

**Федеральное агентство по образованию**  
Белгородский государственный технологический университет  
им. В. Г. Шухова

**П. А. Трубаев**

**Автоматизированное  
проектирование  
энерготехнологического  
оборудования**

**Учебное пособие**



Белгород  
2005

Федеральное агентство по образованию  
Белгородский государственный технологический университет  
им. В. Г. Шухова

П. А. Трубаев

## Автоматизированное проектирование энерготехнологического оборудования

Утверждено советом университета  
в качестве учебного пособия для студентов  
специальности 100800 – Энергетика теплотехнологии

Белгород  
2005

УДК 658.5 / 5.011.56 03

ББК 34.42

Т 77

Рецензенты:

д-р техн. наук, профессор П. В. Беседин

д-р техн. наук, профессор В. А. Кузнецов

канд. техн. наук П. В. Журавлев

**Трубаев, П. А.**

Т77

Автоматизированное проектирование энерготехнологического оборудования: учеб. пособие / П. А. Трубаев. – Белгород: Изд-во БГТУ им. В. Г. Шухова, 2005. – 146 с.

Рассмотрены основы моделирования высокотемпературных и теплоэнергетических установок, основы автоматизированного проектирования и автоматизации чертежных работ. Описаны работа в системе AutoCAD и создание чертежей с помощью языка программирования AutoLISP. Приводятся примеры автоматизированного проектирования элементов теплоэнергетического оборудования.

Ил. 43. Табл. 42. Библиогр.: 12 назв.

УДК 658.5 / 5.011.56 03

ББК 34.42

© Белгородский государственный  
технологический университет  
(БГТУ) им. В. Г. Шухова, 2005

© П. А. Трубаев, 2005

## Введение

Компьютеры стали неотъемлемой частью жизни и профессиональной деятельности человека. Качественное развитие и удешевление вычислительной техники, произошедшее за последние 20 лет, позволило значительно расширить сферу их применения. В настоящее время можно говорить о массовом внедрении информационных технологий в производство, науку и быт. Но при этом наблюдается парадоксальное явление – современная вычислительная мощность компьютеров используется на доли процента, компьютер на производстве решает задачи подготовки документации и проведения бухгалтерских расчетов, а для решения инженерных задач часто вообще не используется.

Это вызвано тем, что самостоятельно компьютер задачи не решает, а лишь реализует заложенные в нем инструкции. Поэтому большое значение в информационных технологиях имеет качество и адекватность реализованных алгоритмов и методик. Ведь, как сказал писатель А. Хейли, «компьютер ошибаться не может, но ошибаются люди, закладывая в него программы». В настоящее время сформировались две обособленные группы людей – практики в определенной предметной области, которым необходимо информатизировать свою деятельность, но которые не имеют знаний и навыков в разработке новых и применении стандартных программ, и компьютерщики с информационной подготовкой, не являющиеся специалистами в предметной области применения компьютеров. Поэтому на практике часто используется разделение процесса создания информационных продуктов на изолированные этапы и привлечение специалистов из разных областей – инженеров (предметников) и программистов, когда инженер ставит задачу, а программист ее решает. Но, как отмечает А. А. Самарский<sup>1</sup>, такой подход приводит к недостаточной эффективности получаемых информационных систем. Рассмотрим причины этого.

Н. С. Бахвалов<sup>2</sup> считает, что «правильное формулирование задачи – это научная проблема, не менее сложная, чем само решение задачи». Чтобы разработать техническое задание на создание информационного продукта необходимо произвести формализацию и обобщение задачи с учетом методов ее реализации на компьютере, определить совокупность исходных данных, учесть все возможные варианты расчета и ограничения, выбрать способы получения данных о свойствах веществ, то есть фактически разработать алгоритм реше-

---

<sup>1</sup> Самарский Александр Андреевич (р. 1919) – академик АН СССР и РАН, д-р техн. наук, профессор, член Президиума РАН, заведующий кафедрами вычислительных методов МГУ и математического моделирования МФТИ. Герой Социалистического Труда, лауреат Ленинской, Государственной и Ломоносовской премий, награжден семью орденами. Область научной деятельности – математическая физика и вычислительная математика. Один из основоположников отечественного математического моделирования. Подготовил 40 докторов наук, автор более 30 монографий. Организатор института математического моделирования РАН, журнала «Математическое моделирование».

<sup>2</sup> Бахвалов Николай Сергеевич (р. 1934) – академик РАН, д-р техн. наук, профессор, заведующий кафедрой вычислительной математики МГУ. Лауреат Государственной премии СССР, награжден орденом «Знак Почета». Область научной деятельности – вычислительная и прикладная математика. Подготовил восемь докторов наук, автор пяти учебников и монографий. Создал специальность «Прикладная математика».

ния задачи. Это кроме фундаментальных знаний в предметной области требует определенной математической и информационной подготовки, которой практики обычно не обладают. Поэтому вместо постановки задачи они могут выдвинуть лишь требования к результатам, которые им бы хотелось получить.

С другой стороны, для сложных объектов с большим числом взаимовлияющих связей, которыми характеризуется современное производство, большая часть особенностей конкретной задачи проявляется только на этапе ее разработки. Поэтому, как отмечает В. С. Зарубин<sup>1</sup>, в современной практике нет четкого разграничения между постановкой задачи, разработкой алгоритма ее решения и реализацией на компьютере, и задача может корректироваться на любом этапе ее решения, в том числе и на последнем – при проверке адекватности работы информационной системы. Разработчики-программисты, не являющиеся специалистами в предметной области, формулировку задачи сомнению не подвергают и не способны заметить возникающие противоречия и возможные способы совершенствования реализации задачи, направляя главные силы на эффективную реализацию численных методов и программного кода.

Рассмотренное положение, как отмечает В. С. Зарубин, приводит к необходимости для разработчика информационных систем сочетать глубокие знания в предметной области, вычислительной математике и информационных технологиях. Следовательно, необходима отдельная подготовка специалистов, способных разрабатывать информационные системы для технологических процессов, основываясь на фундаментальных знаниях в предметной области.

Задачей учебного пособия является изучение методологии автоматизации конструкторских работ при проектировании энерготехнологического оборудования, то есть применение информационных технологий в профессиональной деятельности инженеров теплоэнергетиков и теплотехников. В первой главе рассматриваются теоретические основы моделирования высокотемпературных и теплоэнергетических установок, автоматизированного проектирования, автоматизации чертежных работ. Во второй и третьей главах на основе систем AutoCAD и VisualLISP описываются методы создания чертежей на компьютере и автоматизации черчения. В заключительной, четвертой главе на примерах рассматриваются некоторые методы и способы решения задач при автоматизации проектирования.

---

<sup>1</sup> **Зарубин Владимир Степанович** (р. 1933) – д-р техн. наук, профессор, заведующий кафедрой прикладной математики МВТУ им. Н. Э. Баумана. Область научной деятельности – термомеханика материалов. Автор более 20 учебных пособий и учебников, серии монографий. Является соредактором серии учебников «Математика в техническом университете».

# 1. ОСНОВЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

## 1.1. ЭТАПЫ ПРОЕКТИРОВАНИЯ

*Проектированием* называется разработка описания, достаточного для создания технического объекта, выполняемая на основе заданных требований к объекту. *Проект* – это совокупность документов, удовлетворяющих нормативным требованиям, в которых представлен результат проектирования.

В качестве исходных требований задаются производительность, параметры получаемой продукции, состав и свойства исходных материалов, производственная технология, различные ограничения на функционирование объекта. Результатом проектирования является описание, по которому можно создать объект. Это может быть комплект чертежей, текстовое описание процесса создания объекта, перечень стандартного оборудования и схема его соединения.

Проектирование может включать следующие задачи, реализуемые отдельно или в сочетании друг с другом:

1) структурный синтез – создание технологической схемы объекта, составленной из единичных существующих элементов с известными режимами работы, взаимодействующих между собой;

2) конструктивный расчет – определение геометрических характеристик объекта по необходимым параметрам его работы;

3) поверочный расчет (анализ) – определение параметров работы технологической схемы, объекта или отдельных элементов при известных конструктивных параметрах.

Например, при проектировании высокотемпературной печи при структурном синтезе выбирается схема технологической линии, включающая установки для термической подготовки сырья, обжига сырья и утилизации тепла получаемых продуктов. Отдельные установки могут быть выбраны из перечня готового оборудования или их размеры определяются на этапе конструктивного расчета, при этом учитываются характеристики других элементов технологической линии. При оценке различных вариантов решений производится определение характеристик работы текущей схемы или конструкции, то есть поверочный расчет.

Процесс проектирования разделяется на несколько последовательных этапов или стадий.

По виду получаемого описания объекта можно выделить четыре этапа.

1. Разработка *технического задания*, в котором излагаются все требования к объекту, которые необходимо учесть при проектировании. Разработка технического задания может включать предпроектные исследования по выбору или созданию технологий

следования по выбору или созданию технологии получения требуемого продукта, изучение используемых материалов.

2. Разработка *эскизного проекта*, где определяется принципиальная схема объекта и его основные параметры. На этой стадии для получения требуемой информации могут быть изготовлены макетные установки, на которых исследуется работа объекта.

3. Разработка *технического проекта*, в котором окончательно определяются все необходимые конструктивные параметры объекта.

4. Разработка *рабочего проекта*, в котором по техническому проекту формируется комплект выходной документации, передаваемый заказчику.

5. Изготовление и доводка опытного образца, когда в процессе промышленного изготовления и эксплуатации объекта производится окончательное уточнение проекта перед серийным производством.

По характеру выполняющихся работ различают следующие стадии проектирования.

1. *Научно-исследовательские работы* (НИР) – проводится поиск способов функционирования объекта, получают данные, необходимые для проектирования, но отсутствующие в справочной и научной литературе.

2. *Опытно-конструкторские работы* (ОКР) – при их осуществлении производится проработка с целью получения конструктивных параметров объекта.

3. *Рабочее проектирование* – прорабатываются схемные, конструкторские и технологические решения, обеспечивающие полученные конструктивные параметры.

Связь стадий и этапов проектирования представлена на рис. 1.

	Этапы	Стадии
Проектирование ↓	Техническое задание	НИР
	Эскизный проект	
	Технический проект	ОКР
	Рабочий проект	
	Изготовление и доводка опытного образца	Рабочее проектирование

**Рис. 1. Стадии проектирования**

Проектирование состоит из получения ряда *проектных решений*, являющихся описанием отдельных узлов или частей объекта. Каждая стадия проектирования (эскизный проект, технический проект и т. д.) состоит из получения одного или нескольких проектных решений. Действия, необходимые для получения проектного решения называют *проектной процедурой*. Примером проектных процедур являются получение проекта отдельного узла или элемента, выбор оборудования из

базы данных. В разных проектных процедурах можно выделить ряд одинаковых действий, например изображение типового графического элемента, расчет теплового баланса или теплообмена для однотипных элементов, компоновка оборудования, проведение экономического расчета и т. д. Стандартные действия, входящие в различные проектные процедуры, называются *проектными операциями*. Формализация проектных операций позволяет значительно повысить эффективность и снизить трудоемкость проектирования.

Стадии проектирования прорабатываются итерационно, то есть после получения проектного решения оно анализируется и, если обнаружена невозможность соблюдения требований технического задания или неэффективность полученного решения, осуществляется возврат на предыдущие проектные процедуры или этапы проектирования для уточнения и корректирования данных.

## 1.2. АВТОМАТИЗАЦИЯ ПРОЕКТИРОВАНИЯ

Развитие вычислительной техники позволило автоматизировать многие виды работ, в том числе и проектирование. По степени автоматизации различают следующие виды проектирования.

*Неавтоматизированное проектирование* включает использование компьютера только для выполнения проектных операций. При этом все работы могут быть выполнены на компьютере, но при этом используются отдельные программы, не имеющие связи между собой. Например, в одной программе производится расчет, в другой – построение чертежа по результатам этого расчета.

*Частично автоматизированное проектирование* включает выполнение на компьютере проектных решений (как всех, так и их части) без организации автоматического выполнения последовательности проектных решений и без обмена данных между системами (программами), выполняющими проектные решения. В этом случае на компьютере полностью выполняется часть проекта, например выбор технологической схемы или расчет отдельного узла; человек при этом задает исходные данные, а программа проводит все необходимые расчеты и по ним формирует необходимую выходную документацию.

*Автоматизированное проектирование* включает полное выполнение проекта на компьютере. При этом пользователь задает исходные данные и получает сформированные документы рабочего проекта. Участие человека в проектировании заключается в том, что на промежуточных этапах ему сообщаются полученные результаты и запрашивается направление дальнейшей работы, например в виде выбора одного варианта из нескольких. *Автоматическим* называется проектирование, все стадии которого выполняются без участия человека.

Автоматизация позволяет снизить сроки проектирования, стоимость проекта, увеличить качество расчетов и выходной документации. С помощью систем автоматизированного проектирования (САПР) возможно исследование многих вариантов с целью выбора наилучшего, тогда как при автоматизированном проектировании в лучшем случае сравниваются 2–3 варианта. Но эффективность систем автоматизированного проектирования зависит от качества реализованных процедур проектных решений и математического обеспечения САПР. Автоматизация проектирования требует в первую очередь создания математических моделей установок, применения оптимизационных расчетов, формализации методов оценки и выбора проектных решений, в том числе и на основе экспертных систем.

### 1.3. СОСТАВ САПР

САПР включает следующие составные части:

- математическое обеспечение;
- программное обеспечение;
- информационное обеспечение;
- лингвистическое обеспечение;
- методическое обеспечение;
- техническое обеспечение;
- организационное обеспечение.

*Математическое обеспечение* включает математические модели объектов проектирования, численные методы решения и оптимизации этих моделей и алгоритмы выполнения проектных процедур (то есть последовательность получения необходимых решений). Это самая важная и сложная часть САПР энерготехнологического оборудования, от которой зависит эффективность получаемых проектов, а значит, и эффективность системы автоматизированного проектирования.

Обычно для каждой системы математическое обеспечение разрабатывается индивидуально. Оно основывается на математических моделях процессов, происходящих в проектируемом объекте и на имеющихся нормативных методиках расчета. Поэтому разработка математического обеспечения может быть выполнена только специалистом в области, относящейся к проектируемому объекту, а не инженером-автоматчиком или программистом.

*Программное обеспечение* (ПО) включает совокупность программ, выполняющих проектные процедуры. Оно делится на три группы:

- общесистемное ПО, организующее работу компьютера и оборудования (операционные системы, драйверы и т.д.);

– базовое ПО, состоящее из стандартных программ или программ, решающих стандартные задачи (например, создание графических изображений);

– прикладное ПО, реализующее математическое обеспечение САПР.

*Информационное обеспечение* включает данные, используемые в проектировании. Это может быть каталог оборудования, свойства веществ и материалов, экономические данные и т.д. Однотипные данные организуются в виде банков данных, включающих:

– базу данных, то есть набор структурированных данных, записанных в единой форме;

– систему управления базой данных (СУБД), то есть программу для создания и изменения базы данных и получения информации из нее.

*Лингвистическое обеспечение* определяет способ общения пользователей и САПР. Оно может быть реализовано в следующих формах:

– диалоговый режим, когда пользователь вводит исходные данные и управляет процессом проектирования в виде ответа на ряд вопросов, выводимых системой;

– пакетный режим, реализуемый с помощью специальных программных языков. Они с помощью специальной формы записи позволяют определять порядок выполнения проектных процедур, создавать схемы объекта и его отдельных узлов, дополнять математическое обеспечение системы. В этом случае сначала готовится текст программы, который затем обрабатывается системой с выполнением всех заданных в программе действий.

*Методическое обеспечение* включает документацию по эксплуатации САПР.

*Техническое обеспечение* – это совокупность технических средств, на которых функционирует САПР. Оно обычно состоит из компьютеров, на жестких дисках которых хранятся программное и информационное обеспечение. САПР может реализовываться на базе одного компьютера или сети из нескольких компьютеров, каждый из которых выполняет отдельные действия. Техническое обеспечение также включает:

– средства вывода текущей информации (графические дисплеи);

– средства ввода информации и исходных данных (клавиатура, манипулятор «мышь», дигитайзер, световое перо);

– средства вывода информации (принтеры, плоттеры, графопостроители);

– средства удаленного хранения информации.

*Организационное обеспечение* – документация, регламентирующая работу проектной организации, использующей САПР.

## 1.4. АВТОМАТИЗАЦИЯ ЧЕРТЕЖНЫХ РАБОТ

Составной частью САПР в большинстве случаев является система получения чертежей и технической документации. Для этого используются специальные графические редакторы, ориентированные на чертежные работы. В большинстве случаев чертежные работы технических объектов выполняются в программе AutoCAD, поэтому в дальнейшем примеры будут рассмотрены именно для этой системы. Изготовление чертежей на компьютере позволяет сократить сроки и трудоемкость проектирования, повысить качество и точность проектов.

Трудоемкость создания индивидуального чертежа на компьютере обычно в 1,5–2 раза выше, чем его выполнение с помощью обычных чертежных инструментов. Повышение эффективности работы может быть достигнуто в следующих случаях:

- изображение с помощью компьютера отдельных трудоемких элементов, таких, как текст, штриховка;
- автоматическая простановка размеров;
- повторение в чертеже одинаковых фрагментов, которые можно скопировать;
- использование уже готовых фрагментов из других чертежей;
- при редактировании чертежа, так как отпадает необходимость перерисовывать неизменяющиеся части.

Значительное повышение эффективности работы в десятки и сотни раз достигается при использовании средств автоматизации построения чертежей. Автоматизация чертежных работ может выполняться путем параметризации или программирования.

Параметризация – это создание чертежей, в которых размеры и взаимное расположение элементов описываются символьными параметрами. Параметризированный чертеж семейства однотипных деталей называется комплексной деталью. Комплексную деталь затем можно вставлять в чертежи, задавая ее конкретные размеры.

При программировании построение чертежа выполняется не в графическом режиме, а описывается с помощью программы на специальном языке. Программирование также позволяет создавать семейство однотипных чертежей, но в отличие от параметризованных чертежей оно обладает более широкими возможностями по созданию различных вариантов конструкции.

В соответствии с уровнем автоматизации черчения можно выделить три категории специалистов, использующих системы САПР:

- пользователи, работающие с уже готовыми САПР в диалоговом режиме и использующие их только для автоматического получения типовых чертежей;

– конструкторы, подготавливающие свои чертежи на ЭВМ с использованием программ компьютерного черчения (конструкторских САПР) или графических редакторов;

– разработчики, создающие средства для получения чертежей, которые впоследствии будут использовать другие конструкторы. Задача разработчика заключается в создании средств автоматической генерации чертежа по заданным технологическим параметрам или размерам или создание графической базы данных; при этом может применяться как параметризация, так и программирование.

Очевидно, что пользователям не обязательно иметь представления о методах создания чертежей на компьютере. Конструкторы должны иметь хорошее знание применяемого графического редактора. Разработчики кроме использования графических редакторов должны также владеть программированием, методами автоматизации черчения и создания САПР.

## 1.5. ПАРАМЕТРИЗАЦИЯ ЧЕРТЕЖЕЙ И КОМПЛЕКСНЫЕ ДЕТАЛИ

*Параметризацией* называется разработка правил для определения размеров детали. Параметризованный чертеж, в котором размеры заданы в виде символьных параметров и который включает все возможные варианты исполнения семейства однотипных деталей, называется *комплексной деталью*.

Для классификации деталей используют следующие признаки, которые перечислены в соответствии с их важностью:

- геометрическая форма (например, тела вращения или не тела вращения);
- параметрический признак (например, отношение диаметра к длине);
- форма детали или ее наименование (например, вал коленчатый, колесо зубчатое). Согласно ЕСКД, каждой детали может быть присвоен шестизначный номер (классификатор), структура которого представлена на рис. 2. Обычно параметризуются чертежи семейства деталей, имеющих единый классификатор.

Признак	Геометрическая форма	Параметрические признаки, функции детали, ее наименование			
		Подкласс	Группа	Подгруппа	Вид
Наименование иерархического уровня	Класс				
Количество цифр в классификаторе	2	1	1	1	1

Рис. 2. Структура классификатора

Комплексная деталь создается как обычный чертеж, но при этом размеры описываются по правилам, определенным в системе, используемой для параметризации. Например, размеры могут быть заданы в виде математических формул или в виде переменных. Если у детали может быть несколько вариантов исполнения, наличие и взаимное расположение отдельных элементов в комплексной детали задаются с помощью специальных параметров или ставятся в зависимость от размеров. Одним из способов описания геометрически изменяющейся детали является добавление в комплексную деталь всех возможных элементов. Затем при вставке детали в чертеж все размеры ненужных элементов задаются равными нулю, и элементы не выводятся.

Для каждой комплексной детали создается специальный вид, содержащий эскиз детали (слайд) и перечень размеров и параметров, которые необходимо задать. Если у детали несколько вариантов исполнения, слайд и список размеров создаются для каждого варианта.

Вставка комплексной детали в чертеж заключается в выборе необходимого вида и задании требуемых размеров и параметров. Для стандартизированных деталей размеры в комплексной детали задаются таблично, и пользователь выбирает нужный тип из списка.

### **Классификация параметризованных размеров**

Чертеж комплексной детали создается на основе чертежей отдельных деталей, табличных значений размеров для стандартизированных деталей и зависимостей между размерами детали и взаимным расположением элементов. Поэтому первой задачей при параметризации является установление способов определения всех размеров и параметров, необходимых для создания чертежа.

Комплексная деталь включает следующие виды параметров:

- геометрические размеры;
- параметры наличия элемента на чертеже или выбора одного элемента из нескольких;
- параметры взаимного расположения элементов, например точка расположения элемента, расстояние между элементами, условия параллельности, пересечения и т. п.

Геометрические размеры параметризованной детали классифицируются следующим образом:

1. Параметризованные размеры, меняющиеся в зависимости от варианта детали. Они могут быть следующего вида:

- а) определяющие – устанавливают вариант детали из семейства однотипных деталей. Например, это диаметр болта или фланца;
- б) зависимые – выбираются из таблицы значений параметров или рассчитываются по определяющим размерам;

в) полузависимые – для каждого варианта могут меняться в определенном диапазоне, конкретное значение из диапазона выбирается при вставке детали в чертеж. Например, это размеры шпонок, проточек, длина резьбы и др.;

г) независимые – могут задаваться произвольно и для разных вариантов могут быть как одинаковыми, так и разными.

2. Непараметризованные параметры, не зависящие от варианта детали:

а) конструктивные размеры, остающиеся постоянными для всех типов деталей;

б) размерные, определяющие местоположение, длину, отступы для размерных линий и другого текста;

в) свободные, определяющие расположение таблиц, разрезов, видов и так далее, и не связанные с конструкцией детали.

Параметризованный чертеж содержит большое количество размеров. Поэтому символьное обозначение размеров систематизируют следующим образом:

а) для чертежа одной детали

*Тип размера – Номер элемента;*

б) для чертежа из нескольких деталей

*Краткое обозначение (номер) детали – Тип размера – Номер элемента.*

Тип размера обычно обозначается следующими символами:

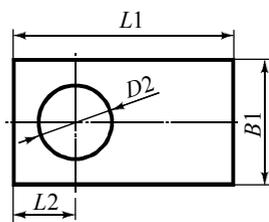
<i>L</i> – длина;	<i>R</i> – радиус;	<i>U</i> – угол;
<i>B</i> – ширина;	<i>D</i> – диаметр;	<i>C</i> – фаска.

Все элементы детали нумеруют, и размеры одного элемента обозначаются одинаковым индексом. Например, на детали (рис. 3) можно выделить два элемента – пластину (элемент № 1) и отверстие (элемент № 2), соответственно устанавливается обозначение размеров. Если бы деталь, изображенная на рис. 3, входила в качестве составляющей в сложный чертеж, впереди размеров также добавлялось бы краткое обозначение детали или ее номер, например,

*BarL1, BarB1, BarL2, BarD2*

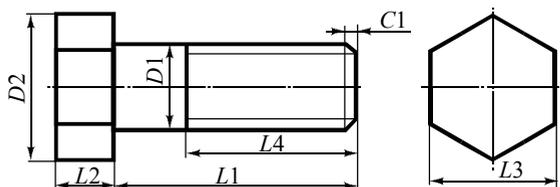
или

*Det1\_L1, Det1\_B1, Det1\_L2, Det1\_D2.*



**Рис. 3. Систематизация параметризованных размеров**

На рис. 4 представлен пример комплексной детали с параметризованными размерами, а в табл. 1 – стандартные размеры этой детали.



**Рис. 4. Параметризованная деталь**

*Таблица 1*

**Таблица параметров детали**

Тип	Размеры						
	Определяющие		Зависимые			Полу-зависимые	Независимые
	Диаметр $D1$	Длина $L1$	Длина $L2$	Длина $L3$	Диаметр $D2$	Длина $L4$	Фаска $C1$
Болт М42×110	42	110	26	55	$\frac{L3}{\cos 30^\circ}$	Определяется конструктором	45°
Болт М30×90	30	90	19	46			45°
Болт М24×80	24	80	15	36			45°

В комплексную деталь включают также информацию о масштабе и месте ее вставки в чертеж. Для определения места вставки на детали задается точка привязки, затем на чертеже, куда помещается деталь, указывается расположение этой точки.

### Средства создания комплексных деталей

Средства создания комплексных деталей в наиболее распространенную систему компьютерного черчения AutoCAD не входят и реализуются в многочисленных дополнительных программных пакетах.

Использование комплексной детали включает два этапа:

- создание комплексной детали (выполняется разработчиком);

– вставка комплексной детали в чертеж с заданием конкретных размеров (выполняется пользователем или конструктором).

Создание комплексной детали всегда основано на использовании уже готового чертежа детали. Рассмотрим четыре основных метода, применяющихся в системах параметризации. Первый из описанных ниже подходов реализуется в системе GLisp, второй – в системе ACADence-p, третий – в системах ACADP и p-Design, четвертый – в системе p-Design.

**Метод 1.** Создается чертеж-прототип, в котором на размерных линиях вместо числовых размеров вводятся символьные обозначения для определяющих параметров и формулы для зависимых параметров, автоматически генерируется программа на языке программирования AutoLISP. Использование комплексной детали заключается в запуске программы, которая запрашивает необходимые размеры и по ним создает чертеж.

**Метод 2.** Задается список параметров, в виде формул определяется способ расчета зависимых параметров от определяющих и производится сопоставление параметров с размерами имеющегося чертежа-прототипа. При использовании модели система запрашивает значения необходимых параметров, загружает чертеж-прототип и модифицирует его в соответствии с заданными параметрами.

**Метод 3.** В AutoCAD создается чертеж в обычной форме, но в процессе работы специальными способами указываются размеры, которые будут параметризованы. Система записывает все действия в называемый сценарий работы, а затем при использовании комплексной детали повторяет этот сценарий, запрашивая необходимые размеры.

**Метод 4.** Изменяются размеры готового чертежа, для этого он загружается в AutoCAD, указываются размеры, которые нужно изменить, и их новые значения. Это можно реализовать и с помощью команды AutoCAD STRETCH (РАСТЯНИ), но в системах параметризации, реализующих такой подход, дополнительно изменяются диаметры, учитываются симметричные размеры (то есть между двумя точками, симметричными относительно заданной оси) и можно задавать часть чертежа, которая должна меняться.

## 1.6. ПРОГРАММИРОВАНИЕ ЧЕРТЕЖНЫХ РАБОТ

Программой называется последовательность действий, записанная в виде текста, которая может быть выполнена на компьютере. При программировании чертежных работ разрабатывается специальная программа, в которой в текстовом виде записаны действия, необходимые для создания чертежа. Программирование намного сложнее непосред-

ственного изображения чертежа в графическом редакторе или разработки параметризированной детали, так как написание программы не сопровождается графическими построениями, а результаты работы программы становятся видны только после ее запуска.

В AutoCAD существуют два способа программирования действий:

- сценарии (пакетные файлы);
- языки программирования.

### Сценарии и пакетные файлы

*Сценарием* называется сохраненная последовательность работы, которая может быть в дальнейшем повторно воспроизведена. Файл, в котором записан сценарий работы, называется пакетным.

В AutoCAD пакетный файл представляет собой текстовый файл с расширением «.scr», в котором записываются выполняющиеся команды и их параметры. Для написания пакетного файла необходимо хорошо знать последовательность информации, вводимой в AutoCAD, так как последовательность символов должна строго соответствовать информации, вводимой в командную строку. Например, стоящие рядом два пробела в пакетном файле эквиваленты нажатию двух пробелов в AutoCAD. Поэтому пакетный файл не должен содержать лишних пробелов или пустых строк.

Пакетный файл является текстовым, поэтому создается в редакторе NotePAD, встроенных редакторах файловых менеджеров FAR или Norton Commander. При создании в редакторе Word файл необходимо записывать командой меню «Сохранить как» с выбором формата «Только текст (\*.txt)». Пример пакетного файла, задающего формат чертежа A2 и выводящего рамку (символ «\_» обозначает нажатие клавиши «Пробел», символ «↵» – клавиши «Enter», после ввода второй строчки клавиша «Enter» не нажимается) :

```
LIMITS _0,0 _210,297 ↵  
LINE _20,4 _292,5 _292,205 _20,205 _Close
```

В AutoCAD есть два способа запуска пакетных файлов.

#### 1. Команда SCRIPT (ПАКЕТ).

Режим работы команды SCRIPT определяется значением системной переменной FILEDIA. Если ее значение равно 1, что является установкой по умолчанию, то после вызова команды выводится диалоговое окно для выбора файла. При ее значении, равном 0, запрос имени файла производится в командном режиме.

#### 2. Выполнение пакетного файла при запуске AutoCAD.

При загрузке AutoCAD после переключателя «/b» указывается имя автоматически выполняемого пакетного файла (без расширения scr), например команда для запуска AutoCAD с выполнением файла ramka.scr выглядит следующим образом:

```
acad /b ramka
```

Один из способов вызова AutoCAD с указанием дополнительного параметра – изменение в Windows свойств ярлыка программы. Для этого необходимо щелкнуть на ярлыке правой кнопкой манипулятора «мышь» и выбрать пункт «Свойства» появившегося меню. Можно создать несколько ярлыков программы, различающихся запускаемым пакетным файлом, для разных условий работы.

Пакетные файлы можно использовать вместо шаблонов для настройки параметров рисунка и построения стандартных элементов. Преимущество пакетных файлов в том, что их количество, в отличие от шаблона, используемого в основе рисунка, может быть более одного.

### **Язык программирования**

Другой способ автоматизации создания чертежей – использование языка программирования. В отличие от сценариев и пакетных файлов, языки программирования позволяют использовать переменные и массивы, производить расчеты, запрашивать у пользователя данные, для выполнения повторяющихся действий создавать подпрограммы, организовывать выполнение действий после проверки заданных условий, выполнять действия циклически.

Программы создаются на специальных языках программирования, которые делятся на языки программирования высокого и низкого уровня. Языки программирования низкого уровня приближены к командам ЭВМ и в настоящее время практически не применяются. Языки высокого уровня обладают наглядностью, так как их синтаксис близок к естественной форме записи действий, и универсальностью, так как программа может выполняться на разных типах ЭВМ. Программа с языка программирования с помощью специальной программы-транслятора переводится на язык команд ЭВМ, выполняемых на компьютере. Программы-трансляторы бывают двух видов:

- интерпретаторы, в которых последовательно осуществляется перевод на язык команд ЭВМ каждой строчки текста программы и сразу производится выполнение строки;

- компиляторы, в которых сначала на язык команд ЭВМ переводится вся программа, и только после этого она исполняется.

Каждый из этих способов обладает своими преимуществами. Интерпретатор позволяет постепенно отлаживать программу, так как на-

личие ошибки останавливает работу программы только тогда, когда станет выполняться строка с этой ошибкой. При компиляции программа начнет работать только тогда, когда все ошибки в ней будут исправлены. При интерпретации программа начинает работать сразу после запуска, тогда как для компиляции необходимо время, которое может быть довольно значительным. Но выполнение программы в режиме интерпретации в десятки и сотни раз медленнее, чем при компиляции.

Процесс написания программы состоит из следующих этапов:

1. Постановка задачи (определение изображаемых деталей, входных данных, формул для расчета, всех необходимых условий и вариантов).

2. Параметризация чертежа, выделение определяющих и зависимых размеров, определение способов расчета координат всех точек, которые будут необходимы для черчения.

3. Создание алгоритма, то есть точной последовательности действий, записанных в произвольной форме. Алгоритм нужен для установления структуры и порядка работы сложных программ, он может быть оформлен в виде блок-схемы или пронумерованного списка действий.

В программах необходимо использовать блочный принцип, то есть разделять большую задачу на ряд логически обособленных подзадач. Разработка программы при этом состоит в последовательной разработке отдельных блоков, что значительно упрощает создание сложной программы.

4. Написание на основе алгоритма программы и ее ввод в компьютер.

5. Отладка программы, то есть процесс исправления ошибок. Ошибки могут быть двух видов. Синтаксические ошибки выявляются на этапе компиляции или интерпретации. Логические ошибки связаны с неправильным алгоритмом работы. В этом случае синтаксическая форма записи верна и программа работает, но результаты ее работы ошибочны. Особенно трудно выявлять скрытые логические ошибки, которые проявляются только при определенном сочетании исходных данных. Выявление логических ошибок осуществляется тестированием, когда проверяются результаты работы программы при различных исходных данных.

Программирование создания чертежей в AutoCAD может выполняться двумя способами:

– с помощью содержащегося в AutoCAD языка программирования AutoLISP, который имеет команды (функции) для математических расчетов, организации диалога с пользователем, выполнения любых команд AutoCAD, работы с объектами, имеющимися на чертеже;

– созданием файла-чертежа в любом языке программирования (записью в файл в установленном формате всех объектов чертежа) и открытием этого чертежа в AutoCAD.

## 1.7. МОДЕЛИРОВАНИЕ ВЫСОКОТЕМПЕРАТУРНЫХ И ТЕПЛОЭНЕРГЕТИЧЕСКИХ УСТАНОВОК

### Системный анализ технических объектов

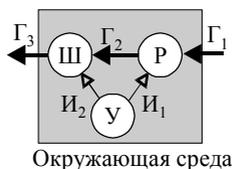
Высокотемпературная установка состоит из ряда элементов, в которых протекают взаимосвязанные процессы. Например, паровой котел обычно включает топку, экономайзер, несколько ступеней пароперегревателей, воздухоподогреватель. Проектирование каждого из этих элементов включает однотипные расчеты теплового баланса по газу и воде (пару), гидравлический и аэродинамический расчет, расчет теплопередачи. Моделирование и проектирование в таком случае удобно осуществлять на основе методов системного анализа. Рассмотрим методологию системного анализа сложных объектов.

