньшей координатой x и УМ – с меньшей координатой y .

Модификаторы направления отсчета угла: ПОЧС – задается отсчет угла по часовой стрелке; ПРЧС – задается отсчет угла против часовой стрелки.

Модификаторы, указывающие координату точки, заданную в геометрическом определении: ХКООРД – координата x ; УКООРД – координата y .

Точка, определенная прямоугольными координатами:

имя = координата x , координата y , координата z

имя. = координата x , координата y

Например (рис. 2.5):

$\text{ТЧА} = -5, 15, 10.7$

$\text{ТОЧКА ТЗ} = 24, -8.2$

Точка, определенная пересечением двух прямых:

имя = [ПЕРЕСЕЧ,] прямая₁, прямая₂

Например (рис. 2.6):

$\text{ТЧ2} = \text{ПР1}, \text{ПР2}$

Точка, определенная пересечением прямой и окружности:

имя = [ПЕРЕСЕЧ,] прямая, окружность, (ХБ|УБ|ХМ|УМ)

Например (рис. 2.7):

$\text{ТОЧКА Т23} = [\text{ПЕРЕСЕЧ,}] \text{ПР7}, \text{КР2}, \text{УБ}$

Примечание. Случай касания прямой и окружности рассматривается как пересечение этих двух объектов.

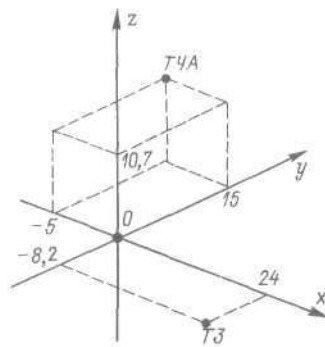


Рис. 2.5

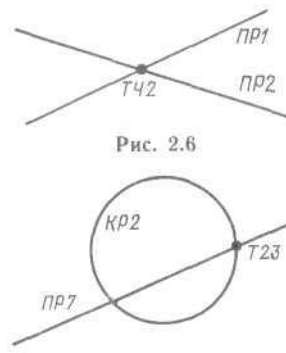


Рис. 2.6

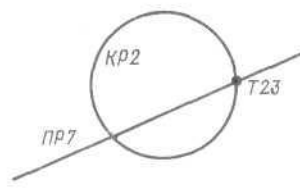


Рис. 2.7

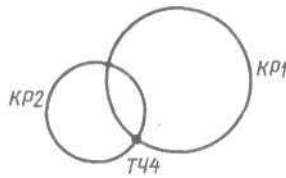


Рис. 2.8

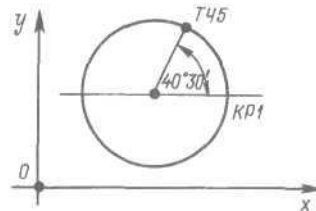


Рис. 2.9

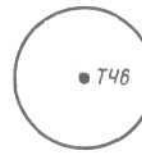


Рис. 2.10

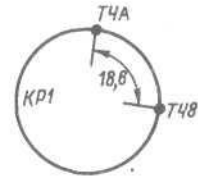


Рис. 2.11

Точка, определенная пересечением двух окружностей:

имя = [ПЕРЕСЕЧ,] окружность₁, окружность₂ {ХБ|УБ|ХМ|УМ}

Например (рис. 2.8):

ТЧ4 = КР1, КР2, УМ

Точка, определенная пересечением окружности и ее радиуса, проведенного под заданным углом к оси x:

имя = окружность, $\left\{ \frac{\ddot{I}\ddot{D} \times \tilde{N}}{\ddot{I}\ddot{I} \times \tilde{N}} \right\}$, угол

Например (рис. 2.9):

ТЧ5 = КР1, ПРЧС, 40.5

Точка, являющаяся центром ранее определенной окружности:

имя = [ЦЕНТР], окружность

Например (рис. 2.10):

ТЧ6 = КР17

Точка, находящаяся на окружности на известном расстоянии от другой точки этой же окружности:

имя = окружность, точка $\left\{ \frac{\ddot{I}\ddot{I} \times \tilde{N}}{\ddot{I}\ddot{D} \times \tilde{N}} \right\}$, расстояние

Например (рис. 2.11):

ТЧ8 = КР1, ТЧ4, ПОЧС, 18.6

Примечание. Расстояние линейное, измеряется по дуге окружности.

Точка, находящаяся на прямой и имеющая заданную координату x или y:

имя = прямая, {ХКООРД|УКООРД}, число

Например (рис. 2.12):

ТЧ9 = ПР1, ХКООРД, 80

Точка, находящаяся на прямой, на определенном расстоянии от другой точки этой же прямой:

имя = прямая, точка, расстояние, {ХБ|УБ|ХМ|УМ}

Например (рис. 2.13):

ТЧ10 = ПР8, ТЧ4, 23, ХМ

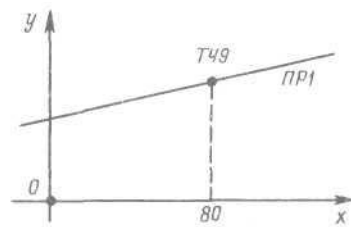


Рис. 2.12

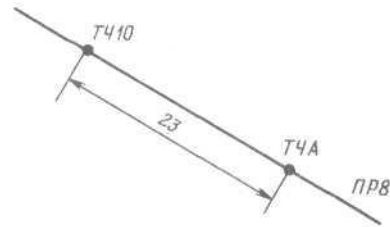


Рис. 2.13

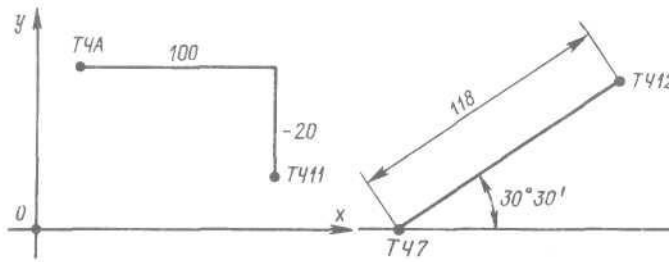


Рис. 2.14

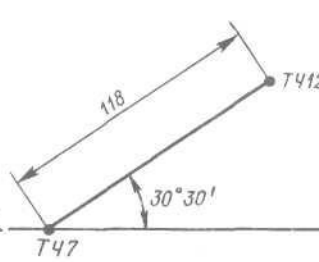


Рис. 2.15

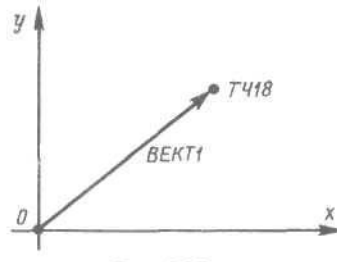


Рис. 2.16

Точка, заданная приращениями координат известной точки:

имя = точка, ПЕРЕНОС, приращение x, приращение y

Например (рис. 2.14):

TЧ11=TЧА, ПЕРЕНОС, 100, -20

Точка, находящаяся на определенном расстоянии от известной точки под углом к оси x:

имя = точка, $\left\{ \frac{\ddot{I}\ddot{I} \times \ddot{N}}{\ddot{I}\ddot{D} \times \ddot{N}} \right\}$, угол, расстояние

Например (рис. 2.15):

TЧ12 = TЧ7, ПОЧС, 30.5, 118

Точка, координаты которой получены преобразованием координат известной точки в соответствии с указанной матрицей:

имя = точка, матрица

Например:

TЧ16 = TЧА, МК6

Точка, определенная ранее заданной точкой:

имя = точка

Например:

TЧ17 = TЧ13A

Точка, для которой ранее определенный вектор является радиус-вектором:

имя = вектор

Например (рис. 2.16):

TЧ18 = ВЕКТ1

2.9.3 Задание координаты z в геометрических определениях точек (оператор ПЛЗ)

Оператор ПЛЗ предназначен для задания координаты z тех точек, в геометрических определениях которых нет явно заданной третьей координаты:

Формат:

ПЛЗ- $\left\{ \begin{array}{c} \div \ddot{e} \ddot{n} \ddot{e} \ddot{i} \\ \ddot{e} \ddot{i} \ddot{y} \\ \ddot{i} \ddot{O} \ddot{A} \ddot{f} \end{array} \right\}$

где *имя* — имя плоскости, определенной ранее; ОТМЕН — служебное слово, обозначающее отмену действия оператора ПЛЗ.

Действие оператора состоит в следующем. Если в операторах определения точки координата z не задана явно, то процессор вычислит ее исходя из того условия, что определяемая точка должна лежать в плоскости, заданной оператором ПЛЗ. Если оператор ПЛЗ не был задан или был выполнен оператор ПЛЗ ОТМЕН, то все координаты z точек, не заданные явно, принимают нулевое значение. Действие оператора ПЛЗ распространяется на все точки, определенные после него, до использования другого оператора ПЛЗ или оператора ПЛЗ ОТМЕН.

На рис. 2.17 изображены точки, определенные в приведенном ниже фрагменте программы:

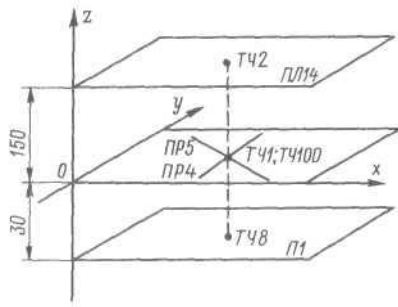


Рис. 2.17

```

ТЧ1=ПР5, ПР4
ПЛОСК П1, ПЛ14
ПЛ14=150
П1 = -30
ПЛЗ ПЛ14
ТЧ2 = ПР5, ПР4
ПЛЗ П1
ТЧ8 = ПР5, ПР4
ПЛЗ ОТМЕН
ТЧ100 = ПР5, ПР4

```

Координаты z точек ТЧ2 и ТЧ8 равны соответственно 150 и -30 , так как они заданы оператором ПЛЗ с указанием плоскости, в которой лежит каждая из этих точек. Координата z точки ТЧ1 по умолчанию равна нулю; координата точки ТЧ100 также равна нулю, так как перед ее геометрическим определением задан оператор ПЛЗ ОТМЕН, отменяющий действие последнего оператора ПЛЗ.

2.9.4 Определение прямой

В определениях прямой используют следующие группы модификаторов:

1.Модификаторы, выбирающие одну прямую из двух, касающихся окружности: ХМ – выбирается прямая, точка касания которой имеет меньшую координату по x ; ХБ – большую координату по x ; УМ – меньшую координату по y и УБ – большую координату по y .

2.Модификаторы, указывающие, с какой стороны определяемая прямая касается окружности: СПРАВА – точка касания прямой находится справа от центра окружности, если смотреть в направлении от первого геометрического объекта правой части определения ко второму; СЛЕВА – прямая касается окружности слева (при тех же условиях рассмотрения геометрических объектов).

3.Модификаторы, указывающие положение прямой относительно другой: ПАРЛЕЛ – параллельно; ПЕРП – перпендикулярно.

Прямая, проходящая через две точки, заданные своими координатами:

имя = координата x_1 , координата y_1 , координата x_2 , координата y_2

Например (рис. 2.18):

ПР20=-15, 10, 35.6, 70

Прямая, проходящая через две точки:

имя = точка₁, точка₂.

Например (рис. 2.19):

ПР21=ТЧА, ТЧБ

Прямая, проходящая через точку и касающаяся окружности:

имя = точка, { $\tilde{\text{НАЧАЛО}}$ }, [КАСАТ], окружность

Например (рис. 2.20):

ПР22 = ТЧА, СПРАВА, КРЗ

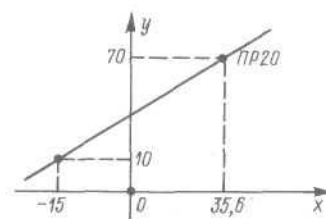


Рис. 2.18

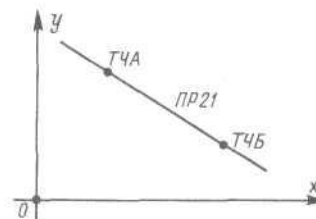


Рис. 2.19

Прямая, касающаяся двух окружностей:

имя = { $\tilde{\text{НАЧАЛО}}$ }, окружность₁, { $\tilde{\text{НАЧАЛО}}$ }, окружность₂

Например (рис. 2.21):

ПР23 = СЛЕВА, КР1, СПРАВА, КР3,

Прямая, проходящая через точку под определенным углом к оси x:

имя = точка, угол

Например (рис. 2.22):

ПР23 = ТЧ5, – 15

Прямая, проходящая через точку под определенным углом к заданной прямой:

имя = прямая, точка, $\left\{ \frac{\vec{I}\vec{I} \times \vec{N}}{\vec{I}\vec{D} \times \vec{N}} \right\}$, угол

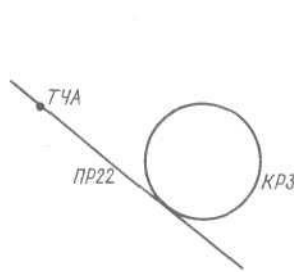


Рис. 2.20

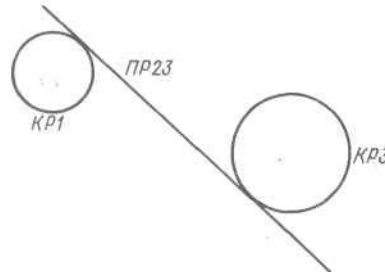


Рис. 2.21

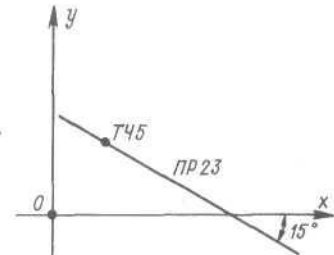


Рис. 2.22

Например (рис. 2.23):

ПР25 = ПРА, ТЧВ, 44.5

Прямая, проходящая через точку и перпендикулярная к другой прямой:

имя = точка, ПЕРП, прямая

Например (рис. 2.24):

ПР26 = ТЧ6, ПЕРП, ПР1

Прямая, проходящая через точку и параллельная другой прямой:

имя = точка, ПАРЛЕЛ, прямая

Например (рис. 2.25):

ПР27 = ТЧ1, ПАРЛЕЛ, ПР14

Прямая, проходящая параллельно другой прямой на заданном расстоянии от нее:

имя = прямая, расстояние, |ХБ|УБ|ХМ|УМ)

Например (рис. 2.26):

ПР28 = ПРА, 43,6, УМ

Прямая, касающаяся окружности и составляющая с осью x заданный угол:

имя = [КАС], окружность, {ХБ|УБ|ХМ|УМ}, угол

Например (рис. 2.27):

ПР290 = КАС, КР1, УМ, – 15

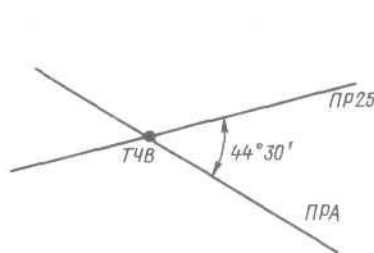


Рис. 2.23

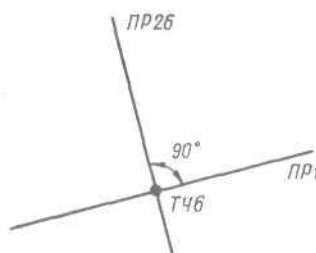


Рис. 2.24

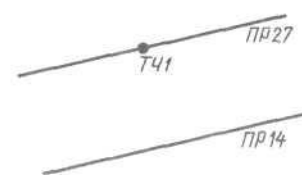


Рис. 2.25

Прямая, проходящая через точку параллельно вектору:

имя = точка, вектор

Например (рис. 2.28):

ПР310 = ТЧ1, ВЕК2

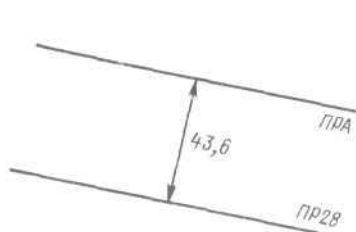


Рис. 2.26

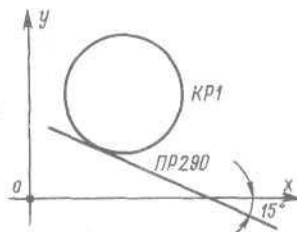


Рис. 2.27

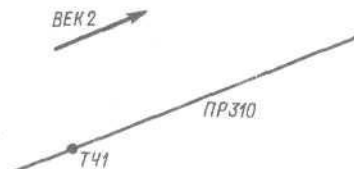


Рис. 2.28

Прямая, полученная преобразованием заданной прямой в соответствии с матрицей:

имя = прямая, матрица

Например:

ПР32 = ПР10, МАТР1

Прямая, совпадающая с прямой, определенной ранее:

имя = прямая

Например:

ПР13 = ПР376

Прямая, параллельная одной из осей координат и находящаяся на заданном расстоянии от нее:

имя = $\left\{ \begin{matrix} \ddot{O}\ddot{I}\ddot{A}\ddot{D} \\ \ddot{O}\ddot{I}\ddot{A}\ddot{D} \end{matrix} \right\}$, расстояние

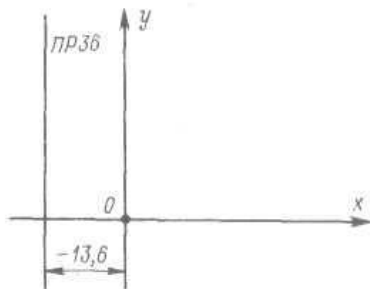


Рис. 2.29

Например (рис. 2.29):

ПР36 = УПАР, -13,6

2.9.5 Определение окружности

В определениях окружности используют следующие группы модификаторов:

1.Модификаторы БОЛШ и МЕНШ, служащие для выбора из двух окружностей одной – с большим или меньшим радиусом соответственно.

2.Модификаторы, указывающие положение определяемой окружности относительно другой, являющейся опорным элементом: ВНЕ – окружность находится вне заданной окружности; ВНУТРИ – окружность находится внутри заданной окружности.

3.Модификаторы, указывающие положение окружности относительно прямой: ХМ – центр окружности находится слева от прямой, ХБ – справа от прямой, УБ – выше прямой и УМ – ниже прямой.

4.Те же модификаторы ХМ, ХБ, УБ, УМ, но служащие для выбора из двух окружностей одной по соответствующей координате ее центра.

Модификаторы 2 и 3-й групп относятся к следующим непосредственно за ними опорным элементам геометрического определения. Применение модификаторов ВНУТРИ и ВНЕ имеет свои особенности, отмеченные ниже.

Окружность, совпадающая с окружностью, определенной ранее:

имя = окружность

Например:

КР4 = КР100

Окружность, определенная координатами центра и радиусом:

имя = координата x, координата y, радиус

Например (рис. 2.30):

КР4 = - 13,6, 20, 5

Окружность, определенная радиусом и точкой, являющейся ее центром:

имя = точка, радиус

Например (рис. 2.31):

КР4 = ТЧ3, 15

Окружность, определенная точкой, являющейся ее центром, и касающаяся заданной прямой:

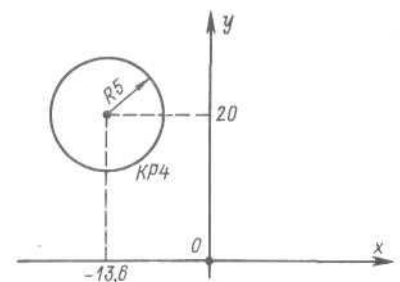


Рис. 2.30

имя = точка, [КАС], прямая

Например (рис. 2.32):

КР34 = ТЧ6, КАС, ПР8

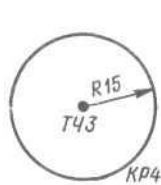


Рис. 2.31

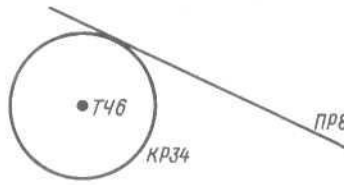


Рис. 2.32

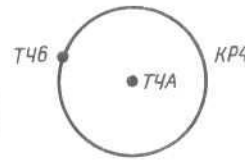


Рис. 2.33

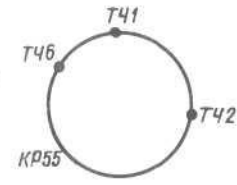


Рис. 2.34

Окружность, определенная точкой, являющейся ее центром, и проходящая через другую заданную точку:

имя = точка₁, точка₂

Например (рис. 2.33):

КР4 = ТЧ4, ТЧ6

Окружность, проходящая через три точки:

имя = точка₁, точка₂, точка₃

Например (рис. 2.34):

КР55 = ТЧ1, ТЧ6, ТЧ2

Окружность, определенная точкой, являющейся ее центром, и касающаяся другой окружности:

имя = точка, $\left\{ \frac{\overset{\circ}{A}\overset{\circ}{I}\overset{\circ}{O}}{\overset{\circ}{A}\overset{\circ}{I}\overset{\circ}{E}\overset{\circ}{O}} \right\}$, окружность

Например (рис. 2.35):

КР6 = ТЧ1, МЕНШ, КРА1

КР6А = ТЧ1, БОЛШ, КРА1

Окружность заданного радиуса, касающаяся двух пересекающихся прямых:

имя = $\left\{ \frac{\overset{\circ}{O}\overset{\circ}{A}}{\overset{\circ}{O}\overset{\circ}{I}} \right\}$, прямая₁, $\left\{ \frac{\overset{\circ}{O}\overset{\circ}{A}}{\overset{\circ}{O}\overset{\circ}{I}} \right\}$, прямая₂, радиус

Например (рис. 2.36):

КР47 = УБ, ПР1, ХБ, ПР2, 18

КР7 = УМ, ПР1, ХБ, ПР2, 18

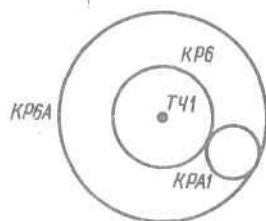


Рис. 2.35

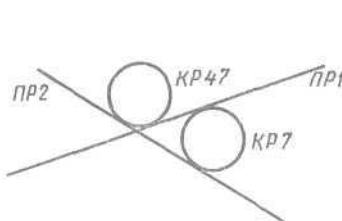


Рис. 2.36

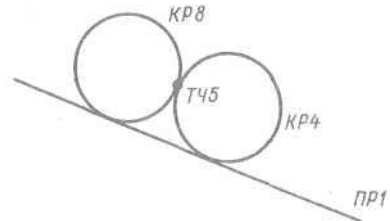


Рис. 2.37

Окружность заданного радиуса, касающаяся прямой и проходящая через заданную точку:

имя = прямая, $\left\{ \frac{\overset{\circ}{O}\overset{\circ}{A}}{\overset{\circ}{O}\overset{\circ}{I}} \right\}$, точка, радиус

Например (рис. 2.37):

КР8 = ПР1, УБ, ТЧ5, 20

КР4 = ПР1, УМ, ТЧ5, 20

Окружность заданного радиуса, касающаяся прямой и окружности:

$$\text{имя} = \left\{ \frac{\tilde{O}\tilde{A}}{\tilde{O}\tilde{I}}, \frac{\tilde{O}\tilde{A}}{\tilde{O}\tilde{I}} \right\}, \text{ прямая}, \left\{ \frac{\tilde{O}\tilde{A}}{\tilde{O}\tilde{I}}, \frac{\tilde{O}\tilde{A}}{\tilde{O}\tilde{I}} \right\}, \left\{ \frac{\hat{A}\hat{I}\hat{A}}{\hat{A}\hat{I}\hat{O}\hat{O}\hat{D}\hat{E}} \right\}, \text{ окружность, радиус}$$

Например (рис. 2.38):

КР1=УБ, ПР1, ХМ, ВНЕ, КРО, 10

КР2 = УБ, ПР1, ХМ, ВНУТРИ, КРО, 10

КР3 = УБ, ПР1, ХБ, ВНУТРИ, КРО, 10

КР4 = УБ, ПР1, ХБ, ВНЕ, КРО, 10

КР5 = УМ, ПР1, ХМ, ВНЕ, КРО, 10

КР6 = УМ, ПР1, ХМ, ВНУТРИ, КРО, 10

КР7 = УМ, ПР1, ХБ, ВНУТРИ, КРО, 10

КР8 = УМ, ПР1, ХБ, ВНЕ, КРО, 10

Окружность заданного радиуса, касающаяся двух других окружностей:

$$\text{имя} = \left\{ \frac{\tilde{O}\tilde{A}}{\tilde{O}\tilde{I}}, \frac{\tilde{O}\tilde{A}}{\tilde{O}\tilde{I}} \right\}, \left\{ \frac{\hat{A}\hat{I}\hat{A}}{\hat{A}\hat{I}\hat{O}\hat{O}\hat{D}\hat{E}} \right\}, \text{ окружность}_1, \left\{ \frac{\hat{A}\hat{I}\hat{A}}{\hat{A}\hat{I}\hat{O}\hat{O}\hat{D}\hat{E}} \right\}, \text{ окружность}_2, \text{ радиус}$$

Например (рис. 2.39):

КР1=УБ, ВНЕ, КРА, ВНЕ, КРБ, 15

КР2 = УБ, ВНЕ, КРА, ВНУТРИ, КРБ, 15

КР3 = УМ, ВНЕ, КРА, ВНУТРИ, КРБ, 15

КР4 = УМ, ВНЕ, КРА, ВНЕ, КРБ, 15

КР5 = УМ, ВНУТРИ, КРА, ВНЕ, КРБ, 15

КР6 = УБ, ВНУТРИ, КРА, ВНЕ, КРБ, 15

КР7 = УБ, ВНУТРИ, КРА, ВНУТРИ, КРБ, 15

КР8 = УМ, ВНУТРИ, КРА, ВНУТРИ, КРБ, 15

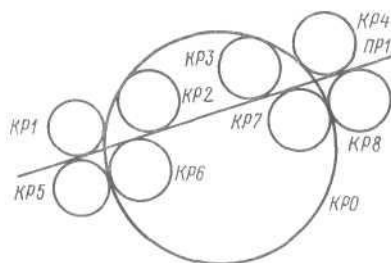


Рис. 2.38

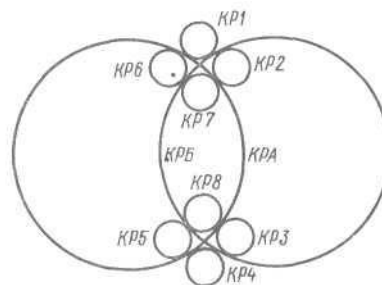


Рис. 2.39

Примечание. В тех случаях, когда определяемая окружность не может быть построена внутри опорных (слишком велик ее диаметр или опорные окружности не пересекаются), модификатор ВНУТРИ указывает положение опорных окружностей относительно определяемой окружности.

Например (рис. 2.40, а, б):

КР9 = УМ, ВНУТРИ, КР31, ВНУТРИ, КР40, 150

КР10 = УБ, ВНУТРИ, КР50, ВНЕ, КР17, 60

КР11=УМ, ВНУТРИ, КР50, ВНУТРИ, КР17, 60

Окружность, касающаяся трех прямых:

$$\text{имя} = \left\{ \frac{\tilde{O}\tilde{A}}{\tilde{O}\tilde{I}}, \frac{\tilde{O}\tilde{A}}{\tilde{O}\tilde{I}} \right\}, \text{ прямая}_1, \left\{ \frac{\tilde{O}\tilde{A}}{\tilde{O}\tilde{I}}, \frac{\tilde{O}\tilde{A}}{\tilde{O}\tilde{I}} \right\}, \text{ прямая}_2, \left\{ \frac{\tilde{O}\tilde{A}}{\tilde{O}\tilde{I}}, \frac{\tilde{O}\tilde{A}}{\tilde{O}\tilde{I}} \right\}, \text{ прямая}_3$$

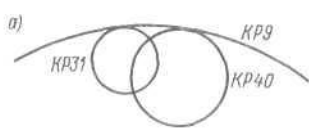


Рис. 2.40

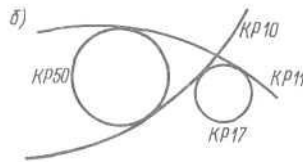
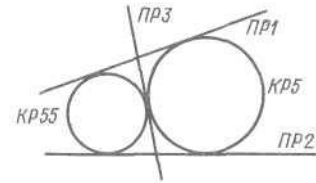


Рис. 2.41



Например (рис. 2.41):

KP5 = УМ, ПР1, УБ, ПР2, ХБ, ПР3

KP55 = УМ, ПР1, УБ, ПР2, ХМ, ПР3

Окружность заданного радиуса, проходящая через две точки:

$$\text{имя} = \left\{ \frac{\tilde{O}\hat{A}}{\tilde{O}\hat{I}}, \frac{\tilde{O}\hat{A}}{\tilde{O}\hat{I}} \right\}, \text{точка}_1, \text{точка}_2, \text{радиус}$$

Например (рис. 2.42):

KP1 = ХБ, ТЧ1, ТЧ2, 50

KP2 = ХМ, ТЧ1, ТЧ2, 30

Окружность заданного радиуса, проходящая через точку и касающаяся другой окружности:

$$\text{имя} = \text{точка}, \left\{ \frac{\tilde{O}\hat{A}}{\tilde{O}\hat{I}}, \frac{\tilde{O}\hat{A}}{\tilde{O}\hat{I}} \right\}, \left\{ \frac{\hat{A}\hat{I}\hat{A}}{\hat{A}\hat{I}\hat{O}\hat{O}\hat{E}\hat{E}} \right\}, \text{окружность}, \text{радиус}$$

Например (рис. 2.43):

KP3 = ТЧ1, УБ, ВНУТРИ, КРА, 25

KP5 = ТЧ1, УМ, ВНУТРИ, КРА, 25

KP6 = ТЧ9, УМ, ВНУТРИ, КРБ, 14

KP9 = ТЧ9, УМ, ВНЕ, КРБ, 14

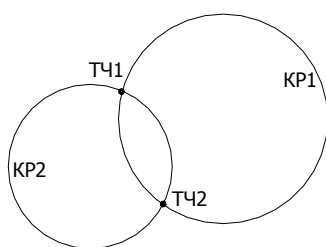


Рис. 2.42

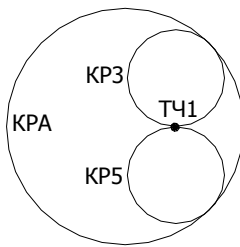


Рис. 2.43

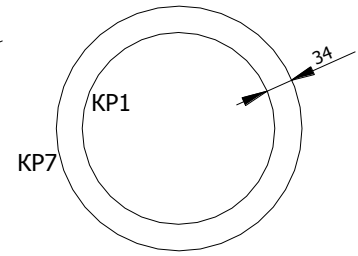
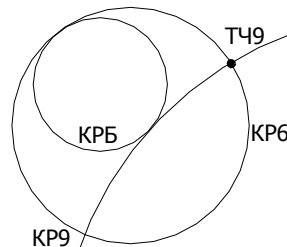


Рис. 2.44

Примечание. В том случае, когда точка, указанная в определении, находится внутри опорной окружности, использование модификатора ВНЕ не имеет смысла. Если опорная точка находится вне опорной окружности, возможно как внутреннее, так и внешнее касание. Модификатор ВНУТРИ в этом случае указывает положение опорной окружности относительно определяемой (окружность КР6 в приведенном примере).

Окружность, концентричная заданной окружности:

$$\text{имя} = \text{ПАРЛЕЛ}, \left\{ \frac{\hat{I}\hat{A}\hat{I}\hat{O}}{\hat{A}\hat{I}\hat{E}\hat{O}} \right\}, \text{окружность}, \text{разность радиусов}$$

Например (рис. 2.44):

KP7 = ПАРЛЕЛ, КР1, БОЛШ, 34

Окружность, полученная преобразованием заданной окружности в соответствии с матрицей:

имя = окружность, матрица

Например:

KP7 = КР2, МАТР6

2.9.6 Определение вектора

В определениях вектора используют модификаторы, указывающие направление вектора: ХБ

– вектор направлен в сторону возрастания координаты x ; ХМ – в сторону уменьшения координаты x ; УБ – в сторону возрастания координаты y и УМ – вектор направлен в сторону уменьшения координаты y .

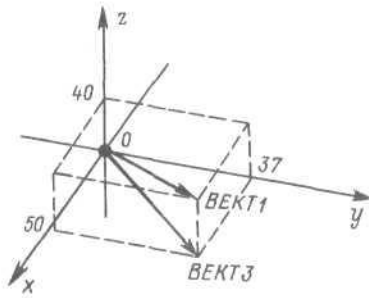


Рис. 2.45

Вектор, определенный проекциями на оси, начало которого находится в начале координат:

имя = проекция x , проекция y , [проекция z]

Например (рис. 2.45):

ВЕКТ1=50, 37, 40

ВЕКТ3 = 50, 37

Вектор, определенный своими начальной и конечной точками:

имя = точка₁, точка₂

Например (рис. 2.46):

ВКТР = ТЧ1, ТЧ2

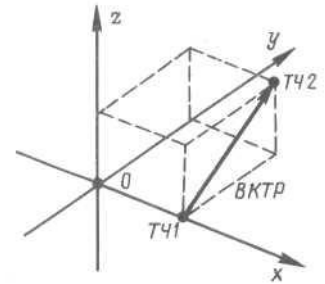


Рис. 2.46

Вектор, равный сумме двух других векторов:

имя = вектор₁, ПЛЮС, вектор₂

Например (рис. 2.47):

ВЕКТ4 = ВЕКР1, ПЛЮС, ВКТ2

Вектор, равный разности двух векторов:

имя = вектор₁, МИНУС, вектор₂

Например (рис. 2.48):

ВЕКТ5 = ВЕКР1, МИНУС, ВКТ2

Вектор, равный произведению заданного вектора и числа:

имя = число, вектор

Например (рис. 2.49):

ВЕКТ7 = 3, ВЕ2

Вектор, равный произведению двух векторов:

имя = вектор₁, УМНОЖ, вектор₂

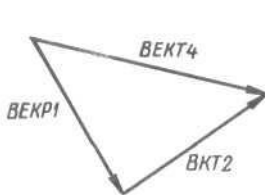


Рис. 2.47

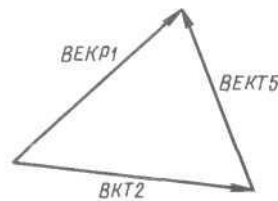


Рис. 2.48

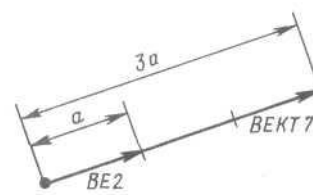


Рис. 2.49

Например:

ВЕКТ11=ВА36, УМНОЖ, ВР7

Вектор, равный ранее определенному вектору:

имя = вектор

Например:

ВЕР4 = ВКР6

Вектор, полученный преобразованием ранее определенного вектора в соответствии с заданной матрицей:

имя = вектор, матрица

Например:

ВЕКТОР В6 = ВЕКТ, М2

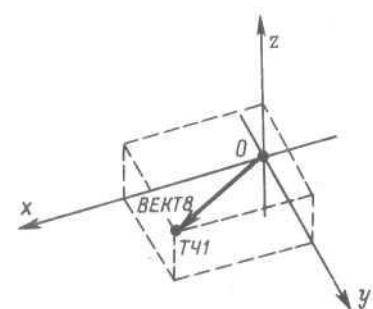


Рис. 2.50

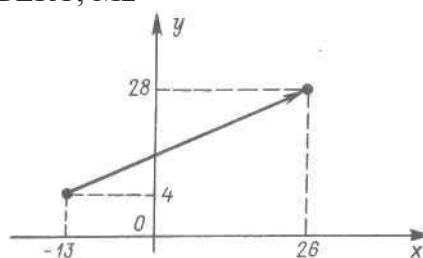


Рис. 2.51

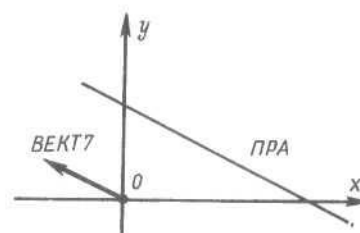


Рис. 2.52

Вектор, являющийся радиус-вектором заданной точки:

имя = точка

Например (рис. 2.50):

ВЕКТ8 = ТЧ1

Вектор, заданный координатами своих начальной и конечной точек:

имя = координата x_1 , координата y_1 , координата x_2 , координата y_2

Например (рис. 2.51):

ВЕКТ В1 = -13, 4, 26, 28

Вектор, параллельный заданной прямой:

имя = прямая, {ХБ|УБ|ХМ|УМ}

Примечание. Длина вектора равна единице, начало его находится в начале координат, например (рис. 2.52):

ВЕКТ7 = ПРА, ХМ

2.9.7 Определение матрицы

Матрица определяет способ преобразования координат. Это преобразование может быть параллельным переносом исходной системы координат, ее поворотом, симметрией или комбинацией этих преобразований. Любое преобразование координатной системы может быть описано с помощью системы трех уравнений:

$$a_{11}x_1 + a_{12}y_1 + a_{13}z_1 + a_{14} = x_2$$

$$a_{21}x_1 + a_{22}y_1 + a_{23}z_1 + a_{24} = y_2$$

$$a_{31}x_1 + a_{32}y_1 + a_{33}z_1 + a_{34} = z_2$$

Всегда можно определить значения таких коэффициентов $a_{11}, a_{12}, \dots, a_{34}$, что при подстановке в систему уравнений значений координат произвольной точки x_1, y_1, z_1 в локальной системе координат будут получены координаты этой точки x_2, y_2, z_2 в абсолютной (исходной) системе координат.

Коэффициенты данной системы уравнений используются для представления матрицы преобразования в канонической форме и выводятся на печать в следующей последовательности:

$$a_{11}a_{12}a_{13}$$

$$a_{21}a_{22}a_{23}$$

$$a_{31}a_{32}a_{33}$$

$$a_{14}a_{24}a_{34}$$

Первые три элемента каждого столбца определяют направление осей локальной системы координат по отношению к исходной (абсолютной) системе координат: a_{11}, a_{21}, a_{31} – единичный вектор оси x_1 ; a_{12}, a_{22}, a_{32} – единичный вектор оси y_1 ; a_{13}, a_{23}, a_{33} – единичный вектор оси z_1 .