*Система* – это совокупность элементов и связей элементов друг с другом и окружающей средой. Система функционирует в окружающей среде и взаимодействует с ней. Она состоит из элементов и связей. *Элементы* являются самостоятельными единицами (объектами). *Связи* – это способы взаимодействия элементов друг с другом и окружающей средой. По отношению к окружающей среде связи делятся на внутренние и внешние. Различают следующие виды связей:

- материальные (сырье, воздух, топливо и др.);
- энергетические (потoki теплоты, электроэнергия);
- информационные (сигналы в электронных цепях).

Элементы обеспечивают переработку и взаимные превращения материала, энергии и информации, а связи – передачу материала, энергии и информации без их изменения.

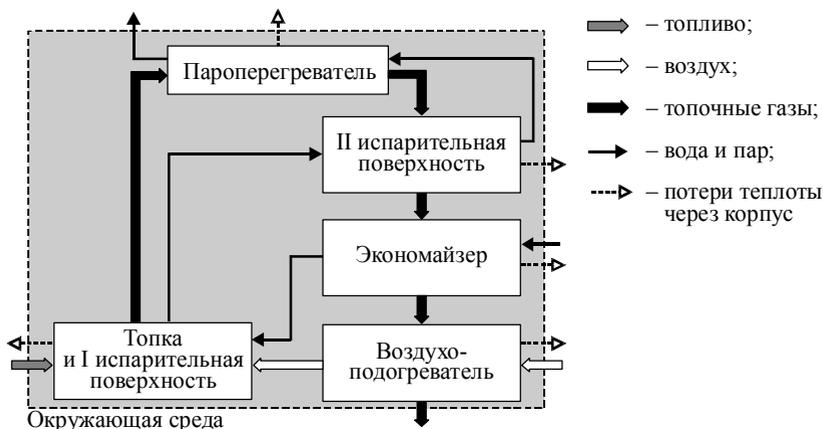
Информационная связь представлена на рис. 5. Примером системы с преобладающими информационными связями служит ЭВМ, целью работы которой является переработка информации.



**Рис. 5. Пример информационной связи:**

Р – расходомер; Ш – шибер; У – управляющий элемент (оператор, автоматический регулятор, ЭВМ);  $\Gamma_1, \Gamma_2, \Gamma_3$  – материальные связи (газовый поток);  $I_1, I_2$  – информационные связи

Пример системы представлен на рис 6. Здесь элементами являются поверхности нагрева, материальными связями – потоки воздуха, топлива, газов, воды и пара, тепловыми связями – потоки теплоты в окружающей среде.



**Рис. 6. Представление прямоточного котла Рамзина как системы**

При исследовании системы ее элементы представляются в виде «черного ящика». При этом происходящее внутри них принимается неизвестным, а работа элемента оценивается по состоянию его связей.

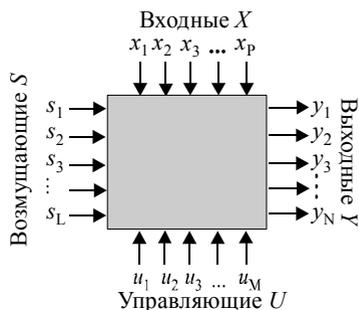
Связи элемента разделяются на *входы* (воздействие, которое на элемент оказывает окружающая среда или другие элементы) и *выходы* (воздействие, которое элемент оказывает на окружающую среду или другие элементы). Каждый вход или выход характеризуют различные параметры, описывающие состояние связи (температура, состав или концентрация, расход и т. д.).

Параметры по своему характеру и отношению к элементу делятся на четыре группы (рис. 7):

- *входные* – параметры, которые могут быть измерены, но не могут быть произвольно изменены (например, состав топлива);

- *управляющие* – параметры, которые могут быть измерены и которые можно изменять для управления процессом (например, расход питательной воды или топлива);

- *возмущающие* – параметры, которые случайным образом изменяются или которые не могут быть измерены (например, скорость ветра, влияющая на потери теплоты через стенки аппарата);



**Рис. 7. Параметры элемента**

- *выходные* – параметры, которые определяются режимом процесса (например, температура отходящих газов, давление и температура пара).

Входные, управляющие и возмущающие параметры соответствуют входам, а выходные – выходам. Входные параметры еще называют *факторами*, а выходные – *откликом*.

По описываемым свойствам параметры разделяются:

- на конструктивные, отражающие структуру и геометрические размеры объекта (объем, диаметр, количество элементов и т. д.);

- физические, описывающие состояние и свойства потоков (состав, температура, концентрация, расход и т. д.);

- параметры элементарных процессов (коэффициент теплоотдачи, трения, константы скоростей химических реакций и т. д.).

Каждый параметр описывается единым образом (табл. 2).

Таблица 2

### Примеры описания параметров

Наименование	Описание		
Название	Температура воды	Энтальпия пара	...
Условное обозначение	$t_v$	$h_{п,п}$	...
Единица измерения	°C	кДж/кг	...
Интервал возможных значений	0...40	2500...4000	...
Тип	Входной	Выходной	...

Разложение элементов на несколько составляющих называется *деструктуризацией*, а объединение нескольких элементов в один – *синтезом*. Работу любой установки можно разделить на ряд стадий, соответствующих перемещению или превращению материала, энергии или информации, называемых процессами.

Процессы, не поддающиеся деструктуризации, называются *типовыми* (или элементарными, физическими). В теплоэнергетике и теплотехнологии различают пять типовых процессов:

- движение;
- теплообмен;
- массообмен;
- изменение агрегатного состояния;
- химические превращения.

Объединение нескольких типовых процессов называется макропроцессом.

Методология исследования и анализа объектов посредством их деструктуризации до стандартных элементов и представления в качестве системы называется *системным анализом*. При этом объект рассмат-

ривается не как единое целое, а как взаимосвязь составляющих его элементов. Этапами системного анализа являются:

- выделение исследуемой системы, ее элементов и связей;
- разработка математической модели системы, включающая моделирование типовых процессов, моделирование макропроцессов, синтез технологической системы;
- разработка критерия оптимизации для проектных решений.

### **Понятие о моделях и моделировании**

Для изучения и проектирования промышленных процессов и аппаратов применяют модели. При этом изучаемый предмет представляется в некоторой форме, отличной от реальной, но отображающей свойства реального предмета.

*Модель* – это объект, заменяющий оригинал во время исследования. *Моделированием называется* разработка модели и получение с ее помощью информации об объекте.

Модели и моделирование разделяются на две группы – материальные (вещественные) и математические.

*Материальные модели* – это уменьшенные копии оригиналов, их отдельных частей или представление процессов оригиналов с помощью иных физических явлений. При построении материальных моделей используется теория подобия, определяющая условия распространения результатов моделирования на оригинал.

*Математическими моделями* называют комплекс математических зависимостей и логических выражений, отображающих характеристики изучаемого объекта. Обычно математическая модель имеет вид системы алгебраических (конечных) и дифференциальных уравнений. Целью математического моделирования является установление зависимости выходных параметров  $Y$  от входных  $X$ , управляющих  $U$  и возмущающих  $S$ . Общий вид математической модели следующий:

$$Y=f(X, U, S).$$

Модель должна обладать адекватностью, то есть соответствием характеристик модели и реального объекта с требуемой степенью точности.

Математическая модель содержит три составные части: формализованное описание, математическое описание и моделирующий алгоритм.

*Формализованное (физическое) описание* – это отражение физической природы моделируемого объекта. При составлении физического описания производится представление объекта в виде системы, выделяются процессы, которые подлежат отражению в модели, формули-

руются допущения и упрощения. Совокупность описываемых процессов определяют параметры объекта, включаемые в модель.

*Математическое описание* – выражение формализованного описания на языке математики в виде системы уравнений и функциональных отношений, включающих различные параметры модели.

*Моделирующий алгоритм* описывает последовательность действий для решения системы уравнений математического описания. При большой сложности математической модели от эффективности моделирующего алгоритма зависит как время расчета, так и сама возможность нахождения решения.

В соответствии с составными частями модели и принципами системного анализа математическое моделирование включает четыре взаимосвязанных этапа:

1. Постановка задачи моделирования (определение цели моделирования и необходимой точности результатов, выделение параметров процессов объекта, отражаемых в модели, представление объекта в виде технологической системы, задание допущений и ограничений).

2. Составление математического описания, представляемого в виде системы уравнений.

3. Выбор метода решения, составление алгоритма и программы.

4. Проверка адекватности (достоверности) модели путем сравнения параметров изучаемого объекта с результатами, полученными на модели в идентичных условиях.

Задачи математического моделирования делятся на прямые, обратные и задачи идентификации.

*Прямые задачи* – по известным зависимостям определяются численные значения параметров объекта. В прямых задачах выделяют *конструктивные расчеты* (определение технологических схем или конструктивных размеров аппаратов по заданным технологическим характеристикам) и *поверочные расчеты* (определение технологических параметров процессов, происходящих в аппаратах с заданными конструктивными размерами).

*Обратные задачи* заключаются в определении по имеющимся зависимостям вида уравнения, описывающего зависимость.

*Задачи идентификации* – это выбор модели, наиболее точно описывающей процесс, из ряда имеющихся моделей, а также определение или уточнение коэффициентов в имеющейся модели. Идентификация является частью обратных задач. В отличие от них, в задачах идентификации вид зависимости уже известен и обычно необходимо определить только коэффициенты этой зависимости.

По виду математического описания различают математические модели с *детерминированным и статистическим описанием*.

Детерминированное описание основано на описании объектов с помощью физических, химических и иных фундаментальных законов. Эти модели являются универсальными и могут описывать любые режимы работы объекта. Детерминированное описание получается с помощью решения прямой задачи, оно характеризуется большой сложностью. Точность таких моделей зависит от адекватности описания процессов в объекте.

Статистическое описание основано на обработке экспериментальных данных, полученных при исследовании объекта. Статистическое описание строится на основе решения обратной задачи или задачи идентификации. Уравнения, полученные с помощью статистического описания, называют также эмпирическими, или экспериментальными.

Разработка экспериментальных зависимостей заключается в том, что, изменяя ряд параметров объекта, определяют изменения нужных характеристик. Параметры, которые изменяют, называют *факторами*, их может быть несколько. Параметр, который измеряется при различных значениях факторов, называется *отклик*, в экспериментальной зависимости он должен быть один. Исследование объекта заключается в задании ряда значений факторов и получении для каждой совокупности факторов отклика. Отдельный акт по получению отклика по значениям факторов называется экспериментом, или *экспериментальной точкой*. Совокупность значений факторов всех элементов, проводимых при одном исследовании, называется *планом эксперимента*.

При описании модели с помощью статистических методов важно правильно построить план эксперимента и подобрать общий вид уравнения зависимости отклика от факторов. Важным показателем при разработке статистических моделей является обеспечение возможности интерполяции и экстраполяции на основе полученных зависимостей. Интерполяция – это соответствие рассчитанных значений реальным внутри диапазона измерений. Экстраполяция – аналогичное соответствие вне диапазона измерений. Достоинство статистического описания заключается в простоте и наличии стандартных методов для разработки (планирование эксперимента, корреляционный и регрессионный анализ). В ряде случаев модели сложных процессов можно получить только статистическими методами. Недостатки моделей, полученных с помощью статистического описания, следующие:

- а) применимость только в диапазоне изменения факторов, в котором производился эксперимент, и малая надежность экстраполяции;
- б) отсутствие универсальности, так как зависимости применимы только для тех условий и того процесса или производства, где они бы-

ли получены и в общем случае не могут быть перенесены на другие агрегаты и технологические линии;

в) возможность влияния неучтенных в модели факторов (например, температуры обжига в рассматриваемом примере);

г) получение результатов возможно только при наличии действующих объектов;

д) статистические зависимости не отражают физической сущности процессов.

По временным признакам разделяют *статические* и *динамические модели*. В статических моделях параметры модели во времени не изменяются. Процессы, не изменяющиеся во времени, называются стационарными. Все непрерывные производства характеризуются стационарными процессами и описываются статическими моделями. Отличительная особенность статических объектов – наличие постоянных входа и выхода материалов и потоков материалов внутри объекта.

В динамических моделях параметры изменяются во времени. Процессы, изменяющиеся со временем, называются нестационарными. Динамические модели описываются дифференциальными уравнениями, включающими производную по времени. Обычно они имеются в периодических производствах и переходных режимах от одного стационарного состояния в другое.

По пространственным признакам модели делят на *модели с распределенными* и *сосредоточенными параметрами*. В моделях с сосредоточенными параметрами все характеристики принимаются равными в разных точках аппарата (например, температура воздуха в замкнутом объеме) и в математическом описании отсутствуют пространственные координаты. В моделях с распределенными параметрами характеристики меняются в разных точках аппарата. Для задания точки в пространстве необходимы три координаты, но если с изменением какой-нибудь координаты характеристики остаются неизменными, одну или две координаты исключают. Таким образом, модели делятся на одномерные, двухмерные и трехмерные.

Уравнения, входящие в математическое описание, можно разделить на четыре группы.

1. Уравнения сохранения массы и энергии, выражающие распределение в потоках вещества и теплоты. Для каждого элемента на основании его связей записывается материальный и тепловой баланс. При этом связи элементов делятся на входные и выходные (обозначаемые так же, как приход и расход). В уравнение балансов обязательно включают все связи рассматриваемого элемента. К этой же группе относятся стехиометрические балансы химических реакций.

2. Уравнения, описывающие типовые процессы, которые включают:  
а) уравнения движения (дифференциальные уравнения неразрывности и Навье – Стокса, расчет гидродинамического сопротивления);

б) уравнения теплообмена (дифференциальные и конечные уравнения теплопроводности и конвективного теплообмена, уравнения переноса теплоты излучением);

в) уравнения массообмена (уравнения массообмена аналогичны уравнениям теплообмена с заменой температур на концентрации, а коэффициентов теплопроводности, теплоотдачи и теплопередачи – на коэффициенты массопроводности или диффузии, массоотдачи и массопередачи);

г) уравнения, описывающие химические превращения (уравнения равновесия и уравнения химической кинетики).

В этом перечне не приведены уравнения, описывающие процессы изменения агрегатного состояния. Данный процесс описывается уравнениями, относящимися к материальным и тепловым балансам и химическому равновесию.

3. Уравнения для расчета свойств веществ в зависимости от температуры и других параметров и уравнения для расчета коэффициентов в уравнениях, описывающих типовые процессы (например, зависимость коэффициента теплоотдачи от температуры).

4. Ограничения для параметров процесса. Ограничения не входят в уравнения математического описания, а задают условия, которым должно удовлетворять решение. Ограничения бывают естественные, определяющие все возможные значения параметра, исходя из физических условий, и искусственные, определяемые из требований к значению параметра в конкретных условиях производства. Например, концентрация вещества может принимать значения от 0 до 1 – это естественное ограничение. Если по техническим условиям концентрация должна быть от 0,2 до 0,4, то это является искусственным ограничением.

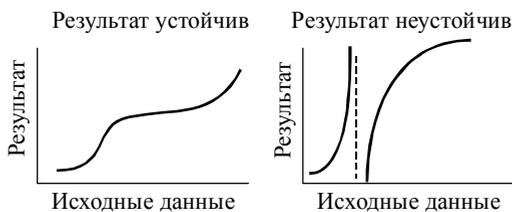
После составления системы уравнений математического описания необходимо выбрать метод решения и составить алгоритм решения и программу. Методы решения разделяются на аналитические и численные.

*Аналитические методы* позволяют получить точное значение неизвестных путем эквивалентных преобразований уравнений с целью выразить значение неизвестных через известные величины.

*Численные методы* позволяют получить приближительное решение с заданной степенью точности. Это осуществляется заменой уравнений на более простые (например, дифференциальных на алгебраические) или поиском решения путем выполняемого по специальным методикам подбора или перебора значений переменных.

*Алгоритм* – это точный порядок действий над входными данными, предназначенный для получения необходимого решения. Разработанный алгоритм должен удовлетворять ряду условий, заключающихся в получении результата при любых допустимых вариантах расчета. Условия корректности алгоритма заключаются в следующем:

1. Сходимость – решение должно быть получено при задании любых допустимых исходных данных.
2. Однозначность – при одинаковых исходных данных должно получаться одинаковое решение.
3. Устойчивость решения – оно должно непрерывно изменяться при изменении исходных данных (рис. 8).



**Рис. 8. Устойчивость результата**

4. Вычислительная устойчивость алгоритма – при допустимых исходных данных в ходе вычислений не должно возникать ошибок (например, деления на ноль или переполнения).

Классификация методов решения уравнений математического описания и алгоритмы методов приведены в работе [1].

При разработке программ используется структурный принцип. Он заключается в разделении расчета на отдельные блоки, оформляемые в виде процедур или функций. Каждый блок соответствует отдельной задаче моделирования. Это может быть расчет отдельного аппарата или процесса, реализация стандартного метода решения уравнения, отдельные расчеты, повторяемые в нескольких частях программы. Таким образом, выполняемая часть программы должна представлять собой вызов ряда процедур или функций. Сложные блоки также разбиваются на ряд подблоков.

Можно выделить четыре главных блока, которые должны обязательно присутствовать в программе:

- 1) процедуры и функции для расчета свойств веществ в зависимости от их состава или температуры;
- 2) ввод исходных данных, необходимых для расчета;
- 3) проведение расчета;
- 4) вывод полученных результатов.

Структурное построение программы позволяет упростить разработку модели, так как создавать отдельные блоки проще, чем всю их совокупность. На первом этапе разработки программы можно упростить ряд блоков. Например, в функции расчета теплоемкости в зависимости от температуры использовать не формулу, а задать постоянное число. Также можно в части блоков расчеты заменить на более простой вариант или вообще без расчетов присваивать результатам необходимые значения. После создания работоспособной программы эти блоки дорабатываются и расширяются до полного варианта расчета. Такой принцип работы называется «от простого к сложному».

Структурный принцип позволяет изменять или заменять блоки, не влияя на работу остальных частей программы. Например, в начале создания программы все внимание сосредоточивается на самом расчете. Поэтому блок ввода исходных данных можно сделать в виде простого присвоения переменным необходимых значений. После разработки расчета блок ввода можно усовершенствовать, создав интерфейс современного уровня. Однотипные блоки можно заменять, изменяя название процедуры для их вызова. Например, можно создать две процедуры: для расчета колосникового холодильника и для расчета рекуператорного холодильника. Затем, в зависимости от типа рассчитываемой печи, необходимо использовать нужную процедуру. Отлаженные блоки могут быть использованы и при решении других задач.

Чтобы использовать указанные преимущества, необходимо выполнить ряд требований структурного программирования. Процедурам и функциям исходные данные должны передаваться только через список их параметров. Результаты расчета должны возвращаться как результат функций или через список параметров (в языке программирования Паскаль для этого переменные, содержащие результаты, должны быть параметрами-переменными и описываться в списке параметров с использованием служебного слова `var`). Все необходимые промежуточные переменные, необходимые для расчета, должны описываться внутри процедур (то есть быть локальными). Таким образом, процедуры и функции не должны использовать никакие глобальные переменные.

Ряд правил позволяет облегчить разработку программ и увеличить наглядность их текста:

1. Имена переменных, процедур, функций должны отражать их назначение (например, `HeatBalans`, `Economizer`, `Fuel` и т. д.).

2. В программе необходимо использовать комментарии, в которых описываются назначение объявляемых переменных и констант, действия сложных операторов и порядок работы программы.

3. В одной строке следует помещать один оператор. В программе необходимо использовать систему отступов. При этом операторы записываются, начиная с одной позиции. Операторы, расположенные внутри циклов или внутри операторных скобок, а также содержимое разделов описания записываются с отступом в 2–3 позиции. Такая запись значительно увеличивает наглядность текста и логической структуры программы.

### **Оптимизация технических систем**

Конечной целью математического моделирования является оптимизация исследуемого объекта, то есть достижение наилучшего конструктивного исполнения или рабочего режима. Успех оптимизации определяет правильная формулировка задачи, которая включает:

- выбор критерия оптимизации;
- установление ограничений;
- выбор оптимизирующих факторов;
- запись целевой функции.

*Критерий оптимизации* характеризует работу системы, поэтому он формируется из выходных параметров. Критерий должен удовлетворять трем требованиям:

- 1) быть единственным;
- 2) быть числовым;
- 3) оптимальное значение должно быть экстремальным (то есть максимумом или минимумом).

Оптимизируемые параметры могут быть экономическими (прибыль, себестоимость, рентабельность) – их применяют в оптимизации производства в целом – и технологическими (расход топлива, производительность), применяемыми при оптимизации одного аппарата или узла. Необходимо стремиться применять экономические параметры оптимизации, так как улучшение одного из технологических показателей может привести к ухудшению других. Например, мероприятия по снижению расхода топлива в печи могут привести к уменьшению температуры горения и температурного напора между газами и материалом. В результате снизится производительность. Так как часть расходов на производство является постоянной и от производительности не зависит, их доля в удельной себестоимости возрастет, причем на величину, которая может быть выше стоимости сэкономленного топлива.

*Ограничения* определяют условия, которые всегда необходимо соблюдать, независимо от значения критерия оптимизации. Параметры, на которые накладывают ограничения, могут быть только управляющие и выходные.

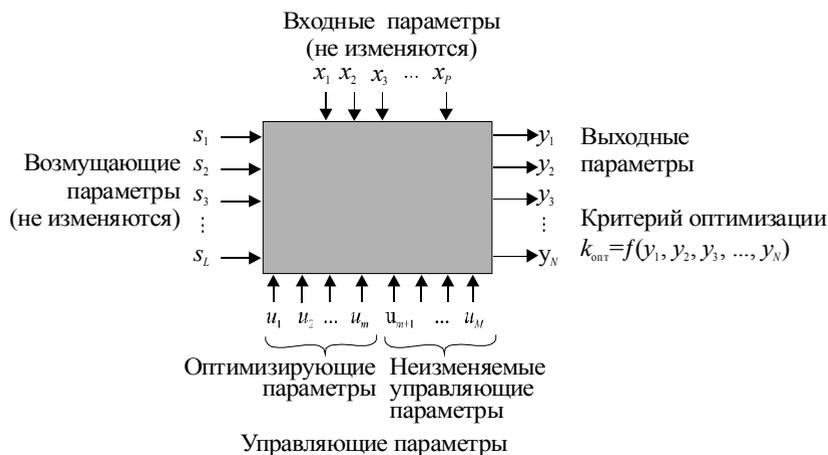
Ограничения параметра  $P$  могут быть в виде одностороннего предела

$$P \geq P_{\min} \quad \text{или} \quad P \leq P_{\max}$$

или интервала

$$P_{\min} \leq P \leq P_{\max}.$$

Оптимизирующие факторы относятся к управляющим параметрам. Задача оптимизации заключается в нахождении значений оптимизирующих факторов, при которых достигается заданный экстремум критерия оптимизации (при соблюдении заданных ограничений). В процессе поиска изменяются только оптимизирующие факторы. Остальные управляющие параметры, а также входные и возмущающие параметры при оптимизации задаются постоянными (рис. 9). Поэтому найденные значения являются оптимальными только в этих условиях, при их изменении оптимум также может измениться.



**Рис. 9. Оптимизация элемента или аппарата**

*Целевая функция* – это критерий оптимизации, рассматриваемый как функция от управляющих, входных и возмущающих параметров.

Рассмотрим отличия математической модели, критерия оптимизации и целевой функции. Представим оптимизируемый элемент в виде схемы (см. рис. 9). Математическая модель элемента представляет собой зависимость набора выходных параметров  $Y$  от входных  $X$ , управляющих  $U$  и возмущающих  $S$ :

$$Y = f(X, U, S).$$

Оптимизирующие факторы состоят из управляющих параметров:

$$U_{\text{опт}} = \{u_1, u_2, \dots, u_m\}.$$

Критерий оптимизации является функцией от выходов системы:

$$k_{\text{опт}} = f(Y).$$

Целевая функция представляет собой зависимость критерия оптимизации от входов элемента:

$$k_{\text{опт}} = f(X, U, S).$$

Рассмотрим пример. Потери теплоты через изоляцию газохода круглого сечения описываются уравнением теплопроводности через цилиндрическую стенку:

$$q = 2\pi \left( \frac{1}{\alpha_1(r+h)} + \frac{1}{\lambda} \frac{r+h}{r} + \frac{1}{\alpha_2 r} \right)^{-1} (t_2 - t_1),$$

где  $r$  – радиус трубы;  $h$  – толщина изоляции;  $t_2$  – температура внутренней среды;  $t_1$  – температура окружающей среды;  $\alpha_1, \alpha_2$  – коэффициенты теплоотдачи на наружной и внутренней поверхностях;  $\lambda$  – коэффициент теплопроводности изоляции.

Необходимо найти толщину изоляции, обеспечивающую минимальные потери теплоты. В рассматриваемой задаче входными параметрами являются радиус газохода, его температура и коэффициент теплопроводности обмазки; управляющим параметром – толщина обмазки; возмущающим – температура воздуха (так как она изменяется в зависимости от времени года и суток). Коэффициенты теплоотдачи  $\alpha_1$  и  $\alpha_2$  параметрами не являются, так как определяются температурой внутренней и внешней среды, их скоростями движения  $v_1, v_2$  и радиусом.

Модель процесса имеет общий вид

$$q = f(t_2, t_1, v_1, v_2, r, h, \lambda).$$

При увеличении толщины изоляции в общем случае (при  $\lambda/\alpha_1 < h$ ) теплотопотери будут уменьшаться, но одновременно будет увеличиваться стоимость изоляции. Поэтому в качестве критерия оптимизации используем приведенные годовые затраты, отнесенные к одному метру газохода:

$$k_{\text{опт}} = 0,15KЗ + TЗ.$$

Текущими затратами является стоимость теряемой теплоты в год:

$$TЗ = \tau_p Q_{\text{ч}} C_{\text{т}},$$

где  $\tau_p$  – количество часов работы газохода в год;  $Q_{\text{ч}} = q \cdot 3600 \cdot 10^{-6} / 4,19$  – количество теряемой теплоты в час, Гкал (величина  $q$  имеет размерность Вт);  $C_T$  – стоимость 1 Гкал теплоты.

Капитальные затраты – стоимость изоляции, которая может быть определена через ее массу, плотность  $\rho$  и стоимость 1 кг изоляции  $C_{\text{и}}$ :

$$KЗ = \pi \left( (r + h)^2 - r^2 \right) \rho C_{\text{и}}$$

Целевой функцией является следующая зависимость:

$$k_{\text{опт}} = f(t_2, t_1, \nu_1, \nu_2, r, h, \lambda, \tau_p, C_T, C_{\text{и}}, \rho).$$

Изменяя  $h$ , находим минимум  $k_{\text{опт}}$ , при этом параметры  $t_2, t_1, \nu_1, \nu_2, r, \lambda, \tau_p, C_T, C_{\text{и}}, \rho$  задаются постоянными.

При оптимизации одного параметра остальные, как правило, ухудшаются. Поэтому, как говорилось ранее, критерием оптимизации должен быть экономический показатель. Если это сделать невозможно, но необходимо одновременное улучшение нескольких параметров системы, они должны объединяться в один критерий оптимизации. Это производится с помощью метода весовых коэффициентов:

$$k_{\text{опт}} = \sum_{i=1}^N \langle Y_i \rangle m_i,$$

где  $k_{\text{опт}}$  – обобщенный критерий оптимизации;  $N$  – число оптимизируемых параметров;  $\langle Y_i \rangle$  – нормализованные параметры, приведенные к единому виду экстремума и размерности;  $m_i$  – весовые коэффициенты, определяющие значимость (важность) соответствующего параметра.

Единая размерность необходима для сопряжения различных параметров. В качестве нее можно выбрать экономическую шкалу, расход энергии и так далее; в этом случае весовые коэффициенты обычно равны 1, то есть отсутствуют. Если единую шкалу подобрать невозможно, то параметры необходимо пронормировать – заменить их на безразмерную величину, изменяющуюся от нуля до единицы:

$$\langle Y \rangle = \frac{Y - Y_{\min}}{Y_{\max} - Y_{\min}},$$

где  $Y_{\min}, Y_{\max}$  – диапазон изменения значения параметра.

Весовые коэффициенты  $m_i$  всегда положительны и характеризуют значимость данных свойств при достижении оптимума. Если параметры приведены к единой размерности, то сумма коэффициентов  $m_i$  должна быть равна единице:

$$m_1 + m_2 + \dots + m_N = 1.$$

Все параметры, входящие в обобщенный критерий оптимизации, должны иметь экстремум одного вида. Для изменения вида экстремума с максимума на минимум или наоборот необходимо изменить знак перед параметром на противоположный. Если оптимальное значение не экстремальное, а локальное и равно  $Y_{\text{опт}}$ , используют отклонение от оптимального значения:

$$\langle Y \rangle = \frac{\sqrt{(Y - Y_{\text{опт}})^2}}{Y_{\text{опт}}}.$$

Например, необходимо выбрать режим, обеспечивающий минимальный удельный расход топлива на единицу продукции  $x_t$ , кг/т, при максимальной производительности  $G$ , т/ч. В качестве единой шкалы можно выбрать экономическую. Если стоимость 1 т продукции  $C_{\text{п}}$ , а килограмма топлива –  $C_t$ , то критерий оптимизации имеет вид

$$k_{\text{опт}} = C_{\text{п}}G - C_t x_t G = (C_{\text{п}} - C_t x_t)G.$$

Для приведения экстремума к единому виду (максимуму) перед стоимостью топлива изменен знак.

Если единую шкалу подобрать невозможно (например, продукция является промежуточной и на нее отсутствует цена), параметры нормируются.

Например, удельный расход топлива в рассматриваемом процессе изменяется от 200 до 240 кг/т, а производительность – от 46 до 54 т/ч. Критерий оптимизации будет иметь вид

$$k_{\text{опт}} = (G - 46)/8 - (x_t - 200)/40 = 0,125G - 0,025x_t - 10,75.$$

Так как свободный член и масштаб на положение оптимума не влияют, критерий оптимизации можно записать в следующем виде:

$$k_{\text{опт}} = 5G - x_t.$$

В обобщенный критерий оптимизации нельзя включать более трех-четырёх параметров, а приведение к единому виду лучше производить с помощью единой шкалы. При нормировке параметров успех оптимизации во многом зависит от правильного подбора весовых коэффициентов, так как даже их незначительное изменение часто приводит к существенному изменению оптимума.

Классификация методов поиска оптимума для детерминированных моделей и алгоритмы методов оптимизации приведены в работе [1].

## Особенности моделирования парогенерирующих установок

Задачей конструктивного расчета теплотехнологических, парогенерирующих и теплообменных установок является нахождение площади теплопередающей поверхности, по которой определяются размеры установки. Площадь теплообмена  $F$  определяется из уравнения теплопередачи

$$Q = k \Delta t F,$$

где  $Q$  – требуемое количество передаваемой теплоты, определяемое из уравнения теплового баланса;  $k$  – коэффициент теплопередачи между горячим и холодным теплоносителями или между топочными газами и нагреваемым веществом;  $\Delta t$  – температурный напор (разница температур).

При поверочном расчете при известной площади  $F$  находится количество передаваемой теплоты  $Q$  и производительность установки.

Математическую модель парогенерирующей установки (ПГУ) в соответствии с принципами системного анализа и структурного программирования можно разделить на ряд подсистем различного уровня иерархии (рис. 10).

В состав модели входят следующие основные уравнения теплового расчета (штрихом обозначен вход узла или элемента, двумя штрихами – выход):

1) материальный баланс по энергоносителю

$$\Sigma G' = \Sigma G'',$$

где  $G$  – массовый расход воды (пара);

2) материальный баланс по топочным газам

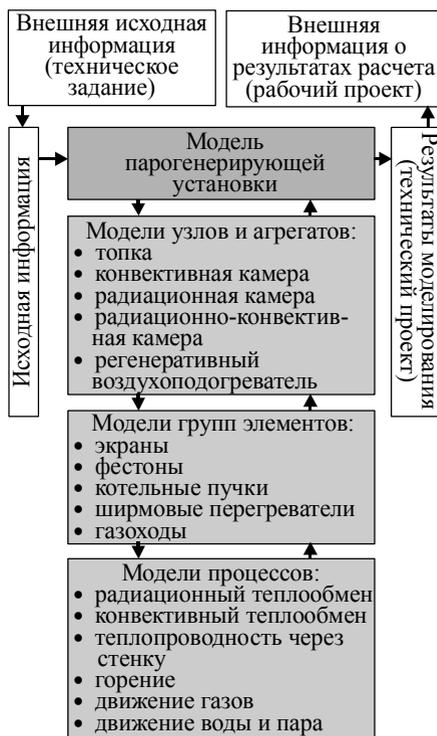


Рис. 10. Схема модели ПГУ и ее связь с этапами проектирования

$$m'_{\text{H}_2\text{O}} + m'_{\text{CO}_2} + m'_{\text{N}_2} + m'_{\text{O}_2} + m'_T + m'_B + m'_3 =$$

$$= m''_{\text{H}_2\text{O}} + m''_{\text{CO}_2} + m''_{\text{N}_2} + m''_{\text{O}_2} + m''_B + m''_T + m''_3,$$

где  $m_{\text{H}_2\text{O}}$ ,  $m_{\text{CO}_2}$ ,  $m_{\text{N}_2}$ ,  $m_{\text{O}_2}$  – масса продуктов горения;  $m''_B$ ,  $m''_T$ ,  $m''_3$  – масса воздуха, топлива и золы;

3) аэродинамический баланс для газов и гидравлический баланс по воде и пару

$$p - p'' + 0,5(v')^2 \rho' - 0,5(v'')^2 \rho'' + \rho' gh' - \rho'' gh'' + \Delta p_c = 0,$$

где  $p$  – абсолютное давление;  $v$  – скорость;  $\rho$  – плотность;  $h$  – высота подъема;  $\Delta p_c$  – потери давления на преодоление трения и местных сопротивлений;

4) тепловой баланс

$$\sum h'_T - \sum h''_T + q_T = \sum h''_B - \sum h'_B + q_{\text{п}},$$

где  $h_T$ ,  $h_B$  – энтальпия газов и воды (пара);  $q_T$  – теплота от сгорания топлива;  $q_{\text{п}}$  – потери теплоты через стенки установки и др.

5) уравнение теплопередачи

$$k \Delta t F = \sum h'_T - \sum h''_T + q_T = \sum h''_B - \sum h'_B + q_{\text{п}}.$$

Неизвестными в модели могут быть площадь теплообмена или расход воды (пара) и температура газов на выходе из установки (элемента).

Дополнительно в модель включаются прочностные расчеты, конструктивные расчеты для поверхностей нагрева и узлов. В модели могут быть учтены следующие ограничения:

- по технологическим требованиям к установке, описанным в техническом задании;
- по технологии изготовления (толщина стенок трубок, балок и т.д.);
- по условиям надежности эксплуатации (термическое напряжение в элементах, коррозия, эрозия);
- экологические (состав отходящих газов).

В результате моделирования ПГУ определяются:

- потери давления потоков;
- изменение энтальпии потоков;
- средние скорости потоков;
- максимальная температура стенки и труб, диаметр труб;
- расходы воды (пара), топлива и газов.

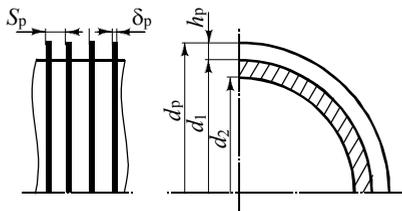
## Особенности моделирования и проектирования кожухотрубных теплообменников

Теплоэнергетическое оборудование включает кожухотрубные теплообменники, которые используются как рекуператоры в системах теплоснабжения и теплотехнологических системах, входят в состав тепловых насосов. Газотрубные котлы по конструкции и методам расчета также могут быть отнесены к кожухотрубным теплообменникам. Обычно в таких теплообменниках пучки трубы расположены в шахматном порядке, трубы могут быть оребренные. По трубам движется один из теплоносителей, в межтрубном пространстве – другой. Например, в испарителе и конденсаторе по трубам движется вода или рассол, в межтрубном пространстве – фреон. Характеристики оребренных труб теплообменников тепловых насосов представлены в табл. 3 и на рис. 11.

Таблица 3

**Характеристики труб теплообменников**

Наружный диаметр $d_1$ , мм	16	17,7	23
Внутренний диаметр $d_2$ , мм	13	14,7	20
Высота ребра $h_p$ , мм	2,2	1,9	1,5
Наружный диаметр ребра $d_p$ , мм	20,4	21,5	26
Толщина ребра $\delta_p$ , мм	0,5	0,5	0,5
Шаг между ребрами $S_p$ , мм	2,04	2,03	2
Коэффициент $\varepsilon_p$	2,67	2,12	1,93
Площадь наружной оребренной поверхности трубы длиной 1 м $F_1$ , м <sup>2</sup>	0,1769	0,1737	0,190
Площадь внутренней поверхности трубы длиной 1 м $F_2$ , м <sup>2</sup>	0,0408	0,0462	0,068



**Рис. 11. Конструктивные размеры оребренных труб**

Характеристики оребренных труб рассчитываются по следующим выражениям:

- площадь внутренней поверхности трубы длиной 1 м

$$F_2 = \pi d_2,$$



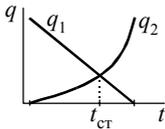
Для расчета коэффициентов теплоотдачи  $\alpha_1$  и  $\alpha_2$  требуются скорость, температура и свойства теплоносителей. Скорость теплоносителей обычно принимается согласно известным значениям.

В связи с законом сохранения энергии тепловые потоки должны быть равны друг другу (с учетом потерь теплоты  $q_{\text{п}}$  через наружные стенки аппарата). Расчет теплообменников заключается в нахождении неизвестной температуры стенки  $t_{\text{ст}}$  решением уравнения (рис. 13)

$$q_1 = q_2 + q_{\text{п}}$$

и определении удельного теплового потока

$$q = q_2.$$



**Рис. 13.** Решение уравнения теплопередачи относительно температуры стенки

В расчетах термическим сопротивлением стенки можно пренебречь, так оно составляет менее 1% от термического сопротивления теплообменников.

Средняя температура теплоносителей определяется по среднелогарифмическому температурному напору. Согласно рис. 12,

$$t_{1\text{cp}} = t'_1 - k(t'_1 - t''_1);$$

$$t_{2\text{cp}} = t''_2 - k(t''_2 - t'_2),$$

где масштабный коэффициент  $k$

$$k = \frac{t'_1 - t''_2 - \Delta t}{t'_1 - t''_1 - t''_2 + t'_2}$$

По удельному тепловому потоку  $q$ , Вт, теплоотдающая поверхность  $F$ , м<sup>2</sup>,

$$F = \frac{Q}{q}$$

где  $Q$  – тепловая нагрузка теплообменника, Вт.

Число труб в теплообменнике должно обеспечивать принятую скорость теплоносителя  $v_2$ , проходящего в трубах.

Число труб в одном ходе, округленное до целого числа,

$$n = \frac{4G_2}{\pi d_2^2 v_2 \rho_2},$$

где  $G_2$  – расход теплоносителя, кг/с;  $\rho_2$  – плотность теплоносителя.

Длина труб в одном ходе  $L_x$

$$L_x = \frac{F}{\pi d_2 n m}$$

где  $m$  – число ходов; для упрощения конструкции теплообменника оно должно быть минимальным. Получаемая длина труб в одном ходе  $L_x$  должна быть в пределах, характерных для проектируемых теплообменников. Если изменением числа ходов  $m$  этого добиться невозможно, то задают другую скорость воды  $v_2$  и повторяют расчет теплообмена и конструктивных параметров.

После определения числа труб производят компоновку трубок в сечении теплообменника и определяют размеры трубной решетки и диаметр теплообменника.

Размещение труб может быть ромбическое (по вершинам равносторонних треугольников), или по концентрическим окружностям (рис. 14). Шаг трубок  $S$  принимается равным  $(1,3 \dots 1,5)d_1$ , но не менее  $d_1 + 6$  мм.

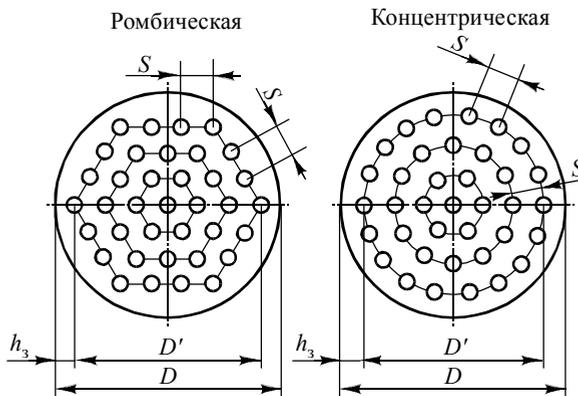


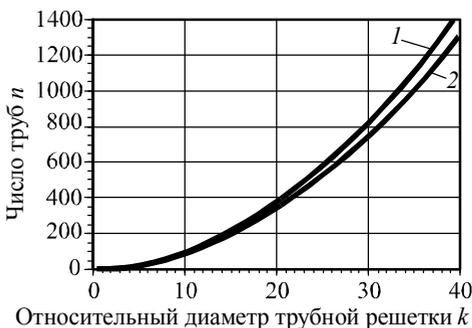
Рис. 14. Компоновка труб в теплообменнике

Для теплообменников с одним ходом расстояние  $D'$  между крайними трубами определяется по рис. 15. Размещение труб в многоходовом теплообменнике должно учитывать установку перегородок, поэтому компоновка и определение расстояния  $D'$  между крайними трубами для многоходовых теплообменников производится графическим путем.

Внутренний диаметр теплообменника  $D$  определяется по выражению

$$D = D' + d_1 + 2h_3,$$

где  $h_3$  – зазор между трубами и корпусом, 6 мм и более.



**Рис. 15. Число труб при различном относительном диаметре трубной решетки для одноходовых теплообменников при ромбической (1) и концентрической (2) компоновке**

Также для теплообменника рассчитываются диаметры входных и выходных патрубков для фреона и воды (по скорости теплоносителя в них).

При проектировании теплообменников оптимизация происходит по приведенным затратам, включающим капитальные затраты (стоимость теплообменника, которая может быть определена по его массе) и текущие затраты (которые включают затраты на перекачку теплоносителей через теплообменник). Оптимизирующими параметрами при этом могут быть диаметр труб и скорость теплоносителей.

## 2. АВТОМАТИЗАЦИЯ ЧЕРТЕЖНЫХ РАБОТ В СИСТЕМЕ AUTOCAD

Система AutoCAD – универсальная графическая система, в основу структуры которой положен принцип открытой архитектуры, позволяющий адаптировать и развивать многие функции системы применительно к конкретным задачам и требованиям. Это мощная универсальная среда для автоматизации инженерно-графических работ, которая позволяет:

- разрабатывать двухмерные чертежи;
- осуществлять трехмерное моделирование;
- создавать библиотеки чертежей и типовых форм;
- изменять и дополнять систему путем изменения меню, ввода новых команд, что дает возможность строить на базе AutoCAD системы автоматизированного проектирования.

### 2.1. ОБЩИЕ ПРИНЦИПЫ РАБОТЫ В СИСТЕМЕ

#### Запуск системы AutoCAD и создание нового чертежа

После запуска программы на экран выводится диалоговое окно, предлагающее открыть чертеж, записанный на диске, или создать новый чертеж (рис. 16).

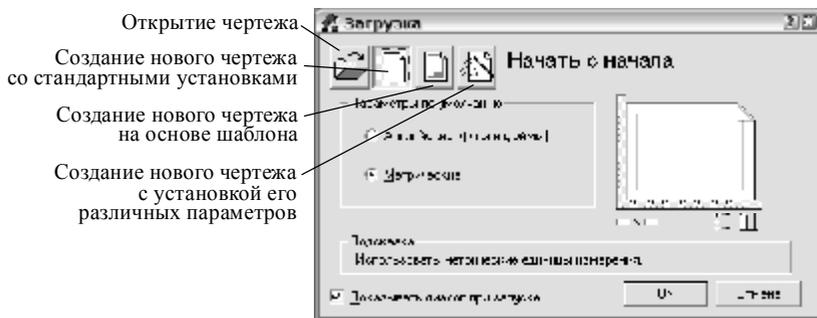


Рис. 16. Диалоговое окно при запуске AutoCAD

Файлы чертежей, создаваемые AutoCAD на дисках, имеют расширение «dwg». В каждом чертеже AutoCAD хранится множество параметров – размер области рисования, масштабные коэффициенты, параметры для вывода размерных надписей и др. Каждый из параметров

хранится в соответствующей ему системной переменной. Для нового чертежа эта информация берется из шаблона.

Шаблон называется чертеж, в котором записаны начальные установки чертежа и могут иметься графические объекты. Например, возможно создание шаблонов для различного размера бумаги с уже имеющимися штампами и т. д. В качестве шаблона можно использовать любой имеющийся чертеж. Обычно чертежи, которые будут использоваться в качестве шаблона, имеют расширение не «dwt», а «dwt». Вся информация, хранящаяся в шаблоне, переносится в новый чертеж, но при изменении чертежа сам шаблон не меняется.

При запуске возможно создание нового чертежа следующими способами:



– на основе стандартного шаблона;



– на основе одного из имеющихся шаблонов;



– изменением в диалоговом окне наиболее важных параметров.

Любой из параметров, выбранный в начале работы или взятый из шаблона, может быть в дальнейшем изменен. Это производится с помощью команды SETVAR (УСТПЕРЕМ) или непосредственным вводом в командную строку имени необходимой системной переменной.

После начала работы повторный вызов стартового диалогового окна производится следующим образом:

◆ Панель инструментов «Standard» («Стандартные инструменты»): 

◆ Команда меню: «File\New ...» («Файл\Новый ...»)

а открытие файла так:

◆ Панель инструментов «Standard» («Стандартные инструменты»): 

◆ Команда меню: «File\Open ...» («Файл\Открыть ...»)

Сохранение чертежа на жестком диске производится командой:

◆ Панель инструментов «Standard» («Стандартные инструменты»): 

◆ Команда меню: «File\Save» («Файл\Сохранить»)

При первом сохранении будет предложено задать имя чертежа. Сохранение чертежа под другим именем производится командой:

◆ Команда меню: «File\Save As ...» («Файл\Сохранить Как ...»)

Рекомендуется в процессе работы регулярно сохранять чертеж на диске для предотвращения потерь результатов работы при сбое в работе компьютера или электрической сети.

## Вид экрана при работе в AutoCAD

Вид окна программы представлен на рис. 17.

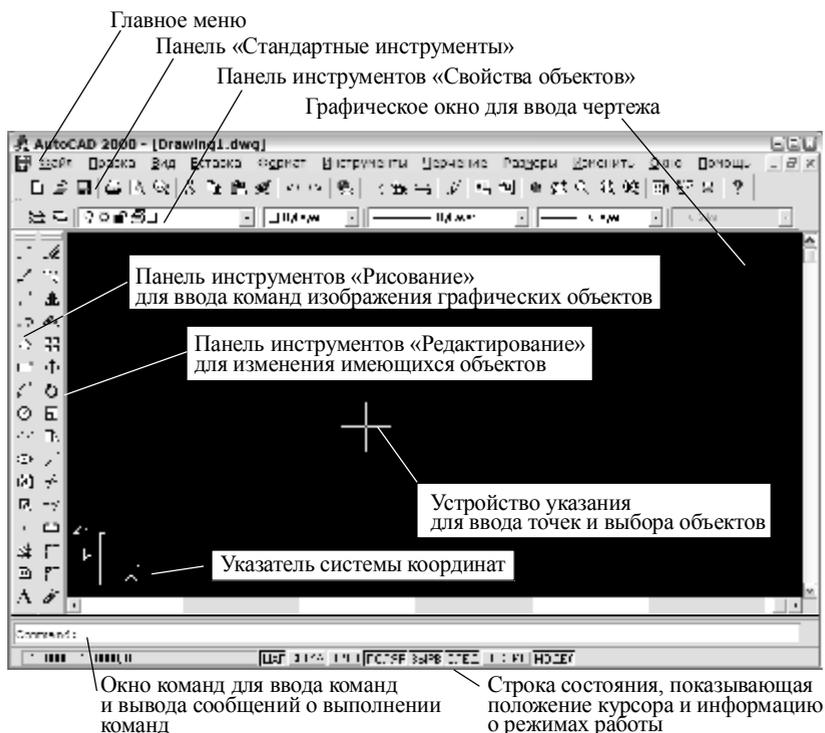


Рис. 17. Окно программы AutoCAD 2000

Окно содержит следующие стандартные элементы:

- меню;
- панели инструментов;
- графическое окно;
- окно команд;
- информационную строку.

*Главное меню* содержит несколько разделов, в каждый из которых входит несколько команд или подразделов. В дальнейшем для указания команды меню будут перечислять все разделы, где эта команда содержится. Например, запись

«Вид\Показать\Увеличить»

означает, что сначала необходимо выбрать раздел меню «Вид», затем подраздел «Показать», в котором находится команда «Увеличить».

*Панели инструментов* содержат команды, представленные в виде пиктограмм (иконок). Команда выполняется одиночным щелчком мыши на пиктограмме. При наведении на пиктограммы курсора выводится подсказка о названии команды.

Панели, изображаемые на экране, можно показывать и скрывать с помощью пункта меню:

- ◆ Команда меню: «View\Toolbars ...» («Вид\Панели инструментов ...»)

или после щелчка правой кнопкой мыши на любой из видимых панелей инструментов.

Панели инструментов могут быть закреплены у края окна или быть плавающими (рис. 18). Для изменения закрепления необходимо переместить панель к краю окна или от него.



**Рис. 18. Панели инструментов**

*Графическое окно* предназначено для показа чертежа или для его создания с помощью манипулятора «мышь». В нем находится указатель системы координат, показывающий направление координатных осей.

*Окно команд* предназначено для ввода команд и их дополнительных параметров. Размер окна команд по высоте можно изменить, переместив с помощью мыши его верхнюю границу. Обычно для работы достаточно высоты, вмещающей две-три строки. Количество строк текстового окна при начале работы с AutoCad задается на странице «Display» («Экран») диалогового окна, вызываемого с помощью команды меню Tools\Options (Инструменты\Опции).

Окно команд может быть выведено в виде отдельного текстового окна с помощью клавиши «F2», которая является переключателем между текстовым и графическим окном. Отдельное текстовое окно удобно при выводе на экран командой большого количества информации или для просмотра предыдущих команд.

*Строка состояния (информационная строка)* предназначена для показа координат курсора и текущих режимов. Каждый из выводимых режимов может быть переключен щелчком мыши на нем.

## Координаты курсора и системы координат

Текущие значения координат отображаются в левой части строки состояния тремя способами:

- в динамическом режиме, когда обновление координат происходит постоянно по мере перемещения курсора;
- в статическом режиме, когда координаты обновляются только при указании точки;
- в полярной системе координат в виде «расстояние < угол»; этот режим возможен только при рисовании объектов, для которых необходимо указать две или более точки.

Переключение режима производится клавишами «F6» или «Ctrl+D». Он может быть изменен с помощью системной переменной COORDS, которая для указанных режимов соответственно равна 0, 1, 2.

В AutoCAD имеется две системы координат: МСК (WCS) – неподвижная мировая система координат и ПСК (UCS) – подвижная (изменяемая) пользовательская система координат (рис. 19).

В МСК ось  $x$  направлена горизонтально вправо, а ось  $y$  – вертикально вверх. В трехмерных координатах ось  $z$  направлена перпендикулярно поверхности экрана по направлению к пользователю.

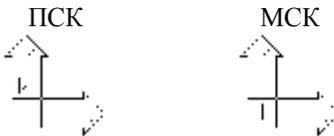


Рис. 19. Пиктограммы систем координат

Все координаты точек в AutoCAD задаются в ПСК. Первоначально МСК и ПСК совпадают. ПСК удобно изменять при работе со сложным элементом чертежа, привязав начало ее координат с характерной точкой этого элемента (например, центра или нижнего левого угла). Это позволит при расчете координат точек не учитывать смещение детали от левого нижнего угла чертежа (начала оси МСК).

Изменение положения ПСК производится командой **UCS (ПСК)** с параметрами New (Новая) и Origin (Начало):

- ◆ Команда меню: «Tools\New USC\Origin» («Инструменты\Создать ПСК\Начало»)

Для возврата в МСК необходимо ввести команду **UCS (ПСК)** с параметром World:

- ◆ Команда меню: «Tools\New USC\MCS» («Инструменты\Создать ПСК\МСК»)

С помощью команды USC и пунктов меню «Инструменты» можно также определить несколько ПСК и переключаться между ними, изменить направления осей ПСК.

Включить или отключить вывод пиктограммы системы координат можно с помощью пункта меню:

◆ Команда меню: «View\Display\UCS Icon\...» («Вид\Отображение\Ярлык ПСК\...»)

или с помощью системной переменной UCSICON, значение 1 которой соответствует включенной пиктограмме, значение 0 – выключенной.

### Вызов команд

Все графические построения в AutoCAD производятся вводом команд и включают два этапа: ввод имени команды и ввод дополнительных параметров (табл. 4):

Таблица 4

Пример ввода команды

Приглашение в командной строке	Вводимая информация	Описание
Command:	Circle ↵	Ввод команды для построения круга
Specify center point for circle:	10,20 ↵	Ввод параметра – центра круга
Specify radius of circle:	10 ↵	Ввод параметра – диаметра круга
Command:		Приглашение для ввода следующей команды

Команды могут быть введены тремя различными способами:

- вводом имени команды в командную строку;
- щелчком пиктограммы команды на панели инструментов;
- выбором команды из главного меню.

Первый из способов является основным, а вторые два предназначены для облегчения работы. Так, после выбора команды в меню или щелчка на ее пиктограмме имя команды всегда появляется в командной строке. В AutoCAD 2000 и 2002 все команды – англоязычные, в том числе и в программе, локализованной для России. Поэтому после выбора русского названия команды в меню, например «Линия», в командной строке появится английское название «Line».

После ввода в командную строку имени команды или ее параметров необходимо нажать клавиши «Enter» или «Пробел». Их действие одинаково, только после нажатия «Enter» ввод продолжается на следующей строке, а после нажатия «Пробел» – в текущей.

После имени команды задаются ее параметры, которые вводятся в командной строке или – для отдельных команд – в появляющемся диалогом окне. Общий вид приглашения к вводу параметра следующий:

Наименование параметра or (или) [Другие варианты] <Значение по умолчанию>

Например, при вводе радиуса окружности выводится приглашение:

Specify radius of circle or [Diameter] <10.0000>

Оно сообщает о необходимости ввести радиус («Specify radius of circle») и предлагает значение по умолчанию (<10.0000>), которое будет принято, если не вводить значение, а нажать клавиши «Enter» или «Пробел».

В качестве параметров могут быть следующие данные:

– число: размер, радиус, число, угол и т. д. Все размеры вводятся в условных единицах чертежа. Угол задается в единицах и в направлении, определенном командой UNITS. По умолчанию используются градусы, отсчет ведется против часовой стрелки;

– координаты точки (point). Указываются координаты по осям  $x$  и  $y$ , разделенные запятой, после чего нажимается клавиша «Enter» или «Пробел», например:

10,20.5 ↵

Кроме абсолютных координат могут быть введены относительные в виде смещения по осям  $x$  и  $y$  от текущего положения (координаты последней введенной точки. При вводе относительных координат перед значениями ставится знак «@»), например:

@2,-2 ↵

Так же могут быть введены и полярные координаты в виде расстояния от текущей точки и направления в градусах, разделенных символом «<»», например:

@2.5<60 ↵

Кроме задания с клавиатуры координаты точки могут быть введены с помощью манипулятора «мышь» (перемещением курсора в необходимое место и нажатием левой кнопки мыши).

Большинство команд AutoCAD может выполняться с различными вариантами параметров. Например, для окружности можно вводить не радиус, а диаметр. Параметры для переключения к другим вариантам ввода приводятся в приглашении в квадратных скобках. Если вариантов несколько, они разделяются наклонной чертой. Для выбора необходимо в ответ на приглашение ввести название варианта полностью или только его прописные буквы.

Например, после ввода команды выбора масштаба ZOOM в командной строке выдается подсказка:

Specify corner of window, enter a scale factor (nX or nXP), or  
[All/Center/Dynamic/Extents/Previous/Scale/Window]

В квадратных скобках приводятся режимы масштабирования. Например, для выбора показа всех графических объектов (вариант «Extens») необходимо полностью ввести слово или ввести только символ «E», выделенный в варианте прописной буквой.

Обратите внимание, что новая команда может быть только когда в командной строке находится приглашение «Command:». Так, в примере, приведенном в табл. 4, после ввода команды CIRCLE ожидается ввод координаты центра окружности. Если вместо этого ввести имя новой команды, то возникнет ошибка. Поэтому для ввода новой команды необходимо завершить ввод всех опций предыдущей команды или прервать ее нажатием клавиш «Esc» или «Ctrl-Break».

Для ускорения работы некоторым командам присвоены укороченные псевдоимена, которые можно вводить вместо полного имени команды. Псевдоимена описаны в файле acad.pgp. Приведем некоторые из них:

A..... ARC	MT..... MTEXT
C..... CIRCLE	OS..... OSNAP
CHA.. CHAMFER	PL ..... PLINE
DT .... DTEXT	POL .... POLYGON
EL..... ELLIPSE	R..... REDRAW
-H..... HATCH	REC .... RECTANGLE
L ..... LINE	SN..... SNAP
LT..... LINETYPE	ST ..... STYLE
-LT.... -LINETYPE	T ..... MTEXT
MI..... MIRROR	Z ..... ZOOM

Для повторного вызова последней команды можно нажать клавишу «Enter». Для прерывания любой команды служит клавиша «Esc».

### Отмена действий

В AutoCAD имеется возможность отмены действия последней или нескольких последних команд.

Команда **U (O)** отменяет последнее действие:

- ◆ Панель инструментов «Standard» («Стандартные инструменты»): 
- ◆ Команда меню: «Edit\Undo» («Редактирование\Отмена»)

Команда **UNDO (ОТМЕНИ)** отменяет несколько последних действий. После вызова команды предлагается ввести количество отменяемых действий (в виде числа) или выбрать другие варианты работы программы:

Enter the number of operations to undo or [BEgin/End/Mark/Back]

Это могут быть следующие варианты:

Begin – отметка начала группы команд, группа потом будет удаляться как одна команда;

End – отметка конца группы команд;

Mark – сохранение состояния чертежа;

Back – возврат к чертежу, сохраненному параметром «Mark».

Команда **OOPS (ОЙ)** восстанавливает объекты, удаленные командой ERASE (СОТРИ).

## 2.2. УСТАНОВКА ПАРАМЕТРОВ И МАСШТАБА ЧЕРТЕЖА

Ряд команд AutoCAD позволяет установить режимы рисования, облегчающие создание чертежа.

Команда **LIMITS (ЛИМИТЫ)** ограничивает размер области, на которой создается чертеж:

◆ Команда меню: «Format\Drawing Limits» («Формат\Лимиты»)

Область задается координатами левого нижнего и правого верхнего углов прямоугольника. Обычно левый нижний угол задается в точке (0,0), а размер области задается по размерам чертежного листа. Размерами области задается и расположение листа – портретное или ландшафтное (рис. 20).

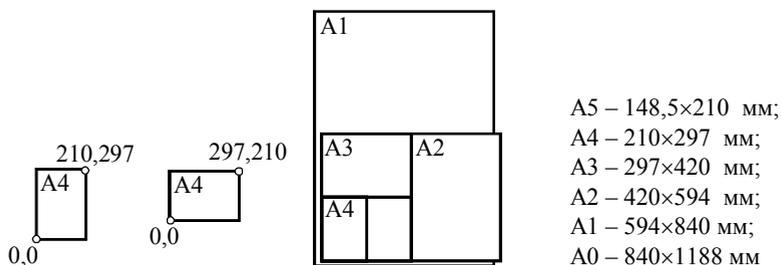


Рис. 20. Формат бумаги

При вводе первого угла области возможно включение и выключение режима ограничения области. При включенном режиме точки, задаваемые за пределами области, игнорируются с выводом сообщения «\*\* Вне лимитов».

Команда **UNITS (ЕДИНИЦЫ)** определяет единицы измерения, используемые в чертеже:

◆ Команда меню: «Format\Units» («Формат\Единицы»)

Определяются:

- формат числа (например, научный или десятичный);
- количество знаков после запятой;
- единицы измерения углов (например, градусы или радианы);
- количество знаков после запятой для углов;
- направление начала отсчета углов;
- отсчет углов по часовой или против часовой стрелки.

При вводе команды в командную строку параметры последовательно запрашиваются, при вызове команды через меню параметры устанавливаются в диалоговом окне.

Команда **SNAP (ШАГ)** используется для повышения точности при работе с мышью. Она включает режим координатной привязки – перемещения с заданным шагом. При этом координаты указателя могут принимать строго определенные значения и изменяются только на величину заданного шага, например на 1 мм. После вызова команды можно задать величину шага и все координаты будут округляться до значения, кратного этой величине. Если шаг 1, координаты могут принимать только целые значения, если шаг 5 – значения 0, 5, 10, 15 и т.д. В команде также может быть выбран один из параметров (табл. 5).

Таблица 5

### Параметры команды SNAP

Параметр	Описание
On Вкл	Включение режима привязки
Off Откл	Выключение режима привязки. Включение и отключение режима производится также клавишей «F9», сочетанием клавиш «Ctrl+B» или двойным щелчком мыши на строке «ШАГ» в информационной панели в нижней части окна AutoCAD. О включении режима информирует яркость строки «ШАГ» в информационной панели
Aspect Аспект	Установление различных шагов по каждой оси
Rotate Поворот	Режим пошагового перемещения курсора под заданным углом к осям
Style Стиль	Позволяет перейти к изометрическому режиму с перемещением курсора по изометрическим плоскостям

Команда **GRID (СЕТКА)** выводит на экран сетку из точек. Сетка предназначена для облегчения создания чертежа, на печать она не выводится. После ввода команды производится задание шага сетки, ввод нуля означает установление шага, равного координатной привязке, добавление после числа английского символа X означает, что задан

коэффициент изменения текущей сетки (например, 2X – увеличение в 2 раза, 0.5X – уменьшение в 2 раза). Другие параметры команды представлены в табл. 6.

Таблица 6

### Параметры команды GRID

Параметр	Описание
On Вкл	Включение сетки
Off Откл	Выключение сетки. Включение и отключение режима производится также клавишей «F7», сочетанием клавиш «Ctrl+G» или двойным щелчком мыши на строке «СЕТКА» в информационной панели в нижней части окна AutoCAD. О включении режима информирует яркость строки «СЕТКА» в информационной панели
Aspect Аспект	Установление различных шагов по каждой оси

Команда **ORTHO (ОПТО)** включает и выключает режим рисования только вертикальных или горизонтальных прямых. Информация о текущем значении показана яркостью строки «ОПТО», расположенной в информационной панели, двойной щелчок мышью на этой строке переключает режим на противоположный. Режим также может быть переключен клавишами «F8» или «Ctrl-O».

Команда **SETVAR (УСТПЕРЕМ)** позволяет просмотреть и изменить значения системных переменных:

- ◆ Команда меню: «Tools\Inquiry\Set Variable» («Сервис\Справка\Переменная»)

Системные переменные определяют текущую настройку AutoCAD – форматы чисел, форматы вывода размеров, режимы и т.д. Всего в AutoCAD около 300 переменных.

После имени команды вводится имя переменной, затем AutoCAD сообщает о ее текущем значении и предлагает ввести новое. Для отказа от изменения значения необходимо нажать клавиши «Enter» или «Пробел». Если вместо имени переменной ввести знак вопроса и шаблон имени, будут выведены все системные переменные и их значения, удовлетворяющие шаблону (например, «\*» обозначает все переменные, «DIM\*» – все переменные, начинающиеся на «DIM»).

### Выбор масштаба чертежа

При работе в AutoCAD масштаб чертежа можно менять.

Команда **ZOOM (ПОКАЖИ)**:

- ◆ Команда меню: «View\Zoom\...» («Вид\Покажи\...»)

увеличивает или уменьшает изображение на экране. После команды вводятся параметры, указанные в табл. 7.

Таблица 7

### Параметры команды ZOOM

Параметр	Описание
Scale factor Масштабный коэффициент nX	Число, задающее абсолютное увеличение, единица соответствует всему изображению
All Все	Относительное увеличение, например 2X, 0.3X, за единицу принято текущее изображение
Extents Границы	На экран выводятся все графические объекты
Window Рамка	Указываются левый нижний и правый верхний углы изображения, которое необходимо увеличить на весь экран
Previous Предыдущий	Восстанавливает предыдущий масштаб изображения
Left (Center) Левый (Центр)	Указывается точка, которая будет левым нижним углом (центром) изображения. После этого запрашивается масштаб изображения
Dynamic Динамика	На экран выводится весь чертеж и рамка, которую можно перемещать и изменять ее размеры. Нажатие левой кнопки манипулятора «мышь» переключает режимы изменения рамки и ее перемещения, нажатие правой кнопки манипулятора «мышь» или клавиш «Enter» или «Пробел» изменяет масштаб

Масштаб изображения может быть изменен при нажатии пиктограммы  панели инструментов «Standard» («Стандартные инструменты»). После этого перемещение манипулятора «мышь» с нажатой левой кнопкой вверх приводит к увеличению масштаба, а вниз – к уменьшению.

Команды REDRAW (ОСВЕЖИ) или REDRAWALL (ВСЕОСВЕЖИ) очищают экран маркеров, отмечающих точки, и других временных элементов:

- ◆ Панель инструментов «Standard» («Стандартные инструменты»): 
- ◆ Команда меню: «View\Redraw» («Вид\Освежи»)

Команда REDRAW очищает текущее устройство вывода, а команда REDRAWALL – все устройства при работе с несколькими мониторами.

Команды REGEN (РЕГЕН) или REGENALL (ВСЕРЕГЕН) производят пересчет экранных координат всех объектов чертежа, после чего чертеж заново выводится на экран:

- ◆ Команда меню: «View\Regen» («Вид\Регенерировать»)

## Работа со слоями

При создании сложного чертежа в AutoCAD можно использовать несколько слоев. Слоем называется часть чертежа для размещения объектов. Слои аналогичны прозрачной кальке, листы которой накладываются друг на друга. Для всех слоев применяется одна и та же система координат. Каждый слой можно сделать невидимым, заморозить (сделать невидимым и игнорировать при регенерации чертежа), заблокировать (запретить изменять объекты), всем объектам слоя можно назначить одинаковый тип линий или цвет для более наглядного представления чертежа. Обычно на отдельных слоях размещают разные детали сложного чертежа, иногда на отдельные слои выносят штриховку и размеры.

Свойства текущего слоя представлены в виде пиктограмм на панели инструментов «Свойства объектов» (рис. 21).



Рис. 21. Панель инструментов «Свойства объектов»

Управление слоями осуществляется командой **LAYER (СЛОЙ)**:

- ◆ Панель инструментов «Object Properties» («Свойства объекта»): 
- ◆ Команда меню: «Format\Layer» («Формат\Слой»)

Команда Layer вызывает диалоговое окно «Layer Properties Manager» («Менеджер свойств слоя»). Для задания параметров в командной строке без вызова диалогового окна используется команда со знаком дефис -LAYER (-СЛОЙ).

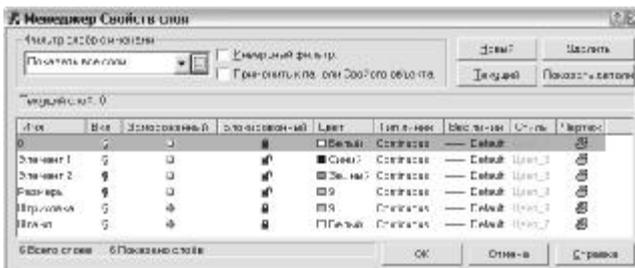


Рис. 22. Диалоговое окно «Менеджер свойств слоя»

В диалоговом окне «Менеджер свойств слоя» (рис. 22) представлен список всех слоев. Для изменения свойств слоя необходимо дважды щелкнуть манипулятором «мышь» на пиктограмме в соответствующей

колонке. Параметры диалогового окна и соответствующие им параметры команды перечислены в табл. 8.

Таблица 8

**Диалоговое окно «Менеджер свойств слоя»**

Параметр	Описание
?	Вывести список имеющихся слоев
Make Создать	Создать новый слой
Set Установи	Изменение текущего слоя на другой, имеющийся на чертеже. После выбора нового слоя на него будут помещаться все вновь создаваемые объекты
New Новый	Создает новый слой и делает его текущим
ON Вкл	Делает слой видимым или невидимым
Freeze (Thaw) Заморозь (Разморозь)	Замораживает (размораживает) слой
Lock (Unlock) Заблокируй (Разблокируй)	Блокирует (разблокирует) слой
Color, Ltype, LWeight Цвет, Типлинии, Веслинии	Устанавливает цвет, тип линии и толщину линии всех объектов слоя, кроме тех, для которых цвет был задан отдельно командой COLOR (ЦВЕТ), тип линии командой LINETYPE (ТИПЛИНИИ)
Plot Чертеж	Включение и выключение вывода слоя на печать

На этой панели инструментов также показаны цвет, тип линий и толщина линий вновь создаваемых объектов. Надпись «ByLayer» («ПоСлою») в этих полях обозначает, что параметры соответствуют параметрам слоя.

Переместить объект на другой слой можно с помощью команды CHANGE (ИЗМЕНИ). Все неиспользуемые слои удаляются командой PURGE (УДАЛИ).

### 2.3. ПОСТРОЕНИЕ ГРАФИЧЕСКИХ ОБЪЕКТОВ

Команда **LINE (ОТРЕЗОК)**:

- ◆ Панель инструментов «Draw» («Рисование»): 
- ◆ Команда меню: «DrawLine» («Черчение\Линия»)

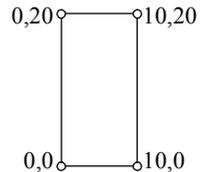
Команда строит отрезок или ломаную линию, состоящую из прямолинейных отрезков. Параметры команды представлены в табл. 9.