Четвертая строка определяет положение начала координат локальной системы: Ом, «24, ОЗ4 – координаты x, y, z начала локальной системы координат в исходной (абсолютной) системе.

Результатом матричного преобразования является геометрический объект, который имеет в локальной системе координат, полученной преобразованием исходной системы, параметры исходного геометрического объекта.

При использовании комбинированных матричных преобразований, задаваемых произведением матриц, следует помнить, что матричное умножение не коммутативно, т. е. конечный результат зависит от последовательности элементарных преобразований.

На рис. 2.53, а–в представлено последовательное преобразование исходной системы координат (о) матрицами поворота (б) и переноса (в), а на рис. 2.53, г–е дано преобразование системы матрицами переноса (д) и поворота (е). В геометрических определениях матрицы используются следующие модификаторы: ПОВХУ – определяет поворот осей x и y вокруг оси z ; ПОВУЗ – поворот осей y и z вокруг оси x и ПОВЗХ – поворот осей z и x вокруг оси y (направление отсчета углов поворота – в соответствии с рисунком).

Матрица, определяющая координатную систему, полученную поворотом на заданный угол исходной системы координат вокруг одной из ее осей:

имя = {ПОВХУ|ПОВУЗ|ПОВЗХ}, угол

Например (рис. 2.54):

МАТР МТ2 = ПОВЗХ, 30.5

Матрица, определяющая координатную систему, полученную параллельным

переносом исходной системы координат:

имя = ПЕРЕНОС, координата x, координата y, [координата z]

Например (рис. 2.55):

МАТРЗ = ПЕРЕНОС, 10, 15, 20

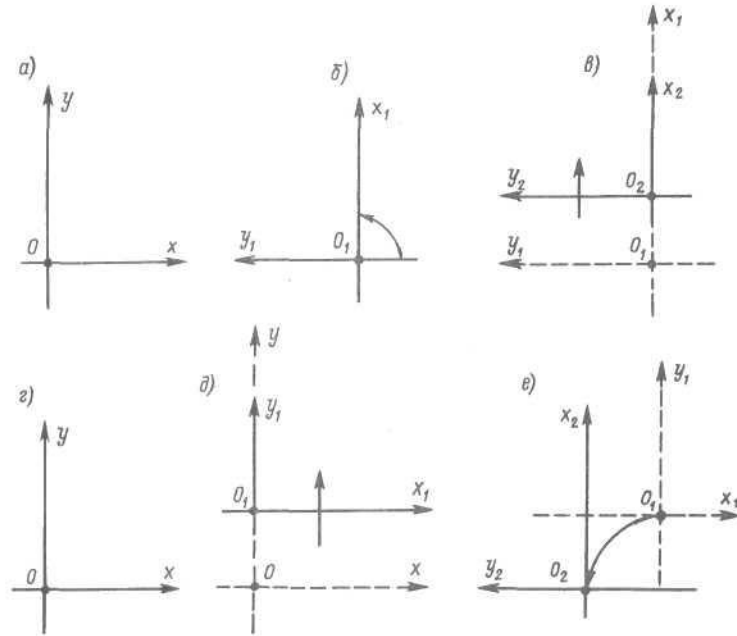


Рис. 2.53

Матрица, определяющая координатную систему, полученную параллельным переносом исходной системы координат и последующим поворотом на заданный угол вокруг одной из координатных осей:

$$eiy = \begin{Bmatrix} \tilde{I}\tilde{A}\tilde{O}\tilde{O} \\ \tilde{I}\tilde{A}\tilde{O}\tilde{C} \\ \tilde{I}\tilde{A}\tilde{C}\tilde{O} \end{Bmatrix}, \quad \begin{matrix} \acute{o}\acute{a}\tilde{i}\tilde{e}, & \tilde{I}\tilde{A}\tilde{B}\tilde{A}\tilde{I}\tilde{I}\tilde{N} & , & \tilde{e}\tilde{I}\tilde{d}\tilde{a}\tilde{e}\tilde{i}\tilde{a}\tilde{d}\tilde{a} & \tilde{o}, & \tilde{e}\tilde{I}\tilde{d}\tilde{a}\tilde{e}\tilde{i}\tilde{a}\tilde{d}\tilde{a} & \acute{o}, & \tilde{e}\tilde{I}\tilde{d}\tilde{a}\tilde{e}\tilde{i}\tilde{a}\tilde{d}\tilde{a} & z \end{matrix}$$

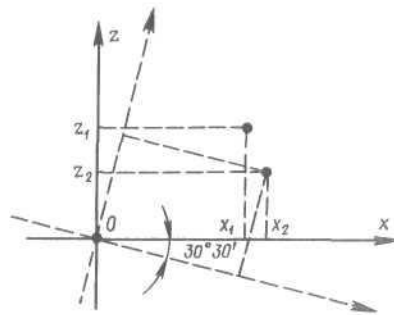


Рис. 2.54

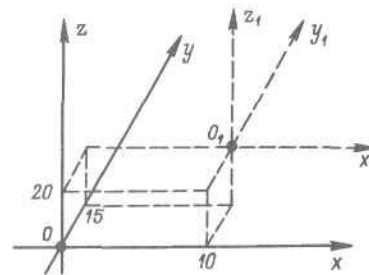


Рис. 2.55

Например (рис. 2.56):

МАТР = ПОВЗХ, 30.5, ПЕРЕНОС, 10, 15, 20

Матрица, являющаяся произведением двух других матриц:

имя = матрица x1, матрица2

Например (рис. 2.56):

МАТР=МАТРЗ, МТ2

Примечания: 1. Произведение матриц эквивалентно применению сначала матрицы₂, а затем преобразованию полученного результата в соответствии с матрицей₁. 2. Матрица, заданная в примере (см. рис. 2.56), является произведением матриц, определенных в первых двух примерах (см. рис. 2.54 и 2.55).

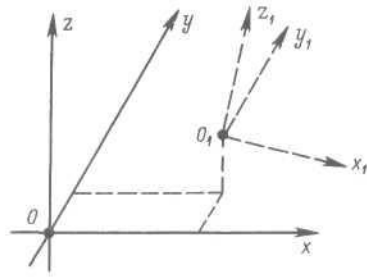


Рис. 2.56

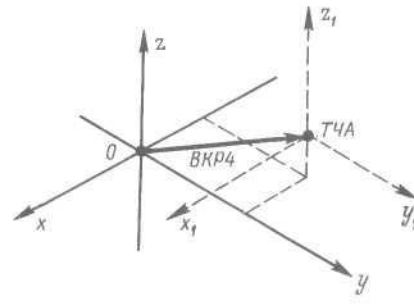


Рис. 2.57

Матрица, равная ранее определенной матрице:

имя = матрица

Например: МАТР4 = МТ6

Матрица, определяющая перенос начала координат в точку, заданную явно или вектором:

имя = ПЕРЕНОС, {точка/вектор}

Например (рис. 2.57):

МКТ = ПЕРЕНОС, ТЧА

МКТ = ПЕРЕНОС, ВКР4

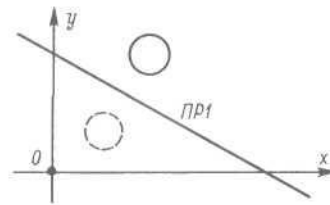


Рис. 2.58

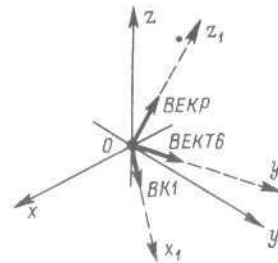


Рис. 2.59

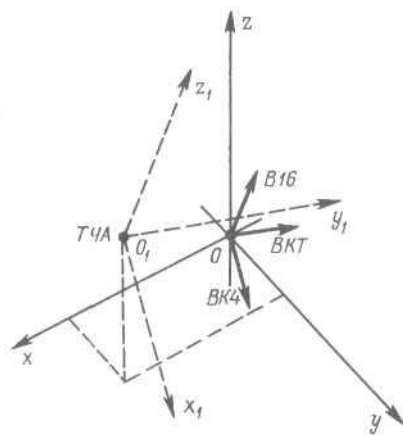


Рис. 2.60

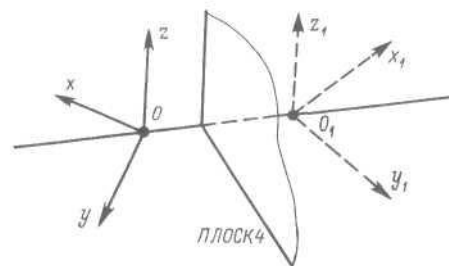


Рис. 2.61

Матрица, определяющая поворот исходной системы координат вокруг одной из осей и последующий перенос начала координат в точку, заданную явно или вектором:

имя = {точка | вектор}, { ПОВХУ|ПОВУЗ|ПОВЗХ }, угол

Матрица, определяющая симметрию относительно заданной прямой:

имя = прямая

Например (рис. 2.58):

МАТР8 = ПР1

Матрица, определяющая координатную систему, начало которой совпадает с началом координат исходной системы, а направления осей заданы векторами:

имя = вектор₁, вектор₂, вектор₃

Например (рис. 2.59):

МАТР9 = ВК1, ВЕКТ6, ВЕКР

Матрица, определяющая координатную систему, начало которой находится в точке, заданной явно или вектором, а направление осей заданы соответствующими векторами:

имя = вектор₁, вектор₂, вектор₃, ПЕРЕНОС, {точка/вектор₄}

Например (рис. 2.60):

М2 = ВК4, ВКТ, В16, ПЕРЕНОС, ТЧА

Матрица, определяющая координатную систему, являющуюся зеркальным отображением исходной относительно заданной плоскости:

имя = плоскость

Например (рис. 2.61):

МАТР2 = ПЛОСК4

Матрица, определяющая преобразование координат, обратное преобразованию, определенному заданной матрицей:

имя = ОБРАТИ, матрица

Например; если ТЧ2 = МАТР2, ТЧ1 и М12 = ОБРАТН, МАТР2, то справедливо геометрическое определение: ТЧ1=ТЧ2, М12

2.9.8 Определение плоскости

В геометрических определениях плоскости используются следующие модификаторы: ХУПАР – определяемая плоскость параллельна плоскости xu , УЗПАР – параллельна плоскости uz и ЗХПАР – плоскости zx . Отсутствие модификатора эквивалентно использованию модификатора ХУПАР.

Плоскость, параллельная одной из координатных плоскостей и проходящая через заданную точку:

имя = {ХУПАР|УЗПАР|ЗХПАР}, точка

Например:

ПЛОСК П1 = УЗПАР, ТЧ3

Плоскость, параллельная одной из координатных плоскостей и находящаяся на заданном расстоянии от нее:

имя = {ХУПАР |УЗПАР |ХЗПАР}, число

Например:

ПЛОСК8 = ЗХПАР, -29.6

Плоскость, совпадающая с плоскостью, определенной ранее:

имя = плоскость

Например:

ПЛОСК ПЛТ = ПЛОСК8

Плоскость, определенная коэффициентами ее нормального уравнения:

имя = число₁, число₂, число₃, число₄

Плоскость, проходящая через три точки, не лежащие на одной прямой:

имя = точка₁, точка₂, точка₃

Например:

ПЛ54 = ТЧ1, ТЧ18, ТЧ46

Плоскость, проходящая через заданную точку перпендикулярно вектору:

имя = вектор, точка

Например (рис. 2.62):

ПЛОСК8 = ВЕКР, ТЧ2

Плоскость, параллельная двум векторам и проходящая через заданную точку:

имя = вектор₁, вектор₂, точка

Например (рис. 2.63):

ПЛ6 = ВЕКТ1, ВЕКИ2, ТЧ1

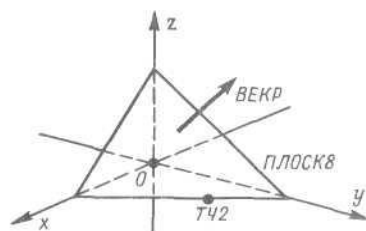


Рис. 2.62

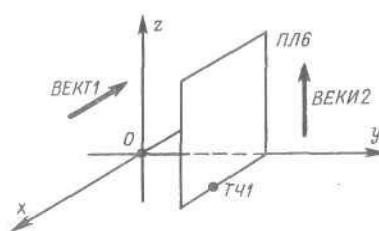


Рис. 2.63

Плоскость, параллельная другой плоскости и проходящая через заданную точку:

имя = точка, ПАРЛЕЛ, плоскость

Плоскость, полученная преобразованием заданной плоскости в соответствии с указанной матрицей:

имя = плоскость, матрица

2.9.9 Вложенное определение

Вместо имени ранее определенного геометрического объекта в правой части геометрического определения или в операторе движения можно задать «вложенное определение». Оно состоит из типа определяемого объекта, необязательного имени объекта и заключенного в скобки варианта геометрического определения соответствующего типа.

Если у такого геометрического объекта отсутствует имя, то значения его геометрических параметров сохраняются временно и используются процессором только для промежуточных вычислений.

На рис. 2.64 изображена окружность К.Р2, которая описана с использованием вложенных определений:

КР2 = ТОЧКА (ПРЯМАЯ ПР2 (ТЧ1, ТЧ2), ПР1), РАДИУС, 2.5

Данное геометрическое определение равноценно трем другим определениям:

ПР2 = ТЧ1, ТЧ2

ТЧА = ПР2, ПР1

КР2 = ЦЕНТР, ТЧА, РАДИУС, 2.5

Различие состоит лишь в том, что значение точки ТЧА во втором варианте записи сохранится, а при использовании вложенного определения точка пересечения прямых ПР1 и ПР2 не может быть использована в программе повторно. Глубину вложения можно увеличивать. В каждом из геометрических определений она может достигнуть двадцати.

2.10 Программирование движений инструмента

2.10.1 Общие сведения

В языке Техтран существуют два типа движения:

1) движение *точка–точка* (поточечное движение), при котором перемещение инструмента программируется прямым заданием либо целевой точки, либо величины смещения;

2) непрерывное движение, при котором перемещение программируется перечислением элементов траектории инструмента (т. е. участков прямых, окружностей, частей контуров) с указанием направления движения.

Возможность управления поверхностью детали позволяет программировать обработку контуров, имеющих наклонное «дно», благодаря чему с помощью языка Техтран возможно (с определенными ограничениями) описывать обработку деталей достаточно сложной пространственной формы. Во всех операторах движения вместо имен геометрических объектов можно использовать вложенные определения объектов тех же типов.

2.10.2 Исходное положение инструмента

Прежде чем программировать движение инструмента, необходимо задать его исходное положение. Для этого служит оператор ИЗ.

Оператор имеет следующий формат:

$$\text{ИЗ} \left\{ \begin{array}{l} \text{точка} \\ \text{точка} \end{array} \right\} |$$

где *точка* – имя точки; x, y, z – координаты точки.

В качестве координат можно использовать арифметические выражения. Если отсутствует координата z , то она считается равной нулю. В операторе ИЗ допускается использовать вложенные определения. В одной программе может присутствовать несколько операторов ИЗ.

Например, возможны следующие варианты оператора ИЗ:

ИЗ 10, 20

ИЗ А, В

ИЗ $(x + y)/2, (x - y)/2$

2.10.3 Движение точка – точка

При использовании этого оператора программист явно указывает каждое перемещение инструмента либо приращениями относительно предыдущего положения, либо точкой результирующего положения центра (вершины) инструмента. Этот тип движения реализуется операторами ВТОЧКУ и ПРИРАЩ. В дальнейшем этот тип движения в некоторых случаях будет именоваться как «поточечное движение».

Оператор ВТОЧКУ вызывает перемещение в заданную точку и имеет формат:

$$\text{ВТОЧКУ} \left\{ \begin{array}{l} \hat{o}\hat{i}\hat{z} \\ \hat{o}, \hat{o}, \hat{z} \end{array} \right\}$$

где *точка* – имя точки; *x*, *y*, *z* – координаты точки.

Вместо чисел в качестве координат точки можно использовать арифметические выражения. Если в операторе указаны только две координаты, то координата *z* считается равной нулю. На рис. 2.65 приведен пример движения *точка – точка*:

ТЧЗ = 35, 10
ИЗ 0, 0
ВТОЧКУ 0, -28
ВТОЧКУ 35, 10, 12
ВТОЧКУ ТЧЗ

Оператором ПРИРАЩ указывается перемещение относительно текущего положения на заданный вектор. Оператор имеет следующий формат:

$$\text{ПРИРАЩ} \left\{ \begin{array}{l} \hat{a}\hat{x}\hat{y}\hat{z} \\ \hat{o}, y, [z] \\ z \end{array} \right\}$$

где *вектор* – имя вектора; *x*, *y*, *z* – проекции вектора на координатные оси.

Вместо чисел в качестве проекций можно использовать арифметические выражения. Если в операторе ПРИРАЩ координаты *x* и *y* или координата *z* равны нулю, то перемещений по соответствующим осям не происходит. Поэтому оператор ПРИРАЩ 0, 0, А эквивалентен оператору ПРИРАЩ А.

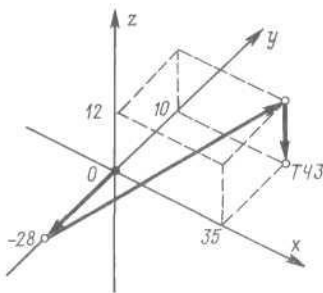


Рис. 2.65

При многократном использовании оператора ПРИРАЩ следует соблюдать осторожность, так как ошибка в начале последовательности влияет на остальные положения инструмента.

Движение, изображенное на рис. 2.65, можно записать при помощи оператора ПРИРАЩ следующим образом:

ИЗ 0, 0
ПРИРАЩ 0, -28, 0
ПРИРАЩ 35, (28+10), 12
ПРИРАЩ -12

При описании движения допускается сочетание операторов

ВТОЧКУ и ПРИРАЩ:

ИЗ 0, 0
ПРИРАЩ 0, -28, 0
ВТОЧКУ 35, 10, 12
ПРИРАЩ -12

Сокращенная запись операторов поточечного движения – это замена одним оператором нескольких записанных подряд операторов поточечного движения. Операнды этих операторов разделяются в этом случае двоеточием. Можно чередовать различные формы задания точек (приращения), например:

ВТОЧКУ ТЧ1: 6.2, А + В, 0 : ТОЧКА (ПР1, ПР2) : ТЧЗ
ПРИРАЩ 5,1 : 0,3*КОС (45) : ВЕКТ1

Следует иметь в виду, что сокращенная запись этих операторов движения затрудняет понимание диагностических сообщений, имеющих ссылку на номер строки.