### Параметры команды LINE

Параметр	Описание
First point (From point)	Ввод координаты начальной точки. Если в ответ на запрос нажать клавишу «Enter» («Пробел»), построение продолжится от конца предыдущей линии. Если предыдущим объектом была дуга, построение продолжится по касательной к концу этой дуги и будет запрошена длина отрезка (Length of line)
Next point (To point)	Ввод координаты следующей точки. Для отказа от ввода и окончания построения линии необходимо нажать клавишу «Enter» («Пробел») или правую кнопку манипулятора «мышь»
Undo	Удаление последнего отрезка ломаной
Close	Замыкание ломаной (соединение последней точки с начальной). Параметр появляется после построения двух отрезков

Пример построения прямоугольника размером  $10 \times 20$ , где первая и вторая точка задаются с помощью абсолютных координат, а третья и четвертая – с помощью относительных:

```
Command: Line ↵
Specify first point: 0,0 ↵
Specify next point or [Undo]: 10,0 ↵
Specify next point or [Undo]: @0,20 ↵
Specify next point or [Close/Undo]: @-10,0 ↵
Specify next point or [Close/Undo]: C ↵
```



С помощью команды LINE строится линия стандартной ширины. Для вывода линий различной ширины используется команда PLINE, описанная далее.

Также в AutoCAD возможна настройка линий при их печати на плоттере. Для этого каждой толщине линии назначается свой цвет. Цвет вновь создаваемых объектов устанавливается командой COLOR (ЦВЕТ), вызывающей диалоговое окно, или командой -COLOR (-ЦВЕТ), запрашивающей номер цвета в командной строке. Первые семь номеров задают стандартные цвета: 1 – красный; 2 – желтый; 3 – зеленый; 4 – голубой; 5 – синий; 6 – сиреневый; 7 – белый или черный, инверсный относительно фонового цвета графического экрана.

При печати чертежа каждому цвету может быть сопоставлен свой тип линии, в том числе и толщина. Для настройки необходимо вызвать диалоговое окно печати (пункт меню File/Print или Файл/Печать),

группе «Таблица стилей» («Plot Style Table») выбрать существующий стиль печати или создать новый стиль нажатием кнопки «Создать» («New»), затем вызвать окно изменения стиля нажатием кнопки «Изменить» («Edit»), где и задать для нужных цветов требуемую толщину.

Также изменить стиль печати можно при помощи команд меню «Файл\Менеджер стилей чертежей» («File\Plot Style Manager») или «Файл\Параметры страницы» («File\Page Setup»).

Команда **CIRCLE (КРУГ)**:

- ◆ Панель инструментов «Draw» («Рисование»): 
- ◆ Команда меню: «Draw\Circle ...» («Черчение\Круг ...»)

По умолчанию окружность строится по центральной точке (center point) и радиусу, вместо ввода которого можно задать ввод диаметра. Пример построения окружности с центром в точке 100, 100 и диаметром 20:

Command: Circle ↵  
 Specify center point for circle or [3P/2P/Ttr (tan tan radius)]: 100,100 ↵  
 Specify radius of circle or [Diameter]: D ↵  
 Specify diameter of circle: 20 ↵

Вместо ввода первой точки может быть выбран другой режим (табл. 10).

*Таблица 10*

**Параметры команды CIRCLE**

Параметр	Описание
2P	Задаются две точки, лежащие на диаметре 
3P	Задаются три точки, принадлежащие окружности 
Ttr	Строится окружность, касающаяся двух других существующих объектов. В этом режиме необходимо выбрать манипулятором «мышь» два объекта (линии, окружности или дуги), которые будут касательными для окружности, затем задать радиус

Команда **ARC (ДУГА)**:

- ◆ Панель инструментов «Draw» («Рисование»): 
- ◆ Команда меню: «Draw\Arc\ ...» («Рисование\Дуга ...»)

Команда позволяет несколькими способами строить дуги. Возможные варианты построения можно выбрать с помощью команды меню «Рисование\Дуга\...», в этом случае будут запрошены только необходимые данные, режимы команды переключать не надо. При работе с командной строкой режимы переключаются в процессе ввода данных с помощью следующих параметров, представленных в табл. 11. Примеры построения дуги приведены в табл. 12.

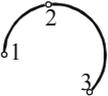
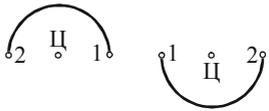
Таблица 11

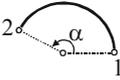
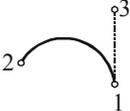
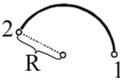
### Параметры команды ARC

Параметр	Описание
Center (Центр)	Центр дуги
Angle (Угол)	Угол, ограничивающий дугу
Direction (Направление)	Касательная к первой точке
Radius (Радиус)	Радиус дуги
End (Конец)	Вторая точка
chord Length (Длина хорды)	Длина хорды, соединяющей начальную и конечную точки

Таблица 12

### Примеры использования команды ARC

Режим построения дуги (подраздел меню «Черчение\Дуга»)	Описание и пример построения с помощью командной строки
1	2
По трем точкам (3 Points; 3 Точки) 	Режим построения дуги по умолчанию Command: Arc Specify start point of arc or [Center]: 5,5 Specify second point of arc or [Center/ENd]: 10,10 Specify end point of arc: 12,0
По центру и двум точкам (Start, Center, End; Начало, Центр, Конец) 	Задаются центр и две точки. Направление дуги от начальной к конечной точке всегда против часовой стрелки Command: Arc Specify start point of arc or [Center]: 20,10 Specify second point of arc or [Center/ENd]: CE Specify center point of arc: 10,10 Specify end point of arc or [Angle/chord Length]: 0,10
По центру, начальной точке и углу (Start, Center, Angle; Начало, Центр, Угол) 	Угол задается в градусах в направлении против часовой стрелки. Если необходимо задать направление дуги по часовой стрелке, вводится отрицательный угол

1	2
	Command: Arc Specify start point of arc or [CEnter]: 20,10 Specify second point of arc or [CEnter/ENd]: CE Specify center point of arc: 10,10 Specify end point of arc or [Angle/chord Length]: A Specify included angle: 140
По центру, точке и длине хорды, соединяющей начальную и конечную точки (Start, Center, Length; Начало, Центр, Длина) 	По умолчанию строится меньшая из двух возможных дуг, то есть угол которой меньше $180^\circ$ . Если же задать длину хорды отрицательной, то будет построена большая дуга.  Command: Arc Specify start point of arc or [CEnter]: 20,10 Specify second point of arc or [CEnter/ENd]: CE Specify center point of arc: 10,10 Specify end point of arc or [Angle/chord Length]: L Specify length of chord: 19
По двум точкам и углу (Start, End, Angle; Начало, Конец, Угол) 	Угол задается в градусах в направлении против часовой стрелки. Если необходимо задать направление дуги по часовой стрелке, вводится отрицательный угол  Command: Arc Specify start point of arc or [CEnter]: 20,10 Specify second point of arc or [CEnter/ENd]: EN Specify end point of arc: 5,15 Specify center point of arc or [Angle/Direction/Radius]: A Specify included angle: 160
По двум точкам и касательной линии (Start, End, Direction; Начало, Конец, Направление) 	Касательная указывается для первой точки и задается точкой, через которую она проходит  Command: Arc Specify start point of arc or [CEnter]: 20,10 Specify second point of arc or [CEnter/ENd]: EN Specify end point of arc: 5,15 Specify center point of arc or [Angle/Direction/Radius]: D Specify tangent direction for the start point of arc: 20,20
По двум точкам и радиусу (Start, End, Radius; Начало, Конец, Радиус) 	Command: Arc Specify start point of arc or [CEnter]: 20,10 Specify second point of arc or [CEnter/ENd]: EN Specify end point of arc: 5,15 Specify center point of arc or [Angle/Direction/Radius]: R Specify radius of arc: 10

Существуют еще три способа построения дуги, в которых первым параметром вводится центр (вместо ввода первой точки необходимо ввести параметр «CE»):

Center, Start, End	Центр, Начало, Конец
Center, Start, Angle	Центр, Начало, Угол
Center, Start, Length	Центр, Начало, Длина

Окружности и дуги так же, как и линии, изображаются линиями стандартной толщины; построить дугу заданной толщины можно командой PLINE.

Команда **PLINE (ПЛИНИЯ)**:

- ◆ Панель инструментов «Draw» («Рисование»): 
- ◆ Команда меню: «Рисование\Полилиния» («Draw\Pline»)

Команда позволяет строить ломаную линию, состоящую из прямых отрезков и дуг с различной толщиной линии. Полилиния рассматривается как один примитив, что облегчает ее копирование и другие виды редактирования. Полилинию необходимо также применять для обеспечения плавного перехода между сегментами различной толщины.

После ввода команды запрашивается первая точка линии. Если в ответ на запрос нажать клавишу «Пробел» («Enter»), в качестве начальной точки будет использована последняя точка предыдущего построения.

После ввода первой точки запрашивается следующая точка полилинии:

Дуга/Замкни/Полуширина/Длина/ОТМени/Ширина/<Конечная точка>  
Arc/Close/Halfwidth/Length/Undo/Width/<Endpoint of line>

Вместо ввода точки можно задать или изменить ширину линии, перейти в режим рисования дуги или выполнить другие действия (табл. 13). Заканчивается построение полилинии вводом вместо координаты точки пустого значения (клавишей «Пробел» или «Enter») или нажатием правой клавиши манипулятора «мышь».

*Таблица 13*

**Параметры команды ПЛИНИЯ**

Параметр	Описание
1	2
Width Ширина	Задание ширины полилинии в начале и конце следующего сегмента. Можно задать два разных значения, тогда ширина полилинии будет изменяться. Если задать два одинаковых значения, толщина полилинии будет постоянной. Все последующие сегменты полилинии будут изображаться с шириной, равной ее последнему указанному значению

1	2
Halfwidth Полуширина	Задание половины ширины полилинии (ширина будет в 2 раза больше указанного значения)
Arc Дуга	Переход в режим рисования дуг. После этого становится возможным использовать параметры построения, аналогичные команде ARC (ДУГА). Если дуга является первым сегментом полилинии, она строится аналогично команде ARC, в противном случае начальной точкой дуги является конечная точка предыдущего сегмента. По умолчанию дуга строится по касательной к предыдущему сегменту. Дополнительно возможно построение дуги по радиусу, центральному углу и направлению хорды. Возврат к построению прямых сегментов осуществляется параметром Line (ОТРЕЗОК)
Length Длина	Изображение линии заданной длины, продолжающей направления предыдущего сегмента. Если предыдущий сегмент был дугой, линия продолжается по касательной к ней
Close Замкни	Замыкание полилинии (соединение последней точки с ее началом)
Undo ОТМени	Отмена изображения последнего сегмента

Пример изображения горизонтальной линии длиной 50 единиц с увеличивающейся шириной от 2 до 5 единиц (рис. 23):

```

Command: PLINE ↵
Specify start point: 100,100 ↵
Current line-width is 0.0000
Specify next point or [Arc/Close/Halfwidth/Length/Undo/Width]: W ↵
Specify starting width <0.0000>: 2 ↵
Specify ending width <2.0000>: 5 ↵
Specify next point or [Arc/Close/Halfwidth/Length/Undo/Width]: 150,100 ↵
Specify next point or [Arc/Close/Halfwidth/Length/Undo/Width]: ↵

```

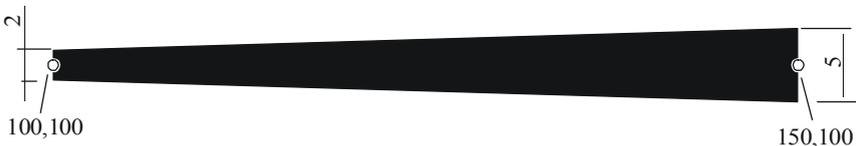


Рис. 23. Полилиния переменной толщины

Команда **LINETYPE** (**ТИПЛИН**):

- ◆ Команда меню: «FormatLineType ...» («Формат/Тип линии ...»)

Позволяет в диалоговом окне «Менеджер типа линий» («Linetype manager») выбрать тип линии, которым будут изображаться все создаваемые далее объекты (рис. 24).

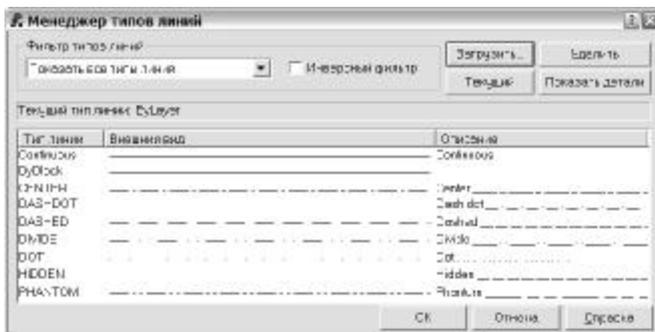


Рис. 24. Диалоговое окно «Менеджер типа линий»

Типы линий, описанные в AutoCAD, можно увидеть на рис. 24. Для каждого типа есть варианты с масштабом в 2 раза меньше и в 2 раза больше, масштаб линии также может быть задан в диалоговом окне «Менеджер типа линий» при нажатии кнопки «Show Details» («Показать детали»).

Для использования тип линии должен быть загружен из библиотеки нажатием кнопки «Load» («Загрузить») диалогового окна «Менеджер типа линий». После загрузки необходимый тип можно выбрать и сделать его текущим нажатием кнопки «Current» («Текущий»).

Типы линий описаны в файле acad.lin. По образцу этого файла можно создать свой файл с типами линий и использовать его в чертежах.

Для выбора типа линий с заданием параметров в командной строке используется системная переменная CELTYPE или команда – **LINETYPE** (**-ТИПЛИН**), параметры которой описаны в табл. 14.

Таблица 14

#### Параметры команды **-LINETYPE**

Параметр	Описание
?	Вывод всех типов линий, находящихся в библиотеке
Create	Создание нового типа линии
Load	Загрузка типа линии из библиотеки
Set	Установка типа линии для создаваемых далее объектов

Установка масштаба всех линий чертежа выполняется командой LТСАLЕ (ЛМАСШТАБ) или с помощью системной переменной LТСАLЕ, а масштаб вновь создаваемых объектов – системной переменной СЕLТСАLЕ.

## 2.4. ОБЪЕКТНАЯ ПРИВЯЗКА

Объектная привязка – это указание при задании точки не ее координат, а определенных точек уже созданных объектов: центров окружностей и дуг, концов и середин линий или их сегментов, пересечения объектов и др. Этот режим обычно используется для задания точных координат при работе с манипулятором «мышь».

Имеется два способа использования объектной привязки:

1. Включение однократной объектной привязки при выборе одной точки. Для этого необходимо в ответ на запрос координаты точки:

– ввести наименование режима объектной привязки (табл. 15);

– или выбрать кнопку на панели «Привязка объекта», соответствующую нужному режиму (см. табл. 15);

– или выбрать режим привязки из контекстного меню, вызываемого правой кнопкой манипулятора «мышь» при нажатой клавише «Shift» (рис. 25).

2. Включение режима автопривязки, выполняемой для каждой вводимой точки, с помощью:

– клавиши «F3»;

– кнопки «Otrack» («ВЫРВ»), расположенной на строке состояния в нижней части окна AutoCAD;

– диалогового окна «Drafting Settings» («Параметры черчения»), закладки «Object Snap» («Объектная привязка»), параметра «Object Snap On» («Автопривязка»). Диалоговое окно «Drafting Settings» («Параметры черчения») вызывается с помощью соответствующего пункта меню из раздела «Tools» («Инструменты») или командой DSETTINGS.

При включенной объектной привязке, если перемещаемый манипулятором «мышь» графический курсор будет находиться вблизи

Свойства привязки:	Свойства привязки:
Начало:	End
Фильтры:	None - End
Контур:	Endpoint
Середина:	Midpoint
Пересечение:	Intersection
Касательная к дуге/окружности:	Perpendicular to Object
Расширение:	Extension
Центр:	Center
Квадрат:	Quadrant
Касательная:	Tangent
Циркуль:	Radius Center
Нарисован:	Radius
Угол:	Angle
Горизонталь:	Horizontal
Вертикаль:	Vertical
Ничего:	None
Режимы привязки:	Object Settings

Рис. 25. Контекстное меню режимов объектной привязки

нужной точки, то она будет отмечена маркером, возле которого будет выведена информация о режиме привязки (рис. 26). Режимы объектной привязки описаны в табл. 15.

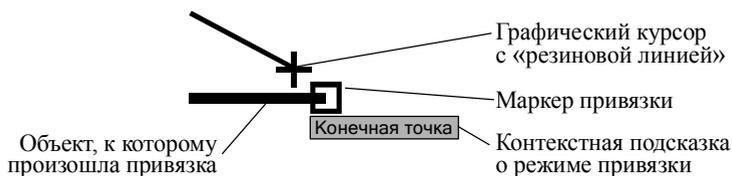


Рис. 26. Объектная привязка

Таблица 15

**Режимы объектной привязки**

Наименование, пиктограмма панели инструментов «Привязка объектов»	Краткое обозначение параметра	Описание
1	2	3
Endpoint Конточка 	END КОН	Привязка к ближайшей конечной точке линии или дуги, полилинии, границе области и трехмерного тела
Midpoint Середина 	MID СЕР	Привязка к середине линии, дуги, полилинии, границе области и трехмерного тела
Intersection Пересечение 	INT ПЕР	Привязка к существующему пересечению двух линий, дуг, окружностей или воображаемому пересечению их продолжений. В последнем случае, если в область графического курсора попал объект, но нет точки его пересечения с другим объектом, будет запрос на указание второго объекта
Apparent Intersect Видимое пересечение 	APP	Привязка для двух непересекающихся объектов в трехмерном пространстве, которые выглядят пересекающимися на текущем виде
Extension Отслеживание 	EXT	Используется совместно с другими режимами для привязки к продолжениям объектов

1	2	3
Center (Центр) 	CEN ЦЕН	Привязка к центру окружности, дуги или эллипса, происходящая при выборе видимой части объекта
Quadrant Квадрант 	QUA КВА	Привязка к ближайшей точке, расположенной под углом 0, 90, 180 или 270° от центра дуги, окружности или эллипса. Если объект находится в блоке, учитывается угол поворота блока. Также учитывается угол поворота системы координат
TANgent Касательная 	TAN КАС	Привязка к точке на окружности или дуге, которая при соединении с последней точкой вычерчиваемого объекта образует касательную
Perpendicular Нормаль 	PER НОР	Если привязка используется для указания первой точки линии или окружности, происходит построение объекта, перпендикулярного объекту, к которому выполняется привязка. Если привязка используется для указания второй точки отрезка или окружности, на объекте, к которому происходит привязка, указывается точка, образующая с первой точкой нормаль (перпендикуляр) к этому объекту
Parallel Параллели 	PAR ПАР	Привязка, создающая линии параллельно существующим объектам
Insertion Точка вставки 	INS ТВС	Привязка к точке вставки текста или блока
Node (Узел) 	NOD УЗЕ	Привязка к точке, созданной командой POINT («ТОЧКА»)
Nearest Ближайшая 	NEA БЛИ	Привязка к точке на линии, дуге или окружности, которая является ближайшей к позиции курсора
Quickly Быстрая	QUI БЫС	Используется совместно с другими режимами привязки для выбора первой из найденных точек привязки, тогда как без этого режима ищутся все возможные точки и выбирается ближайшая к графическому курсору
None (Нет) 	NON НИЧ	Отменяет все режимы объектной привязки

Объектная привязка обычно возможна только для видимых на экране объектов. Ее нельзя использовать для объектов на невидимых или замороженных слоях и для пробелов на штриховых линиях. В командной строке можно задать нескольких режимов привязки, разделяя их запятой, при этом будет выбираться ближайший из возможных вариантов. Переход между активными режимами автопривязки осуществляется клавишей «Tab».

Режим объектной привязки можно использовать для получения координат принадлежащих объектам точек, определения точки пересечения объектов, расчета расстояния между объектами.

Цвет и размер маркера объектной привязки можно установить в диалоговом окне «Drafting Settings» («Параметры черчения») на закладке «Object Snap» («Объектная привязка») или с помощью команды APERTURE (АПЕРТУРА).

## 2.5. ВВОД ТЕКСТА

Текстовая информация на чертеж помещается командами:

**TEXT (ТЕКСТ)** – ввод однострочного текста;

**DTEXT (ДТЕКСТ)** – ввод многострочного текста:

◆ Команда меню: «Draw\Text\MText» («Рисование\Текст\Однострочный»)

**MTEXT (МТЕКСТ)** – ввод многострочного текста с возможностью его форматирования:

◆ Панель инструментов «Draw» («Рисование»): 

◆ Команда меню: «Draw\Text\Dtext» («Рисование\Текст\Многострочный»)

Параметры команд TEXT (ТЕКСТ) и DTEXT (ДТЕКСТ) одинаковы (табл. 16). Форматирование текста в этих командах определяется текущим текстовым стилем.

*Таблица 16*

**Параметры команд TEXT и DTEXT**

Параметр	Описание
1	2
Justify / Style / <Startpoint> Выравнивание / Стиль / <Начальная точка>	Ввод начальной точки, начиная с которой будет вводиться текст, с возможностью перехода к выбору выравнивания или текстового стиля
Height Высота	Высота текста в выбранных единицах измерения. Запрашивается, если в текстовом стиле задана высота 0. Может быть задана числом в текстовом окне команд или визуально на экране с помощью указателя

1	2
Rotation Angle Угол поворота	Угол наклона строки текста: при угле 90° текст располагается вертикально с направлением снизу вверх, при угле 180° текст перевернут. Угол наклона символов задается в текстовом стиле
Text Текст	Ввод текста. Для ввода русских символов необходимо в текстовом стиле задать русифицированный шрифт. В команде TEXT ввод заканчивается по нажатию клавиши «Enter», в команде DTEXT – при двукратном нажатии клавиши «Enter»

При вводе текста автоматически устанавливается выравнивание по левой границе. При выборе параметра «Выравнивание» возможно задание различных режимов, приведенных в табл. 17.

Таблица 17

### Режимы выравнивая текста

Параметр	Описание
Align ВПИсанный	Размещение текста между двумя границами путем изменения размеров символов
Fit Поширине	Задаются две точки – левая и правая границы, текст размещается между ними. Изменяется ширина символов, высота остается равной заданной
Center Центр	Центрирование текста относительно начальной точки (аналогично параметру «BC») <b>ТЕКСТ</b>
Middle Середина	Начальная точка является центром строки как по ширине, так и по высоте (аналогично параметру «MC») <b>ТЕКСТ</b>
Right ВПРаво	Выравнивание текста по правой границе, над начальной точкой всегда будет конец строки (аналогично параметру «BR») <b>ТЕКСТ</b>
TL / TC / TR ВЛ / ВЦ / ВП	Начальная точка сверху текста (символ «Т» – top). По ширине текст выравнивается по левой границе, центру или правой границе <b>ТЕКСТ ТЕКСТ ТЕКСТ</b>
ML / MC / MR СЛ / СЦ / СП	Начальная точка по высоте в центре текста (символ «М» – middle). По ширине выравнивание текста происходит по левой границе, центру или правой границе <b>ТЕКСТ ТЕКСТ ТЕКСТ</b>
BL / BC / BR НЛ / НЦ / НП	Начальная точка снизу текста (символ «В» – bottom). По ширине выравнивание текста происходит по левой границе, центру или правой границе <b>ТЕКСТ ТЕКСТ ТЕКСТ</b>

В тексте, вводимом с помощью шрифтов, поставляемых с системой AutoCAD (например, «txt»), строкой %% обозначаются специальные символы, например:

- %%d – градус (°);
- %%p – допуск (±);
- %%c – диаметр (Ø);
- %%% – процент (%);
- %%nnn – символ с десятичным кодом nnn.

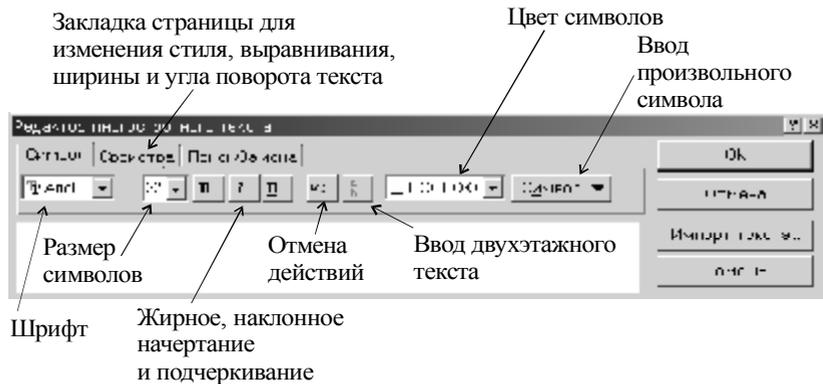
Для других шрифтов это выполняется не всегда, потому что на месте перечисленных символов могут быть записаны другие, например в русифицированных шрифтах – русские символы. Так, в оригинальном шрифте «txt» указанные символы имеются, а в русифицированном шрифте «txt» или шрифте «txtr» их нет.

Команда МТЕХТ (МТЕКСТ) позволяет вводить текст с возможностью разного форматирования отдельных символов и слов. В параметрах задаются две точки – границы текста по ширине. При достижении границы текст будет перенесен на новую строку. Границы текста по вертикали при этом не задаются, и абзац будет продолжен на необходимое количество строк.

При задании второй точки возможен переход к выбору высоты символов, выравнивания текста, угла поворота текста, текстового стиля и заданию ширины текста в виде числа:

Height/Justify/ Rotation/Style/Width  
Высота/Выравнивание/Поворот/Стиль/Ширина

После задания границ на экране появляется окно ввода текста (рис. 27).



**Рис. 27. Окно ввода многостраничного текста**

Для изменения формата вводимого текста необходимо выбрать необходимые шрифт, размер и начертание. Для изменения формата существующего текста его предварительно необходимо выделить (манипулятором «мышь» с нажатой левой кнопкой или клавишами перемещения курсора с нажатой клавишей «Shift»). Например, значку Ø соответствует символ «Ж» шрифта «Symbol». Ввод произвольного символа осуществляется следующим образом.

1. Нажимается кнопка «Символ» и выбирается пункт «Другой ...».
2. В появившемся окне «Таблица символов» выбирается шрифт, отмечаются необходимые символы, затем они вставляются в строку «Копировать символы» путем двойного нажатия левой кнопки мыши или нажатием кнопки «Выбрать».
3. После ввода всех символов они помещаются в буфер обмена нажатием кнопки «Копировать».
4. Окно «Таблица символов» закрывается.
5. В окне AutoCAD «Редактор многострочного текста» вставляется строка из буфера обмена нажатием клавиш «Shift+Insert» (при этом должен быть выбран необходимый шрифт).

Для ввода дроби или одной строчки текста под другой необходимо ввести верхнюю и нижнюю части, разделив их символами «/» для дроби или «^» для двух строчек текста, выделить текст и нажать кнопку .

Команда **STYLE (СТИЛЬ)** создает или изменяет стили, используемые для текста:

- ◆ Команда меню: «Format\Text Styles» («Формат\Текстовые стили»)

По умолчанию в чертеже используется текстовый стиль STANDARD.

При вызове команды появляется диалоговое окно, где можно выбрать или задать новое имя стиля, шрифт и его начертание. Если задать высоту символов, равной нулю, она будет каждый раз запрашиваться в командах TEXT и DTEXT. Если задать высоту, отличную от нуля, в командах ввода текста она запрашиваться не будет.

После изменения стиля он становится текущим. При изменении текстового стиля в чертеже изменится весь текст, введенный ранее с этим стилем.

Текстовые стили можно изменить не в диалоговом, а в командном режиме, для этого используется команда -STYLE (-СТИЛЬ). Параметры этой команды приведены в табл. 18.

### Параметры команды -STYLE

Параметр	Описание
Text style name (or ?) Имя текстового стиля (или ?)	Запрашивается имя стиля. Список всех стилей можно вывести на экран, задав знак «?». Если будет введено имя существующего стиля, то он будет изменен с изменением всего текста, созданного ранее этим стилем
Font style Имя шрифта	Укажите полное имя шрифта, установленного в Windows, или имя файла, содержащего шрифт. В AutoCAD имеются собственные шрифты в файлах с расширением *.shx, находящиеся в подкаталоге «Support» каталога программы
Height Высота	Высота символов. При задании «0» команды ТЕХТ и DTEXT при работе с этим стилем будут запрашивать высоту, при задании другого числа высота запрашиваться не будет
Width Степень сжатия- растяжения	Ширина символов, единица соответствует стандартной ширине
Obliging angle Угол наклона	Угол наклона символов
Backwards? Справа налево?	Ответ «Y» («Д») задает зеркальное отображение (переворот) текста по горизонтали
Upsidedown? Перевернутый?	Ответ «Y» («Д») задает вертикальный вывод текста (символ под символом)

## 2.6. РЕДАКТИРОВАНИЕ ОБЪЕКТОВ ЧЕРТЕЖА

### Выбор объектов

Команды редактирования требуют выбора объектов, который необходимо произвести после приглашения «Выберите объекты» («Select objects»).

Выбрать можно один объект или несколько объектов, называемых набором. Набор формируется последовательным выбором необходимых объектов.

При выборе можно непосредственно указать точку, которая принадлежит объекту. Лучше для этого использовать точки, координаты которых вводились при создании объекта, например концы линий. Если через точку проходят несколько объектов, то выбран будет тот, который создан первым. Также вместо указания точки возможен ввод параметра для перехода в другие режимы выбора (табл. 19).

## Режимы выбора объектов

Параметр	Описание
Window Рамка	Выбираются объекты, полностью попавшие в прямоугольную область, задаваемую двумя точками
Last Последний	Выбирается последний созданный объект
Crossing Секрамка	Выбираются объекты, которые полностью или частично попадают внутрь прямоугольной рамки, задаваемой двумя точками
BOX БОКС	Выбираются объекты, попавшие в прямоугольную область, задаваемую двумя точками. Если точки задаются слева направо, эквивалентно режиму Рамка, если справа налево – режиму Секрамка
ALL Все	Выбор всех объектов текущего слоя
Fence Линия	Выбираются объекты, которые пересекает линия, задаваемая двумя точками
Wpolygon PMн-угол	Выбираются объекты, полностью попавшие внутрь выпуклого (с непересекающимися сторонами) многоугольника произвольной формы
Spolygon CMн-угол	Аналогично режиму PMн-угол, но выбираются также и объекты, частично попавшие в многоугольник
Group Группа	Выбираются объекты из заданной группы. Группы объектов можно создать с помощью команды GROUP (ГРУППА), вызывающей диалоговое окно, или команды -GROUP (-ГРУППА), все параметры которой задаются в командной строке. Каждой группе присваивается имя, по которому выбираются все объекты группы
Add Добавь	Возврат из режима удаления объектов в режим добавления объектов в набор
Remove Удали	Переключение в режим удаления из набора. Выбранные объекты будут удаляться из набора, если они в нем имеются. Для отмены режима удаления и возврата в режим добавления к набору необходимо ввести опцию Add (Добавить)
Multiple Несколько	В этом режиме объекты при выборе на экране не отмечаются, выход из режима осуществляется нажатием клавиши «Enter» («Пробел») в ответ на запрос координат очередной точки. Также режим позволяет выбирать пересекающиеся объекты, задавая несколько раз точку их пересечения
Previous Текущий	Выбираются объекты, отобранные при предыдущем выборе
Undo Отмени	Удаляет заданное число объектов, поступивших в набор последними
Auto Авто	Если введенная точка попадает на объект, этот объект добавляется в набор. Если точка не попадает на объект, осуществляется переход в режим BOX (Режим Auto используется по умолчанию в начале выбора)
Single Единственный	После ввода этого параметра процесс выбора прекращается при нахождении первого (единственного) объекта

После выбора каждого очередного объекта на экран выводится информация:

- selected (выбрано) – количество объектов в наборе;
- found (найдено) – количество найденных последней операцией выбора объектов;
- duplicate (вторично) – количество найденных объектов, которые уже имеются в наборе;
- removed (удалено) – количество удаленных из набора объектов.

Добавление объектов в набор прекращается при нажатии клавиши «Enter» («Пробел») в ответ на запрос выбора нового объекта или новой точки.

Создать набор объектов, который затем будет использоваться с помощью режима «Previous» («Текущий»), возможно с помощью команды SELECT (ВЫБЕРИ).

### Перемещение, копирование и удаление объектов

Команда **MOVE (ПЕРЕНЕСИ)** перемещает объекты:

- ♦ Панель инструментов «Edit» («Редактирование»): 
- ♦ Команда меню: «Изменить\Перенести» («Edit\Move»)

Команда **COPY (КОПИРУЙ)** копирует объекты, оставляя на месте оригиналы:

- ♦ Панель инструментов «Edit» («Редактирование»): 
- ♦ Команда меню: «Изменить\Копировать» («Edit\Copy»)

После выбора объектов запрашиваются параметры (табл. 20).

Таблица 20

Параметры команд MOVE и COPY

Параметр	Описание
1	2
Base point or displacement Базовая точка или перемещение	Ввод базовой точки, относительно которой будет задаваться перемещение, или смещения объектов по осям $x$ и $y$
Second point of displacement Вторая точка перемещения	Запрос второй точки, куда будет перемещен или скопирован объект так, чтобы базовая точка заняла место указанной второй точки



1	2
	Если в ответ на запрос нажать клавишу «Enter» («Пробел»), то введенные при запросе первой точки значения будут пониматься как смещение по осям $x$ и $y$ от исходного положения объекта
Multiple Несколько	Параметр может быть выбран при запросе базовой точки. В этом случае вторая точка запрашивается несколько раз, и в каждой указанной точке создается новый объект. Копирование прекращается при нажатии клавиши «Enter»

Команда **ARRAY (МАССИВ)** создает несколько копий выбранных объектов, расположенных по кругу или в несколько горизонтальных и вертикальных рядов:

- ◆ Панель инструментов «Draw» («Рисование»): 
- ◆ Команда меню: «Edit\Array» («Изменить\Массив»)

Сначала предлагается выбрать объект, затем запрашивается способ создания массива:

Rectangular/polar array (<R>/<C>).  
Прямоугольный или круговой массив (<П>/<К>)

Для прямоугольного массива запрашиваются:

- число строк (number of rows);
- число столбцов (number of columns);
- размер ячейки или расстояние между строками (unit cell of distance rows);
- расстояние между столбцами (distance between columns).

Исходный объект будет расположен в углу полученного массива. Если задаются положительные расстояния между строками и столбцами, новые объекты располагаются справа и вверху относительно исходного элемента, если отрицательные – слева и снизу.

Для кругового массива запрашивается:

- центр массива (centre point of array);
- число элементов (number of items)
- угол заполнения (angle to fill), угол между первым и последним элементами, при положительном значении угла новые элементы создаются в направлении против часовой стрелки относительно исходного элемента, если угол отрицательный – в направлении по часовой стрелке;
- поворачивать объекты при копировании (Rotate objects as the are copied).



Системная переменная `dimt1h` управляет направлением текста на размерных линиях. Если `dimt1h=1` (Да), текст всегда будет горизонтальным, если `dimt1h=0` (Нет), текст при зеркальном отображении будет вдоль размерной линии.

Команда **SCALE (МАСШТАБ)** осуществляет изменение размеров объектов. После выбора объектов запрашивается точка, которая при изменении масштаба будет сохранять свое положение, а также масштабный коэффициент (размеры будут умножаться на это число):

Base point (Базовая точка)  
<Scale factor>/Reference (<Масштаб>/Ссылка)

При выборе варианта «Ссылка» указываются старый размер и новый размер.

Команда **STRETCH (РАСТЯЖИ)** позволяет перемещать часть объектов, сохраняя их связи с другими объектами чертежа.

Вначале предлагается выбрать объекты с помощью опций Window (Рамка) или Crossing (Кроссрамка):

Select object to stretch by window  
(Выберите растягиваемые объекты с помощью рамки)

После этого выбора в набор можно добавлять или удалять объекты обычным способом:

Выберите объекты (Select object)

После окончания выбора предлагается указать базовую точку и ее новое местоположение:

Base point (Базовая точка)  
New point (Новая точка)

Все концы линий, дуг, полос, сегментов, полилиний, попавшие в рамку выбора, перемещаются, а не попавшие – остаются на месте.

### Редактирование объектов

Команда **CHANGE (ИЗМЕНИ)** позволяет изменять размеры, положения и свойства линий, окружностей, текста. В команде необходимо указать изменяемые объекты и точку изменения:

Select object (Выберите объекты)  
Properties /<Change point> (Параметры /<точка изменения>)

*Выбран отрезок:* конец отрезка, ближайший к точке, используемой при выборе отрезка (или при отсутствии данной точки, ближайшей к точке изменения), переместится в точку изменения. При включении режима ORTHO конец отрезка переместится до уровня, на котором расположена точка изменения.

*Выбрана окружность:* ее диаметр изменится таким образом, что точка изменения будет лежать на окружности. Если не вводить точку изменения (нажать пробел), будет запрошен новый радиус окружности.

*Выбран текст:* можно задать новую точку привязки (которая будет являться точкой изменения), стиль, высоту, угол наклона и ввести новый текст.

Если вместо точки изменения выбрать опцию Properties, то возможно изменение следующих свойств выбранных объектов:

- Цвет (Color);
- LineType (Тип линии);
- Высота (Thickness).

Команда **PEDIT (ПОЛПРЕД)** предназначена для преобразования в полилинии отрезков и дуг и редактирования полилиний. Вначале необходимо указать полилинию для редактирования:

Select polyline (Выберите полилинию)

Если выбран отрезок или дуга, произойдет запрос о преобразовании его в полилинию. После выбора возможны следующие действия, представленные в табл. 21.

Таблица 21

### Параметры команды PEDIT

Параметр	Описание
Close (Замкни)	Делает полилинию замкнутой
Open Разомкни	Размыкает полилинию, построение которой было закончено опцией «Замкни»
Join Присоедини	Присоединяет к незамкнутой полилинии отрезки, дуги и другие незамкнутые полилинии, если их начальные точки являются конечной точкой текущей полилинии
Width Ширина	Изменяется ширина полилинии, новая ширина указывается числом или двумя точками
Fit curve Сглаживание	Заменяют полилинию гладкой кривой. Отменяется операцией Decurve
Spline curve Аппроксимация	Заменяет полилинию кривой, описываемым квадратным (если системная переменная Splinetype = 5) или кубическим (Splintype = 6) полиномом. Отменяется опцией Decurve
Undo Отмена	Отменяет предыдущую операцию или преобразование отрезка и дуги в полилинию
Edit vertex Вершина	Редактирование сегментов (узлов) полилинии
Exit Выход	Осуществляет выход из команды

Команда **BREAK (РАЗОРВИ)** удаляет часть отрезка, окружности, дуги, полилинии между двумя указанными точками.

Если при выборе объекта указывалась точка, ему принадлежащая, она принимается в качестве первой и сразу закрашивается. Если все же необходимо ввести первую точку, вместо ввода второй точки можно выбрать параметр «F» («П»).

Преобразование полилиний в отдельные отрезки и дуги осуществляется с помощью команды **EXPLODE (РАСЧЛЕНИ)**.

### **Сопряжение нескольких объектов**

Команда **TRIM (ОБРЕЖЬ)** производит обрезание объектов по указанной границе, которая указывается после запроса. В команде сначала выбираются режущие кромки – отрезки, дуги, окружности и полилинии (поэтому такие объекты необходимо создать до использования команды TRIM). Затем закрашиваются объекты, которые необходимо обрезать. После окончания выбора выбранные объекты обрезаются по линии, задаваемой режущими кромками. Удаляется та часть объекта, которая была указана при выборе.

Команда **EXTEND (УДЛИНИ)** продлевает отрезки, незамкнутые полилинии и дуги до пересечения с указанной границей. В ответ на запрос граничных кромок указываются объекты – отрезки, дуги, окружности и полилинии. Затем указывается удлиняемый объект, после чего произойдет удлинение ближайшего к указанной точке конца объекта до ближайшей границы.

Команда **FILLET (СОПРЯГИ)** соединяет два объекта (отрезки полилинии, дуги окружности) дугой заданного радиуса. При необходимости объекты продлеваются до точки соединения или обрезаются. После ввода или команды необходимо выбрать два объекта для сопряжения и параметры:

- Полилиния (Polyline), режим сглаживания углов полилинии;
- Радиус (Radius), задание радиуса сглаживающей дуги.

Сопрягаться будут ближайшие к указанным при выборе точкам концы объектов. Если задать радиус сопряжения ноль, будет построено пересечение объектов с обрезанием ненужных частей.

Команда **CHAMFER (ФАСКА)** обрезает две прямые на заданном расстоянии от точки их пересечения и соединяет полученные концы отрезком. Команда запрашивает первую и вторую прямую. Вместо выбора первого отрезка возможны другие варианты:

- Полилиния (Polyline), обрезание углов полилинии;
- Длина (Distance), задание величин фасок, последовательно запрашивается длина 1-й и 2-й фасок.

При задании фасок, равных 0, происходит сопряжение прямых или удаление фасок из полилиний.

## Получение информации о чертеже

Команда **AREA (ПЛОЩАДЬ)** позволяет вычислить площадь и периметр заданной области. Площадь может задаваться тремя или более точками. Окончание ввода точек осуществляется нажатием клавиш «Пробел» или «Enter». Выбор параметра

Entity (Примитив)

позволяет вычислить площадь внутри окружности или замкнутой полилинии, выбор объектов предлагается сделать после ввода параметра.

При выполнении команды могут быть рассчитаны площади ряда фигур с накоплением рассчитанных значений площадей. Режим наполнения регулируется опциями:

- Добавить (Add), последующие площади будут прибавляться к общей сумме;
- Вычесть (Subtract), последующие площади будут вычитаться из общей суммы.

Таким образом может быть рассчитана площадь сложных фигур, имеющих выемки.

После вычисления выдается следующая информация:

- площадь (Area);
- периметр (Perimeter);
- общая площадь (Total area).

Общая площадь и периметр последней области после выполнения команды содержатся в системных переменных Area и Perimeter.

Команда **DBLIST (БДСПИСОК)** выводит полный список объектов, содержащихся в чертеже. Команда **LIST (СПИСОК)** выводит информацию о выбранных объектах. Приостановить выдачу списка можно комбинацией «Ctrl+C» или клавишей «Esc».

Команда **1D (КООРД)** выводит координаты точки, указанной на рисунке манипулятором «мышь» или относительными координатами. Команда **DIST (ДИСТ)** вычисляет расстояние между двумя указанными точками. В командах 1D и DIST возможно использование объектной привязки для расчета различных координат и значений.

## 2.7. ПРОСТАНОВКА РАЗМЕРОВ

В AutoCAD имеются следующие типы размеров (рис. 29):

- 1) линейные горизонтальные и вертикальные, а также параллельные заданной линии;
- 2) линейные цепные;
- 3) указывающие радиус и диаметр;
- 4) базовые, у которых совпадает одна из выносных линий;
- 5) угловые.

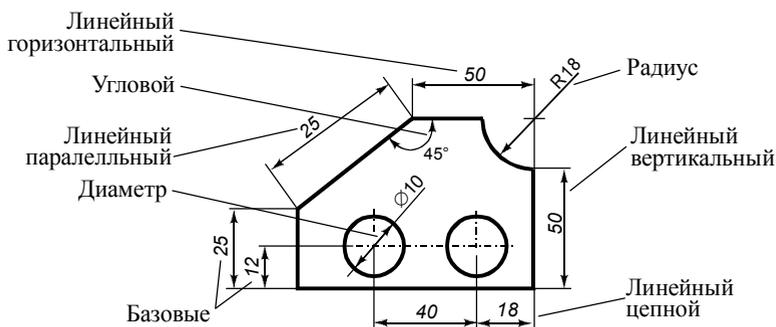


Рис. 29. Типы размеров

Каждый размер состоит из ряда элементов (рис. 30).



Рис. 30. Элементы размера

Размеры можно строить двумя способами:

а) вводом команды для построения нужного размера;

б) переходом в режим построения размеров командой **DIM** (РАЗМЕР). Во втором случае приглашение к вводу команды «Command» («Команда») заменяется на приглашение «Dim» («Размер»). В этом режиме сначала вводится параметр, соответствующий нужному типу размера, а затем все необходимые дополнительные параметры. Параметр Undo (Отмени) позволяет последовательно отменить последние действия в этом режиме. Переход в режим ввода команд осуществляется параметром Exit (Выход).

Размеры привязываются к объектам чертежа или могут создаваться самостоятельно, без связи с другими примитивами. Построение размеров включает следующие стадии:

- ввод команды или параметра для построения необходимого размера;
- указание начала выносных линий или объекта, для которого размер будет строиться;
- указание положения размерной линии, которая будет проходить через заданную точку. Положение размерного текста относительно выносных линий также определяется этой точкой;
- при необходимости ввод размерного текста.

Основные команды построения размеров и параметры, задаваемые в команде DIM (РАЗМЕР), приведены в табл. 22.

Таблица 22

### Построение размеров

Тип размера	Команда для построения размера	Параметр в режиме построения размеров	Примечание
1	2	3	4
Линейный	DIMLINEAR РЗМЛИНЕЙНЫЙ	Horizontal Горизонтальный Vertical Вертикальный	Требует ввода трех точек для начала выносных линий и размерной линии. В ответ на запрос о первой точке можно нажать пробел и выбрать объект, в том числе дугу или окружность, для которой будет задан размер. Вид размера, вертикальный или горизонтальный, определяется расположением объекта или заданных точек для выносных линий. Дополнительными параметрами можно задать вид размера (вертикальный или горизонтальный), что необходимо, если автоматически вид определяется неправильно, а также угол поворота размерного текста (параметр «Угол») и размерной линии (параметр «Повернутый»)
Параллельный	DIMALIGNED РЗМПАРАЛ	Aligned Парал	Размерная линия параллельна указанному объекту или прямой, на которой расположены начальные точки выносных линий. Дополнительно можно задать угол поворота размерного текста (параметр «Повернутый»)
Ординатный	DIMORDINATE РЗМОРДИНАТА	Ordinate Ордината	Указывает расстояние по вертикали или горизонтали между двумя точками, например расстояние между базовой точкой и объектом
Радиус	DIMRADIUS РЗМРАДИУС	Radius Радиус	Требует указания окружности или дуги; через указанную при выборе точку будет проходить размерная линия. Затем запрашивается положение размерного текста. Положение точки (внутри или снаружи дуги) определяет вид размерной линии

1	2	3	4
Диаметр	DIMDIAMETER РЗМДИАМЕТР	Diameter Диаметр	Требует указания окружности или дуги. Затем запрашивается положение размерного текста. Положение точки (внутри или снаружи дуги) определяет вид размерной линии
Угловой	DIMANGULAR РЗМУГЛОВОЙ	Угловой Angular	<p>Определяет угол между двумя линиями, размерная линия имеет вид дуги. Может быть построен тремя способами:</p> <p>а) выбрать дугу или окружность, причем точка выбора будет являться первой точкой угла, затем задать вторую точку угла. Центр угла будет совпадать с центром объекта;</p> <p>б) выбрать два отрезка или два сегмента ломаной;</p> <p>в) в ответ на запрос выбора объектов нажать клавишу «Пробел» и указать три точки – вершину угла и две точки его сторон</p>
Базовый	DIMBASELINE РЗМБАЗОВЫЙ	Baseline Базовый	Перед построением базового или цепного размера необходимо нанести первый линейный размер. После этого базовые и цепные размеры будут строиться от этого размера
Цепной	DIMCONTINUE РЗМЦЕПЬ	Continue Цепь	

К командам простановки размера можно отнести следующие:

**LEADER (ВЫНОСКА)** – вводится поясняющий текст для элемента чертежа;

**TOLERANCE (ДОПУСК)** – указывает допустимые отклонения размеров при изготовлении деталей.

Размерным текстом называется любой текст, связанный с размерами, в том числе размерные числа, допуски (как размерные, так и допуски формы и расположения), префиксы, суффиксы, а также одно- и многострочные пояснительные надписи. По умолчанию системой предлагается значение, соответствующее указанному элементу или введенным положениям выносных линий. Точность и единицы измерения предлагаемого значения могут быть заданы предварительно командой **UNITS (ЕДИНИЦЫ)**. Вместо текста, предлагаемого по умолчанию, можно ввести свое значение или вообще отказаться от наличия текста.

Для однострочного размерного текста используется текстовый стиль, установленный в диалоговом окне «Надписи» команды **DDIM**

**(ДИАЛРАЗМ)**. Многострочный текст пишется текущим текстовым стилем с учетом всех модификаций, заданных в текстовом редакторе.

Вид выводимых размеров определяет ряд системных переменных (табл. 23). Они могут быть изменены в диалоговом окне, выводимой командой DDIM (ДИАЛРАЗМ) или с помощью команды изменения системных переменных SETVAR (УСТПЕРЕМ).

Таблица 23

**Системные переменные, влияющие на построение размеров**

Системная переменная	Описание
DIMTAD (P3MTHPJL)	Вертикальное положение размерного текста: 0 – на размерной линии; 1 – сверху размерной линии; 2 – снаружи размерной линии
DIMTIH (P3MTMEJG)	Расположение текста: Off (Нет) – параллельно размерной линии; On (Да) – всегда горизонтально
DIMTIX (P3MTMB)	Расположение текста относительно выносных линий: Off (Нет) – выбирается системой автоматически; On (Да) – текст всегда располагается между выносными линиями
DIMTOFL (P3MRJMB)	Расположение текста за выносными линиями: Off (Нет) – выбирается системой автоматически; On (Да) – текст всегда располагается за выносными линиями
DIMTSZ (P3MBJ3AC)	Если переменная равна 0, на концах размерной линии изображаются стрелки, если больше 0 – засечки, причем число определяет их размер
DIMASZ (P3MBJCTP)	Размер стрелки на концах размерной линии
DIMTXT (P3MTEKCT)	Размер текста
DIMRND	Величина округления размеров
DIMSCALE (P3MVCIN)	Общий масштабный коэффициент. При его применении изменяются все размеры чертежа
DIMLFAC	Масштабный коэффициент, используемый для расчета новых размеров относительно единиц чертежа

## 2.8. ШТРИХОВАНИЕ

В AutoCAD можно использовать ассоциативную и неассоциативную штриховку. При ассоциации штриховка связана с объектами чертежа, которые являются ее границами. При изменении границ штриховка также изменяется. Неассоциативная штриховка не зависит от контура границы и является отдельным самостоятельным объектом.

Команда **НАТЧН (ШТРИХ)** позволяет наносить только неассоциативную штриховку. Команда выполняется только из командной строки, после ее вызова запрашиваются следующие параметры:

а) Enter a pattern Name (выбор типа штриховки). Возможен выбор одного из следующих параметров:

– ввод имени типа штриховки, список доступных стилей вызывается при вводе символа «?»;

– Solid – выбор сплошной штриховки;

– User Defined – штриховка параллельными линиями (в этом случае дополнительно запрашивается угол наклона, расстояние между линиями и использовать ли двойную штриховку);

б) Select object (Выбор границ штриховки). Границы должны представлять отрезки, дуги или полилинии, область штриховки должна быть замкнутой (в противном случае могут быть заштрихованы другие области чертежа). При нажатии клавиши «Пробел» в ответ на запрос границ после дополнительного вопроса осуществляется переход в режим построения полилинии, которая и будет являться границами штриховки. При построении полилинии ее необходимо завершать командой «Замкни».

Команда **ВНАТЧН (КШТРИХ)** позволяет наносить ассоциативную и неассоциативную штриховку:

♦ Панель инструментов «Редактирование»: 

♦ Команда меню: «DrawHatch» («Рисование\Штриховка»)

Параметры команды (тип, масштаб и др.) задаются в диалоговом окне. Для задания области штриховки в этой команде указывается точка, принадлежащая штрихуемой области (кнопка «Указание точек»), или выбираются объекты, являющиеся границами (кнопка «Выбор объектов»). При нажатии кнопки «Дополнительные опции» можно выбрать, как будут обрабатываться области, расположенные внутри штрихуемой (группа «Стиль контура»):

а)  – нормальный стиль, при наличии нескольких вложенных друг в друга контуров происходит чередование заштрихованных и не заштрихованных областей;

б)  – внешний стиль, штрихуется только внешняя область, все внутренние области не заштрихованы;

в)  – игнорирующий стиль, штрихуется вся область со всеми внутренними объектами.

Для ввода параметров команды без вызова диалогового окна с использованием только командной строки при вызове команды перед ней указывается дефис: -ВНАТСН (-КШТРИХ).

## 2.9. БЛОКИ

Блоком называется набор графических объектов, имеющий имя. Созданный блок можно неоднократно вставлять в чертеж. После объединения объектов в блок он может быть неоднократно вставлен в текущий чертеж с указанием масштаба по каждой оси и угла поворота или записан на диск, а потом вставлен в другой чертеж. В каждом блоке есть базовая точка (точка привязки), при вставке блока запрашивается положение этой точки в чертеже.

Создание блоков является средством автоматизации черчения, позволяющим дублировать однотипные объекты или использовать записанные ранее на диск блоки. С помощью блоков на диске может быть создана библиотека чертежей или типовых элементов.

Рассмотрим команды для работы с блоками.

Команда **BLOCK (БЛОК)** создает в текущем чертеже блок:

- ◆ Панель инструментов «Draw» («Рисование»): 
- ◆ Команда меню: «Draw\Block\Создать» («Черчение\Блок\Make»)

После вызова команды появляется диалоговое окно, где необходимо:  
– ввести имя блока, если введено имя существующего блока, то после выполнения команды все ранее вставленные блоки с таким именем будут в чертеже заменены на новые;

– задать координаты базовой точки («Base point»);

– выбрать объекты («Selected objects»), при этом возможны следующие варианты:

а) Оставить (Retain) – сохранить объекты в чертеже после создания блока;

б) Сделать блоком (Convert to block) – заменить объекты на блок;

в) Удалить (Delete) – удалить объекты из чертежа.

Команда **BLOCK (БЛОК)** записывает блок на диск. После вызова команды появляется диалоговое окно, в котором можно выбрать существующий блок или, как в команде **BLOCK**, объединить в блок объекты, а также задать имя и местоположение файла, в котором блок будет записан.

Команда **INSERT (ВСТАВЬ)** вставляет блок в чертеж:

- ◆ Панель инструментов «Draw» («Рисование»): 
- ◆ Команда меню: «Insert\Block» («Вставка\Блок»)

В появившемся диалоговом окне выполняются следующие действия:

- выбор существующего в текущем чертеже блока или файла, записанного на диск (это может быть как ранее записанный блок, так и любой другой чертеж, базовая точка чертежа по умолчанию 0,0 и может быть изменена командой BASE (БАЗА));

- выбор точки вставки (Insertion point), в которую будет помещена базовая точка блока;

- задание масштабного коэффициента (Scale) по каждой из осей. Отметив параметр «Uniform Scale» («Равные масштабы») можно задать единый масштаб для всех осей. Значение больше 1 означает увеличение, от 0 до 1 – уменьшение, меньше 0 – зеркальное отображение по оси;

- задание угла поворота (Rotation) блока относительно точки вставки.

Если в диалоговом окне отметить параметр «Explode» («Расчленить»), то после вставки блок будет разбит на отдельные объекты.

Для каждого из параметров вставки блока может быть выбран режим «Specify On-screen» («Указать на экране»); в этом случае после закрытия диалогового окна системой AutoCAD запрашиваются параметры для выбора точки вставки, общего коэффициента масштабирования и коэффициентов для каждой из осей, угла поворота и режима буксировки:

Specify insertion point or [Scale/X/Y/Z/Rotate/PScale/PX/PY/PZ/PRotate]  
Точка вставки или [Масштаб/X/Y/Z/Поворот/ПМасштаб/ПХ/ПУ/ПЗ/ППоворот]

Параметры, начинающиеся с символа «P», влияют на отображение (буксировку) блока, показываемого около курсора в момент вставки. Если блок имеет большие размеры, с помощью параметров буксировки его изображение можно уменьшить для более удобного выбора места вставки.

Для вставки блока без использования диалогового окна используется команда **-INSERT (-ВСТАВЬ)**. Команда запрашивает имя блока, если блока с заданным именем не существует, AutoCAD будет искать файл с таким именем на диске.

Команда **MINSERT (МВСТАВЬ)** позволяет вставить в чертеж блок в виде массива, дополнительные параметры команды аналогичны команде ARRAY (МАССИВ).

После объединения объектов в блок они становятся недоступны для изменения или удаления. Для редактирования блока его необходимо разделить на составные объекты командой EXPLODE (РАСЧЛЕНИ), провести необходимые действия, а затем снова объединить в блок, сохранив старое имя, при этом во всех ранее вставленных блоках будут произведены такие же изменения.

Для организации работы с блоками и чертежами, записанными на диск, применяется AutoCAD DesignCenter, вызываемый:

- командой ADCENTER;
- клавишами «Ctrl+2»;
- командой меню Tools\AutoCAD DesignCenter (Инструменты\Центр проектирования AutoCAD);
- пиктограммой  панели инструментов «Стандартная»).

AutoCAD DesignCenter позволяет искать, просматривать и использовать чертежи, записанные на диск, или их элементы. Например, из выбранного чертежа можно поместить в текущий чертеж отдельные объекты или блоки. Это позволяет создавать библиотеки, содержащие в одном чертеже множество блоков.

### 3. ЯЗЫК ПРОГРАММИРОВАНИЯ AutoLISP

Язык программирования AutoLISP является частью системы AutoCAD. Он может быть использован для самых разных целей – от простых арифметических вычислений в процессе работы в среде AutoCAD до создания сложных систем автоматического построения чертежей.

В языке AutoLISP есть все стандартные средства языка программирования – условные переходы, логические проверки, циклы, переменные. Это позволяет перейти от создания чертежа в графическом режиме к описанию методики построения чертежа, которую потом можно применять с заданием разнообразных исходных данных и конструктивных параметров. Кроме того, AutoLISP позволяет изменять и дополнять саму систему AutoCAD – вводить в нее новые команды, изменять систему меню, модифицировать систему помощи. Это является основой для разработки на основе AutoCAD систем автоматического проектирования для конкретных прикладных задач.

Язык программирования Lisp был разработан в 60-х гг. XX в. для решения задач искусственного интеллекта. Он является функциональным языком, то есть программа представляет собой описание и вызов различных функций. Функции могут быть как элементарные (встроенные), так и описанные программистом. Язык AutoLISP является модернизацией языка Lisp для работы в системе AutoCAD. В AutoCAD 2000 и более поздних версиях появился VisualLISP – среда программирования, предназначенная для упрощения процесса создания программ и объединяющая редактор для ввода текста программ на AutoLISP и средства для отладки программ.

#### 3.1. ОСНОВНЫЕ ПОНЯТИЯ ЯЗЫКА AutoLISP

##### Списки и функции

Элементарной конструкцией AutoLISP является список. Список – это набор разделенных пробелами параметров, заключенный в круглые скобки. Можно привести следующие примеры списков:

(12 15 4.5)

("San" "Man")

Любым из параметров списка может быть другой список, причем глубина вложенности неограниченна

(1 2 (3 4))

Если первым элементом списка является имя функции, то этот список называется выражением. За именем функции в выражении идет спи-

сок аргументов, передаваемых функции. В качестве аргументов могут быть конкретные значения, списки, другие выражения, но их тип должен строго соответствовать типу требуемого в данном месте аргумента.

В качестве примера приведем выражение для вычисления суммы двух чисел 2 и 3 и расчета квадратного корня из 25:

```
(+ 2 3)
(Sqrt 25)
```

В первом случае вызывается функция сложения, имеющая имя "+", и ей передается два числа – 2 и 3. Во втором случае функции Sqrt передается один аргумент – число 25. После выполнения функции выражение возвращает результат вычисления. Если выражение не может быть вычислено или не должно давать никакого результата, то возвращается nil (специальное значение, обозначающее отсутствие или неопределенность данных).

Если в качестве аргумента функции требуется ввести число, вместо него может быть записано выражение, возвращающее значение требуемого типа. Например, для расчета выражения  $\sqrt{2+3}$  записывается функция Sqrt, в которой вместо аргумента – выражение для вычисления суммы

```
(Sqrt (+ 2 3))
```

Если необходимо рассчитать гипотенузу прямоугольного треугольника со сторонами 4 и 3 по математическому выражению  $\sqrt{4 \cdot 4 + 3 \cdot 3}$ , то выражение AutoLISP будет иметь вид



Допускается любой уровень вложенности, например выражение

```
(+ 2 3 (* 4 (- 5 (* 6 7))))
```

эквивалентно арифметической записи  $2 + 3 + 4(5 - 6 \cdot 7)$ .

Для одних функций число аргументов фиксировано, для других количество аргументов не ограничено. Например, в рассмотренной функции сложения "+" число аргументов произвольно:

```
(+ 2 3 4) – возвращает 9
(+) – возвращает 0
(+ 7) – возвращает 7
```

Функции AutoLISP для математических расчетов описаны в табл. 24.

### Функции AutoLISP для записи математических выражений

Функция	Описание и математический эквивалент
<b>Арифметические функции</b>	
(+ a b ...)	Сумма, $a + b + \dots$
(- a b)	Разность, $a - b$
(- a)	Изменение знака, $-a$
(- a b c ...)	Вычитание из первого аргумента всех остальных, $a - b - c - \dots$ или $a - (b + c \dots)$
(* a b ...)	Произведение, $ab$
(/ a b)	Частное, $a/b$
(/ a b c)	Деление первого аргумента на сумму всех остальных, $a/(b + c)$
(1+ a)	Увеличение аргумента на единицу, $a + 1$
(1- a)	Уменьшение аргумента на единицу, $a - 1$
<b>Математические функции</b>	
(Abs a)	Модуль, $ a $
(Exp a)	Экспонента, $e^a$
(Expt a b)	Степень, $a^b$
(Log a)	Натуральный логарифм, $\ln a$
(Sqrt a)	Квадратный корень, $\sqrt{a}$
(Max a b ...)	Максимальное число из списка аргументов
(Min a b ...)	Минимальное число из списка аргументов
<b>Тригонометрические функции</b>	
(Sin a)	Синус, $\sin a$
(Cos a)	Косинус, $\cos a$
(Atan a)	Арктангенс, $\arctg a$
(Atan a b)	Арктангенс от частного аргументов, $\arctg(a/b)$
<b>Функции преобразования</b>	
(Fix a)	Преобразование аргумента в целое число отбрасыванием дробной части, $[a]$
(Float a)	Преобразование аргумента в вещественное число, например в выражениях, в которых результат должен быть вещественным
(Rem a b)	Остаток от деления $a$ на $b$ , $a \bmod b$
(Gcd a b)	Наибольший общий делитель чисел $a$ и $b$
<b>Функции для геометрических расчетов</b>	
(точки – списки из трех или двух значений с координатами $x$ , $y$ и $z$ )	
(Polar t a r)	Возвращает координаты точки, лежащей от заданной точки $t$ на расстоянии $r$ с углом $a$ , измеряемым в радианах и отсчитываемым от оси $x$
(Angle t1 t2)	Угол в радианах наклона прямой, проходящей через точки $t1$ и $t2$ к оси $x$
(Distance t1 t2)	Расстояние между точками $t1$ и $t2$
(Inters t1 t2 t3 t4)	Координата точки, являющейся пересечением двух прямых, проходящих через точки $t1$ , $t2$ и через точки $t3$ , $t4$

Необходимо обратить внимание на то, что результат арифметических выражений будет такого же типа, который имеют его аргументы. Если в выражении используются целые числа, результат также будет целым, а полученная дробная часть будет отброшена. Поэтому целые числа, которые участвуют в расчете вещественных выражений, необходимо записывать с дробной частью, например «5.0». Это свойство демонстрируется следующим примером:

```
( / 5 3) ; Результат – 1
( / 5.0 3.0) ; Результат – 1.66667
( / (float 5) (float 3)) ; Результат – 1.66667
```

Для получения вещественного результата достаточно, чтобы хотя бы один из аргументов был вещественным. Поэтому рассмотренный выше пример можно записать в следующем виде:

```
( / 5.0 3) ; Результат – 1.66667
( / 5 3.0) ; Результат – 1.66667
```

Но лучше целые числа всегда записывать в вещественном виде. Например, необходимо рассчитать выражение  $(\frac{2}{3} + 1)/4$ , которое равно 0,4167. Если записать его в виде

```
( / (+ ( / 2 3) 1) 4.0),
```

в AutoLISP выражение будет рассчитываться в следующем порядке:

- 1) расчет частного  $\frac{2}{3}$ , результатом которого будет 0, так как аргументы функции целые;
- 2) прибавление к частному единицы, результат 1;
- 3) деление суммы на 4, результат 0,25.

Таким образом, результат выражения будет вещественным, но неверным.

### Комментарии

В текст программ могут быть вставлены комментарии, которые поясняют назначение отдельных выражений, порядок использования программы, размерность величин и т.д. Они могут начинаться в любом месте строки с символа «точка с запятой», после которого весь текст до конца строки рассматривается как комментарий. Например,

```
; Расчет размеров отверстий
(SetQ D (* 0.25 L) ); Диаметр отверстий составляет 25% от длины
```

При разработке программ в их тексте необходимо обязательно использовать комментарии, поясняющие назначение переменных и выполняемые программой действия. Это значительно облегчит написание и отладку сложных программ, а также их модификацию в дальнейшем.

### Запись координат точек

Координаты точки в системе AutoCAD – это не отдельные значения координат по осям  $x$ ,  $y$  и  $z$ , а список из этих трех значений. Для создания списков из отдельных значений существует специальная функция List, поэтому, если в программе AutoLISP или в ответ на запрос AutoCAD требуется задать координаты точки, необходимо использовать следующую форму записи:

(List  $x$   $y$ ) или (List  $x$   $y$   $z$ )

где  $x$ ,  $y$ ,  $z$  – числа или выражения для их расчета. Если координата  $z$  в списке опущена, она принимается равной нулю.

Например,

Команда: ОТРЕЗОК  
От точки: (List 0 0)  
К точке: (List 50 75)

### Типы и инициализация переменных

В AutoLISP могут быть использованы переменные различных типов. Тип переменной является динамическим. Он автоматически определяется при инициализации переменной в соответствии с присваиваемым ей значением и может быть в дальнейшем изменен, если переменной будет присвоено значение другого типа. В AutoCAD существуют следующие типы:

Int – целый;  
Real – вещественный;  
Str – строковый;  
Sym – символьный;  
List – список.

Для присвоения переменным значений существует стандартная функция AutoLISP SetQ. Первым аргументом функции является имя переменной, вторым – значение переменной. Например, алгебраические выражения  $X = 1$  и  $S = "A"$  в AutoLISP будут выглядеть следующим образом:

(SetQ X 1)  
(SetQ S "A")

Одна функция SetQ может использоваться для присвоения значений нескольким переменным, например

(SetQ X 1 S "A")

Вместо значений переменных могут использоваться выражения, функции, списки, причем допускается любой уровень вложенности.

Например, переменной  $Y$  может быть присвоен результат вычисления выражения  $5\cos(\pi/3)$ :

```
(SetQ Y (* 5 (cos (/ 3.1415 3))))
```

Переменные, которым присвоены значения, могут использоваться в качестве аргументов функций и элементов списков. Например, введем диаметр круга – переменную  $D$  и рассчитаем площадь круга  $S$ :

```
(SetQ D 5)
(SetQ S (/ (* 3.1415 D D) 4))
```

Если переменной присвоить значение `nil`, она уничтожается, освобождая занимаемую память, и больше ее использовать будет невозможно. Например, после расчета площади переменная  $D$  может быть удалена из памяти:

```
(SetQ D nil)
```

Для просмотра значения переменной из командной строки AutoCAD необходимо ввести восклицательный знак и за ним имя переменной, например

```
!X
!Y
```

В консольном окне VisuaLISP (см. подразд. 3.2) достаточно просто ввести имя переменной.

Функция `Eval` вычисляет и возвращает значение выражения, указанного в качестве аргумента. Ее надо использовать, если в программе в качестве возвращаемого значения выражения должно быть значение переменной, например

```
(Eval X)
(Eval Y)
```

### Ссылки на переменные

Кроме значения, хранящегося в переменной, возможно сослаться и на саму переменную. Ссылка на переменную  $X$  возвращается функцией

```
(Quote X)
```

В тексте программы на AutoLISP допускается упрощенная форма записи ссылки:

```
'X
```

Ссылка обозначает место в памяти, где хранится данная переменная.

Принципиальное отличие значения переменной и ссылки на переменную рассмотрим на примере. Координата точки в AutoCAD представляется в виде списка. Пусть в двух переменных  $x$  и  $y$  содержатся соответствующие значения координат. Создадим переменную  $p1$ , содержащую координаты точки

```
(SetQ p1 (list x y) )
```

Переменной  $p1$  будут присвоены численные значения переменных  $x$  и  $y$ , которые установлены на момент выполнения выражения. Определение координаты  $p1$  через ссылки на переменные  $x$  и  $y$

```
(SetQ p1 (list 'x 'y) )
```

означает, что координатами точки  $p1$  будут являться переменные  $x$  и  $y$ , и при изменении этих переменных соответственно изменится и  $p1$ .