2.10.4 Непрерывное движение

При программировании непрерывного движения на языке Техтран описывается движение инструмента относительно трех поверхностей (рис. 2.66, а): поверхности детали (ПД); направляющей поверхности (НП); ограничивающей поверхности (ОП). На рис. 2.66, б приведено взаимное положение инструмента и поверхности. Рабочая часть инструмента при этом находится в постоянном контакте с поверхностью детали и направляющей поверхностью. Конечное положение инструмента после выполнения очередного оператора непрерывного движения определяется ограничивающей поверхностью (рис. 2.67, а). Прямые и окружности, составляющие контур детали, представляют собой проекции НП и ОП на горизонтальную плоскость (рис. 2.67, б).

Поверхность детали представляет собой плоскость – горизонтальную или наклонную. По умолчанию ПД совпадает с плоскостью xOy . В дальнейшем для простоты изложения НП и ОП отождествляются со своими проекциями на плоскость xOy .

Оператор ИНСТР (описание инструмента) предназначен для задания геометрических параметров инструмента, необходимых для расчета траектории его движения относительно контура и поверхности детали. Оператор имеет следующий формат:

$$\text{ИНСТР диаметр} \left[\begin{Bmatrix} 0 \\ \delta \end{Bmatrix} \right]$$

где диаметр – диаметр инструмента; радиус – радиус скругления режущей кромки.

Радиус скругления не может быть больше радиуса цилиндрической части фрезы. Отсутствие оператора ИНСТР эквивалентно заданию инструмента нулевого диаметра. Новый оператор ИНСТР отменяет действие предыдущего. При непрерывном движении изменение диаметра инструмента запрещено.

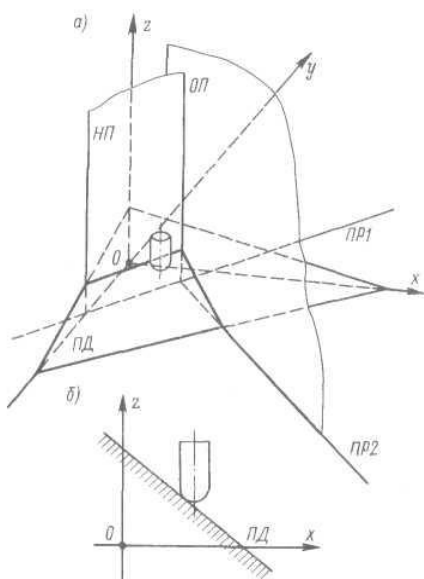


Рис. 2.66

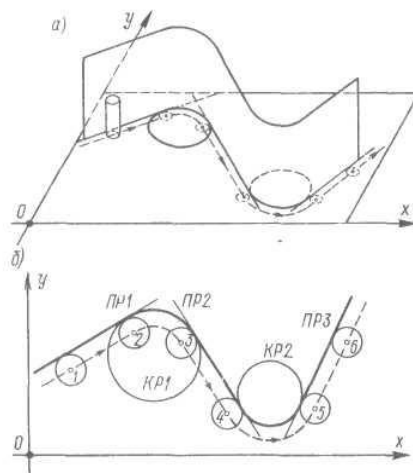


Рис. 2.67

На рис. 2.68 и 2.69 изображены различные резцы и фрезы и их описание на языке Техтран. На рис. 2.70, а, б представлены схемы электроэрозионной обработки и формирования зоны резания и описания на языке Техтран (A – ширина реза, $A=d+2\delta$). Точкой отмечен геометрический центр инструмента. Траекторией движения инструмента является линия, по которой перемещается в пространстве геометрический центр инструмента.

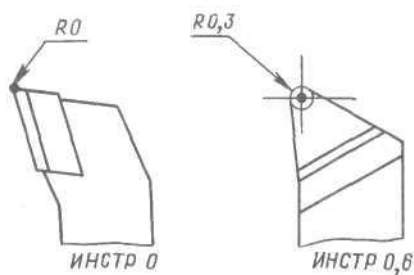


Рис. 2.68

Оператор МЕТОД используют для задания метода интерполяции и режима аппроксимации окружности отрезками прямой.

Формат оператора:

$$\text{ЛИНЕЙН} \left\{ \begin{array}{l} \text{ЭЭІАЭІ} , \text{АЭНЭЭІ} , \div \text{ēñēī}_1 \\ \text{ЭЭІЕДОА} \left\{ \begin{array}{l} [\text{ЭАААД}] \\ \text{ІАЭНДАА} \end{array} \right\} , \div \text{ēñēī}_2 \end{array} \right\}$$

где ЛИНЕЙН –линейная интерполяция (аппроксимация

окружностей отрезками прямых); ЛИНКРУГ – линейно-круговая интерполяция (аппроксимация кривых дугами окружностей); К.ВАДР – линейно-круговая интерполяция с делением окружностей на квадранты; НЕК.ВАДР – то же без деления на квадранты; *число₁* – максимальная длина отрезков, аппроксимирующих окружность; *число₂* – максимальный радиус окружности, при котором автоматически происходит переход к линейной интерполяции.

На рис. 2.71, а, б представлены линейная (а) и линейно-круговая (б) интерполяции.

Параметр МАКСЛИН применяется для ограничения длины аппроксимирующих отрезков (рис. 2.72):

МЕТОД ЛИНЕЙН, МАКСЛИН, 4, 35.

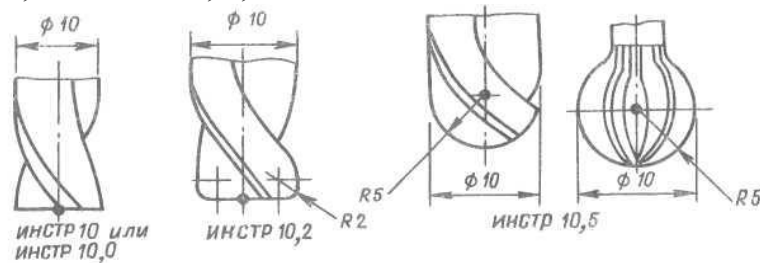


Рис. 2.69

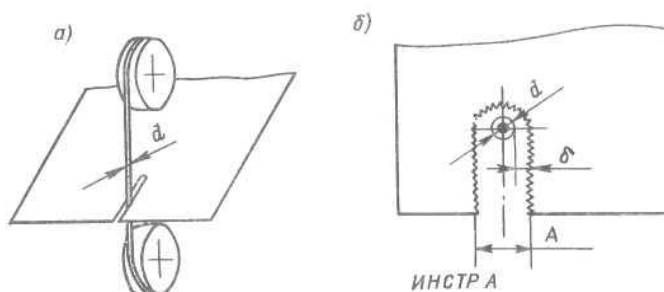


Рис. 2.70

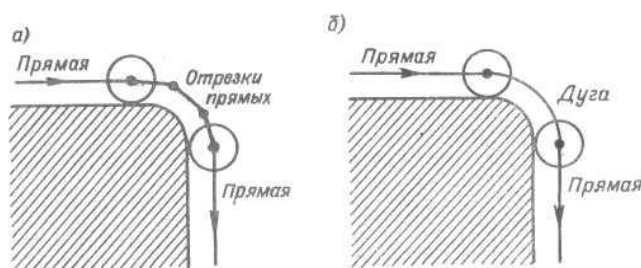


Рис. 2.71

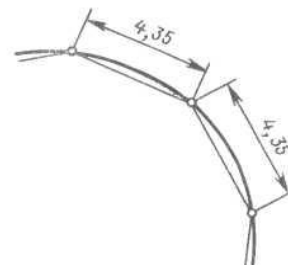


Рис. 2.72

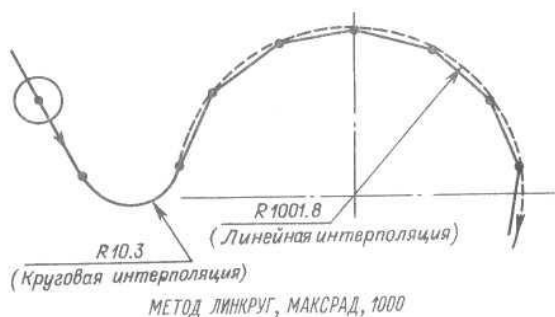


Рис. 2.73

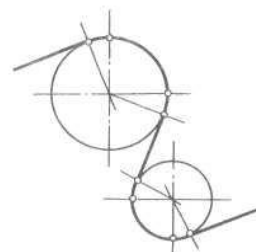


Рис. 2.74

На рис. 2.73 показано применение параметра МАКСРАД в операторе МЕТОД. Точками ограничены участки траектории, заданные в управляющей программе отдельными кадрами:

МЕТОД ЛИНКРУГ, МАКСРАД, 1000.

При использовании модификатора КВАДР процессор системы разбивает дуги окружностей, составляющих траекторию, на участки, лежащие в отдельных квадрантах (рис. 2.74).

Отсутствие в программе оператора МЕТОД эквивалентно применению оператора МЕТОД ЛИНКРУГ, КВАДР. Круговая интерполяция разрешена только в одной из координатных плоскостей или в плоскостях, параллельных им. Круговая интерполяция с делением на квадранты возможна только в плоскостях, параллельных плоскости xOy .

Если поверхность детали не параллельна плоскости xOy , следует задавать оператор МЕТОД, ЛИНЕЙН, даже если контур не содержит дуг окружностей.

Операторы НАРДОП и ВНДОП используются для задания точности линейной аппроксимации дуг окружности, которые заменяются отрезками прямых. НАРДОП число – точность аппроксимации касательными – наружный допуск (рис. 2.75, а); ВНДОП число – точность аппроксимации хордами – внутренний допуск (рис. 2.75, б).

При одновременном использовании этих операторов производится аппроксимация секущими (рис. 2.75, а). Если не заданы ни НАРДОП, ни ВНДОП, то система устанавливает следующие их значения:

НАРДОП.0, 05

ВНДОП 0

Оператор ИДИ задает начало непрерывного движения. Оператор ИДИ выводит инструмент в рабочее положение относительно управляющих поверхностей $НП$, $ПД$ и $ОП$. Положение инструмента относительно каждой из них указывается модификаторами ДО, НА, ЗА. Оператор имеет следующий формат:

$$\begin{matrix} \ddot{E} \ddot{A} \ddot{E} \end{matrix} \left\{ \begin{matrix} \ddot{A} \ddot{f} \\ \dot{f} \ddot{A} \\ \zeta \ddot{A} \end{matrix} \right\} \ddot{f} \left[\left\{ \begin{matrix} \ddot{A} \ddot{f} \\ \dot{f} \ddot{A} \end{matrix} \right\} \ddot{f} \ddot{A} \right] \left[\left\{ \begin{matrix} \ddot{A} \ddot{f} \\ \dot{f} \ddot{A} \\ \zeta \ddot{A} \end{matrix} \right\} \ddot{f} \right]$$

По команде ИДИ инструмент выводится в требуемую точку по кратчайшему пути. Если $НП$ и $ОП$ имеют две точки пересечения, то выбирается ближайшая из них. На рис. 2.76 показано применение модификаторов ДО, НА, ЗА при движении с одной (а–в) и с двумя поверхностями (г–д) соответственно, а иллюстрация этих примеров на языке Техтран приведена ниже:

ИЗ ТЧ1; ИДИ ДО ПР1 (а)

ИЗ ТЧ1; ИДИ НА ПР1 (б)

ИЗ ТЧ1; ИДИ ЗА ПР1 (в)

ИЗ ТЧ1; ИДИ ДО ПР1 ДО ПР2 (г)

ИЗ ТЧ1; ИДИ ДО ПР1 ЗА ПР2 (д)

При необходимости направление подхода может указываться операторами НАПРВ (в направлении вектора) или НАПРТ (в направлении точки).

На рис. 2.77, а, б приведены соответственно иллюстрации операторов:

ИДИ НА ПРХ ДО ПЛ1 ЗА ПР3

ИДИ НА ПРХ НА ПЛ1 ЗА ПР3

Если в операторе ИДИ задана поверхность детали, то при непрерывном движении инструмент будет находиться в контакте с ней в соответствии с заданным модификатором. Команды поточечного движения (ВТОЧКУ и ПРИРАЩ), вызывающие отход от поверхности детали по координате z , будут игнорироваться. Отвести инструмент от $ПД$ можно только после оператора ПОВДЕТ ОТМЕН.

Если же $ПД$ в операторе ИДИ опущена, то поверхностью детали становится плоскость, параллельная плоскости xOy и отстоящая от нее на значение текущей координаты z . При этом разрешено перемещать инструмент по вертикальной оси.

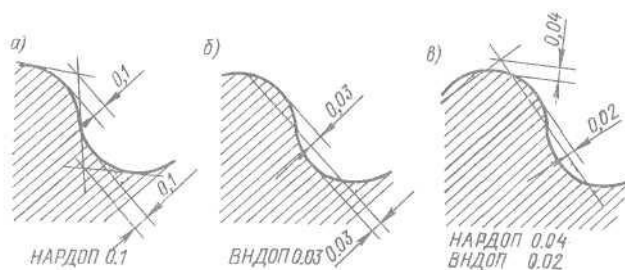


Рис. 2.75

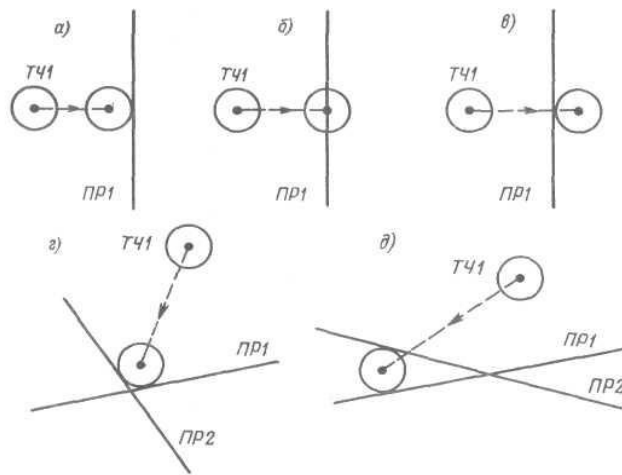


Рис. 2.76

Различие в использовании оператора ИДИ с одной, двумя и тремя поверхностями иллюстрирует рис. 2.78:

перемещение 0–1:

ИЗ ТЧ1; ИДИ ДО ПР1

перемещение 0–2:

ИЗ ТЧ1; ИДИ ДО ПР1 ДО ПР2

перемещение 0–3:

ИЗ ТЧ1; ИДИ ДО ПР1 ДО ПЛ1

перемещение 0–4:

ИЗ ТЧ1; ИДИ ДО ПР1 ДО ПЛ1 ДО ПР2

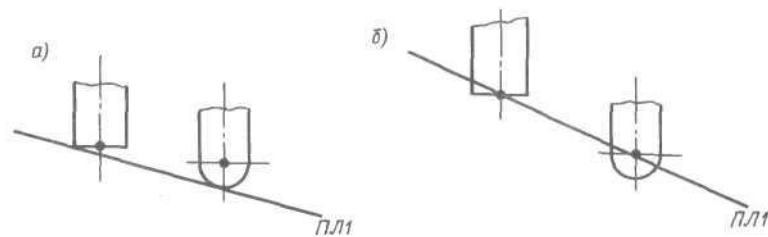


Рис. 2.77

В операторе ИДИ с двумя или тремя поверхностями в качестве ОП должна быть указана поверхность, которая будет направляющей в следующей команде движения.

Операторы НАПРТ и НАПРВ являются вспомогательными и используются совместно с оператором ИДИ в следующих случаях:

1) если при использовании оператора ИДИ с одной поверхностью (кроме поверхности детали) надо осуществить подход не к ближайшей точке поверхности;

2) если при использовании оператора ИДИ с двумя поверхностями (не считая поверхности детали) необходимо выбрать одну из двух возможных точек пересечения НП и ОП;

3) если инструмент уже находится в требуемом положении относительно НП;

4) если в операторе ИДИ в качестве НП указан объект типа КОНТУР.

В первом случае отсутствие оператора, задающего направление подхода, вызовет подход к ближайшей точке поверхности, а в остальных трех случаях приведет к ошибке.

Операторы НАПРТ и НАПРВ не управляют перемещением по третьей координате, т. е. направление подхода к НП определяется проекцией вектора подхода на горизонтальную плоскость.

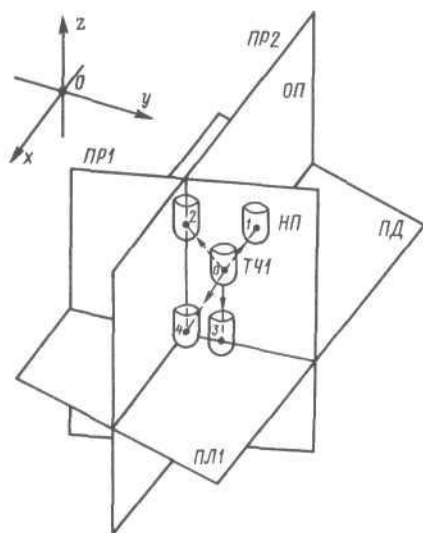


Рис. 2.78

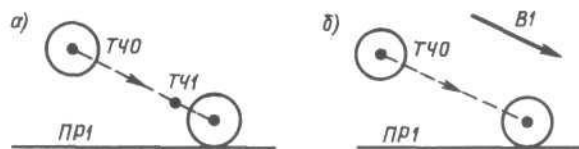


Рис. 2.79

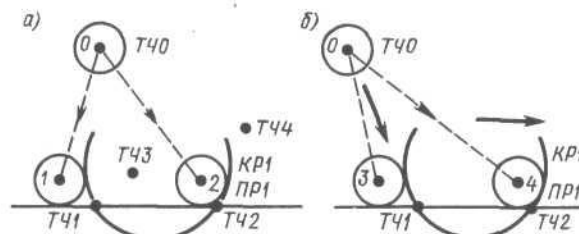


Рис. 2.80

Формат оператора НАПРТ:

$$\text{НАПРТ} \left\{ \begin{array}{l} \partial \hat{a} \div \hat{e} \hat{a} \\ \tilde{o}, \acute{o} \end{array} \right\}$$

где *точка* – имя точки; x, y – переменные или арифметические выражения, значения которых являются соответственно координатами x и y точки.

Формат оператора НАПРВ:

$$\hat{I} \hat{A} \hat{I} \hat{B} \hat{A} \hat{A} \left\{ \begin{array}{l} \hat{a} \hat{a} \hat{e} \hat{o} \hat{i} \hat{d} \\ \tilde{o}, \acute{o} \end{array} \right\}$$

где *вектор* – имя вектора; x, y – переменные или арифметические выражения, значения которых являются соответственно проекциями x и y вектора.