Ссылаться можно не только на переменные, но и на списки и выражения. Так, запись

```
(SetQ b '(+ x y) )
```

обозначает, что  $b$  есть выражение  $b = x + y$ . С помощью функции вычисления выражений `Eval` в любой момент можно вычислить сумму  $x + y$  с использованием текущих значений этих переменных. Например, определим переменные

```
(SetQ x 1)
```

```
(SetQ y 2)
```

```
(SetQ b '(+ x y) )
```

```
(Eval b)
```

```
(SetQ x 10)
```

```
(Eval b)
```

При первом вызове функция `(Eval b)` вернет значение 3, при втором – 12.

### **Ввод значений переменных с клавиатуры**

В программе на AutoLISP значения переменных или координаты точек могут быть введены с клавиатуры. Для этого существует ряд специальных функций, приведенных в табл. 25.

### Функции ввода данных с клавиатуры

Функция	Описание
(GetInt Текст)	Ввод целого числа. Параметр «Текст» выводится в командную строку для подсказки пользователю о необходимых действиях.
(GetReal Текст)	Ввод вещественного числа
(GetString Флаг Текст)	Ввод строки. Если флаг отсутствует, ввод заканчивается при нажатии клавиш «Пробел» или «Enter». Если флаг установлен (им может быть любое значение, отличающееся от nil), строка может содержать пробелы и ввод заканчивается при нажатии клавиши «Enter»
(GetKWord Текст)	Ввод ключевых слов, определенных перед этой функцией с помощью функции InitGet (описание см. ниже)
(GetPoint Точка Текст)	Ввод координаты точки. Если задан параметр «Точка» (в виде переменной или списка из значений координат $x$ , $y$ , $z$ , образованного функцией list), то перемещение курсора на экране сопровождается «резиновой линией» из этой точки. Если параметр «Точка» не задан, «резиновая линия» не изображается. Координаты могут быть введены с клавиатуры или с помощью мыши. Пример функции: (getpoint (list 10 25 0) "Введите точку ") или (getpoint "Введите точку ")
(GetCorner Точка Текст)	Ввод координаты точки. При задании параметра «Точка» перемещение курсора на экране сопровождается «резиновым прямоугольником»
(GetAngle Точка Текст)	Ввод значения угла между линией, проведенной через две точки, и осью $x$ неподвижной системы координат. Если параметр «Точка» задан, необходимо ввести только вторую точку, если нет – ввести две точки
(GetOrient Точка Текст)	В отличие от GetAngle учитывает изменение оси в текущей системе координат
(GetDist Точка Текст)	Ввод расстояния между двумя точками. Необходимо ввести вторую точку, если задан параметр «Точка», или две точки в противном случае

В программе введенное значение, чтобы оно не было потеряно, необходимо присвоить переменной.

Примеры:

– ввод целого числа, которое присваивается переменной *n*:

```
(SetQ n (GetInt "Введите число точек ") )
```

– ввод точки, значение введенной точки присваивается переменной *p1*:

```
(SetQ p1 (GetPoint "Введите первую точку ") )
```

– ввод точки с изображением на экране резиновой линии от точки *p1*, значение введенной точки присваивается переменной *p2*:

```
(SetQ p2 (GetPoint p1 "Введите вторую точку ") )
```

– ввод точки с изображением на экране резиновой линии от начала координат, значение введенной точки присваивается переменной *p3*:

```
(SetQ p3 (GetPoint (list 0 0) "Введите вторую точку ") )
```

Работой функций ввода можно управлять с помощью функции

```
(InitGet Флаг КлючевыеСлова)
```

InitGet действует на одну следующую функцию ввода. Флаг задает тип вводимого значения: 1 – запрещено вводить пустое значение, нажимая в ответ на запрос клавиши «Пробел» или «Enter»; 2 – запрет ввода нуля; 4 – запрет ввода отрицательных чисел; 8 – отмена контроля попадания точек в границы чертежа; 16 – возврат трехмерных координат; 32 – «резиновые» линии и прямоугольники изображаются пунктиром.

Ключевые слова задают список допустимых слов, которые могут вводиться в функции GetKWord (подобно списку возможных режимов команд AutoCAD). Слова разделяются пробелами. Если для слова необходимо предусмотреть краткую форму ввода, требуемая часть слова записывается заглавными символами или краткая форма приводится через запятую после полной формы, например

```
(InitGet 7 "ВЕНтиль ЗАДвижка ШИБер")
```

или

```
(InitGet 7 "Вентиль,ВЕН ЗАДвижка,ЗАД ШИБер,ШИБ ")
```

Символ 7 в примере запрещает ввод любых других значений, кроме ключевых слов.

## Вывод информации на экран

Для вывода информации на экран могут быть использованы функции, представленные в табл. 26.

Таблица 26

### Функции вывода на экран

Функция	Описание
(Prompt Текст)	Вывод на экран текстовой строки
(Print Аргумент)	Вывод на экран значения переменной или выражения, начиная с новой строки, после выведенного текста добавляется пробел
(Print Аргумент)	Вывод на экран значений переменной или выражения, начиная с текущей позиции без перехода на новую строку
(PrintC Аргумент)	Вывод на экран значений переменной или выражения, начиная с текущей позиции без перехода на новую строку. Строки выводятся без кавычек
(Write-Char Код)	Вывод символа по заданному коду, например для вывода специальных символов, отсутствующих на клавиатуре
(Terpri)	Переход на новую строку
(GrText Зона Текст Подсветка)	Вывод текста в области экрана согласно значению параметра «Зона»: 0 – в строку меню, -1 – в информационную строку, -2 – в строку координат экрана. Если необязательный параметр «Подсветка» не равен нулю, текст будет выделен. (GrText) восстанавливает значение измененной области
(Alert Текст)	Выводит на экран окно с указанным текстом. Для разделения текста на строки используются символы «\n».

Функции Prompt и PrintC допускают использование в составе строк управляющих символов: \n – переход на новую строку; \t – переход на начало текущей строки; \t – табуляция, перемещение на 10 позиций.

Пример использования функций вывода приведен в табл. 27.

Таблица 27

### Пример вывода на экран

Выражения	Выводимая информация
(Prompt "Введите ") (Prompt "числа\n?")	Введите числа ?
(Prompt "Введите температуру, ") (Write-Char 176) (Prompt "C ")	Введите температуру, °C
(Print (+ 1 3) ) (Print (+ 2 5) ) (Print (+ 1 3) ) (Print (+ 2 5) )	4 7 47
(Print "Строка") (PrintC "Строка")	"Строка" Строка

## Запись сложных выражений

Если записывать сложные математические выражения, функция AutoLISP может получиться довольно громоздкой. Но язык AutoLISP не накладывает ограничения на количество пробелов внутри списков и позволяет размещать список на нескольких строках. Это дает возможность структурировать программу, делая ее текст более наглядным. Ряд правил позволяет облегчить составление сложных математических выражений.

1. Перед составлением программы исходное математическое выражение необходимо проанализировать, определив порядок его расчета, и записывать выражение на AutoLISP согласно этому порядку.

2. Для списков каждая открывающая скобка должна иметь закрывающую. Соответствующие пары скобок лучше записывать или на одной строке, или в разных строках друг под другом, а каждую функцию помещать в отдельной строке. Например, для выражения  $x = \frac{1 + 0,78 \cdot 25}{1 + 0,36 \cdot 50}$  соответствующая функция AutoLISP будет иметь вид

```
(SetQ x
  (/
    (+ 1 (* 0.78 25))
    (+ 1 (* 0.36 50))
  )
)
```

Такое правило записи позволяет наглядно показать глубину действия и баланс скобок.

При записи выражения для каждой открывающейся скобки сразу же следует записывать закрывающуюся, и только потом вписывать параметры внутри этой пары скобок. Это позволит всегда соблюдать правильный баланс скобок.

3. Выражения можно разбить на части, введя для каждой части промежуточную переменную, рассчитать значения этих переменных, а потом с их использованием – значение выражения. Например, для приведенной выше дроби можно ввести отдельные переменные  $r1$  и  $r2$  для числителя и знаменателя, рассчитать их, а затем определить их частное:

```
(SetQ r1 (+ 1 (* 0.78 25))
(SetQ r2 (+ 1 (* 0.36 50))
(SetQ x (/ r1 r2))
(SetQ r1 nil)
(SetQ r2 nil)
```

Две последние функции уничтожают временные переменные и освобождают память, отведенную для них (это необходимо делать при недостатке памяти и работе в AutoCAD для среды DOS).

### Функции для работы с текстом

Часто бывает необходимо вывести на чертеж текстовую информацию, включающие значения переменных. Но команды ТЕКСТ, ДТЕКСТ и МТЕКСТ могут выводить только данные типа STR (строка). Поэтому все числа для использования в таких операторах необходимо преобразовать в текстовый вид и сформировать из них и дополнительного текста одну строку. Функции для выполнения таких операций представлены в табл. 28.

Таблица 28

### Функции для работы со строками

Функция	Описание
1	2
<b>Преобразование чисел в текст</b>	
(Itoa Целое число)	Преобразует целое число в строку текста
(Rtos Вещественное Число Формат Точность)	Преобразует вещественное число в строку текста. Формат совпадает с возможными значениями системной переменной AutoCAD LUNITS и может быть: 1 – научный (мантисса и порядок, разделенные латинским символом «E»); 2 – десятичный; 3 – инженерный; 4 – архитектурный; 5 – с дробной частью (см. пункт меню AutoCAD «Формат\Единицы»). Точность определяет количество знаков после запятой. Формат и точность могут не указываться, тогда будут использованы настройки AutoCAD
(AngTos Угол Формат Точность)	Преобразует числовое значение угла в строку. Формат совпадает с возможными значениями системной переменной AutoCAD AUNITS и может быть: 0 – градусы; 1 – градусы, минуты, секунды; 3 – градусы; 4 – радианы; 5 – топографические единицы (см. пункт меню AutoCAD «Формат\Единицы»). Параметр «Точность» задает число знаков после запятой. Формат и точность могут не указываться, тогда будут использованы настройки AutoCAD
(Chr Число)	Возвращает символ по его коду
<b>Преобразование текста в числа</b>	
(Atol Строка)	Преобразовывает число, записанное в текстовом виде, в целое число, если строка числом не является, возвращает нуль

1	2
(AtoF Строка)	Преобразует текстовую запись числа в вещественное число, если строка числом не является, возвращает нуль
(Ascii Строка)	Возвращает код первого символа в заданной строке
<b>Объединение и разделение строк</b>	
(StrCat Строка1 Строка2 ...)	Объединяет несколько строк в единую строку
(SubStr Строка Начало Длина)	Выделяет из строки часть, начиная с символа с номером «Начало». Если параметр «Длина» не задан, выделяется один символ. Если параметр «Длина» превышает возможное число символов или не задан, выделяется текст до конца строки
(StrLen Строка)	Возвращает длину (число символов) строки

Например, в программе описана переменная  $D$  целого типа, значение которой надо использовать при выводе следующего текста:

Диаметр горелки  $D$  мм

Выражения для формирования и вывода такой строки будут следующие:

```
; Преобразование значения в текстовый вид
(SetQ S1 (Itoa D))
; Соединение трех строк в одну
(SetQ S2 (StrCat "Диаметр горелки " S1 " мм"))
; Вывод на экран строки S2 из точки с координатами 10,10
; Высота текста 7, угол наклона 0
(Command "ТЕКСТ" "10,10" "7" "0" S2)
```

### Функции для работы со списками

Для работы со списками в AutoLISP есть ряд функций, позволяющий образовывать списки и извлекать из них отдельные элементы, которые представлены в табл. 29.

Таблица 29

### Функции AutoLISP для работы со списками

Функция	Описание и математический эквивалент
1	2
(List Элемент1 Элемент 2 ...)	Возвращает список из элементов Элемент1, Элемент2, ..., указанных в качестве аргументов

1	2
(Cons Элемент Список)	Добавляет «элемент» в начало «Списка» и возвращает обновленный список
(Append Список1 Список2 ...)	Объединяет все элементы списков, указанных в качестве аргументов, в единый список. В отличие от функции Append, функция List элементы из списков, указанных в качестве аргументов, не выделяет, а образует новый список с элементом – другим списком
(Reverse Список)	Переставляет элементы списка в обратном порядке и возвращает обновленный список
(Length Список)	Возвращает длину списка
(Car Список)	Возвращает первый элемент списка
(Last Список)	Возвращает последний элемент списка
(Nth Номер Список)	Возвращает элемент списка с заданным номером, нумерация начинается с нуля
(Member Элемент Список)	В списке ищется элемент и возвращается новый список, от найденного элемента до конца исходного списка, если элемент не найден, возвращается nil
(Cdr Список)	Удаляет первый элемент и возвращает обновленный список

Рассмотрим пример обработки списка.

1. Создадим список из пяти чисел и поместим его в переменную  $L$ :

(SetQ L (List 10 11 12 13 14) )

2. Пометим в переменную  $E1$  элемент списка с номером один, то есть число 11:

(SetQ E1 (Nth 1 L) )

3. Удалим из списка  $L$  второй элемент (имеющий номер один). Для этого с помощью функции Member создадим новый список, включающий элементы от второго до конца исходного списка  $L$ , затем функцией Cdr удалим из этого списка первый элемент, функцией Cons добавим в полученный усеченный список первый элемент исходного списка  $L$  и результат поместим обратно в переменную  $L$ :

(SetQ L (Cons (Car L) (Cdr (Member E1 L) ) ) )

Если необходимо сортировать список, переставлять, удалять и добавлять элементы, необходимо использовать функции организации циклов.

## Логические функции и функции управления порядком выполнения программы

AutoLISP имеет функции для организации условных операторов (то есть выполнения или невыполнения определенных действий в зависимости от заданных условий) и циклов (многократного повторения заданных действий). В этих функциях используются логические выражения. Если условие выполняется, то результатом логического выражения будет «Т», то есть true (истина), если условие не выполняется, выражение возвращает nil (пустое значение). Логические операции включают сравнение, логические операции и проверку типа аргумента.

Функции сравнения приведены в табл. 30.

Таблица 30

### Функции сравнения

Функция	Описание
(= a b ...)	Результат Т, если все элементы равны, иначе nil. Аргументами могут быть два и более числа или строки
(/= a b)	Результат Т, если элементы не равны, иначе nil. Аргументы – два числа или две строки
(< a b ...) (<= a b ...) (> a b ...) (>= a b ...)	Результат Т, если каждый следующий элемент больше (больше или равен, меньше, меньше или равен) предыдущего, иначе nil
(ZeroP a)	Возвращает Т, если аргумент равен 0, иначе – nil
(MinusP Число)	Возвращает Т, если аргумент – отрицательное число, иначе – nil
(Equal a b Допуск)	Сравнивает числа, строки, списки, координаты точек. Если аргументы равны, возвращает Т, иначе – nil. Необязательный параметр «Допуск» определяет точность сравниваемых чисел (допустимую разность между ними)

Пример записи функций сравнения:

```
(SrtQ x1 5 x2 5.01)
(= x1 x2) ; Результат nil
(Equal x1 x2) ; Результат nil
(Equal x1 x2 0.1) ; Результат Т
(> x1 0) ; Результат Т
```

Функции AutoLISP для логических операций приведены в табл. 31. С помощью логических операций можно учитывать одновременно несколько условий. Например, если необходимо проверить двухстороннее неравенство  $0 \leq x \leq 1$  ( $x \in [0; 1]$ ), составляются два логических условия:  $x \geq 0$  и  $x \leq 1$ . Двухстороннее условие будет истинно, если будут выполняться они оба:

(And (>= x 0) (<= x 1))

**Функции логических операций**

Функция	Описание
(And a b ...)	Логическое «И». Возвращает Т, если все аргументы равны Т, если хотя бы один из них равен nil, возвращает nil
(Or a b ...)	Логическое «ИЛИ». Возвращает Т, если хотя бы один из аргументов не равен nil, если они все равны nil, также возвращает nil
(Not a)	Логическое «НЕТ». Если аргумент равен nil, возвращает Т, во всех других случаях возвращает nil

Если же надо проверить условие непопадания значения  $x$  в интервал от 0 до 1, то есть что  $x \notin [0; 1]$ , можно использовать функцию Or или not:

$$\begin{aligned} &(\text{not } (\text{And } (>= x 0) (<= x 1))) \\ &(\text{Or } (< x 0) (> x 1)) \end{aligned}$$

Ряд функций AutoLISP позволяет проверять тип аргумента (табл. 32). Эти функции применяются в тех случаях, когда возможный тип аргумента заранее может быть неизвестен, например, при работе с существующими объектами чертежа или при разработке новых функций для анализа полученных ими аргументов (см. пример на с. 118).

**Функции проверки типа аргумента**

Функция	Описание
(Atom a)	Возвращает nil, если аргумент является списком, иначе – Т
(BoundP a)	Возвращает Т, если аргумент не равен nil, иначе возвращает nil
(Null a)	Возвращает Т, если аргумент равен nil, иначе возвращает nil
(NumberP a)	Возвращает Т, если аргумент целое или вещественное число, иначе – nil

Для выполнения тех или иных действий в зависимости от результатов проверки условий применяется функция условного выполнения If и функция выбора Cond (табл. 33). В этих функциях в каждом варианте для выполнения может быть записано только одно выражение. Если требуется записать не одно, а несколько выражений, их необходимо объединить функцией Progn.

### Функции условного выполнения и выбора

Функция	Описание
(Cond (Условие1 Выражение1) (Условие2 Выражение2) ... )	Последовательно проверяет условия, выполняет первое из выражений, условие для которого не равно nil
(If Условие (Выражение1) (Выражение2) )	Если условие не равно nil, выполняется выражение 1, иначе – выражение 2
(Progn (Выражение1) (Выражение2) ... )	Объединяет несколько выражений в одно. Используется, когда в функциях Cond, If, While, Repeat необходимо выполнить не одно выражение, а несколько

Пусть требуется рассчитать критерий Нуссельта по выражению

$$Nu = c (PrGr)^n,$$

где  $c = 1,18$ ;  $n = 0,125$  при  $GrPr < 1000$ ;  $c = 0,5$ ;  $n = 0,25$  при  $GrPr \geq 1000$ .

С помощью функции if это выражение может быть записано следующим образом:

```
(SetQ GrPr (* Gr Pr))
(if (< GrPr 1000)
  (SetQ Nu (* 1.18 (expt GrPr 0.125) ) )
  (SetQ Nu (* 0.50 (expt GrPr 0.250) ) )
)
(SetQ GrPr nil)
```

В примере для упрощения текста программы использована временная переменная GrPr, которая после расчета критерия Nu уничтожается. С помощью оператора Cond это же выражение будет выглядеть следующим образом:

```
(SetQ GrPr (* Gr Pr))
(Cond
 (< GrPr 1000)
  (SetQ Nu (* 1.18 (expt GrPr 0.125) ) )
 (>= GrPr 1000)
  (SetQ Nu (* 0.50 (expt GrPr 0.250) ) )
)
(SetQ GrPr nil)
```

Рассмотрим случай, когда в условном операторе необходимо записать несколько выполняемых выражений. Пусть заданы переменные  $L$  и  $H$  – длина и высота детали и переменные  $XMax$  и  $HMax$  – размеры области построения (координаты ее верхнего правого угла). Необходимо рассчитать координаты  $x$  и  $y$  левого нижнего угла детали так, чтобы она разместилась в центре области построения (рис. 31). Но если деталь не помещается в этой области, координаты  $x$  и  $y$  необходимо задать равными нулю и вывести сообщение об этом на экран.

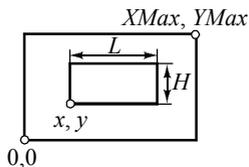


Рис. 31. Пример построения

Программа для решения такой задачи следующая:

```
(If (And (< L XMax) (< H YMax) )
; Действия при выполнении условия
(Progn
  (SetQ x (/ (- Xmax L) 2) )
  (SetQ y (/ (- Ymax H) 2) )
)
; Действия при невыполнении условия
(Progn
  (SetQ x 0 y 0)
  (print "Деталь больше области построения")
  (Terpri)
)
)
```

Для выполнения циклических вычислений используются функции, представленные в табл. 34.

Таблица 34

### Функции организации циклов

Функция	Описание
(While Условие Выражение)	Выражение выполняется до тех пор, пока условие не станет равным nil. Функция While возвращает результат, полученный при последнем выполнении выражения. Условие проверяется перед каждым выполнением выражения, поэтому если при первом вызове функции условие равно nil, выражение выполняться не будет. В выражении должны меняться данные, используемые в условии, иначе цикл никогда не закончится
(Repeat Число Выражение)	Выражение выполняется заданное число раз. Число (в качестве которого можно использовать переменную или выражение) должно быть целым

Рассмотрим пример построения  $N$  окружностей радиусом  $r$ , центры которых равномерно распределены на окружности радиусом  $r_0$ , имеющей центр в точке с координатами  $x_0, y_0$  (рис. 32). Центр первой из создаваемых окружностей должен быть в верхней точке окружности радиусом  $r_0$ . Текст программы для выполнения этой задачи следующий:

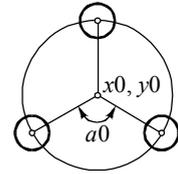


Рис. 32. Пример построения

```
(SetQ N 7 r 10 r0 100 x0 100 y0 100)
; a0 – угол между центрами соседних окружностей
(SetQ a0 (/ (* 2 Pi) N))
; a – угол между центрами первой и текущей окружностей,
; для первой окружности равен 0
(SetQ a 0)
; Цикл из N повторений
(repeat N
  (progn
    ; расчет координат центра текущей окружности по выражениям
    ; x = x0 + r sin a
    ; y = y0 + r cos a
    (SetQ x (+ x0 (* r0 (sin a))))
    (SetQ y (+ y0 (* r0 (cos a))))
    (SetQ p (list x y))
    ; Построение окружности
    (Command "CIRCLE" p r)
    ; Увеличение угла для построения следующей окружности
    (SetQ a (+ a a0))
  )
)
```

В качестве примера применения оператора While рассмотрим упрощенную задачу размещения компенсаторов температурного расширения на трубе. Пусть имеется линия, координаты концов которой  $x_1, y_1$  и  $x_2, y_2$ , причем  $x_2 > x_1$ . Необходимо на этой линии через расстояние  $z$  построить окружности радиусом  $r$ . Общее число окружностей при этом заранее неизвестно и определяется длиной линии и расстоянием  $r$ . Пример решения этой задачи:

```
(SetQ x1 30 y1 30 x2 100 y2 0 z 20 r 5)
; Угол наклона линии
(SetQ a (atan (- y2 y1) (- x2 x1)))
; Расстояние между центрами окружностей по горизонтали и вертикали
(SetQ zx (* z (cos a)))
(SetQ zy (* z (sin a)))
```

```

; Координаты центра первой окружности
(SetQ x x1 y y1)
; Цикл будет выполняться, пока координата центра окружности
; не станет больше координаты конца линии
(While (< x x2)
  ; Построение окружности
  (Command "Circle" (list x y) r)
  ; Координаты середины следующей окружности
  (SetQ x (+ x xz) y (+ y zy))
)

```

### 3.2. РАБОТА В СРЕДЕ VISUALLISP

VisualLISP – это среда программирования, включающая язык AutoLISP, средства создания графического интерфейса программ, редактор для написания программ и средства их отладки.

Вызов VisualLISP осуществляется из AutoCAD командой VLIDE или пунктом меню

Tools\AutoLISP\Visual LISP Editor  
Инструменты\AutoLISP\Редактор Visual LISP

#### Окна VisualLISP

Среда VisualLISP содержит несколько окон (рис. 33), переключение между которыми происходит в разделе меню Window.

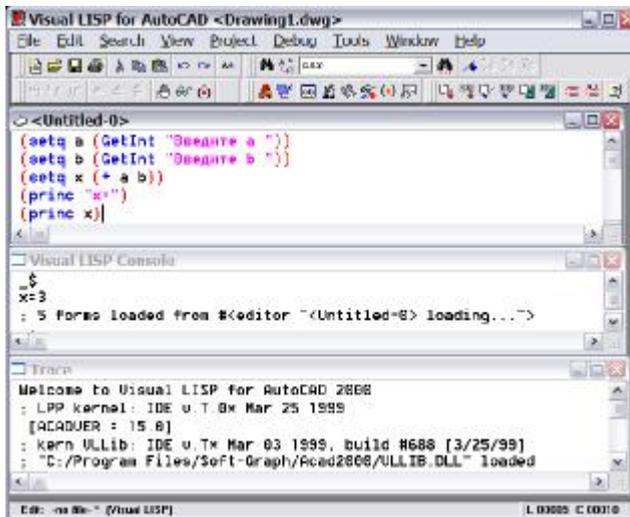


Рис. 33. Окно VisualLISP с примером программы

1. *Консольное окно* (VisualLISP Console) – в нем, подобно командному окну AutoCAD, можно вводить команды AutoLISP. После ввода команды сразу происходит ее выполнение. Также в консольном окне отражаются результаты выполнения программы.

В командных окнах AutoCAD и VisualLISP есть несколько отличий. Например, для отображения значения переменной в AutoCAD вводится имя переменной с восклицательным знаком, а в VisualLISP – просто имя переменной.

В консольном окне выводятся следующие виды приглашений:

\$ – ожидается ввод команды;

(> – показывает, что в выражении не закрыта одна или более скобок.

При вводе команд возможно использование следующих клавиш:

«Ctrl+Enter» – продолжение ввода команды на новой строке;

«Tab (Shift+Tab)» – вывод ранее выполненной команды;

«Esc» – отмена вводимой команды.

2. *Окно программы* – появляется при создании нового файла (пункт меню File\New file) или вызова программы, записанной на диск (пункт меню File\Load File). Предназначено для ввода текста программ, при этом вводимые выражения сразу не выполняются. Для выполнения программы или проверки ее синтаксиса используются команды табл. 35.

Таблица 35

### Команды выполнения и синтаксической проверки программы

Действие	Команда меню (раздел меню «Tools»)	Клавиша	Пиктограмма (панель инструментов «Tools»)
Выполнение всей программы	Load Text in Editor	Ctrl–Alt–E	
Выполнение выделенного фрагмента	Load Selection	Ctrl–Shift–E	
Синтаксическая проверка всей программы	Check Text in Editor	Ctrl–Alt–C	
Синтаксическая проверка выделенного фрагмента	Check Selection	Ctrl–Shift–C	

3. *Отладочное окно* (Trace) – содержит дополнительную информацию об ошибках программы при ее выполнении.

4. *Окно результатов синтаксической проверки программы* (Build Output) – вызывается после выполнения синтаксической проверки.

Быстрое переключение между окнами производится клавишами:

«Alt–1» – окно консоли;

«Alt–2» – окно трассировки;

«Alt–3», «Alt–4», ... – окна первой, второй программ и так далее.

Для облегчения ввода программы ее текст форматируется с использованием цвета: скобки обозначаются красным цветом, операторы – синим, строковые константы – малиновым, имена переменных – черным. При вводе текст форматируется автоматически, для принудительного форматирования, например после вставки фрагмента из буфера обмена, можно использовать команды, приведенные в табл. 36. При вводе удобно пользоваться командами поиска (табл. 37).

Таблица 36

### Команды форматирования текста

Действие	Команда меню (раздел меню «Tools»)	Клавиша	Пиктограмма (панель инструментов «Tools»)
Форматирование всей программы	Format Code in Editor	Ctrl–Alt–F	
Форматирование выделенного фрагмента	Format Code in Selection	Ctrl–Shift–F	

Таблица 37

### Команды поиска

Действие	Команда меню (раздел меню «Edit\Parantheses Matching»)	Клавиша
Найти закрывающую скобку	Match Forward	Ctrl–]
Найти открывающую скобку	Match Backward	Ctrl– [
Выделить текст до конца скобок	Select Forward	Ctrl–Shift–]
Выделить текст до начала скобок	Select Backward	Ctrl– Shift– [

### Отладка программ

Если при выполнении программы обнаружена ошибка, то выполнение прекращается и выводится сообщение об ошибке и выражение, в котором ошибка обнаружена. Просмотреть информацию, выводимую в командную строку, можно в текстовом окне AutoCAD, вызываемом клавишей «F2» или командой меню «Вид\Отображение\Текстовое окно» (View\Display\Text Window). Так как место самой ошибки внутри выражения не отмечается, при составлении программ не следует записывать сложные выражения, которые трудно анализировать, а лучше разбивать их на ряд более простых.

Для отладки программ используются команды раздела меню «Debug» или одноименной панели инструментов (табл. 38).

## Команды отладки

Действие	Команда меню (раздел меню «Debug»)	Клавиша	Пиктограмма (панель инструментов «Debug»)
Пошаговое прохождение программы без захода во вложенные выражения	Step Into	F8	
Пошаговое прохождение программы с заходом во вложенные выражения	Step Over	Shift-F8	
Пошаговое прохождение программы с выходом из вложенных выражений	Step Out	Ctrl-Shift-F8	
Установка или удаление точки останова, при достижении которой выполнение программы приостанавливается. Дальнейшее выполнение программы продолжается после команды «Continue» или команды пошагового прохождения программы	Toggle Breakpoint	F9	
Продолжение выполнения приостановленной программы до следующей точки остановки или конца программы	Continue	Ctrl-F8	
Прекращение выполнения программы	Reset To Top Level	Ctrl-R	

Для возможности пошагового прохождения программы в ней необходимо поставить точку останова, а затем запустить программу на выполнение обычным образом.

При отладке программы следить за значениями переменных можно с помощью команд, находящихся в разделе меню View:

Inspect – вывод значений переменных, списков, наборов, примитивов и выражений AutoLISP или AutoCAD, необходимые объекты должны быть перед вызовом команды выделены в тексте программы;

Symbol Service – вывод значений переменных AutoLISP;

Watch Window – вывод окна, постоянно отображающего значения переменных. Добавление переменных в это окно осуществляется его кнопкой .

При отладке могут быть использованы две функции AutoLISP:

- (Trace Имя Функции) – устанавливает для заданной функции режим трассировки, в котором при каждом вызове функции на экран бу-

дут выводиться значения ее аргументов и возвращаемое функцией значение;

- (UnTrace Имя Функции) – отменяет для функции режим трассировки.

Для нахождения места ошибки в программе удобно использовать следующий метод. Программа на AutoLISP выполняется в режиме интерпретации, то есть после обнаружения ошибки выполнение программы прекращается. Поэтому в текст программы можно вставить операторы вывода на экран, например (print "Точка 1"). Если после запуска программы информация на экран выводится, значит, программа до места вставки оператора выполняется и ошибка находится далее. Если информации на экране нет, значит, ошибка содержится до места вставки оператора.

### **Загрузка в AutoCAD программы AutoLISP**

Готовую записанную на диск программу можно загрузить в AutoCAD с помощью функции Load (которую не следует путать с командой LOAD AutoCAD). Имя программы может быть произвольным, а расширение должно быть стандартным – «.lsp». Таким образом, загрузка имеющегося в каталоге (папке) AutoCAD файла prog.lsp осуществляется выражением AutoLISP:

```
(Load "PROG")
```

При этом расширение «.lsp» можно не указывать.

Если программа расположена не в каталоге (папке) системы AutoCAD, необходимо указать к ней полный путь, начиная с имени диска, например

```
(Load "D:\SAPR\PROG")
```

Программа на AutoLISP также может быть загружена командой AutoCAD APpload, которая вызывает диалоговое окно для выбора файла.

После загрузки программы последовательно выполняются все выражения, описанные в ней. По окончании выполнения в среде AutoCAD остаются все описанные в программе функции и переменные, которые можно использовать в дальнейшей работе.

При старте AutoCAD автоматически выполняются выражения, записанные в файле acad.lsp (файл должен быть в каталоге программы AutoCAD). В этот файл можно записать выражения для присвоения значения переменным и определения новых функций и команд, которые затем можно будет использовать при работе (см. с. 120).

### 3.3. ИСПОЛЬЗОВАНИЕ ЯЗЫКА AUTOLISP В СРЕДЕ AUTOCAD

#### Использование выражений AutoLISP при вводе команд AutoCAD

Использовать переменные и выражения AutoLISP можно при вводе параметров команд AutoCAD, например:

Команда: (SetQ p1 (list 2 3) )

Команда: КРУГ

ЗТ/2Т/ККР/<Центр>: p1

Диаметр/<Радиус>: (+ 2 4)

В ответ на запрос центральной точки в AutoCAD был передан список из двух чисел 2 и 3, хранящийся в переменной p1, а в ответ на запрос радиуса – сумма чисел 2 и 4.

Кроме того, выражения AutoLISP можно использовать и в качестве команды AutoCAD. Например, после ввода функции

Команда: (+ 2 3)

в командную строку будет помещено значение, возвращаемое функцией сложения, то есть число 5.

При вводе выражений AutoLISP внутри круглых скобок допускается любое число пробелов. Это является отличием от ввода команд AutoCAD, где пробел равносителен нажатию клавиши «Enter». Большие выражения нажатием клавиши «Enter» можно переносить на другую строку. При этом AutoCAD напoмнит число незакрытых скобок.

#### Функция Command

В программе на AutoLISP может быть выполнена любая команда AutoCAD. Для этого существует функция

(Command "Имя Команды" Аргументы)

Работа этой функции заключается в последовательном вводе аргументов, заданных в этой функции, в командную строку AutoCAD. При записи аргументов функции Command необходимо соблюдать следующие правила:

1. Список аргументов функции Command должен точно соответствовать параметрам, которых требует команда AutoCAD. Поэтому перед записью выражения можно ввести требуемую команду в командную строку AutoCAD и просмотреть порядок запрашиваемых параметров и формат их записи.

2. В качестве аргументов вводятся строки – требуемые значения, заключенные в двойные кавычки, например

(Command "LINE" "10,20" "20,10" "" )

Внутри выражений AutoLISP количество пробелов не регламентируется. Но внутри аргументов – текстовых строк пробелов быть не должно, так как здесь должны соблюдаться правила записи команд AutoCAD. Поэтому нельзя, например, записывать координату точки как "10, 20", необходимо пробелы удалить – "10,20".

3. Вместо текстовых аргументов могут вводиться переменные и выражения AutoLISP, соответствующие по типу необходимым параметрам. Например, если есть переменные  $p1$  и  $p2$ , содержащие координаты точек, то отрезок может быть построен следующей командой:

(Command "LINE" p1 p2 "" )

Из двух переменных, содержащих координаты  $x$  и  $y$ , координаты точки образуются функцией list

(Command "LINE" (list 0 0) p2 "" )

(Command "LINE" p1 p2 "" )

Вещественные и целые переменные можно вводить в качестве диаметров и радиусов, расстояний, толщин линий и т. д.

4. Если в команде необходимо ввести пробел, например для окончания построения ломаной или для сохранения без изменения текущего параметра, в функции Command необходимо ввести подряд две двойные кавычки без пробелов между ними.

Примеры функции Command приведены в табл. 39.