На рис. 2.79, а, б представлены методы подхода к НП в направлении точки и вектора соответственно:

(а) ИЗ ТЧО

НАПРТ ТЧ1

ИДИ ДО ПР1

(б) ИЗ ТЧО

НАПРВ В1

ИДИ ДО ПР1

В случае, когда НП и ОП пересекаются в двух точках, целевая точка подхода выбирается с помощью операторов НАПРТ или НАПРВ. Если используется НАПРТ, то целевой выбирается точка, ближайшая к точке, заданной в этом операторе. Если используется НАПРВ, то указанный там вектор используется для выбора из двух точек так же как и модификаторы типа ХБ – УМ в геометрических определениях (рис. 2.80), например:

перемещение 0–1 (рис. 2.80, а) – оператор ИДИ задает переход в точку ТЧ1, так как она ближе к ТЧ3, чем ТЧ2:

ИЗ ТЧО

НАПРТ ТЧ3

ИДИ ДО ПР1 ДО КР1

перемещение 0–2 (рис. 2.80, а) – инструмент переходит в точку ТЧ2, так как она ближе к ТЧ4, чем ТЧ1

ИЗ ТЧО

НАПРТ ТЧ4

ИДИ ДО ПР1 ДО КР1

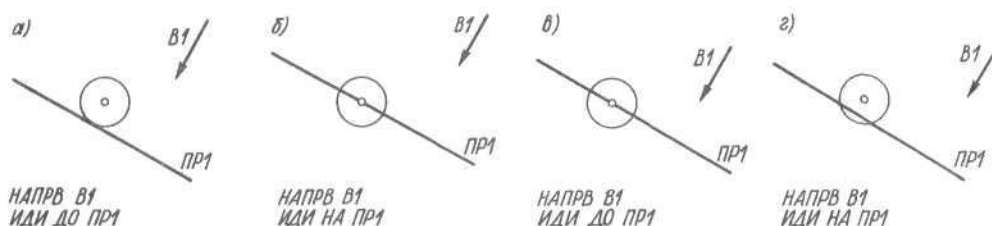


Рис. 2.81

перемещение 0–3 (рис. 2.80,6):

ИЗ ТЧ0

НАПРВ ВЕКТОР (ТЧ0, ТЧ1)

ИДИ ДО ПР1 ДО КР1

перемещение 0–4 (рис. 2.80,6):

ИЗ ТЧ0

НАПРВ ВЕКТОР (1, 0, 0)

ИДИ ДО ПР1 ДО КР1

Если инструмент уже подведен к *НП*, то для вывода инструмента в требуемое положение относительно *НП* или для задания направления, относительно которого будет выбран оператор непрерывного движения, следует указать направление подхода операторами НАПРВ или НАПРТ (рис. 2.81, а, б). При этом запрещен вывод инструмента в направлении, встречном указанному в операторах НАПРВ или НАПРТ (рис. 2.81, в). В исходном положении инструмент может касаться *НП* или его центр может находиться на *НП*; вариант, показанный на рис. 2.81, г, недопустим.

Если в операторе ИДИ в качестве *НП* указан объект типа КОНТУР, то следует указать направление подхода (см. п. 2.11):

ИЗ ТЧ1

НАПРВ В1

ИДИ ДО КОНТ1

Оператор ПОВДЕТ предназначен для управления поверхностью детали при поточечном и непрерывном движении.

Оператор имеет следующий формат:

$$\text{ПОВДЕТ} \left\{ \begin{array}{l} \div \grave{e} \tilde{n} \tilde{e} \tilde{t} \\ \grave{e} \grave{y} \\ \hat{t} \grave{o} \grave{l} \hat{a} \hat{t} \end{array} \right\}$$

число – число или арифметическое выражение, задающее *ПД* как плоскость, параллельную плоскости *xOy* и отстоящую от нее на указанном расстоянии; *имя* – *t*, символическое имя плоскости, которая будет использоваться как *ПД* в последующих; командах движения; ОТМЕН – служебное слово, вызывающее отмену ранее введенной *ПД* и возврат к плоскости *xOy*.

При поточечном движении координата *z* каждой точки будет вычисляться в соответствии с заданной *ПД*. В отличие от оператора ПЛЗ, который задает координату *z* при построении точек, оператор ПОВДЕТ используется с командами движения.

При непрерывном движении применение оператора ПОВДЕТ равносильно заданию в операторе ИДИ поверхности детали с модификатором НА. Вводя новые операторы ПОВДЕТ, можно переходить от одной поверхности детали к другой, не прерывая непрерывного движения. Применять одновременно операторы ПОВДЕТ и *ПД* в операторе ИДИ нельзя.

Независимо от того, как в программе определена поверхность детали (в операторе ИДИ или в операторе ПОВДЕТ), инструмент будет находиться в контакте с ней до тех пор, пока *ПД* не будет отменена командой ПОВДЕТ ОТМЕН.

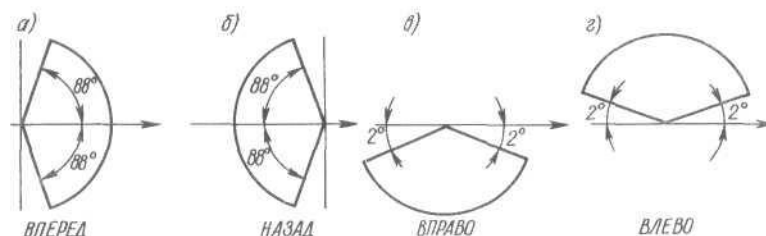


Рис. 2.82

После выхода на направляющую поверхность движение по каждому из геометрических элементов контура описывается в следующем формате:

$$\left\{ \begin{array}{l} \hat{A} \hat{I} \hat{A} \hat{D} \hat{A} \hat{A} \\ \hat{I} \hat{A} \hat{\zeta} \hat{A} \hat{A} \\ \hat{A} \hat{E} \hat{A} \hat{A} \hat{I} \\ \hat{A} \hat{I} \hat{D} \hat{A} \hat{A} \hat{I} \end{array} \right\} \hat{I} \hat{I} \left\{ \begin{array}{l} \hat{A} \hat{I} \\ \hat{I} \hat{A} \\ \hat{\zeta} \hat{A} \\ \hat{E} \hat{A} \hat{N} \end{array} \right\} \left[\div \grave{e} \tilde{n} \tilde{e} \tilde{t} \right] \left[\hat{A} \hat{D} \hat{A} \hat{N} \hat{A} \times \right] \hat{I} \hat{I}$$

ВПЕРЕД, НАЗАД, ВПРАВО, ВЛЕВО – служебные слова, указывающие направление перемещения по направляющей поверхности; *НП* и *ОП* – направляющая и ограничивающая поверхности; *число* – номер пересечения *НП* с *ОП*, до которого задано движение по *НП*; КАС – модификатор, определяющий движение по *НП* до точки касания с *ОП*; ДО, НА, ЗА – модификаторы, определяющие положение инструмента относительно *ОП* (аналогично модификаторам оператора ИДИ).

Направление движения выбирается относительно предыдущего перемещения в соответствии с рис. 2.82, а–г (секторами ограничены области действия служебных слов, указывающих направление перемещения).

При движении по окружности направлением движения считается направление касательной к окружности в точке пересечения *НП* и *ОП*.

Поэтому в случае касания *НП* и *ОП* следует применять только операторы ВПЕРЕД и НАЗАД (рис. 2.83, а, б).

Пример программирования непрерывного движения для контура, изображенного на рис.

2.84 (из положения / в положение 5):

ВПЕРЕД П1 ЗА КР1

ВЛЕВО КР1 КАС К2

ВПЕРЕД КР2 ДО ПР2

ВПРАВО ПР2 НА П1

Поверхности, указанные в операторе непрерывного движения, могут пересекаться и не один раз. Если не указывать номер пересечения, то инструмент будет выведен на (до, за) первое пересечение *НП* и *ОП*. Поэтому, когда требуется остановиться на последующих пересечениях, следует указывать номер пересечения *НП* с *ОП*.

Номер пересечения отсчитывается для каждого оператора движения в отдельности, вне связи с предыдущими операторами непрерывного движения. Причем, если перед выполнением оператора движения инструмент уже касался *НП* и *ОП*, это пересечение считается первым.

Примеры программирования непрерывного движения:

для контура, изображенного на рис. 2.85, а,–

1–2 ВПЕРЕД ПР1 ЗА КР1

или ВПЕРЕД ПР1 ЗА 1 КР1

2–3 ВПЕРЕД ПР1 ЗА 2 КР1

для контура, изображенного на рис. 2.85, б,–

1–2 ВПЕРЕД ПР2 ДО 2 КР2

2–3 ВЛЕВО КР2 НА 2 ПР2

3–4 ВПЕРЕД КР2 НА 2 ПР2

Номер пересечения, указанный в операторе движения, не должен превышать общего числа пересечений данных *НП* и *ОП*, в противном случае ситуация считается ошибочной (например, задание второго пересечения для двух прямых). Исключение составляет третье пересечение окружности с прямой, под которым подразумевается полный обход окружности.

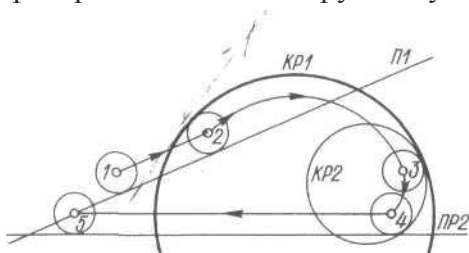


Рис. 2.84

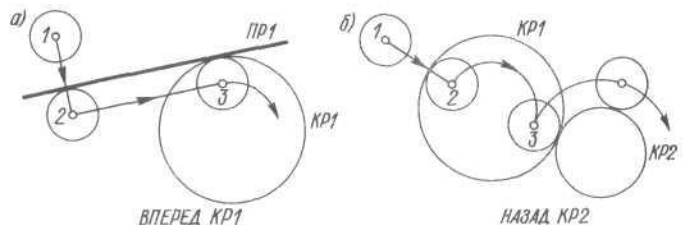


Рис. 2.83

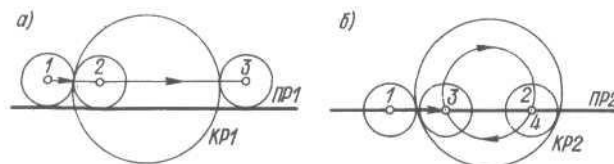


Рис. 2.85

Примеры программирования с «третьим» пересечением:

для контура, изображенного на рис. 2.86, а,–

ВПЕРЕД ПР1 ДО КР2

НАЗАД КР2 ДО 3 ПЕРЕСЕЧ ПР1

для контура, изображенного на рис. 2.86, б,–

ВПЕРЕД ПР2 ЗА КР3

ВЛЕВО КР3 ДО 3 ПР2

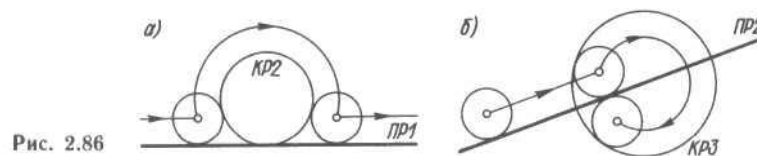


Рис. 2.86

Если поверхность детали является наклонной плоскостью, то команды движения следует задавать так же, как и в случае расположения обрабатываемого контура в плоскости xOy . Истинное перемещение инструмента представляет собой результат проектирования на ПД в направлении оси z эквидистанты к этому контуру (с учетом формы инструмента и его положения относительно ПД).

На рис. 2.87 показано программируемое и истинное перемещения инструмента при заданной поверхности детали для следующей программы:

МЕТОД ЛИНЕЙН

ИЗ ТЧ1

ИДИ ДО ПР1 ДО ПЛ1 ЗА КР1

ВЛЕВО ПР1 ДО ПР2

ВЛЕВО ПР2 ДО КР1

ВЛЕВО КР1 ДО ПР1

ПОВДЕТ ОТМЕН

ВТОЧКУ ТЧ1

Неявное определение ограничивающей поверхности имеет место в том случае, если известно положение инструмента относительно контура обрабатываемой детали и ограничивающая поверхность и относящийся к ней модификатор в промежуточном операторе движения могут быть опущены. Они выбираются системой по направляющей поверхности следующего оператора движения. При этом необходимо задать положение инструмента относительно контура при помощи одного из операторов:

ИНСТНА – инструмент на контуре детали;

ИНСТЛВ – инструмент слева от контура (по направлению движения);

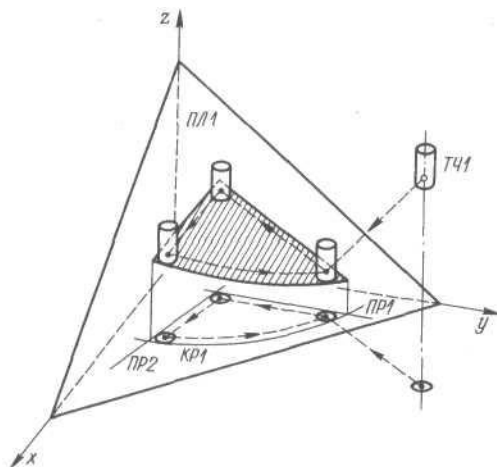


Рис. 2.87

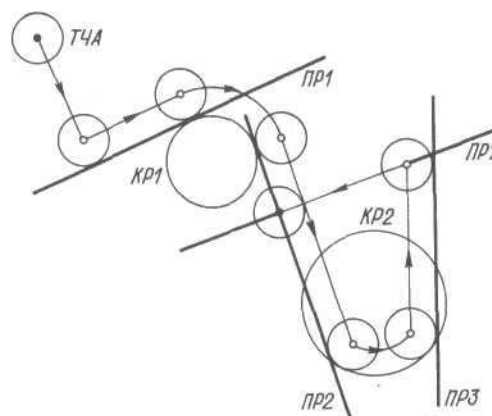


Рис. 2.88

ИНСТПР – инструмент справа от контура (по направлению движения).

Несколько следующих подряд операторов движения одного типа могут заменяться одним, при этом операнды перечисляются через двоеточие. Число пересечений и ограничивающая поверхность с соответствующим модификатором задаются как обычно. При необходимости можно переходить от неявной ограничивающей поверхности к явной и наоборот.

Данный прием программирования позволяет значительно уменьшить объем исходной информации, однако при этом затрудняется понимание диагностических сообщений, связанных с номером строки.

На рис. 2.88 изображен контур, при программировании которого целесообразно использовать операторы непрерывного движения с неявной ограничивающей поверхностью:

ИЗ ТЧА

ИДИ ДО ПР1

ИНСТЛВ

ВЛЕВО ПР1

ВПЕРЕД КР1 : ПР2 ДО 2 ПЕРЕСЕЧ

ВЛЕВО КР2 : ПР3 НА ПР7

ВЛЕВО ПР7 НА ПР2

Последний оператор движения всегда должен содержать явно заданную ограничивающую поверхность. После операторов движения, не содержащих ограничивающей поверхности, запрещено использовать операторы постпроцессора и управления интерполяцией.

Совместное применение операторов поточечного и непрерывного движений имеет некоторые особенности. Поточечное движение прерывает непрерывное движение, поэтому после операторов ПРИРАЩ и ВТОЧКУ невозможно продолжить непрерывное движение при помощи операторов ВПРАВО, ВЛЕВО, ВПЕРЕД, НАЗАД. Для возобновления непрерывного движения следует задать оператор ИДИ.

Исключение составляет перемещение по третьей координате при незаданной поверхности детали (рис. 2.89):

ВПЕРЕД ПР1 ДО КР2

ПРИРАЩ 50

ВПРАВО КР2

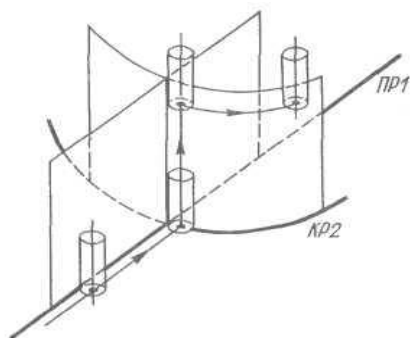


Рис. 2.89

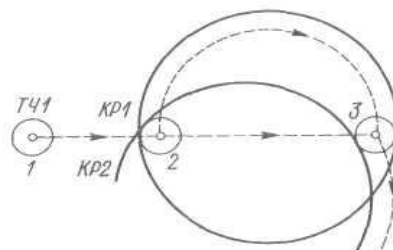


Рис. 2.90

Фиктивное движение, т. е. расчет перемещений инструмента без передачи их в управляющую программу можно запрограммировать с помощью оператора НЕОБР и отменяющего его оператора ОБРАБ. Этот прием используют для того, чтобы переместить инструмент в положение, которое описать явным образом неудобно или невозможно. Например, для случая, изображенного на рис. 2.90, непосредственный переход из положения 1 в положение 3 программируется операторами:

ИЗ ТЧ1

НЕОБР

ИДИ ЗА КР2 ЗА КР1

ВЛЕВО КР1 ДО 2 КР2

ОБРАБ

ВПЕРЕД КР2

При этом исключается движение по окружности *КР1* из положения 2 в положение 3

2.10.5 Преобразование траектории движения инструмента

При программировании сложных деталей бывает удобно кодировать элементы контура не в базовой системе координат детали, а в некоторых местных системах координат. При этом необходимо задать положение каждой местной системы координат в базовой системе координат детали, для чего используются матричные преобразования.

Возможны два способа использования матриц: в геометрических определениях и в командах движения.

В первом случае новый геометрический объект определяется посредством матричного преобразования, производимого над ранее определенным геометрическим объектом. Во втором случае преобразованию подвергается отмеченный участок траектории, для чего используют операторы ТРАНС и ТРАНСРЕЗ.

Оператор ТРАНС предназначен для преобразования направляющих и ограничивающих поверхностей, относительно которых описывается движение. Оператор ТРАНСРЕЗ используют для преобразования точек рассчитанной траектории.

Оператор ТРАНС имеет следующий формат:

$$\text{ÐÐÀÍÑ} \left\{ \begin{array}{l} \text{ìàððèöà} \\ \text{îòìÁí} \end{array} \right\}$$

где матрица – символическое имя матрицы.

Матрица, заданная в этом операторе, должна быть плоской, т. е. это либо матрица поворота в плоскости xOy , либо матрица переноса по координатам x и y , либо матрица симметрии относительно прямой, либо произведение матриц такого вида.

Оператор ТРАНС выполняет преобразование траектории движения инструмента в соответствии с заданной матрицей до применения нового оператора ТРАНС. Оператор ТРАНС ОТМЕН задает возврат к исходной системе координат. Если в программе последовательно применяются несколько операторов ТРАНС, то следует учесть, что каждый следующий оператор определяет преобразование исходной системы координат, а не текущей; оператор ТРАНС ОТМЕН задает возврат к исходной системе координат, а не к действовавшей ранее.

Например, необходимо обработать три одинаковых элемента (1–3), расположенных так, как показано на рис. 2.91. Фрагмент программы на языке Техтран имеет вид:

```
...
команды обработки элемента 1
МАТР М1=ПОВОРОТ, 30
ТРАНС М1
команды обработки элемента 1
МАТР М2 = ПОВОРОТ, 60
ТРАНС М2
команды обработки элемента 1
ТРАНС ОТМЕН
...
```

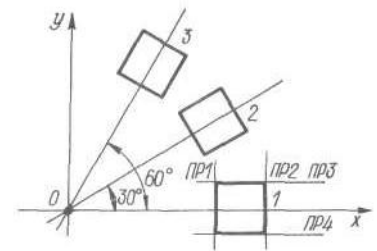


Рис. 2.91

Оператор ТРАНС действует на поверхности движения (прямые, окружности и др.), а не на получаемые точки траектории. Поэтому следует иметь в виду, что неправильное употребление этого оператора в середине непрерывного движения может привести к разрыву траектории (рис. 2.92).