Таблица 39

**Пример записи команд AutoCAD в AutoLISP**

Ввод команд AutoCAD (символ $\square$ – нажатие клавиш «Пробел» или «Enter»)	Выражение AutoLISP
1	2
Команда: ОТРЕЗОК $\square$ От точки: 0,0 $\square$ К точке: 50,25 $\square$ К точке: 50,50 $\square$ К точке: $\square$	(SetQ p2 (list 20 25) ) (Command "ОТРЕЗОК" "0,0" p2 "50,50" "" )
Команда: КРУГ $\square$ ЗТ/ЗТ/ККР/⟨Центр⟩: 20,30 $\square$ Диаметр/⟨Радиус⟩: Д $\square$ Диаметр: 10 $\square$	(SetQ c (list 20 30) ) ; координаты центра (SetQ d 10) ; диаметр (Command "КРУГ" c "Диаметр" d ) ; Центр и диаметр заданы переменными

1	2
Команда: ОТРЕЗОК_ От точки: 0,0_ К точке: 100,0_ К точке: 100,100_ К точке: 0,100_ К точке: Замкни_ 	; Построение квадрата (SetQ x0 0 y0 0) ; координаты нижнего ; левого угла (SetQ a 100) ; длина стороны (Command "ОТРЕЗОК" (list x0 y0) ; точки задаются в виде (list (+ x0 a) y0) ; списка из двух значений (list (+ x0 a) (+ y0 a) ) (list x0 (+ y0 a) ) "Замкни" ; замыкание ломаной ) 
Прерывание текущей команды (нажатие клави- ши «Esc» или «Ctrl+C»)	(Command)

В функции Command может быть записано одновременно несколько команд, например, следующая команда выводит окружность и две осевые линии для нее, восстанавливая затем изображение сплошной линией:

```
(Command
  " КРУГ" "150,200" "50" ""
  "ТИПЛИН" "Установи" "Осевая" ""
  "ОТРЕЗОК" "90,200" "210,200" ""
  "ОТРЕЗОК" "150,140" "150,260" ""
  "ТИПЛИН" "Установи" "Continuous" ""
)
```

Объединять много команд в одной функции не следует, так как при наличии ошибки будет трудно определить, в какой именно из команд она содержится.

Вместо значения аргумента в функции Command может быть задано слово Pause, например

```
(Command "ПЛИНИЯ" "20,40" "Ширина" Pause "" "80,40" "")
```

Это слово обозначает приостановку работы программы и ожидание ввода требуемого значения (точки, радиуса, угла и т.д.) пользователем. После ввода выполнение программы продолжится. В приведенном выше примере пользователь должен задать ширину выводимой линии.

### Отладка функции Command

Ошибки в функции Command могут быть двух видов – неправильный формат записи функции, например лишние символы или отсутствие одной из закрывающих кавычек, и неправильная последовательность задания параметров для команд. В последнем случае выполнение

программы может остановиться, когда вводится параметр недопустимого типа. Но иногда выполнение может продолжаться далее, если соответствие типов соблюдается, но вводится неправильная информация, например, когда при изображении линии пропущена одна из точек. В этом случае сообщения об ошибке не будет, обнаружить ее можно только по неправильному построению чертежа.

Если в функции Command допущена ошибка, для ее определения необходимо проанализировать выводимую в окно команд AutoCAD информацию. При выполнении этой функции в окно команд выводятся запросы для параметров и задаваемая для них информация. Поэтому, проверяя последовательность ввода параметров, можно обнаружить ошибочное место в выражении. Для вывода на экран текстового окна с возможностью просмотра выводимой ранее информации используется клавиша «F2» или команда меню «Вид\Отображение\Текстовое окно» (View\Display\Text Window).

### Доступ к примитивам чертежа

В AutoLISP можно работать с существующими объектами чертежа. Каждый примитив чертежа имеет свое уникальное имя, получить которое можно с помощью функций, указанных в табл. 40.

Таблица 40

#### Функции получения имен примитивов

Функция	Описание
(EntNext ИмяПримитива)	Возвращает имя примитива, следующего за заданным примитивом. Если Имя примитива в качестве параметра не указано, возвращает имя первого примитива. При отсутствии примитива возвращается nil
(EntLast)	Возвращает имя последнего из созданных примитивов рисунка
(EntSel Запрос)	Возвращает имя примитива, указанного пользователем. Запрос – необязательный текст, который будет выводиться на экран

Свойства примитива по его имени могут быть получены в виде списка функцией EntGet. Для каждого типа примитива (линии, окружности, текста и т. д.) существует своя структура этого списка, называемая DXF-структурой, в которой под определенными номерами записаны параметры примитива (размеры, ширина линии и т. д.). Каждый параметр, в свою очередь, представляет список из двух значений – номера параметра, называемого DXF-кодом, и значения параметра.

Если создать на чертеже линию с одним и тремя сегментами, полилинию и окружность следующими командами

```
LINE 10,10 10,20
LINE 15,15 30,15 30,30 C
PLINE 50,50 W 1 1 100,100
CIRCLE 75,75 25
```

то этим примитивам будут соответствовать следующие списки (линии с тремя сегментами соответствуют три примитива):

```
((-1 40024d70>) (0 "LINE") (330 40024cf8>) (5 "2E")
(100 "AcDbEntity") (67 0) (410 "Model") (8 "0") (100 "AcDbLine")
(10 10.0 10.0 0.0) (11 10.0 20.0 0.0) (210 0.0 0.0 1.0))

((-1 40024d78>) (0 "LINE") (330 40024cf8>) (5 "2F")
(100 "AcDbEntity") (67 0) (410 "Model") (8 "0") (100 "AcDbLine")
(10 15.0 15.0 0.0) (11 30.0 15.0 0.0) (210 0.0 0.0 1.0))

((-1 40024d80>) (0 "LINE") (330 40024cf8>) (5 "30")
(100 "AcDbEntity") (67 0) (410 "Model") (8 "0") (100 "AcDbLine")
(10 30.0 15.0 0.0) (11 30.0 30.0 0.0) (210 0.0 0.0 1.0))

((-1 40024d88>) (0 "LINE") (330 40024cf8>) (5 "31")
(100 "AcDbEntity") (67 0) (410 "Model") (8 "0") (100 "AcDbLine")
(10 30.0 30.0 0.0) (11 15.0 15.0 0.0) (210 0.0 0.0 1.0))

((-1 40024d90>) (0 "LWPOLYLINE") (330 40024cf8>) (5 "32")
(100 "AcDbEntity") (67 0) (410 "Model") (8 "0") (100 "AcDbPolyline")
(90 2) (70 0) (43 1.0) (38 0.0) (39 0.0) (10 50.0 50.0) (40 1.0)
(41 1.0) (42 0.0) (10 100.0 100.0) (40 1.0) (41 1.0) (42 0.0)
(210 0.0 0.0 1.0))

((-1 40024d98>) (0 "CIRCLE") (330 40024cf8>) (5 "33")
(100 "AcDbEntity") (67 0) (410 "Model") (8 "0") (100 "AcDbCircle")
(10 75.0 75.0 0.0) (40 25.0) (210 0.0 0.0 1.0))
```

Часть DXF-кодов являются общими для всех объектов, другая часть индивидуальна для каждого типа объекта. Описание отдельных DXF-кодов приведено в табл. 41.

Таблица 41

<b>DXF-коды</b>	
Код	Описание
1	2
-1*	Имя примитива, индивидуально для каждого сеанса работы в AutoCAD
0*	Тип примитива, обычно совпадает с именем команды, которой примитив был создан
5*	Метка – внутренний номер примитива в чертеже, в отличие от имени примитива для заданного чертежа не изменяется

Код	Описание
8*	Номер слоя, на котором размещен объект
10	Координаты начала линии, центр окружности
11	Координаты конца линии
40	Начальная ширина полилинии, радиус окружности
41	Конечная ширина полилинии
60*	Видимость примитива (0 – видимый, 1 – невидимый)
62*	Номер цвета от 0 до 256, ноль обозначает цвет по блоку (BYBLOCK), 256 – цвет по слою (BYLAYER)

\* Коды, общие для всех объектов.

Функцией `assoc` из списка может быть получен нужный подсписок, из которого затем функцией `cdr` может быть извлечен параметр. Пример программы для вывода имен и типов всех примитивов, находящихся в чертеже, следующий:

```

; Получение имени первого объекта
(setq p (entnext))
; Цикл, заканчивающийся когда имя объекта станет равным nil
(while p
  (princ
    ; По имени примитива функцией entget получается список параметров,
    ; из которого функцией assoc извлекается нужный элемент и функцией
    ; cdr удаляется DXF-код, затем параметр выводится на экран
    (print (cdr (assoc -1 (entget p))))
    (princ (cdr (assoc 0 (entget p))))
    ; Получение имени следующего примитива
    (setq p (entnext p))
  )
)

```

Удалить имеющийся примитив можно командой

```
(entdel ИмяПримитива)
```

где `ИмяПримитива` – значение, возвращаемое функциями `EntNext`, `EntLast`, `EntSel`. Для изменения параметров примитивов используется функция `EntMod` (после ее применения изображение примитива на экране необходимо обновить функцией `EntUpd`). Создать новый примитив можно функцией `EntMake`, параметром которой является список, аналогичный возвращаемому функцией `EntGet`.

Подробно работа с объектами чертежа рассмотрена в книгах [1, 12].

### 3.4. СОЗДАНИЕ НОВЫХ ФУНКЦИЙ И КОМАНД

#### Описание новых функций

В AutoLISP могут быть определены новые функции, которые затем можно будет использовать в выражениях. Описание функции выполняется с помощью специальной функции DeFun, формат которой следующий

```
(DeFun ИмяФункции (Аргументы / Локальные переменные)
  Выражения
)
```

Имя функции должно состоять из латинских букв и цифр, начинаться с буквы и не содержать пробела. Аргументы – это список данных, которые будут передаваться функции. Тип аргументов не указывается, и при выполнении функции он определяется по передаваемым данным.

Локальные переменные – это список имен вспомогательных переменных, которые используются внутри функции. Локальные переменные никак на остальную часть программы не влияют. Их имена могут совпадать с именами переменных в программе и других функциях, но любое изменение значения локальной переменной не отражается на значениях других переменных с такими же именами. Локальные переменные можно не описывать, а определять внутри функции в выражении SetQ. Все локальные переменные и переменные, описанные внутри функции, могут использоваться только в самой функции, и при окончании ее работы их значения теряются, а переменные уничтожаются.

Список аргументов и локальных переменных может отсутствовать. Если в функции их нет, после имени должны быть две круглые скобки без текста внутри, например,

```
(DeFun NewFun () ...)
```

Если нет аргументов, но есть список локальных переменных, он должен начинаться со слэша "/":

```
(DeFun NewFun (/ a b) ...)
```

Выражения представляют собой одно или несколько выражений AutoLISP. Функцией возвращается значение, возвращаемое последним из ее выражений.

Опишем новые функции, вычисляющие тангенс, десятичный логарифм и квадратный корень.

```
(DeFun sqr (x)
  (* x x)
)
```

```
(DeFun lg(x)
  ( / (log x) (log 10) )
)

(DeFun tg (a)
  ( / (sin a) (cos a) )
)
```

При вызове функции вместо ее аргумента ( $x$  для квадратного корня и логарифма,  $a$  – для тангенса) подставляется фактическое значение – число, переменное или выражение, которое и будет участвовать во всех расчетах, производимых в функции. Например,

```
(SetQ R (lg 20) ) ; Расчет R=lg 20
(SetQ Q (sqr (- R 5) ) ) ; Расчет Q=(R – 5)2
```

Далее приводится пример функции, не имеющей аргумента и возвращающей число  $\pi$ . В примере используется функция Eval, возвращающая значение аргумента:

```
(DeFun pi ()
  (Eval 3.1415926)
)
```

Для вызова функции, не имеющей аргумента, в выражении приводится ее имя, заключенное в круглые скобки, например,

```
(SetQ L (cos (* 0.5 (pi) ) ) )
```

Рассмотрим пример функции, рассчитывающей массу цилиндрического силоса по его диаметру  $D$ , высоте  $H$  и толщине стенок  $st$ . В списке аргументов функции кроме исходных данных указываются локальные переменные  $r$ ,  $D1$ ,  $S$ ,  $M$ . Значение, которое необходимо вычислить, сначала помещается в переменную  $M$ , затем в последнем выражении, входящем в функцию, это значение возвращается функцией Eval:

```
(DeFun MSilo (D H st \ r D1 S M)
; Плотность стали
; Внутренний диаметр
(SetQ r 7800) (SetQ D1 (- D (* 2 st) ) )
; Площадь сечения, имеющего вид кольца
(SetQ S (* 0.25 3.1415926 (- (* D D) (* D1 D1) ) ) )
; Масса, определяемая по объему и плотности
(SetQ M (* S H r) )
; Значение, возвращаемое функцией
(Eval M)
)
```

Тип аргументов, передаваемых функции, в AutoLISP не задается, и поэтому при вызове функции в качестве ее аргументов могут быть использованы любые данные. Для проверки типа полученных аргументов могут использоваться функции табл. 32. Например, в функции Calc в качестве аргумента  $x$  должно быть целое или вещественное число. Проверка полученного аргумента может быть выполнена следующим образом:

```
(DeFun Calc (x)
  (if (NumberP x)
    ... выражения функции Calc ...
    (progn
      (print "Аргумент функции Calc не является числом")
      (terpri)
      (eval nil)
    )
  )
)
```

Если в качестве аргумента функции передается число, например (Calc 5), выполняются нужные действия. Если передано не число, например (Calc "5"), на экран выводится сообщение «Аргумент функции Calc не является числом» и функция Calc возвращает значение nil.

### Описание новых команд

Функции, описанные указанным выше способом, могут использоваться только в выражениях AutoLISP. Но с помощью функции DeFun возможно создание и новых команд AutoCAD, вводимых в командную строку. Для этого перед именем функции указываются символы «C:». Описание команды не должно содержать аргументов и локальных переменных (при необходимости последние могут быть описаны внутри функции).

Опишем новую команду, которая будет выводить изображение сечения трубы (рис. 34). В качестве исходных данных будет с клавиатуры вводиться центр сечения (переменная  $c$ ), внешний радиус трубы (координата  $r$ ) и толщина стенки (координата  $s$ ). При вводе радиуса использована функция GetDist, которая выводит «резиновую линию» от полученных координат центра до текущего положения указателя.

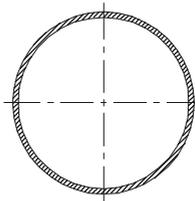


Рис. 34. Сечение трубы

В функции рассчитаны внутренний радиус трубы  $r1$  и в переменной  $x$  задан выступ 10% осевой линии относительно радиуса. При выводе осевой линии и указания точки штриховки используется функция polar, возвращающая точки, лежащие в заданном направлении и на заданном расстоянии относительно центра.

Текст функции следующий.

```
(DeFun C:ProfileT ()
; Ввод размеров сечения
(SetQ c (GetPoint "Координаты центра "))
(SetQ r (GetDist c "Радиус "))
(SetQ s (GetInt "Толщина стенки "))
; Построение окружностей
(Command "КРУГ" c r)
(Command "КРУГ" c (- r s))
; Штриховка
(Command "-КШТРИХ" "C" "LINE" "1" "45" (polar c 0 (- r (* 0.5 s))) "")
; Половина размера осевой линии
(SetQ x (* 1.1 r))
; Тип линии - осевая
(Command "ТИПЛИН" "Установи" "Осевая" "")
; Построение осевых линий
(Command "ОТРЕЗОК" (polar c 1.57 x) (polar c -1.57 x) "")
(Command "ОТРЕЗОК" (polar c 3.14 x) (polar c 0 x) "")
; Тип линии - сплошная
(Command "ТИПЛИН" "Установи" "Continuous" ""))
)
```

После ввода функции с клавиатуры или загрузки файла, в котором эта функция описана, в AutoCad становится доступной новая команда ProfileT, которая может быть вызвана из командной строки (рис. 35).

Еще одно применение команд, создаваемых пользователем, – автоматизация загрузки программ. Каждый раз набирать функцию load для загрузки программы довольно утомительно. Поэтому можно создать специальную команду для этого, дав ей короткое имя, не совпадающее с имеющимися командами AutoCad, например,

```
(DeFun C:ZZ (Load "program") )
```

После ввода этой функции, достаточно в командной строке ввести команду «ZZ», и программа program.lsp будет загружена.



Рис. 35. Использование новой команды

В AutoLISP при вводе нового объекта старый объект с таким же именем уничтожается и заменяется новым. Поэтому, если в функции или команде есть ошибки, достаточно ввести текст функции заново или загрузить файл с исправленной версией функции, и она будет заменена.

### **Создание библиотеки функций**

Все функции и команды, введенные с клавиатуры и загруженные вместе с файлами функцией Load, находятся в оперативной памяти компьютера. Поэтому по окончании работы с AutoCAD они пропадают и для использования в следующем сеансе их необходимо загружать заново.

В AutoCAD имеется возможность автоматической загрузки программ на AutoLISP. Для этого необходимо использовать файл acad.lsp, расположенный в каталоге (папке) системы AutoCAD (если такого файла нет, его необходимо создать). Этот файл автоматически загружается каждый раз при запуске AutoCAD. Поместить в этот файл описания функций и команд возможно двумя способами:

- а) непосредственно записав в файл исходные тексты функций;
- б) записав в файл acad.lsp функцию загрузки файла, содержащего функции, например

(Load "Library")

Если в файле acad.lsp определить функцию C:: Startup, она будет не только загружаться, но и выполняться. В эту функцию можно поместить, например, действия по настройке AutoCAD, построение рамки со штампом и другие стандартные действия.

## 4. АВТОМАТИЗИРОВАННОЕ ПРОЕКТИРОВАНИЕ ЭЛЕМЕНТОВ ТЕПЛОЭНЕРГЕТИЧЕСКОГО ОБОРУДОВАНИЯ

### 4.1. СОЗДАНИЕ ПРОГРАММЫ ПОСТРОЕНИЯ ПАРАМЕТРИЗОВАННОГО ЧЕРТЕЖА

Программа на AutoLISP для построения чертежей может иметь следующую структуру:

- 1) ввод исходных данных или определяющих размеров;
- 2) расчет требуемых размеров (определяющих размеров по исходным данным и зависимых размеров по определяющим размерам);
- 3) расчет координат опорных точек;
- 4) графические построения с использованием опорных точек.

Рассмотрим пример построения чертежа в AutoLISP детали, представленной на рис. 36. Классификация размеров детали дана в табл. 42.

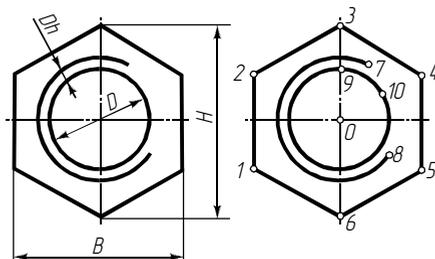


Рис. 36. Исходный чертеж и чертеж с указанными опорными точками

Таблица 42

**Классификация размеров**

Условное обозначение	Тип размера	Расчетная формула
$D$	Определяющий (мм)	—
$B$	Зависимый	$B = 1,5D$ (округленный до целого значения)
$H$	Зависимый	$H = B/\cos 30^\circ = 1,155B = 1,732D$ (округленный до целого значения)
$Dh$	Зависимый	$Dh = 0,1D$ (округленный до 0,5 мм)

В качестве базовой точки с номером 0 и координатами  $x_0, y_0$  выберем центр детали. Выделим опорные точки, лежащие на углах шестиугольника. Также выделим две точки, лежащие на окружности, обозначающей отверстие (для ее построения в виде двух дуг и для простановки размеров), и две точки на концах дуги, обозначающей резьбу (рис. 36).

Координаты опорных точек рассчитываются следующим образом:

$$\begin{array}{ll}
 x_1 = x_0 - 0,5B; & y_1 = y_0 - 0,25H; \\
 x_2 = x_1; & y_2 = y_0 + 0,25H; \\
 x_3 = x_0; & y_3 = y_0 + 0,5H; \\
 x_4 = x_0 + 0,5B; & y_4 = y_2; \\
 x_5 = x_4; & y_5 = y_1; \\
 x_6 = x_0; & y_6 = y_0 - 0,5H; \\
 x_7 = x_0 + 0,25D; & y_7 = y_0 + \sqrt{(0,5D + Dh)^2 - (0,25D)^2}; \\
 x_8 = x_0 - \sqrt{(0,5D + Dh)^2 - (0,25D)^2}; & y_8 = y_0 - 0,25D; \\
 x_9 = x_0; & y_9 = y_0 - 0,5D; \\
 x_{10} = x_0 + D\sqrt{8} = x_0 + 0,35355D; & y_{10} = y_0 + D\sqrt{8} = x_0 + 0,35355D.
 \end{array}$$

Программа изображения чертежа будет следующей.

```

; AutoLISP AutoCAD 2000 (english)
; Очистка экрана и выключение объектной привязки1
(command "Erase" "All" "")
(command "Osnap" "None")
(terpri)
; Ввод размера D2
(SetQ D 100)
;(SetQ D (GetInt "D, mm - ?"))
; Ввод координат опорной точки
(SetQ x0 100 y0 100)
;(SetQ x0 (GetInt "x0 - ?"))
;(SetQ y0 (GetInt "y0 - ?"))
; Расчет зависимых размеров
(SetQ B (* 1.5 D) H (* 1.732 D) Dh (* 0.1 D))
; Округление зависимых размеров3
(SetQ B (fix (+ 0.5 B)))

```

<sup>1</sup> В режиме включенной объектной привязки при построении объектов из программы на AutoLISP могут наблюдаться сбои.

<sup>2</sup> Для ввода значений переменных при написании и отладки программы используется функция SetQ. После отладки программы для ввода могут быть использованы функции ввода данных с клавиатуры.

<sup>3</sup> В AutoLISP нет функции округления, поэтому используется функция отбрасывания дробной части fix, к аргументу которой прибавляется 0,5, чтобы числа с дробной частью более 0,5 округлялись в большую сторону. Функция fix возвращает целое значение, поэтому переменные D и H становятся целыми.

Для округления Dh до 0,5 мм эта величина сначала увеличивается в 2 раза, затем округляется до целого числа и уменьшается в 2 раза. В выражении деления цифра 2 вводится с дробной частью, чтобы результат был вещественным.

```
(SetQ H (fix (+ 0.5 H)))
(SetQ Dh (/ (fix (+ 0.5 (* 2 Dh))) 2.0))
(terpri) (PrnC "B=") (PrnC B) (PrnC " H=") (PrnC H) (PrnC " Dh=") (PrnC Dh);
```

#### **Расчет координат точек**

```
(SetQ x1 (- x0 (/ B 2)))
(SetQ y1 (- y0 (/ H 4)))
(SetQ y2 (+ y0 (/ H 4)))
(SetQ y3 (+ y0 (/ H 2)))
(SetQ x4 (+ x0 (/ B 2)))
(SetQ y6 (- y0 (/ H 2)))
(SetQ x7 (+ x0 (/ D 4)))
(SetQ y7 (+ y0 (sqrt (- (expt (+ (* 0.5 D) Dh) 2) (expt (* 0.25 D) 2)))))
(SetQ x8 (+ x0 (sqrt (- (expt (+ (* 0.5 D) Dh) 2) (expt (* 0.25 D) 2)))))
(SetQ y8 (- y0 (/ D 4)))
(SetQ y9 (- y0 (/ D 2)))
(SetQ x10 (- x0 (* 0.35355 D)))
(SetQ y10 (- y0 (* 0.35355 D)))
```

#### **; Опорные точки**

```
(SetQ p0 (list x0 y0))
(SetQ p1 (list x1 y1))
(SetQ p2 (list x1 y2))
(SetQ p3 (list x0 y3))
(SetQ p4 (list x4 y2))
(SetQ p5 (list x4 y1))
(SetQ p6 (list x0 y6))
(SetQ p7 (list x7 y7))
(SetQ p8 (list x8 y8))
(SetQ p9 (list x0 y9))
(SetQ p10 (list x10 y10))
```

#### **; Установка непрерывной линии<sup>1</sup>**

```
(command "-LINETYPE" "Set" "Continuous" "")
```

#### **; Построение шестиугольника<sup>2</sup>**

```
(command "PLINE" p1 "w" "1.5" "1.5" p2 p3 p4 p5 p6 "Close")
```

#### **; Построение отверстия (окружность)<sup>3</sup>**

```
(command "PLINE" p10 "Arc" "Ce" p0 p9 "Ce" p0 p10 "")
```

#### **; Построение резьбы (дуга)**

```
(command "PLINE" p7 "w" "0.5" "0.5" "Arc" "Ce" p0 p8 "")
```

#### **; Построение осевых линий<sup>4</sup>**

```
(command "-LINETYPE" "Set" "DashDotX2" "")
```

```
(command "PLINE" (list (- x1 5) y0) "w" "0.5" "0.5" (list (+ x4 5) y0) "")
```

```
(command "PLINE" (list x0 (+ y3 5)) (list x0 (- y6 5)) "")
```

#### **; Создание текстового стиля DimTXT для размерного текста<sup>5</sup>**

```
(command "-Style" "DimTXT" "Symbol" "0" "1" "15" "N" "N")
```

---

<sup>1</sup> Перед началом построения необходимо установить непрерывный тип линии, так как в AutoCAD текущим может быть другой тип.

<sup>2</sup> Толщина основных линий 1,5.

<sup>3</sup> Окружность для отверстия строится линиями толщиной 1,5 мм из двух дуг.

<sup>4</sup> Для координат концов линий задается выступ за деталь 5 мм. Толщина осевых линий 0,5.

<sup>5</sup> Стиль необходим для ввода символа «Ø» с помощью шрифта «Symbol».

; Установка системных переменных для размерных линий

(Command

"SETVAR" "DIMTXSTY" "DimTXT"; Стиль текста

"SETVAR" "DIMASZ" "7"; Размер стрелки

"SETVAR" "DIMTAD" "1"; Положение текста относительно линии

"SETVAR" "DIMTXT" "5"; Размер текста

"SETVAR" "DIMTIH" "0"); Текст вдоль вертикальных линий

; Простановка размеров

(command "-LINETYPE" "Set" "Continuous" "")

(Command "DIM" "Vertical" p3 p6 (list (+ x4 20) y0) (ltoa D) "Exit")

(Command "DIM" "Horizontal" p2 p4 (list x0 (+ y3 20)) (ltoa H) "Exit")

(Command "DIM" "Diameter" p10 (StrCat "Ж" (ltoa D)) p0 "Exit")

; Конец программы

## 4.2. ПРОЕКТИРОВАНИЕ ГОРЕЛКИ

Вихревая реверсивная горелка (ВРГ) является газовой горелкой без предварительного смешения топлива и воздуха и предназначена для установки в агрегатах, где тепловыделение из факела должно быть растянуто по длине рабочего пространства (рис. 37). В горелке дроссель можно перемещать вдоль его оси, изменяя проходное сечение сопла и скорость вылета газов.

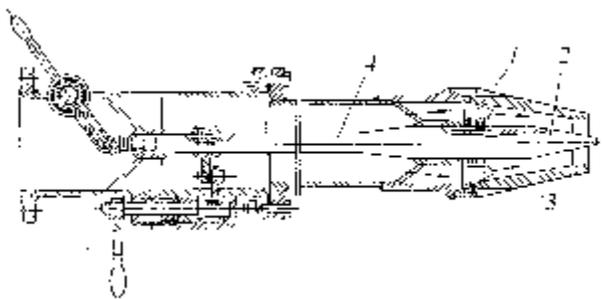


Рис. 37. Горелка ВРГ:

1 – лопатки завихрителя; 2 – дроссель; 3 – сопло; 4 – шток

Расчет горелки сводится к определению диаметра сопла при заданных расходе и характеристиках природного газа. Рассмотрим проектирование упрощенного варианта горелки. Порядок расчета горелки согласно методике ВНИИпромгаза следующий.

Исходные данные:

– расход газа  $V$ , м<sup>3</sup>/ч;

– скорость вылета газа  $w$ , м/с;

– отношение давления газа перед форсункой к давлению в пламенном пространстве печи  $\epsilon$ .

Для горелки также задается ряд констант:  
 – отношение диаметра острого дросселя к диаметру сопла  $k_1$ ;  
 – угол конуса дросселя  $\alpha$ , °;  
 – толщина стенки горелки  $s$ , мм;  
 – длина изображаемой части горелки по отношению к диаметру сопла  $k_2$ .

Порядок расчета:

- диаметр сопла  $d_s = 1000 \sqrt{\frac{4V}{3600\pi\varepsilon\eta(1-k_1^2)}}$ , мм;
- внутренний диаметр горелки  $d_g = 1,2 d_s + 2s$ , мм;
- диаметр дросселя  $d_d = 0,7d_s$ , мм;
- диаметр острого дросселя  $d_n = k_1 d_s$ ;
- длина горелки  $L = k_2 d_s$ .

Для горелки определяющим размером будет  $d_s$ , зависимыми размерами –  $d_g, d_d, d_n, L$ , независимыми –  $s$  и  $\alpha$ . В качестве точки привязки  $p_0$  с координатами  $x_0$  и  $y_0$  примем центр левого (входного) сечения горелки. Координаты всех остальных точек будут рассчитываться относительно точки привязки. Для изображения горелки необходимо определить вспомогательные зависимые размеры  $x_1 \dots x_4$  и опорные точки  $p_1 \dots p_{18}$ . Так как многие опорные точки имеют одинаковые координаты  $x$  и  $y$ , определим дополнительные уровни (значения координат  $x$  и  $y$ )  $z_2 \dots z_{15}$ . Изображение горелки с указанием перечисленных размеров и точек приведено на рис. 38.

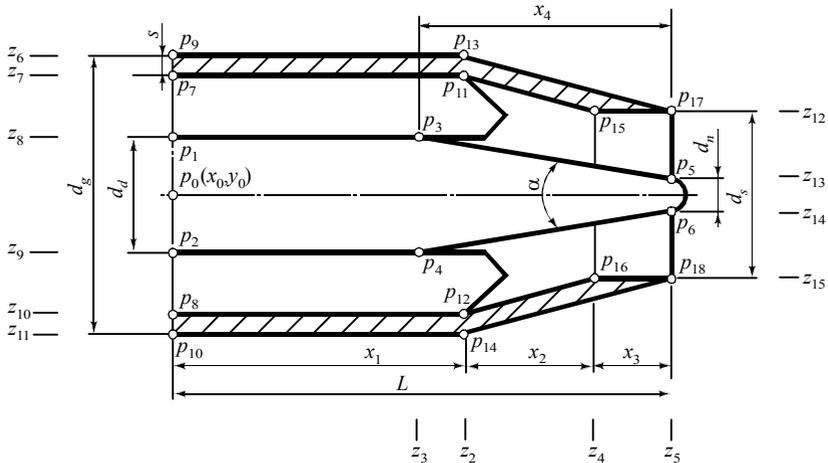


Рис. 38. Размеры и опорные точки чертежа

Вспомогательные зависимые размеры рассчитываются согласно рис. 38, 39 по следующим выражениям:

$$x_1 = L - x_2 - x_3; \quad x_2 = \frac{0,5(d_g - d_s)}{\operatorname{tg} 0,5\alpha}, \quad x_3 = \frac{s}{\sin 0,5\alpha}, \quad x_4 = \frac{0,5(d_d - d_n)}{\operatorname{tg} 0,5\alpha}$$

Программа построения чертежа горелки представлена ниже; в результате работы программы получается чертеж, показанный на рис. 40.