Предполагается вывести инструмент последовательно из Т40 в позиции 1–4:

```
ИЗ Т40
ИДИ ДО КР1
ВЛЕВО КР1 ДО ПР1
ТРАНС МАТР (ПОВОРОТ, 30)
ВПРАВО ПР1 ДО КР2
```

После выполнения оператора 3 инструмент оказался в позиции 2. После выполнения оператора 4 прямая ПР1 заняла положение ПР1', инструмент же остался в позиции 2. В связи с этим выполнить оператор 5 невозможно.

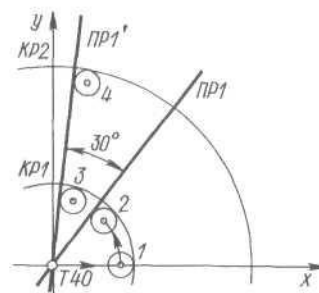


Рис. 2.92

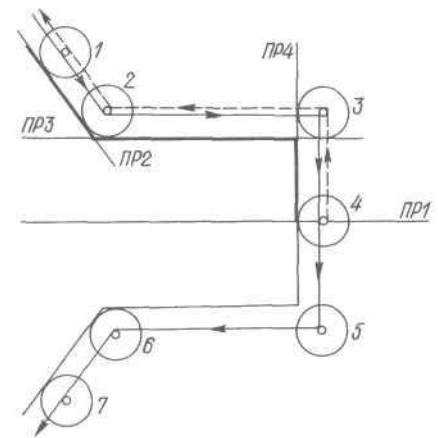


Рис. 2.93

При использовании матрицы симметрии относительно прямой операторы движения, следующие после оператора ТРАНС, должны описывать движение по исходному контуру в обратном направлении (кроме первого перемещения после оператора ТРАНС). При этом не требуется изменять положение инструмента относительно контура, если оно было задано оператором ИНСТЛВ или ИНСТПР.

Движение по контуру, изображенному на рис. 2.93 (— — — траектория, описанная после оператора ТРАНС; — — — фактическая траектория движения инструмента), программируется следующим образом (табл. 2.4).

Оператор ТРАНСРЕЗ имеет следующий формат:

$$\text{ÐÐÀÍÑÐÄÇ} \left\{ \begin{array}{l} \text{èÿ} \\ \text{îòìÁí} \end{array} \right\}$$

где имя – символическое имя матрицы.

Т а б л и ц а 2.4

Оператор программы	Участок траектории	
	запрограммированной	фактической
ИНСТЛВ		
ВПЕРЕД ПР2	-1-2	-1-2
ВЛЕВО ПР3	2-3	2-3
ВПРАВО ПР4	3-4	3-4
НА ПР1		
МАТР М1=ПР1		
ТРАНС М1		
ВПЕРЕД ПР4	4-3	4-5
ВЛЕВО ПР3	3-2	5-6
ВПРАВО ПР2	2-1-....	6-7-....

Матрица, заданная в этом операторе, может быть любой (в том числе и трехмерной). В отличие от оператора ТРАНС оператор ТРАНСРЕЗ осуществляет преобразование по заданной матрице рассчитанных точек траектории. Действие его распространяется до нового оператора ТРАНСРЕЗ или до оператора ТРАНСРЕЗ ОТМЕН.

Различие в результатах, полученных при использовании операторов ТРАНС и ТРАНСРЕЗ в приведенном примере, иллюстрируется на рис. 2.94 (--- → — программируемое перемещение; → — фактическое перемещение). Например:

```

ТЧ0 = 0,0
ТЧ1 = -100,0
ПР1=ТЧ0, 60
МАТР М1=ПОВХУ, -90
ИЗ ТЧ1
НАПРТ ТЧ0
ИДИ НА ПР1
ТРАНС М1
ВПРАВО ПР1 НА ОКРУЖИ (ТЧ0, 50)

```

Оператор ТРАНС выполняет преобразование поверхностей. Поэтому после применения оператора ТРАНС прямая ПР1 примет положение ПР1', а оператор ВПРАВО вызовет перемещение инструмента из ТЧ0 в точку 1 (см. рис. 2.94, а).

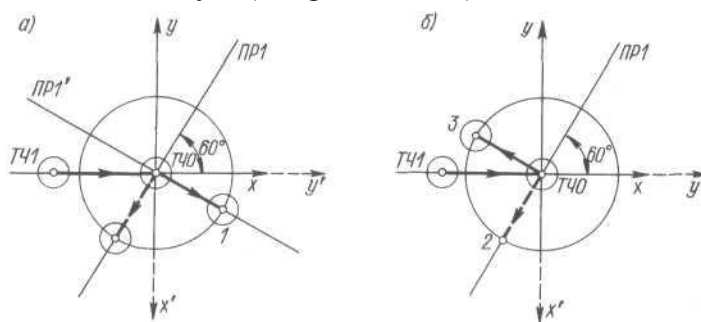


Рис. 2.94

Если в этой же программе заменить оператор ТРАНС М1 оператором ТРАНСРЕЗ М1, то преобразованию будет подвергнута точка траектории 2, которая займет положение 3, куда и переместится инструмент из ТЧ0 (см. рис. 2.94, б).

2.11 Контур

Наряду с элементарными типами геометрических объектов в языке Техтран имеется комбинированный тип — КОНТУР. Геометрические объекты этого типа являются совокупностями отрезков прямых и дуг окружностей. Они могут использоваться в командах непрерывного движения аналогично элементарным геометрическим объектам, т. е. инструмент может быть выведен на контур оператором ИДИ, а движение по нему осуществляется с помощью операторов ВЛЕВО, ВПРАВО, ВПЕРЕД, НАЗАД.

Различают простые и составные контуры. Простой контур — комбинация элементарных геометрических объектов, а составной контур является комбинацией нескольких простых контуров.

Простой контур. Описание простого контура состоит из последовательности операторов

непрерывного движения, ограниченных заголовком контура и служебным словом КОНКОН:

[КОНТУР] *имя* = ИЗ точка, модификатор

оператор₁

оператор₂

...

оператор_{n-1}

оператор_n

КОНКОН

где КОНТУР – служебное слово, задающее тип определяемого объекта (если он еще не известен); *имя* – имя, присвоенное контуру; *точка* – имя первой точки контура (если контур замкнутый, то любой точки контура); *модификатор* – любой из модификаторов ХБ–ХМ–УБ–УМ, задающий направление, относительно которого выбирается оператор₁; оператор₁, оператор₂, ..., оператор_n – операторы движения.

Простой контур называется замкнутым, если начальная и конечная точки контура совпадают, и незамкнутым, если точки не совпадают. Направление движения в операторе₁ определяется относительно модификатора, указанного в заголовке контура. Проще всего в качестве оператора₁ употреблять оператор ВПЕРЕД. Направление движения в последующих операторах указывается относительно предыдущего направления движения по правилам, изложенным в п. 2.10.

Оператор, ($1 \leq i \leq n$) может быть оператором с двумя поверхностями

$$\left\{ \begin{array}{l} \hat{A}\hat{I}\hat{A}\hat{D}\hat{A}\hat{A} \\ \hat{I}\hat{A}\hat{C}\hat{A}\hat{A} \\ \hat{A}\hat{E}\hat{A}\hat{A}\hat{I} \\ \hat{A}\hat{I}\hat{D}\hat{A}\hat{A}\hat{I} \end{array} \right\} \hat{I}\hat{I} \left\{ \begin{array}{l} \hat{A}\hat{I} \\ \hat{I}\hat{A} \\ \hat{C}\hat{A} \\ \hat{E}\hat{A}\hat{N} \end{array} \right\} \left[\div \hat{e}\hat{n}\hat{e}\hat{i} \quad \hat{I}\hat{A}\hat{D}\hat{A}\hat{N}\hat{A}\hat{A} \times \right] \hat{I}\hat{I}$$

либо оператором с одной поверхностью

$$\left\{ \begin{array}{l} \hat{A}\hat{I}\hat{A}\hat{D}\hat{A}\hat{A} \\ \hat{I}\hat{A}\hat{C}\hat{A}\hat{A} \\ \hat{A}\hat{E}\hat{A}\hat{A}\hat{I} \\ \hat{A}\hat{I}\hat{D}\hat{A}\hat{A}\hat{I} \end{array} \right\} \hat{I}\hat{I} \left[\div \hat{e}\hat{n}\hat{e}\hat{i} \quad \hat{I}\hat{A}\hat{D}\hat{A}\hat{N}\hat{A}\hat{A} \times \right]$$

В описании замкнутого контура направляющие поверхности в первом и последнем операторах могут совпадать. В описании контуров кроме операторов движения можно использовать операторы ПОВТОР и ЕСЛИ (см. п. 2.12).

Незамкнутый контур, изображенный на рис. 2.95, описывается таким образом:

КОНТУР КОН 1= ИЗ ТЧ1, ХБ

ВПЕРЕД ПР1

ВЛЕВО КР1 2 ПЕРЕСЕЧ

ВЛЕВО ПР1 НА ПР2

КОНКОН

Применим в описании контура (рис. 2.95) условный оператор для выбора обхода верхней или нижней полуокружности в зависимости от знака переменной *P*:

КОНТУР КОН 1= ИЗ ТЧ1, ХБ

ВПЕРЕД ПР1

ЕСЛИ(*P*>0)ТО

ВЛЕВО КР1 2 ПЕРЕСЕЧ

ВЛЕВО ПР1 НА ПР2

ИНАЧЕ

ВПРАВО КР1 2 ПЕРЕСЕЧ

ВПРАВО ПР1 НА ПР2

КОНЕСЛИ

КОНКОН

Замкнутый контур, изображенный на рис. 2.96, может быть задан двумя способами. Обычная форма записи следующая:

КОНТУРКОН2=ИЗТЧ1,ХБ

ВПЕРЕД ПР1

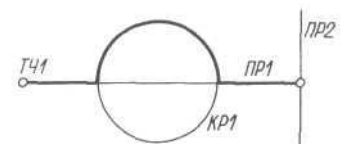


Рис. 2.95

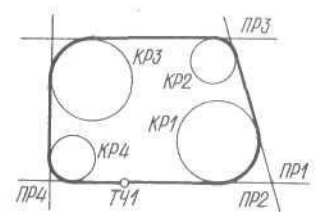


Рис. 2.96

M1=ПРХ
M2 = ПРУ
M3 = M1, M2
K1 =

K= + 1, K1, -1, M1, K1, +1, M3, K1, -1, M2, K1
(или K = K1, -1, M1, K1, M3, K1, -1, M2, K1)

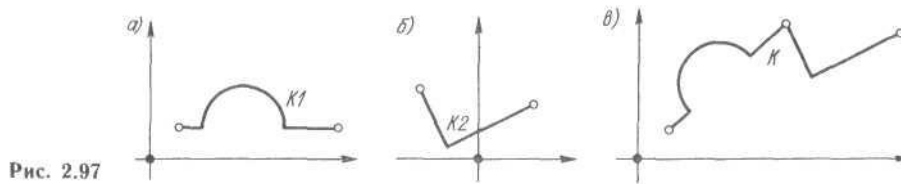


Рис. 2.97

В определениях составных контуров можно использовать операторы ЕСЛИ и ПОВТОР. Применим оператор цикла при построении контура K (рис. 2.100,6), получающегося многократным повторением контура K1 (рис. 2.100, а):

```

КОНТУР K =
  ПОВТОР И = 1,5
    МАТР (ПОВОРОТ, 72*(И -1)), K1
  КОНЦИКЛ
КОНКОН

```

Частным случаем описания составного контура является оператор присваивания: A = Б или КОНТУР A = Б, где A и Б – объекты типа КОНТУР.

Определение табулированной кривой. Табулированная кривая (СПЛАЙН, таблично заданная кривая) – это гладкая кривая, интерполирующая заданный набор точек и состоящая из отрезков прямых и дуг окружностей. Табулированная кривая имеет направление от первой заданной точки к последней (т. е. при построении кривой от точки ТЧ₁ к точке ТЧ_n и от точки ТЧ_n к точке ТЧ₁ результаты совпадать не будут).

Табулированные кривые имеют тип КОНТУР и с ними можно осуществлять любые операции, имеющие смысл для незамкнутых контуров. Так, например, они могут быть компонентами составного контура. Их геометрическое определение в языке Техтран имеет вид:

КОНТУР имя = СПЛАЙН [, прямая₁], точка₁, точка₂, ..., точка_n, [прямая₂]

где КОНТУР – служебное слово, задающее тип определяемого объекта, если он не был объявлен ранее; СПЛАЙН – служебное слово, задающее вид определяемого контура; точка₁, точка₂, ..., точка_n – опорные точки кривой (число точек не ограничено); прямая₁, прямая₂ – имена необязательных прямых, касательных к кривой в начальной и конечной точках.

Как и в определении контура, в определении табулированной кривой можно использовать операторы ПОВТОР и ЕСЛИ. Определение с оператором цикла имеет вид:

```

[КОНТУР] имя = СПЛАЙН, прямая1
  точка1, точка2, ..., точкаk
  ПОВТОР счетчик = нач, кон, шаг
    массив точек (счетчик)
  КОНЦИКЛ
  точкаk+1, точкаk+2, ..., точкаm
  прямая2
КОНКОН

```

Например:
КОНТУР T1=СПЛАЙН, ПР1
 ПОВТОР И = 1,50
 ТЧ(И)
 КОНЦИКЛ
 ПР2
КОНКОН

Определение табулированной кривой с условным оператором имеет вид:

```

[КОНТУР] имя = СПЛАЙН [.прямая1]
  точка1, точка2, ..., точкаk
  ЕСЛИ (логическое выражение) ТО

```

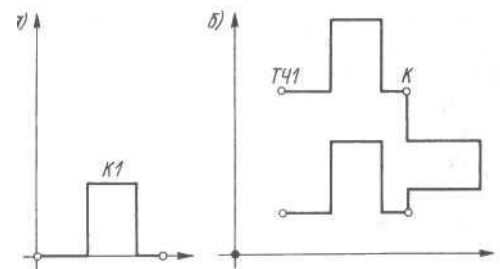


Рис. 2.98

точка_{к+1}, точка_{к+2}, ..., точка_l
 ИНАЧЕ
 точка_{l+1}, точка_{l+2}, ..., точка_m
 КОНЕСЛИ
 точка_{m+1}, точка_{m+2}, ..., точка_n

КОНКОН

Например:

КОНТУР Т2 = СПЛАЙН, ПР1

ЕСЛИ (A >= B) ТО

ТЧ1, ТЧ2, ТЧ3

ИНАЧЕ

ТЧ4, ТЧ5, ТЧ6

КОНЕСЛИ

ПР2

КОНКОН

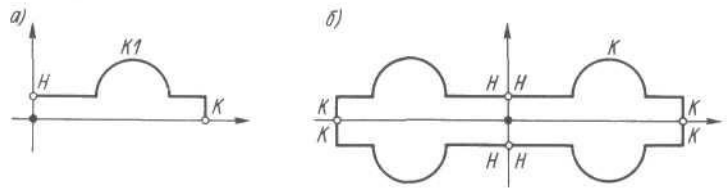


Рис. 2.99

Оператор цикла и условный оператор могут встречаться в определении табулированной кривой в различных комбинациях:

КОНТУР Т3 = СПЛАЙН, ПР1

ТЧ1, ТЧ2, ТЧ3

ЕСЛИ (A >= B) ТО

ПОВТОР И=1, К

ТЧ(И)

КОНЦИКЛ

ИНАЧЕ

ПОВТОР И=1, М

ЕСЛИ (L = 1) ТО

ТК(И)

ИНАЧЕ

ТК(М-И)

КОНЕСЛИ

КОНЦИКЛ

КОНЕСЛИ

ТЧ4, ТЧ5, ТЧ6

ПР2

КОНКОН

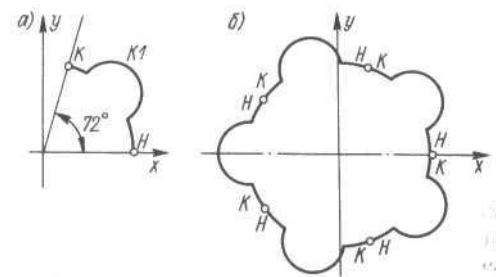


Рис. 2.100

Подход к контуру. Существует два способа подхода к контуру: при помощи оператора ИДИ и при помощи операторов непрерывного движения ВПЕРЕД, НАЗАД, ВЛЕВО, ВПРАВО.

При использовании оператора ИДИ следует указывать направление подхода к контуру командой НАПРВ или НАПРТ (рис. 2.101):

ИЗ ТЧ1

НАПРВ В1

ИДИ ЗА К1

При использовании операторов непрерывного движения инструмент следует вывести на контур с помощью любой команды непрерывного движения с двумя поверхностями. Число пересечений не должно превышать восьми. Если поверхность подхода совпадает с одним из сегментов контура, то инструмент выводится в первую или последнюю точку сегмента.

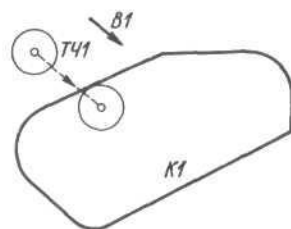


Рис. 2.101

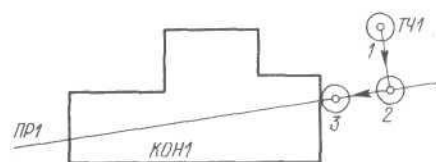


Рис. 2.102

На рис. 2.102 приведен пример подхода к контуру по прямой, заданный следующими командами движения:

ИЗ ТЧ1
 ИДИ НА ПР1
 ВПРАВО ПР1 ДО КОН1

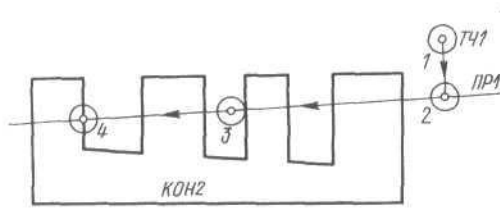


Рис. 2.103

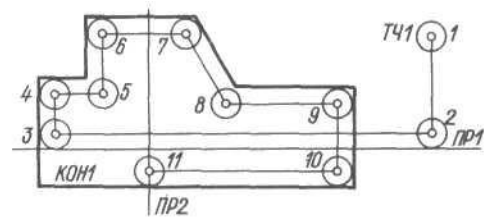


Рис. 2.104

В случае многократных пересечений (рис. 2.103) вывод инструмента из позиции 1 в позицию 3 задается следующим образом:

ИЗ ТЧ1
 ИДИ НА ПР1
 ВПРАВО ПР1 ЗА 4 ПЕРЕСЕЧ КОН2

Перемещение инструмента из позиции 1 в позицию 4 программируется операторами:

ИЗ ТЧ1
 ИДИ НА ПР1
 ВПРАВО ПР1 НА 7 ПЕРЕСЕЧ КОН2

Движение по контуру. Для программирования движения инструмента по контуру можно задать любой из операторов ВПЕРЕД, НАЗАД, ВЛЕВО, ВПРАВО с двумя поверхностями, указав требуемое число пересечений. Ограничивающей поверхностью может быть только прямая или окружность (но не контур и не табулированная кривая).

Например, для контура, показанного на рис. 2.104, перемещение инструмента из позиции 1 в позиции 2 – 11 программируется следующим образом:

ИЗ ТЧ1
 ИДИ ДО ПР1
 ВПРАВО ПР1 ДО 2 ПЕРЕСЕЧ КОШ
 ВПРАВО КОН1 НА 2 ПЕРЕСЕЧ ПР2

Аналогично можно запрограммировать и перемещение инструмента по незамкнутому контуру (рис. 2.105):

ИЗ ТЧ1
 НАПРТ ТЧА
 ИДИ ДО К1
 ВПРАВО К1 НА ПР1

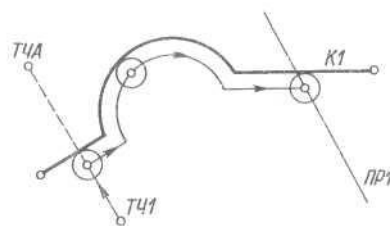


Рис. 2.105

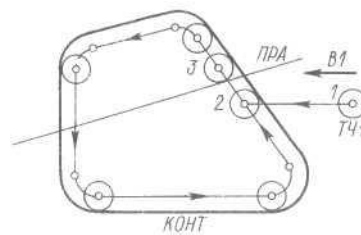


Рис. 2.106

Полный обход всего контура программируется оператором непрерывного движения с одной поверхностью, которой является контур:

$$\left\{ \begin{array}{l} \hat{A}\hat{I}\hat{A}\hat{D}\hat{A}\hat{A} \\ \hat{I}\hat{A}\hat{C}\hat{A}\hat{A} \\ \hat{A}\hat{E}\hat{A}\hat{A}\hat{I} \\ \hat{A}\hat{I}\hat{D}\hat{A}\hat{A}\hat{I} \end{array} \right\} \hat{e}\hat{i}\hat{y}$$

где *имя* – символическое имя контура.

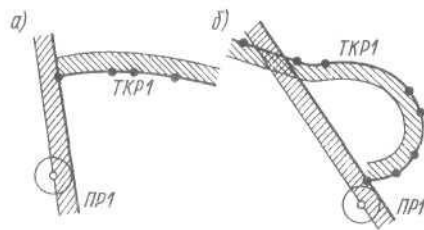


Рис. 2.107

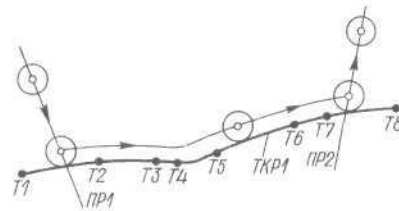


Рис. 2.108

Совершив полный обход контура, инструмент возвращается в положение, в котором он находился перед использованием оператора этого вида.

Например, полный обход контура представленного на рис. 2.106, задается операторами:

ИЗ ТЧ1

НАПРВ В1

ИДИ ЗА КОНТ # поз 1-2

ВПРАВО КОНТ ЗА ПРА # поз. 2-3

ВПЕРЕД КОНТ # поз. 3-2-3

Оператор обхода контура с одной поверхностью допускается употреблять только для замкнутого контура.

Особенности движения по табулированным кривым. Поскольку табулированная кривая является разновидностью контура, то все сказанное выше о командах движения для контуров применимо и к табулированным кривым. Необходимо помнить, что при непрерывном движении по контуру не только сами поверхности, но и их эквидистанты должны иметь требуемое число пересечений. Эквидистанта контура (и табулированной кривой) ограничена нормальными в начальной и конечной точках контура. Например, в случаях, изображенных на рис. 2.107, а, б, невозможен выход в начало табулированной кривой, так как эквидистанты к ПР1 и ТКР1 не пересекаются в начальной точке. Примеры различных способов подхода к табулированной кривой рассмотрены ниже.

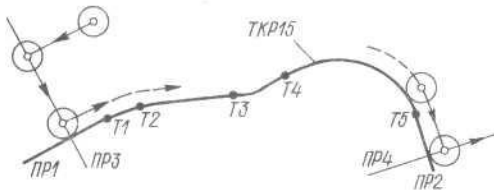


Рис. 2.109

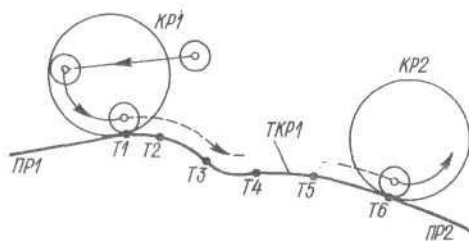


Рис. 2.110

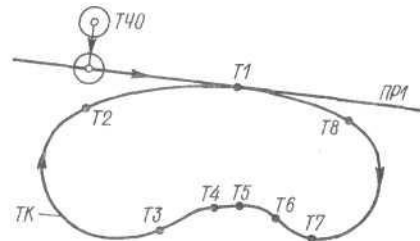


Рис. 2.111

Подход к табулированной кривой и отход от нее *по секущей, не проходящей через граничные точки* (рис. 2.108), является наиболее простым и надежным способом движения по табулированной кривой. Он позволяет исключить начальный и конечный участки сплайна, на которых наиболее вероятно возникновение биений:

КОНТУР ТКР1 = СПЛАЙН, T1, T2, T3, T4, T5, T6, T7, T8

ВПЕРЕД ПР1 ДО ТКР1

ВЛЕВО ТКР1 НА ПР2

ВЛЕВО ПР2...

Подход к табулированной кривой *по касательной* осуществляется с помощью прямых ПР1 и ПР2, входящих в геометрическое определение сплайна и задающих касательные к нему в начальной и конечной точках (рис. 2.109):

КОНТУР ТКР15 = СПЛАЙН, ПР1, T1, T2, T3, T4, T5, ПР2

ИДИ НА ПР3

ВЛЕВО ПР3 ДО ПР1

ВЛЕВО ПР1 КАС ТКР15

ВПЕРЕД ТКР15 КАС ПР2

ВПЕРЕД ПР2 НА ПР4

Выход в начальную точку табулированной кривой *по дуге окружности* (рис. 2.110) производится в следующей последовательности:

КОНТУР ТКР1=СПЛАЙН, ПР1, T1, T2, T3, T4, T5, T6, ПР2

КР1=КАС, ПР1, УБ, Т1, РАДИУС, 20
КР2 = КАС, ПР2, УБ, Т6, РАДИУС, 20

.....
ВЛЕВО КР1 КАС ТКР1
ВПЕРЕД ТКР1 КАС КР2
ВПЕРЕД КР2

При построении *замкнутой табулированной кривой* необходимо в качестве начальной и конечной указывать одну и ту же точку, а также общую касательную, если не допускается излома в этой точке (рис. 2.111):

КОНТУР ТК = СПЛАЙН, ПР1, Т1, Т2, Т3, Т4, Т5, Т6, Т7, Т8, Т1, ПР1
ИЗ ТЧ0
ИДИ НА ПР1
ВЛЕВО ПР1 КАС ТК
ВПЕРЕД ТК

При *многократных пересечениях с табулированной кривой* (рис. 2.112) для выхода из позиции 1 в позицию 4 (через позиции 2–3) используются операторы: ИЗ ТЧ0 ИДИ ДО ПР1 ВЛЕВО ПР1 ДО ТКР ВЛЕВО ТКР НА 4 ПЕРЕСЕЧ ПР1

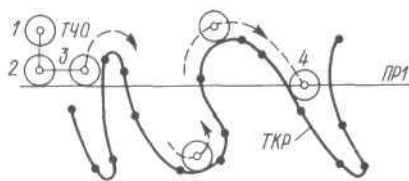


Рис. 2.112

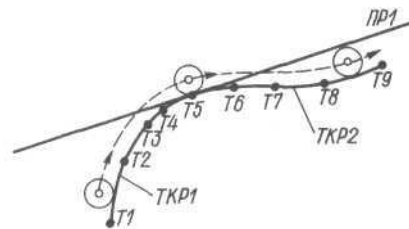


Рис. 2.113

Для *гладкого сопряжения двух табулированных кривых* необходимо задать вырожденное движение по их общей касательной (рис. 2.113): КОНТУР ТКР1=СПЛАЙН, Т1, Т2, Т3, Т4, Т5, ПР1 КОНТУР ТКР2 = СПЛАЙН, ПР1, Т5, Т6, Т7, Т8, Т9 ВПЕРЕД ТКР1 КАС ПР1 ВПЕРЕД ПР1 КАС ТКР2 ВПЕРЕД ТКР2

2.12 Специальные операторы

Управляющие операторы. Эти операторы позволяют изменять порядок выполнения операторов программы, т. е. пропускать (при определенных условиях) или многократно выполнять некоторые ее участки. Такие возможности позволяют составлять на входном языке программы, имеющие сложную логическую структуру. К управляющим операторам относятся операторы НАМЕТКУ, ЕСЛИ и ПОВТОР.

Оператор НАМЕТКУ вызывает безусловную передачу управления на помеченный оператор и имеет следующий формат:

НАМЕТКУ *имя* где *имя* – имя использованной в программе метки.

После этого оператора выполняется не следующий по порядку оператор, а тот, который помечен указанной меткой. Например:

ИЗ ТЧ8
НАМЕТКУ К2
ИДИ НА ПРХ
К2: В ТОЧКУ ТЧА
ПРИРАЩ 20

В приведенном фрагменте программы после оператора ИЗ будет выполнен не оператор ИДИ, а помеченный оператор ВТОЧКУ-

Оператор ЕСЛИ имеет следующий формат:

ЕСЛИ (*логическое выражение*) *оператор*₁ [*; оператор*₂; ...; *оператор*_{*n*}]

где *оператор*₁, *оператор*₂, ... и т. д. – операторы языка Техтран.

В том случае, если логическое выражение, заключенное в скобки, имеет значение ИСТИНА, то последовательно выполняются *оператор*₁, *оператор*₂ и т. д. Если значение логического выражения – ЛОЖЬ, то выполняется оператор, а если он отсутствует, то выполняется оператор, находящийся в следующей строке программы. *Оператор*₁, *оператор*₂, ... и т. д. могут быть только простыми операторами входного языка. Применение составных операторов

типа ПОВТОР, КОНТУР, МАКРО недопустимо. Например:

ЕСЛИ (A>0.5 ИЛИ A<0) НАМЕТКУ М24

Переход на метку М24 выполняется, если значение переменной А больше 0.5 или меньше 0.

Если значения переменной К меньше 10, то будут выполнены операторы присваивания для ТЧ1 и ТЧ2:

ЕСЛИ (К МНШ 10) ТЧ1=0, 0, 5; ТЧ2 = ПРЗ, ПР4

ТЧ6 = ЦЕНТР, КР2

При других значениях переменной К логическое выражение принимает значение ЛОЖЬ, геометрическое определение точки ТЧ1 опускается и выполняются операторы ТЧ2= ... и ТЧ6= ...

Оператор ЕСЛИ–ТО–ИНАЧЕ–КОНЕСЛИ имеет следующий формат:

ЕСЛИ (логическое выражение) ТО

[список операторов₁]

$$\left[\begin{array}{c} \text{ЕІА} \times \text{А} \\ \left[\begin{array}{c} \text{ñïèñîê} \quad \text{ñãðàòîðîâ} \end{array} \right]_2 \end{array} \right]$$

КОНЕСЛИ

Если логическое выражение имеет значение ИСТИНА, то последовательно выполняются операторы списка, расположенного после слова ТО. Если же значение логического выражения – ЛОЖЬ, то выполняются операторы списка, стоящего за словом ИНАЧЕ. Любой из этих списков может быть пустым – в этом случае будет выполняться оператор, следующий за ограничителем КОНЕСЛИ. При отсутствии слова КОНЕСЛИ выдается предупреждение и ошибка корректируется. Например:

1) ЕСЛИ (A>=0) ТО

A = A

ИНАЧЕ

A = - A

КОНЕСЛИ

В результате будет вычислено абсолютное значение А.

2) ЕСЛИ (X>=0) ТО

У = КВКОР (X)

ИНАЧЕ

КОНЕСЛИ

В этом примере отсутствует второй список. Оператор аналогичен оператору ЕСЛИ (X БРВ 0) У = КВКОР (X).

3) ВПЕРЕД ПРХ НА ПРУ

ПР1=ПАРЛЕЛ, ПРХ, А

ЕСЛИ (A>0) ТО

ВЛЕВО ПРУ НА ПР1

ИНАЧЕ

ВПРАВО ПРУ НА ПР1

КОНЕСЛИ

В этом примере выбирается оператор движения ВПРАВО или ВЛЕВО в зависимости от положения прямой ПР1 относительно оси х.

Оператор ПОВТОР предназначен для многократного повторения участка программы, называемого телом цикла. Существуют две модификации оператора ПОВТОР: 1) цикл со счетчиком и 2) цикл ПОКА.

Формат 1-го варианта оператора:

ПОВТОР *перем* = *нач*, *кон* [, *шаг*]

*оператор*₁

*оператор*₂

...

...

*оператор*_п

КОНЦИКЛ [*перем*]

где *перем* – идентификатор вещественной переменной, являющейся управляющей переменной цикла; *нач* – число, переменная или арифметическое выражение, задающие

начальное значение управляющей переменной цикла; *кон* – число, переменная или арифметическое выражение, задающие конечное значение управляющей переменной цикла; *шаг* – число, переменная, или арифметическое выражение, задающие приращение управляющей переменной; по умолчанию приращение равно 1;

оператор₁, *оператор₂*, ..., *оператор_n* – операторы входного языка, составляющие тело цикла.

При первом исполнении цикла *перем* = *нач*, при втором – *перем* = *нач* + *шаг* и т. д. до тех пор, пока удовлетворяется условие: *перем***шаг* ≤ *кон***шаг*. После последнего исполнения цикла управление передается оператору, следующему за оператором КОНЦИКЛ, а управляющая переменная цикла *перем* принимает первое значение, не удовлетворяющее этому условию. Значение управляющей переменной цикла *перем* может быть использовано при вычислениях так же, как значение любой другой вещественной переменной, например:

ПОВТОР И = 1, 10

А (И)=0

КОНЦИКЛ И

В результате выполнения этого цикла первым десяти элементам массива А присваиваются нулевые значения, например:

Т = 6,5

ПОВТОР И =1,7

Т = Т-И

ЕСЛИ (Т МНШ 0) НАМЕТКУ М1

Т = Т + 2.0

КОНЦИКЛ И

М1:Р = Т Т

В процессе выполнения цикла переменная *Т* последовательно принимает следующие значения: 6,5; 5,5; 7,5; 5,5; 7,5; 4,5; 6,5; 2,5; 4,5; -0,5. Затем оператор ЕСЛИ передает управление на оператор с меткой М1, при этом И = 5.

Если начальное значение переменной *Т* принять равным 26,5, то последующие ее значения, вычисленные в цикле, будут равны: 27,5; 27,5; 26,5; 24,5; 21,5; 17,5; 12,5. Затем произойдет нормальный выход из цикла; при этом И = 8, так как это первое значение переменной И, при котором перестает выполняться условие $I \cdot 1^7 \cdot 1$.

Формат 2-го варианта оператора:

ПОВТОР ПОКА (логическое выражение)

оператор₁

оператор₂

⋮

⋮

оператор_m

КОНЦИКЛ

Такая конструкция позволяет повторять выполнение операторов, составляющих тело цикла, до тех пор, пока логическое выражение, находящееся в заголовке цикла, сохраняет значение ИСТИНА. Для нормального завершения такого цикла необходимо, чтобы в теле цикла имелись операторы присваивания, изменяющие значения переменных, которые входят в логическое выражение. Если перед очередным повторением цикла обнаруживается, что логическое выражение имеет значение ЛОЖЬ, то управление передается оператору, следующему за оператором КОНЦИКЛ. Если значение выражения оказывается ложным с самого начала, то операторы, составляющие цикл, не выполняются.

В качестве примера приведем программу вычисления значения корня кубического с использованием метода Ньютона. Полагаем $x_0=1$ и затем вычисляем,

$$x_1 = \frac{2x_0 + \frac{A}{x_0^2}}{3},$$

где *A* – число, из которого необходимо извлечь корень кубический.

Далее полагаем $x_0 = x_1$ и вычисляем новое значение x_2 . Процесс вычислений продолжается до тех пор, пока разница между двумя последовательными значениями *x* не окажется в пределах допустимой ошибки. Например, процесс вычисления корня кубического из числа 4 можно представить следующим образом: $A = 4$; $x_0=1$; $x_1 = (2 + 4/1)/3 = 2$; $x_2 = (4 + 4/4)/3 = 1.667$;

$x_3 = (3.333 + 4/2.778)/3 = 1.591$.

Если продолжить процесс вычислений, то значения x будут приближаться к 1,5874 – корню кубическому из 4. Программа на Техтроне будет иметь вид:

```
ХСТАР = 0
ХНОВ = 1
ПОВТОР ПОКА ( АБС ( ХНОВ-ХСТАР) БЛШ 0.0001)
ХСТАР = ХНОВ
ХНОВ = (2*ХСТАР + 4/ХСТАР**2)/3
КОНЦИКЛ
```

Разрешается использовать вложенные циклы. В этом случае тело внутреннего цикла полностью должно находиться в теле внешнего цикла, т. е. перекрытие циклов не допускается.

Глубина вложений должна быть не больше 10. Рассмотрим случай вложения двух циклов:

```
ВЕЩ А (50,6)
ПОВТОР Н=1,50
ПОВТОР И = 2,6
ХК = И
А (Н,И) = (1.0 + 0,01*ХК)**И
КОНЦИКЛ И
КОНЦИКЛ Н
```

Эта программа вычисляет значения $A(N, I) = (1 + I)^N$ для $I = 0.01; 0.02; 0.03; \dots; 0.06$ и $I = 1, 2, \dots, 50$.

Не допускается применять операторы, лежащие вне тела цикла и передающие управление внутрь цикла. Варианты допустимых передач управления показаны на рис. 2.114.

Использование циклов должно быть согласовано с использованием макросов. Это означает, что цикл, начинающийся (кончающийся) внутри макроса, должен там же и кончаться (начинаться).

Управление печатью. Каждая фаза процессора системы Техтран осуществляет вывод информации на печать. Для управления объемом выводимой информации в языке имеются специальные операторы: ПЕЧТЕКСТ, ПЕЧТР, СЛЕД и ПЧТ1.

Оператор ПЕЧТЕКСТ предназначен для управления распечаткой исходного текста программы на Техтроне (результат работы первой фазы процессора). Оператор имеет следующий формат:

$$\text{ПЕЧТЕКСТ} \left\{ \begin{array}{l} \hat{A}\hat{E}\hat{E} \\ \hat{A}\hat{N}\hat{A} \\ \hat{A}\hat{U}\hat{E}\hat{E} \end{array} \right\}$$

где ВКЛ – распечатка всей вводимой информации кроме расшифровок операторов ВЫЗОВ; ВСЕ – полная распечатка, включая расшифровки операторов ВЫЗОВ; ВЫКЛ – подавление печати вводимой информации.

Отсутствие оператора ПЕЧТЕКСТ эквивалентно применению оператора ПЕЧТЕКСТ ВКЛ. Операторы ДЕТАЛЬ и КОНЕЦ, а также строки, содержащие ошибки, выводятся на печать всегда.

Оператор ПЕЧТР устанавливает режим распечатки траектории движения инструмента и имеет формат:

$$\text{ПЕЧТР} \left\{ \begin{array}{l} \hat{A}\hat{E}\hat{E} \\ \hat{A}\hat{N}\hat{A} \\ \hat{A}\hat{U}\hat{E}\hat{E} \end{array} \right\}$$

При отсутствии в программе оператора ПЕЧТР или при использовании оператора ПЕЧТР ВКЛ на печать выводятся следующие параметры выполняемого движения: данные об инструменте, метод интерполяции и точность линейной аппроксимации, параметры геометрических объектов, образующих контур, координаты начальной точки, координаты точек пересечения эквидистант, координаты опорных точек.

При наличии в программе таблично заданных кривых и контуров формируются сообщения

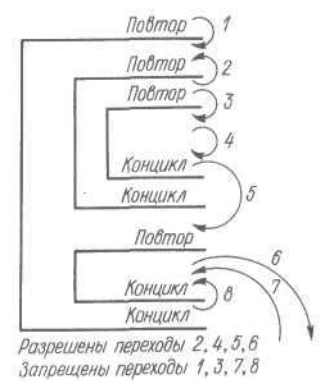


Рис. 2.114

«НАЧАЛО КОНТУРА», «КОНЕЦ КОНТУРА», а также параметры геометрических элементов и координаты точек пересечения этих элементов.

При использовании оператора ПЕЧТР ВСЕ дополнительно выводятся на печать следующие параметры: координаты точек разбиения окружностей на квадранты, промежуточные точки аппроксимации окружности отрезками прямых, сообщение «ДВИЖЕНИЕ ПО КОНТУРУ» и траектория движения по контуру или по таблично заданной кривой.

Использование оператора ПЕЧТР ВЫКЛ приводит к подавлению вывода информации на печать до конца программы или до следующего оператора ПЕЧТР. Исключение составляют строки, содержащие ошибки движения, которые выводятся всегда.

Данные обо всех геометрических переменных выводятся в соответствии с их внутренним представлением (см. п. 2.9).

Оператор СЛЕД выполняет трассировку программы, т. е. печатает результаты выполнения операторов и имеет следующий формат:

$$\text{СЛЕД} \left\{ \begin{array}{l} \hat{A}\hat{E}\hat{E} \\ \hat{A}\hat{N}\hat{A} \\ \hat{A}\hat{U}\hat{E}\hat{E} \end{array} \right\} \left\{ \begin{array}{l} ,1 \\ \mathbf{p} \end{array} \right\}$$

где ВКЛ – включение распечатки результатов выполнения операторов, находящихся вне макроса; ВСЕ – включение распечатки результатов выполнения всех операторов; ВЫКЛ – выключение распечатки результатов выполнения операторов.

Таблица 2.5

Содержание строки	Данные, выводимые на печать
ЕСЛИ (логическое)	ЛОЖЬ или ИСТИНА
НАМЕТКУ	ПЕРЕХОД НА МЕТКУ
ПОВТОР	Минимальное и максимальное значения индекса
КОНЦИКЛ	ПЕРЕХОД НА МЕТКУ
ВЫЗОВ	ВХОД В МАКРО
КОНЕЦ МАКРОСА	ВЫХОД ИЗ МАКРО
Арифметическое выражение	Результат
Геометрическое определение	Имя параметра геометрического объекта
КОНТУР	Параметры геометрических объектов, образующих контур; координаты опорных точек табулированной кривой
Примечание. Перед результатом на печать выводится номер соответствующей строки.	

В режиме «1» печатаются результаты всех операторов, приведенных в табл. 2.5. В режиме «О» печатаются только результаты управляющих операторов. Вся информация выводится в процессе работы второй фазы процессора в формате, приведенном в табл. 2.5.

Оператор ПЧТ1 используется для печати значения некоторой переменной или параметров геометрического объекта (точки, окружности, вектора или прямой) и имеет следующий формат:

ПЧТ1 имя₁, имя₂, имя₃,...

где имя₁, имя₂, ... – имена геометрических объектов или переменных.

Аналогично оператору СЛЕД информация выводится в процессе работы второй фазы процессора в следующем формате:

номер строки имя₁ = внутреннее представление

номер строки имя₂ = внутреннее представление

Оператор СИНОНИМ служит для замены стандартных служебных слов синонимами и имеет следующий формат:

СИНОНИМ (служебное слово₁, синоним₁), %

[(служебное слово₂, синоним₂), %

...

(служебное слово_n, синоним_n)]

В качестве синонимов могут употребляться только имена. Оператор должен быть задан до первого использования синонима. Например, задав оператор СИНОНИМ (ВПЕРЕД, В), (ПЕРЕСЕЧ, П), можно вместо оператора ВПЕРЕД ПР1 ДО 2 ПЕРЕСЕЧ КР1 употреблять в программе оператор В ПР1 ДО 2 П КР1.

Оператор вставки текста «*» позволяет вставить в программу текст, хранящийся во внешней памяти, и имеет следующий формат:

* *имя*

* *литерал*

где *имя* – имя модуля из библиотеки программ; *литерал* – спецификация файла. Например:

*PROG1

*'SY: PROG2'

Конкретный формат оператора зависит от типа используемой ЭВМ.

2.13 Макросы

В языке Техтран можно использовать в программе отдельно подготовленные части исходного текста – макросы. В макросе разрешается применять любые операторы языка, но вместо конкретных значений переменных могут использоваться параметры.

Подключение макроса к программе осуществляется при помощи специального оператора ВЫЗОВ. При этом в тексте программы на месте оператора ВЫЗОВ появляется текст вызываемого макроса (тело), в котором формальные параметры, заданные в теле макроса, заменяются фактическими значениями, указанными в операторе ВЫЗОВ. Допускается определение макросов внутри программы. В этом случае вызов макроса должен производиться из этой же программы.

Структура макросов. Макрос состоит из заголовка, тела и ограничителя. Заголовок макроса представляет собой специальный оператор МАКРО. Тело макроса – произвольная последовательность операторов языка Техтран (кроме операторов ДЕТАЛЬ, КОНЕЦ и ПЕЧТЕКСТ), в которых могут быть использованы параметры макроса. Тело макроса может содержать другие макросы и обращения к ним. Ограничителем макроса является служебное слово КОНМАК.

Параметры макроса. Параметры могут быть операторными и текстовыми. Параметры записываются в произвольном порядке и отделяются друг от друга запятыми.

Операторный параметр представляет собой оператор языка Техтран, заключенный в скобки. При вызове макроса операторные параметры выполняются непосредственно перед телом макроса. В качестве операторных параметров нельзя использовать операторы ДЕТАЛЬ, КОНЕЦ, МАКРО, КОНМАК, ВЫЗОВ, КОНТУР, КОНКОН, ПОВТОР, КОНЦИКЛ, ЕСЛИ. Операторы присваивания, не содержащие разделителей в правой части, в скобки можно не заключать.

Текстовый параметр в отличие от операторного имеет свое имя и значение. Значением текстового параметра является произвольная последовательность символов, не содержащая символа «#». Имя текстового параметра – произвольное имя, допустимое в языке Техтран.

В теле макроса имени текстового параметра должен предшествовать символ макроподстановки «&». При вызове макроса вместо символа «&» и имени параметра в текст будет подставлено его значение. Не разрешается параметризовать часть имени, числа или литерала.

Например, вместо оператора $KP \& KP = 0,0,40$ после текстовой подстановки значения параметра '10' получим: $KP\ 10 = 0,0,40$, что недопустимо.

Правильной является запись: $\&KP = 0,0,40$. При значении параметра 'КРЮ' оператор будет преобразован к виду: $KP10 = 0,0,40$.

Значения текстовых параметров макроса задаются в операторах ВЫЗОВ или МАКРО.

Заголовок макроса. Заголовок представляет собой специальный оператор МАКРО и имеет следующий формат:

МАКРО *имя* [(*список значений*)]

где *имя* – имя макроса (любое имя, допущенное в языке Техтран); *список значений* – список значений текстовых параметров, установленных по умолчанию.

Если значения по умолчанию не задаются, скобки должны быть опущены. Значение по умолчанию используют в том случае, если в операторе ВЫЗОВ значение данного текстового параметра не задано. Элементы списка значений отделяются друг от друга запятыми.

Значение по умолчанию задается одним из следующих способов:

1) '*имя*' = *литерал* (например, 'КР'='КР10' – параметр с именем КР принимает по умолчанию значение, равное КРЮ);

2) '*имя*₁' = & *имя*₂ (например, 'КР1' = &КР2 – параметр с именем КР1 принимает то же значение, что и параметр КР2; параметр КР2 должен быть определен ранее);

3) '*имя*' = *объединение литералов и текстовых параметров* (например, если значение

параметра КР по умолчанию определено следующим образом: 'КР' = 'КР'&М6&АБ'10', то при значениях параметров М6 и АБ'04' и 'В' соответственно, параметр с именем КР примет значение КР04В10).

Примеры заголовков макросов:

МАКРО МТК1

Значений по умолчанию нет.

МАКРО М2 ('ПГ = '1',' П2' = 'ПР'&П1,' ПЗ' = 'КР'&П1&П1)

По умолчанию: П1 = 1; П2 = ПР1; ПЗ = КРП

МАКРО МЗ ('ПГ = 'Г','П2'== 'ПР'&П4&П4,'ПЗ' = 'КР'&П4)

По умолчанию: П1 = 1; П2 = ПР значение П4 значение П4; ПЗ = КР значение П4.

Значение параметра П4 обязательно должно быть определено при вызове макроса. В противном случае возникнет ошибка. Если для параметра П4 нужно задать значение по умолчанию, то это следует сделать до присваивания значений П2 и ПЗ: МАКРО ПЗ ('ПГ = 'Г,' П4' = '2',' П2'='ПР'&П4&П4, % 'ПЗ' = 'КР'&П4)

Вызов макроса. Для выполнения макроподстановки, т. е. вставки в текст исходной программы на Техтроне тела макроса и для замены формальных "параметров фактическими предназначен оператор ВЫЗОВ. Он имеет следующий формат:

$$\text{ВЫЗОВ} \left\{ \begin{array}{l} \text{имя} \\ \text{литерал} \end{array} \right\} [(\text{список параметров})]$$

где *имя* – имя макроса, использованное в операторе МАКРО; *литерал* – спецификация файла, в котором находится текст макроса; *список параметров* – последовательность операторных параметров и значений текстовых параметров, разделенных запятыми. Если параметры в вызове отсутствуют, скобки должны быть опущены.

Имя макроса используется в том случае, когда макрос находится в макробιβлиотеке или определяется в самой программе. Если макрос содержится в отдельном файле, следует использовать литерал – спецификацию этого файла. Формат спецификации файла зависит от типа ЭВМ и операционной системы, например:

ВЫЗОВ М1

ВЫЗОВ 'SY: MACRO 1'

С помощью оператора «:» (двоеточие) можно задавать умолчание для имени макроса. Формат оператора следующий:

: *литерал*₁, *литерал*₂

где *литерал*₁ – строка символов, присоединяемая к имени макроса слева; *литерал*₂ – строка символов, присоединяемая к имени макроса справа.

Например, оператор: 'DK1: [77, 300]', '. ТЕН' позволяет, задавая в программе вызов в формате ВЫЗОВ МАКРО1, передать операционной системе имя файла в виде 'DK1: [77, 300] МАКРО 1. ТЕН'.

Рассмотрим следующий оператор:

ВЫЗОВ М1 (А=1, (ВЕЩ Б = 2), В = 30.5, (ТЧ1 =0,0,0))

В нем заданы четыре операторных параметра. Без скобок можно записывать только параметры, присваивающие вещественные значения.

Если в операторе ВЫЗОВ определено значение текстового параметра, которому присвоено значение по умолчанию в заголовке макроса, то значение по умолчанию игнорируется, например:

ДЕТАЛЬ 'МАКРО'

МАКРО М1 ('ПГ = 'Г,' П2' = & Ш'0')

А = &П1& П2

ТЧ1=А, В

КР1=ТЧ1, В + & П1

ПЧТ1 А, В, ТЧ1, КР1

КОНМАК

ВЫЗОВ М1 ('П1' = '2',' П2' = '25', В = 0)

ВЫЗОВ М1 ('П1' = '2', В = 3)

ВЫЗОВ М1 ('П2' = '35', В=2.5)

ВЫЗОВ М1 (В = 7.4)

КОНЕЦ

Таблица 2.6

Номер вызова макроста	П1	П2	В	А	ТЧ1			КР1		
1	2	25	0	27	27	0	0	27	0	2
2	2	20	3	22	22	3	0	22	3	5
3	1	35	2,5	36	36	2,5	0	36	2,5	3,5
4	1	10	7,4	11	11	7,4	0	11	7,4	8,4

Значения параметров и переменных, выводимые на печать, приведены в табл. 2.6.

Вложенные вызовы. В теле макроста может встретиться оператор ВЫЗОВ с именем другого макроста. При этом происходит вставка тела нового макроста в тело старого. В теле внутреннего макроста нельзя использовать текстовые параметры внешнего макроста, но их можно применять в операторе ВЫЗОВ для задания значений внутренних текстовых параметров, например:

МАКРО М1 ('П1' = ", П2 = ", П3 = ")

ВЫЗОВ М2 ('ПГ' = '&ПГ')

ВЫЗОВ М2 ('ПГ' = '&П2 + &П3')

КОНМАК

МАКРО М2

А = &П1 -1

КОНМАК

ВЫЗОВ М1 ('ПГ' = '10')

ВЫЗОВ М1 ('П2' = 'Г', 'П3' = '2')

Значения параметра и переменной, выводимые на печать (для данного примера), приведены в табл. 2.7.

Локализация переменных в макросте. Для обеспечения полной независимости Макросов от вызывающей программы предусмотрена возможность локализации переменных внутри макросов.

Переменная является локализованной в макросте, если ее значение доступно только внутри этого макроста. В этом случае вне макроста может существовать другая переменная с тем же именем.

Для локализации переменной внутри макроста ее необходимо там явно описать. В противном случае будет использоваться переменная, определенная вне макроста. Если переменная используется впервые, она считается локализованной на текущем уровне вложения макросов.

Приведем пример использования локальных переменных:

А=1

МАКРО М

ВЕЩ А

А = 2

КОНМАК

ВЫЗОВ М

ПЧТ1 А

Поскольку в макросте М имеется явное задание типа переменной А, то она считается локальной в макросте М. Вне макроста ее значение, равное 2, недоступно. Вне тела макроста определена другая переменная А, которой было присвоено значение, равное 1, и именно это значение будет выведено на печать:

А=1

МАКРО М

А = 2

КОНМАК

ВЫЗОВ М

ПЧТ1 А

В этом случае печатается значение А = 2, так как в теле макроста используется переменная, определенная вне его. Расположение тела макроста относительно оператора А=1 здесь не существенно:

МАКРО М

Таблица 2.7

Номер вызова макроста М2	П1	А
1	10	А=10-1
2	+	А=+-1
3	-	А=-1
4	1+2	А=1+2-1
Примечание. В результате 2-го вызова макроста будет выдана ошибка компиляции.		

```

A = 2
КОНМАК
ВЫЗОВ М
ПЧТ1 А

```

В данном примере допущена ошибка, так как значение переменной А не определено. Операторные параметры рассматриваются как операторы, находящиеся внутри макроса.

Переменные, определяемые в операторных параметрах, должны удовлетворять тем же соглашениям, что и переменные тела макроса, например:

```

A = 1
ВЫЗОВ М (ВЕЩ А = 2)
ПЧТ1 А

```

На печать будет выведено А = 1, так как в вызове определена локальная в макросе М переменная.

```

A = 1
ВЫЗОВ М (А = 2)
ПЧТ1 А

```

На печать будет выведено значение А = 2.

Использование меток. Все используемые в макросе метки локализованы внутри этого макроса. Это означает, что передачи управления в макрос и из него запрещены. Разрешены только переходы внутри тела макроса.

2.14 Примеры программ на Техтране

Примеры, приведенные ниже, иллюстрируют возможности процессора САП Техтран. Поэтому в них, как правило, отсутствуют операторы постпроцессора и некоторые вспомогательные перемещения, необходимые с точки зрения технолога.

Пример 1. Ниже приводится текст программы обработки простейшего контура, изображенного на рис. 2.115.

```

ДЕТАЛЬ 'ПРИМЕР 1'
#
# ОПРЕДЕЛЕНИЕ ЭЛЕМЕНТОВ КОНТУРА
#
ТЧ0 = -30, 0
ТЧ1 = 0, 0
ПР1 = ТЧ1, 45
ПР2 = ТЧ1, -45
ТЧ2 = ПР1, ТЧ1, 50, УМ
ТЧ3 = ПР2, ТЧ1, 50, УМ
ПР3 = ТЧ0, ПЕРП, ПР1
КР1 = ТЧ1, ТЧ2, ТЧ3
КР2 = УМ, ПР1, УМ, ПР2, РАДИУС, 10
КР3 = УМ, ПР1, ХМ, ВНУТРИ, КР1, 10
КР4 = УМ, ПР2, ХБ, ВНУТРИ, КР1, 10
#
ИНСТР 6                # ДИАМЕТР ИНСТРУМЕНТА
МЕТОД ЛИНКРУГ          # МЕТОД ИНТЕРПОЛЯЦИИ
ИНСТПР                 # ИНСТРУМЕНТ СПРАВА
#
# ОПИСАНИЕ ТРАЕКТОРИИ ДВИЖЕНИЯ ИНСТРУМЕНТА
#
ИЗ ТЧ0                  # НАЧАЛО ДВИЖЕНИЯ
ИДИ ДО ПР1              # ПОДХОД К КОНТУРУ
ВПРАВО ПР1              # ОБХОД КОНТУРА
ВПЕРЕД КР3 : КР1 : КР4 : ПР2 : %  # СНАРУЖИ ПРОТИВ
    КР2 : ПР1 НА ПР3      # ЧАСОВОЙ СТРЕЛКИ
ВТОЧКУ ТЧ0              # ВОЗВРАТ В ИСХ. ТЧ.
КОНЕЦ                  # КОНЕЦ ПРОГРАММЫ

```

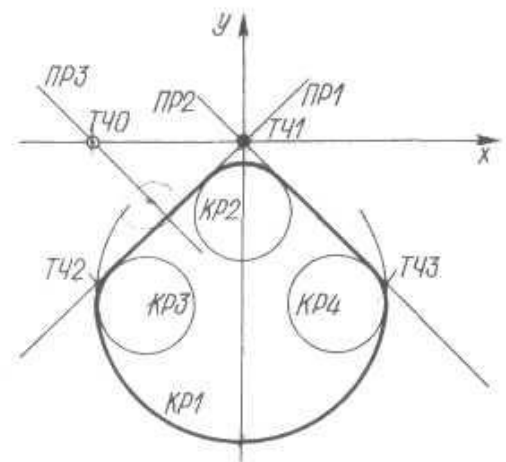


Рис. 2.115