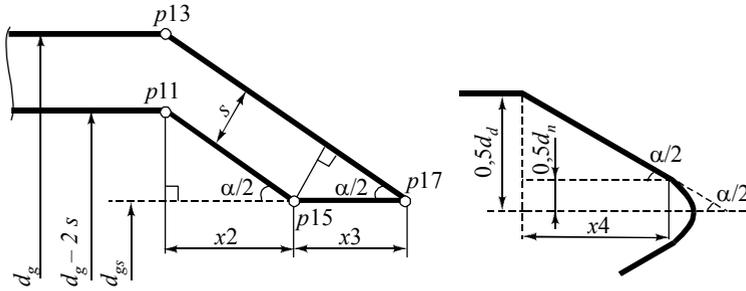


Рис. 39. Расчет вспомогательных размеров

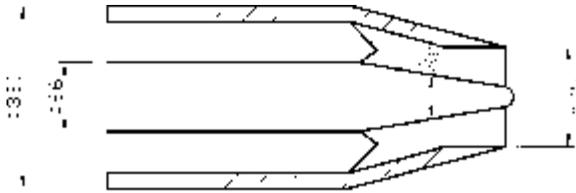


Рис. 40. Чертеж горелки ВРГ, полученный в программе на AutoLISP

; AutoLISP AutoCAD 2000 (english)

; Исходные данные<sup>1</sup>

(SetQ V 15000.0) ; (SetQ V (GetInt "Rate, m3/h ?")) ; расход, м<sup>3</sup>/ч  
 (SetQ e 2.0) ; (SetQ W (GetInt "Preessure, atm ?")) ; давление газа, атм  
 (SetQ W 300.0) ; (SetQ W (GetInt "Speed, m/s ?")) ; скорость вылета, м/с  
 (SetQ A 30) ; (SetQ A (GetInt "Angle, grad ?")) ; угол, град  
 (SetQ S 15.0) ; (SetQ S (GetInt "Wall, mm ?")) ; толщина стенки, мм  
 (SetQ XMax 297 YMax 210) ; размеры листа бумаги

; Расчет размеров

; Конструктивные параметры

(SetQ k1 0.2 k2 4.0) ; коэффициенты, задающие отношения размеров  
 (SetQ H1 0.5 H2 0.2) ; толщина основной и тонкой линий

<sup>1</sup> Исходные данные могут быть заданы в программе или введены пользователем функциями GetInt.

```

; Определяющий размер
(SetQ Ds (fix
  (* 1000.0
    (sqrt
      (/
        (* 4.0 (/ V 3600.0))
        (* pi W e (- 1 (* k1 k1)) )
      )
    )
  ))
)
; Зависимые конструктивные размеры
(SetQ Dd (fix (* 0.7 Ds)) Dg (fix (+ (* 1.2 Ds) (* 2 s))) Dn (fix (* k1 Ds)) L (fix (* k2 Ds)))
; Зависимые вспомогательные размеры
(SetQ A2 (* 3.1415926 (/ A 360.0))) ; Половина угла A в радианах
(SetQ X3 (/ S (sin A2)))
(SetQ X2 (/ (* 0.5 (- Dg ds)) (/ (sin A2) (cos A2))))
(SetQ X4 (/ (* 0.5 (- Ds dn)) (/ (sin A2) (cos A2))))
(SetQ X1 (- L (+ X2 X3)))
; Отключение объектной привязки
(Command "Osnap" "OFF")
; Выбор масштаба1
(Command "Limits" "0,0" (List XMax YMax)
  "Erase" "C" "0,0" (List XMax YMax) ""
  "Zoom" "A")
(SetQ M0 (/ XMax (* 2.0 L)))
(SetQ M
(Cond ((< M0 0.00125) 0.001) ; Масштаб 1:1000
  ((< M0 0.002) 0.00125) ; 1:800
  ((< M0 0.0025) 0.002) ; 1:500
  ((< M0 0.005) 0.0025) ; 1:400
  ((< M0 0.01) 0.005) ; 1:200
  ((< M0 0.0133) 0.01) ; 1:100
  ((< M0 0.02) 0.0133) ; 1:75
  ((< M0 0.025) 0.02) ; 1:50
  ((< M0 0.04) 0.025) ; 1:40
  ((< M0 0.05) 0.04) ; 1:25
  ((< M0 0.0667) 0.05) ; 1:20
  ((< M0 0.1) 0.0667) ; 1:15
  ((< M0 0.2) 0.1) ; 1:10
  ((< M0 0.25) 0.2) ; 1:5
  ((< M0 0.4) 0.25) ; 1:4
  ((< M0 0.5) 0.4) ; 1:2.5
  ((< M0 1.0) 0.5) ; 1:2
  ((< M0 2.0) 1.0) ; 1:1
  ((< M0 2.5) 2.0) ; 2:1
  ((< M0 4.0) 2.5) ; 2.5:1
)
)

```

---

<sup>1</sup> Масштаб выбирается из условия, что деталь занимает 50% листа. Коэффициент *M0* задает отношение размера листа по горизонтали к удвоенной длине детали. Затем это отношение округляется в меньшую сторону до значения масштаба, определяемого стандартом.

```

((< M0 10.0) 4.0) ; 4:1
((< M0 20.0) 10.0) ; 10:1
((< M0 40.0) 20.0) ; 20:1
((< M0 50.0) 40.0) ; 40:1
((< M0 100.0) 50.0) ; 50:1
((< M0 200.0) 100.0) ; 100:1
(* 100.0 (fic/ M0 100)); Произвольный масштаб, больший 100:1
))
; Размеры на чертеже с учетом масштаба1
(SetQ Lm (* L M) Dsm (* Ds M) Dgm (* Dg M) Ddm (* Dd M) Dnm (* Dn M)
Sm (* S M) x1m (* x1 M) x2m (* x2 M) x3m (* x3 M) x4m (* x4 M))
; Базовая точка2
(SetQ x0 (* 0.5 (- XMax Lm)))
(SetQ y0 (* 0.5 (- YMax Dgm)))
; Вспомогательные уровни
(SetQ z2 (+ x0 x1m) z6 (+ z7 Sm) z12 (+ y0 (* 0.5 Dsm))
z3 (+ x0 (- Lm x4m)) z8 (+ y0 (* 0.5 Ddm)) z13 (+ y0 (* 0.5 Dnm))
z4 (+ x0 x1m x2m) z9 (- y0 (* 0.5 Ddm)) z14 (- y0 (* 0.5 Dnm))
z5 (+ x0 Lm) z10 (- y0 (* 0.5 Dgm)) z15 (- y0 (* 0.5 Dsm))
z7 (+ y0 (* 0.5 Dgm)) z11 (- z10 Sm))
; Опорные точки
(SetQ p1 (List x0 z8)) p6 (List z5 z14) p11 (List z2 z7) p16 (List z4 z15)
p2 (List x0 z9) p7 (List x0 z7) p12 (List z2 z10) p17 (List z5 z12)
p3 (List z3 z8) p8 (List x0 z10) p13 (List z2 z6) p18 (List z5 z15)
p4 (List z3 z9) p9 (List x0 z6) p14 (List z2 z11)
p5 (List z5 z13) p10 (List x0 z11) p15 (List z4 z12))
; Изображение сечения корпуса3
(Command
"LineType" "S" "Continuous" ""
"PLine" p9 "w" h1 h1 p13 p17 p15 p11 p7 "c"
"Hatch" "U" "45" "5" "N"
"C" ((list(1- x0)(1+ z6)) (list(1+ z5)(1- z12))) ""
"PLine" p10 p14 p18 p16 p12 p8 "c"
"Hatch" "U" "45" "5" "N" "C" ((list(1- x0)(1+ z15)) (list(1+ z5)(1- z11))) ""
)
; Изображение дресселя
(Command "PLine" p1 p3 p5 "A" p6 "L" p4 p2 "")
; Изображение лопаток завихрителя4
(SetQ a1 (* 0.7 (- z7 z8)))
(SetQ a2 (* 0.3 (- z7 z8)))

```

<sup>1</sup> Все размеры пересчитываются с использованием масштабного коэффициента, в результате получаются размеры, откладываемые на чертеже, далее в расчетах и построении используются только пересчитанные размеры.

<sup>2</sup> Координаты базовой точки, расположенной в центре левого сечения детали, рассчитываются с условием расположения детали в центре листа.

<sup>3</sup> Основные линии изображаются толщиной, заданной переменной  $h1$ . Сразу после изображения сечения производится его штриховка, чтобы объекты, помещаемые на чертеж далее, не исказили результата выполнения команды Hatch.

<sup>4</sup> Вспомогательные переменные  $a1$  и  $a2$  задают размеры лопаток по отношению к их высоте.

```

(Command
"PLine" p11 (list (+ z2 a1) (+ z8 a2)) (list (+ z2 a2) z8) p1 ""
"PLine" p12 (list (+ z2 a1) (- z9 a2)) (list (+ z2 a2) z9) p2 ""
)
; Изображение вертикальных линий – границ выходного сопла1
(Command
"PLine" P17 p5 ""
"PLine" P18 p6 ""
"PLine" p15 "w" h2 h2 p16 ""
"Trim" p3 "" (list z4 y0) ""
)
; Изображение осевой линии и линии обреза слева2
(Command
"LineType" "S" "Center2" ""
"PLine" (list (- x0 3) y0) (list (+ z5 3) y0) ""
"PLine" p7 p8 ""
"LineType" "S" "Continuous" ""
)
; Простановка размеров
(Command
"SetVar" "DIMASZ" "3"; Размер стрелки
"SetVar" "DIMTAD" "2"; Положение текста относительно линии
"SetVar" "DIMTXT" "3"; Размер текста
"SetVar" "DIMTIH" "0"; Текст вдоль вертикальных линий
"DIM1" "HOR" (list x0 (- z11 3))
(list z5 (- z15 3))
(list z2 (- z11 15)) L
"DIM1" "Ver" (list (- x0 3) z6)
(list (- x0 3) z11)
(list (- x0 20) y0)
(StrCat "%%" (rtos L 2 0))
"DIM1" "Ver" (list (- x0 3) z8)
(list (- x0 3) z9)
(list (- x0 10) y0)
(StrCat "%%" (rtos Dd 2 0))
"DIM1" "Ver" (list (+ z5 3) z12)
(list (+ z5 3) z15)
(list (+ z5 15) y0)
(StrCat "%%" (rtos Ds 2 0))
"DIM1" "ANG" (list (* 0.5 (+ z3 z5)) (* 0.5 (+ z8 z13)))
(list (* 0.5 (+ z3 z5)) (* 0.5 (+ z9 z14)))
(list (* 0.5 (+ z3 z5)) y0) A ""
)
; Перерисовка чертежа
(Command "RedRaw" "Zoom" "A" )
; Конец программы

```

---

<sup>1</sup> Линия между точками *p15* и *p16* изображается толщиной, заданной переменной *h1*. Чтобы линия обрывалась на изображении дросселя, используется команда обрезки «Trim».

<sup>2</sup> После изображения осевых линий для дальнейших построений устанавливается непрерывный тип линий.

### 4.3. ПРОЕКТИРОВАНИЕ ПАРО-ПАРОВОГО ТЕПЛООБМЕННИКА

Проектирование осуществляется в два этапа. С помощью программы, разработанной в среде Turbo Pascal, рассчитываются и записываются в текстовом формате на диск определяющие размеры теплообменника. Затем программа на AutoLISP считывает из этого текстового файла размеры и строит по ним чертеж.

#### Назначение паро-парового теплообменника

Для регулирования температуры пара в котлах и парогенерирующих установках применяются следующие устройства:

- впрыскивающие пароохладители с впрыском питательной воды или собственного конденсата, получаемого в теплообменниках;
- паро-паровые теплообменники (ППТО);
- газо-паропаровые теплообменники;
- поверхностные пароохладители.

Паро-паровой теплообменник нашел применение благодаря особенностям теплообмена в радиационных и конвективных поверхностях. Если перегреватель высокого давления имеет развитую радиационную поверхность, то температура в нем при уменьшении нагрузки котла будет расти. Получающийся избыток теплоты в тракте высокого давления передается промежуточному пару в ППТО.

Наиболее распространенной является схема с регулированием байпасом количества подаваемого в ППТО вторичного пара (рис. 41).

ППТО является частью поверхности нагрева пара промежуточного перегревателя, обеспечивающей на номинальном режиме прирост энтальпии до 160 ... 170 кДж/кг. Размещается ППТО вне газового тракта. По тракту пара высокого давления ППТО включают после первой или второй ступени радиационного перегревателя. Снижением температуры пара после теплообменника облегчаются условия работы металла труб второй ступени радиационного перегревателя (или ширм при их включении после этой ступени).

#### Особенности расчета паро-парового теплообменника

В паро-паровых теплообменниках за расчетную поверхность нагрева принимается поверхность труб на стороне обогреваемого пара. Средний

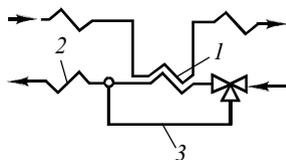


Рис. 41. Схема включения ППТО:  
1 – ППТО; 2 – промежуточный перегреватель; 3 – байпас

температурный напор определяется как среднегеометрическая разность температур с учетом взаимного направления движения обеих сред.

Коэффициент теплопередачи находится по формуле

$$R = \frac{1}{\frac{1}{\alpha_1} \frac{d_{1\text{вн}}}{d_{1\text{нар}}} + \frac{\delta_{\text{т}}}{\lambda_{\text{т}}} \frac{1}{\alpha_2}}, \text{ Вт/м}^2 \cdot \text{К (ккал/м}^2 \cdot \text{ч} \cdot \text{град)},$$

где  $\alpha_1$  и  $\alpha_2$  – соответственно коэффициенты теплоотдачи от греющего пара к наружной стенке труб диаметром  $d_{1\text{нар}}$  и от внутренней стенки труб диаметром  $d_{1\text{вн}}$  к вторичному пару, Вт/м<sup>2</sup>·К (ккал/м<sup>2</sup>·ч·град);  $\delta$ ,  $\lambda$  – толщина и коэффициент теплопроводности стенки труб, м и Вт/м·К (ккал/м·ч·град).

Физические характеристики, входящие в формулы для определения коэффициентов теплоотдачи  $\alpha_1$  и  $\alpha_2$ , берутся по средним температурам соответствующей среды.

Коэффициенты теплоотдачи  $\alpha_1$  и  $\alpha_2$  для паро-паровых теплообменников при обычно имеющем место продольном течении греющего и обогреваемого пара определяются с учетом формы канала. При одностороннем обогреве (охлаждении) кольцевых каналов коэффициент теплоотдачи определяется по формуле

$$\alpha_k = 0,023 \frac{\lambda}{d_3} \left( \frac{w d_3}{\nu} \right)^{0,8} \text{Pr}^{0,4} C_t C_d C_l, \text{ Вт/(м}^2 \cdot \text{ч} \cdot \text{К)},$$

где  $\lambda$  – коэффициент теплопроводности при средней температуре среды, Вт/м·ч·К;  $\nu$  – коэффициент кинематической вязкости при средней температуре потока, м<sup>2</sup>/с;  $w$  – расчетная скорость, м/с;  $d_3$  – эквивалентный диаметр, м; Pr – критерий физических свойств при средней температуре пара;  $C_t$ ,  $C_d$ ,  $C_l$  – коэффициенты, учитывающие температуру потока, течение в широких трубах и кольцевых односторонне обогреваемых каналах.

Для теплообменников типа «труба в трубе» с поворотом пара коэффициенты для  $\alpha_1$  и  $\alpha_2$  приближенно принимаются 1,06 и 1,1.

### **Программа конструктивного расчета паро-парового теплообменника**

Работа программы заключается в расчете количества теплоты  $q$ , передаваемой от первичного пара к вторичному. Производится расчет двух значений  $q$  – по уравнениям теплового баланса и теплообмена (подпрограмма « $dQ$ »). Полученные величины сначала приближаются друг к другу подбором числа пучков труб в теплообменнике, затем решение уточняется изменением длины прямого участка труб теплообменника.

После определения геометрических размеров программа записывает их на диск для использования при построении чертежа теплообменника. Блок-схема программы представлена на рис. 42.

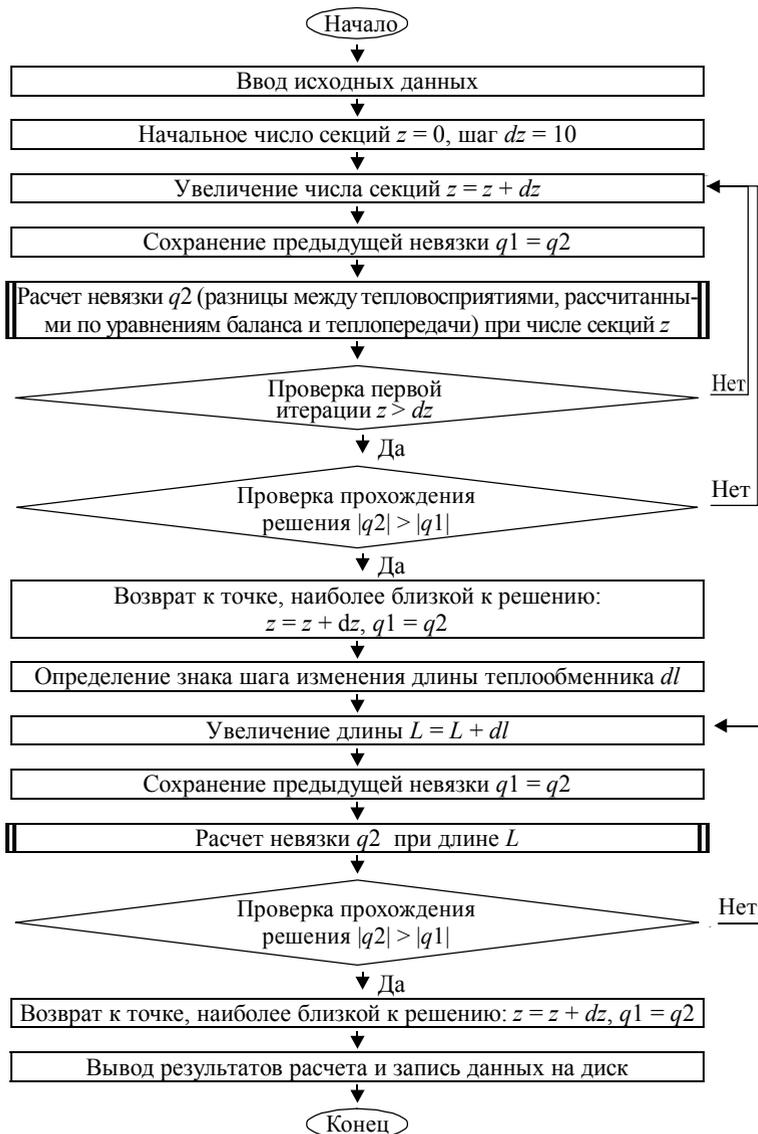


Рис. 42. Блок-схема программы

## Исходный текст программы следующий:

```
{Тепловой расчет вторичного перегревателя}
Program V1;
uses Crt,Steam;
const
  {ПАРОПРОИЗВОДИТЕЛЬНОСТЬ КОТЛА, кг/ч}
  D:real=950000;
  {ПАРАМЕТРЫ ПЕРВИЧНОГО ПАРА}
  t_pp1:real=476;      {Температура на входе, °C}
  p_pp:real=280;      {Давление, кгс/см2}
  {ПАРАМЕТРЫ ВТОРИЧНОГО ПАРА}
  D_vt:real=800000;   {Расход, кг/ч}
  t_vt1:real=307;     {Температура на входе, °C}
  t_vtb:real=340;     {Температура после байспаса}
  p_vt:real=39;       {Давление, кгс/см2}
  {Расход воды на первый и второй впрыск, кг/ч}
  D_vpr1:real=30000;
  D_vpr2 :real=20000;
  {Величина байспаса вторичного пара, %}
  tb:real=60;
  {Диаметры и толщины стенок труб первичного и вторичного пара, м}
  d1:real=0.065;   s1=0.005;
  d2:real=0.100;   s2=0.005;
  {Количество труб в секции и секций}
  ntr:integer=5;
  z:integer=78;
  {Длина прямого участка и радиус изгиба, м}
  L:real=2.930;
  R:real=0.300;

{Функция расчета теплообменника}
function dQ:real;
Var
  i_vt1, i_vt2, i_vtb, t_vt2, Qb, Dpp, i_pp1, i_pp2,
  t_pp2, t_pp, t_vt, dt1, dt2, Delta_t, n, F1, F2, ltr, W_pp, W_vt,
  a1, a2, k1, k2, k3, de2, lst, k, H, Qt :real;
Begin
  {Энтальпия вторичного пара на входе и после байспаса}
  i_vt1:=HSteam(t_vt1,p_vt);
  i_vtb:=HSteam(t_vtb,p_vt);
  {Энтальпия и температура вторичного пара на выходе}
  i_vt2:=i_vt1+(i_vtb-i_vt1)/(1-tb/100);
  t_vt2:=TSteam(i_vt2,p_vt);
  {Тепловосприятие теплообменника}
  Qb:=D_vt*(i_vtb-i_vt1);
  {Расход первичного пара в теплообменнике}
  Dpp:=D-D_vpr1-D_vpr2;
  {Энтальпия первичного пара на входе}
  i_pp1:=Hsteam(t_pp1,p_pp);
  {Энтальпия и температура первичного пара на выходе}
  i_pp2:=i_pp1-Qb/Dpp;
  t_pp2:=TSteam(i_pp2,p_pp);
  {Средние температуры первичного и вторичного пара}
  t_pp:=(t_pp1+t_pp2)/2;
  t_vt:=(t_vt1+t_vt2)/2;
  dt1:=t_pp2-t_vt1;
  dt2:=t_pp1-t_vt2;
```

```

{Температурный напор}
Delta_t:=(dt1-dt2)/ln(dt1/dt2);
{Количество и площадь сечения труб}
n:=ntr*z;
F1:=pi*sqr(d1)/4*n;
F2:=pi*(sqr(d2-2*s2)-sqr(d1))/4*n;
{Эквивалентный диаметр для вторичного пара}
de2:=pi*(sqr(d2-2*s2)-sqr(d1))/(pi*d1+pi*(d2-2*s2));
{Скорость первичного и вторичного пара}
W_pp:=Dpp*VSteam(t_pp,p_pp)/(3600*F1);
W_vt:=(1-0.01*tb)*D_vt*VSteam(t_vt,p_vt)/(3600*F2);
{Коэффициент теплоотдачи от греющего пара к стенке}
k1:=0.023*LSteam(t_pp,p_pp)/d1;
k2:=exp(0.8*ln(W_pp*d1/vuSteam(t_pp,p_pp)));
k3:=exp(0.4*ln(1.43));
a1:=k1*k2*k3*1.06;
{Коэффициент теплоотдачи от стенки к пару}
k1:=0.023*LSteam(t_vt,p_vt)/de2;
k2:=exp(0.8*ln(W_vt*de2/vuSteam(t_vt,p_vt)));
k3:=exp(0.4*ln(0.977));
a2:=k1*k2*k3*1.1;
{Коэффициент теплопроводности стенки}
lst:=55-0.0332*(0.5*(t_vt+t_pp));
{Длина трубы и поверхность нагрева}
ltr:=2*L+pi*R;
H:=pi*ltr*d1*n;
{Коэффициент теплопередачи}
k:=1/((1/a1)*(d1/(d1-2*s1))+s1/lst+(1/a2));
Qt:=k*H*Delta_t;
dQ:=Qt-Qb;
end;

var
  Name,group,variant:string;
  q1,q2,dl:real;
  dz:integer;
  F:text;

BEGIN
  {Ввод исходных данных}
  ClrScr;
  write('Температура первичного пара, °C (450-500) - ');
  readln(t_pp1);
  write('Температура вторичного пара, °C (300-350) - ');
  readln(t_vt1);
  t_vtb:=t_vt1+15;
  write('Первоначальная длина прямого участка, м (2-4) - ');
  readln(L);
  {Выбор числа секций}
  z:=0; dz:=10;
  repeat
    z:=z+dz;
    q1:=q2;
    q2:=dq;
    writeln(z,q2:10:0);
  until (abs(q2)>abs(q1))and(z>dz);
  z:=z-dz;
  {Уточнение длины прямого участка}
  q2:=q1;

```

```

if q2<0 then dl:=0.005 else dl:=-0.005;
repeat
  L:=L+dl;
  q1:=q2;
  q2:=dq;
  writeln(L:0:3,q2:10:0);
until abs(q2)>abs(q1);
L:=L-dl;
writeln;
writeln('РЕЗУЛЬТАТЫ РАСЧЕТА');
writeln('Число секций z = ',z);
writeln('Длина прямого участка L = ',l:0:3);
writeln;
writeln('Нажмите клавишу "Enter" для записи результатов');
readln;
{Запись данных для чертежа}
assign(F,'data1.txt');
rewrite(F);
writeln(F,L*1000:0:0);
writeln(F,d2*1000:0:0);
writeln(F,s2*1000:0:0);
writeln(F,d1*1000:0:0);
writeln(F,s1*1000:0:0);
writeln(F,R*1000:0:0);
writeln(F,ntr);
writeln(F,z);
close(f);
END.

```

### Программа AutoLISP для построения чертежа паро-парового теплообменника

Программа автоматического построения чертежа разработана на языке AutoLISP системы AutoCAD. Алгоритм работы программы включает следующие этапы:

- 1) считывание с диска размеров теплообменника;
- 2) определение масштаба чертежа;
- 3) построение теплообменника;
- 4) простановка размеров.

Исходный текст программы на AutoLISP представлен далее. Чертеж теплообменника, созданный программой, показан на рис. 43.



**Рис. 43. Чертеж ШТО**

**; Размер листа**

(setq xm 210 ym 297)

**; Считывание данных из файла<sup>1</sup>**

(SetQ F (open "e:\\Sapr\\data.txt" "r"))

(SetQ Size\_l (AtoF (Read-Line F))) ; длина секции

(SetQ Size\_d2 (AtoF (Read-Line F))) ; диаметр труб греющего пара

(SetQ Size\_s2 (AtoF (Read-Line F))) ; толщина стенки труб греющего пара

(SetQ Size\_d1 (AtoF (Read-Line F))) ; диаметр труб вторичного пара

(SetQ Size\_s1 (AtoF (Read-Line F))) ; толщина стенки труб вторичного пара

(SetQ Size\_r1 (AtoF (Read-Line F))) ; радиус пучка труб

(SetQ Ntr (Atoi (Read-Line F))) ; число труб в пучке

(SetQ Ztr (Atoi (Read-Line F))) ; число пучков труб

(Close F)

**; Определение масштабного коэффициента sc**

(SetQ M0 (/ xm (\* 2.0 Size\_l)))

(SetQ sc

(Cond ((< M0 0.00125) 0.001) ; 1:1000

((< M0 0.002) 0.00125) ; 1:800

((< M0 0.0025) 0.002) ; 1:500

((< M0 0.005) 0.0025) ; 1:400

((< M0 0.01) 0.005) ; 1:200

((< M0 0.0133) 0.01) ; 1:100

((< M0 0.02) 0.0133) ; 1:75

((< M0 0.025) 0.02) ; 1:50

((< M0 0.04) 0.025) ; 1:40

((< M0 0.05) 0.04) ; 1:25

((< M0 0.0667) 0.05) ; 1:20

((< M0 0.1) 0.0667) ; 1:15

((< M0 0.2) 0.1) ; 1:10

((< M0 0.25) 0.2) ; 1:5

((< M0 0.4) 0.25) ; 1:4

((< M0 0.5) 0.4) ; 1:2.5

((< M0 1.0) 0.5) ; 1:2

((< M0 2.0) 1.0) ; 1:1

((< M0 2.5) 2.0) ; 2:1

((< M0 4.0) 2.5) ; 2.5:1

((< M0 10.0) 4.0) ; 4:1

((< M0 20.0) 10.0) ; 10:1

((< M0 40.0) 20.0) ; 20:1

((< M0 100.0) 50.0) ; 50:1

((< M0 200.0) 100.0) ; 100:1

(\* 100.0 (fic(/ M0 100)))) ; ?00:1

(SetQ scStr

(Cond ((equal sc 1 0.1) "1:1")

((equal sc 2.5 0.1) "4:1") ((equal sc 0.4 0.05)"1:4")

((< sc 1) (StrCat "1:" (RTos (/ 1 sc ) 2 0)))

((> sc 1) (StrCat (RTos sc 2 0) ":1"))))

**; Начальные установки**

(command "LIMITS" "0,0" (list xm ym))

(command "OSNAP" "OFF")

(command "ERASE" "Все" "")

---

<sup>1</sup> Используются функции open и close для открытия и закрытия файла, Read-Line – для считывания строки из файла, atof и atoi – для преобразования строки в вещественное и целое число.

```
(command "ZOOM" "Границы")
(command "-LINETYPE" "Set" "Continuous" "")
(command
  "-STYLE" "DIM" "Symbol" "0" "1" "15" "H" "H"
  "SETVAR"
  "dimtxsty" "DIM" ; стиль текста
  "dimasz" "4" ; размер стрелки
  "dimtih" "0" ; текст вдоль вертикальных линий
  "dimtoh" "0"
  "dimtad" "1" ; положение текста относительно линии
  "dimgap" "2" ; расстояние от текста до линии
  "dimtxt" "5" ; размер текста
  "dimexo" "0"
  "dimtofl" "1")
(command "-STYLE" "Standard" "" "0" "1" "15" "H" "H")
```

**; Пересчет размеров с учетом масштаба**

```
(setq d2 (* Size_d2 sc))
(setq d1 (* Size_d1 sc))
(setq r1 (* Size_r1 sc))
(setq l (* Size_l sc))
```

**; U-образные трубы**

```
(setq ymd (+ 60 (/ (- ym 65) 2)))
(setq x1 (- (/ xm 2) (/ l 2)))
(setq x2 (+ x1 l))
(setq p1 (list x1 ymd))
(setq p2 (list x2 ymd))
(setq y1 (+ ymd r1))
(setq y1l (- ymd r1))
(setq y2 (+ y1 (- (/ d2 2) (/ d1 2))))
(setq y21 (- y1l (- (/ d2 2) (/ d1 2))))
(setq y3 (+ y1 (/ d2 2)))
(setq y31 (- y1l (/ d2 2)))
(setq y4 (+ y3 (/ d1 2)))
(setq y41 (- y31 (/ d1 2)))
(setq y5 (+ y3 (/ d2 2)))
(setq y51 (- y31 (/ d2 2)))
(command "PLINE" (list x2 y1) "Ширина" "0.1" "0.1" (list x1 y1)
  "Дуга" (list x1 y1l)
  "Отрезок" (list x2 y1l) "")
(command "PLINE" (list x2 y2) (list x1 y2)
  "Дуга" (list x1 y21)
  "Отрезок" (list x2 y21) "")
(command "PLINE" (list x2 y4) (list x1 y4)
  "Дуга" (list x1 y41)
  "Отрезок" (list x2 y41) "")
(command "PLINE" (list x2 y5) (list x1 y5)
  "Дуга" (list x1 y51)
  "Отрезок" (list x2 y51) "")
```

**; Фланцы**

```
(setq yy1 (+ y5 2))
(setq yy1l (- y51 2))
(setq yy2 (- y1 2))
(setq yy21 (+ y1l 2))
(setq x3 (+ x2 d2))
(setq x4 (+ x3 (* 2 d2)))
(setq x5 (+ x4 d1))
(setq x1c (/ (+ x2 x3) 2))
(setq x2c (/ (+ x4 x5) 2))
```

```
(setq y1c (+ yy1 0))
(setq y2c (- yy1 0))
(setq y3c (+ yy1 0))
(setq y4c (- yy1 0))
```

**; Простановка размеров**

(command "SETVAR" "dimtoht" "1") ; Расположение текста горизонтально

(command "CIRCLE" (list x1c y1c) "Д" d2)

(command "DIMDIAMETR"

(list x1c (+ y1c (/ d2 2))) "Т"

(strcat "Ж" (RTos Size\_d2 2 0) (chr 180) (RTos Size\_s2 2 0))

(list x4 (+ y1c d2)))

(command "CIRCLE" (list x2c y2c) "Д" d1)

(command "DIMDIAMETR"

(list x2c (- y2c (/ d1 2))) "Т"

(strcat "Ж" (RTos Size\_d1 2 0) (chr 180) (RTos Size\_s1 2 0))

(list x5 (- y2c d1)))

(command "PLINE" (list x2 yy1) (list x2 yy2) (list x3 yy2) (list x3 yy1)

"Arc" "Center" (list x1c y1c) "Angle" "359" "")

(command "PLINE" (list x2 yy1) (list x2 yy2) (list x3 yy2) (list x3 yy1)

"Arc" "Center" (list x1c y2c) "Angle" "359" "")

(command "PLINE" (list x3 y2) (list x4 y2) "")

(command "PLINE" (list x3 y4) (list x4 y4) "")

(command "PLINE" (list x3 y2) (list x4 y2) "")

(command "PLINE" (list x3 y4) (list x4 y4) "")

(command "PLINE" (list x4 yy1) (list x4 yy2) (list x5 yy2) (list x5 yy1)

"Arc" "Center" (list x2c y1c) "Angle" "359" "")

(command "PLINE" (list x4 yy1) (list x4 yy2) (list x5 yy2) (list x5 yy1)

"Arc" "Center" (list x2c y2c) "Angle" "359" "")

(setq xs (+ x1 50))

(command "SETVAR" "dimtoht" "0") ; Текст по линии

(command "DIM"

"HOR" (list x1 ymd) (list x1c y2c)

(list x1c (- y2c 10))

(RTos Size\_l 2 0)

"V" (list xs y1) (list xs y1) (list xs y1)

(RTos (\* 2 Size\_r1) 2 0)

"V" (list (+ xs 30) y5) (list (+ xs 30) y5)

(list (+ xs 30) y5)

(RTos (\* 2 (+ Size\_r1 Size\_d2)) 2 0)

"Exit")

(command "TEXT"

"C" "Standard" (list x1 (- y2c 20)) "5" ""

(StrCat "Число тpyб " (IToA ntr) "x" (IToA ztr)))

**; Осевые линии**

(command "LINETYPE" "Set" "Center" "")

(command "LINE" (list x5 y3) (list x1 y3) "")

(command "ARC" (list x1 y3) "C" (list x1 ymd) (list x1 y3) "")

(command "LINE" (list x1 y3) (list x5 y3) "")

(command "LINE" (list (- x1 30) ymd) (list (+ x5 20) ymd) "")

(command "LINE" (list x1c (+ y1c d2)) (list x1c (- y2c d2)) "")

(command "LINE" (list x2c (+ y3c d1)) (list x2c (- y4c d1)) "")

(command "LINE" (list (- x1c 5) y1c) "@10,0" "")

(command "LINE" (list (- x1c 4) y2c) "@10,0" "")

(command "LINE" (list (- x2c 5) y3c) "@10,0" "")

(command "LINE" (list (- x2c 4) y4c) "@10,0" "")

(command "LINETYPE" "Set" "Continuous" "")

(command "ZOOM" "All")

## Заключение

В учебном пособии рассмотрены основы автоматизированного проектирования и выполнения конструкторских работ на компьютере. За пределами издания остался ряд важных вопросов – разработка математического обеспечения САПР, трехмерное моделирование, AutoCAD DesignCenter (ADC), использование баз данных, вывод чертежей на печатающее устройство и многое другое.

Причина этого – не только ограничение в объеме дисциплины, для изучения которой предназначено учебное пособие. Современные информационные технологии и программные средства очень сложны и предоставляют пользователю для работы большое количество разнообразных инструментов. Согласно известному афоризму Козьмы Прутова, «нельзя объять необъятное» и не существует человека, который владеет всеми возможностями не только AutoCAD, но даже и Microsoft Word или Excel. Это связано не с недостатком образования и профессиональных навыков, а с тем, что специалист использует только ту часть систем, которая ему необходима для решения своих профессиональных задач.

Задача пособия – продемонстрировать возможности, методы и способы автоматизации проектирования в деятельности инженера-теплоэнергетика и теплотехника. Представленные материалы являются отправной точкой в этой области. Лучший способ дальнейшего освоения информационных технологий и превращения в специалиста по AutoCAD и САПР – постоянное расширение применения компьютера в своей профессиональной деятельности, постановка и решение новых более сложных задач. Ведь информатика – это прикладная наука, которую нельзя изучить по книгам. Она осваивается только при практической работе на компьютере. О квалификации специалиста в этой области говорит не количество изученных им программных продуктов и языков программирования, а способность успешно решать конкретные прикладные задачи.

## Библиографический список

1. Бугрименко Г. А. Автоматизация конструирования на ПЭВМ с использованием системы AutoCAD / Г. А. Бугрименко, В. Н. Лямке, Э.-К. С. Шейбокене. – М.: Машиностроение. – 1993. – 336 с.
2. Норенков И. П. Основы автоматизированного проектирования: Учеб. для вузов. – М.: Изд-во МГТУ им. Н. Э. Баумана, 2002. – 333 с.
3. Трубаев П. А. Моделирование и оптимизация технологических процессов производства строительных материалов: Учеб. пособие / П. А. Трубаев. – Белгород, 1999. – 178 с.
4. ГОСТ 22847–2000. Проектирование автоматизированное. Термины и определения. – М., 2000.
5. ГОСТ 23501.101–87. Системы автоматизированного проектирования. Основные положения. – М., 1987.
6. ГОСТ 23501.108–85. Система автоматизированного проектирования. Классификация и обозначение. – М., 1985.
7. Омура Д. AutoCAD 2000: Справочное руководство / Д. Омура, Р. Каллори. – М.: Лори, 2000. – 356 с.
8. Полещук Н. Н. AutoCAD 2002 / Н. Н. Полещук. – СПб.: ВHV, 2004. – 608 с.
9. Финкельштейн Э. AutoCAD 2000: Библия пользователя / Э. Финкельштейн. – М.: Диалектика, 1999. – 1040 с.
10. Полещук Н. Н. VisualLISP и секреты адаптации AutoCAD / Н. Н. Полещук. – СПб.: ВHV, 2001. – 576 с.
11. Кудрявцев Е. М. AutoLISP. Основы программирования в AutoCAD / Е. М. Кудрявцев. – М.: ДМК Пресс, 2001. – 416 с.
12. Джамп Д. AutoCAD. Программирование: Пер. с англ. – М.: Радио и связь, 1992. – 336 с.

## Предметный указатель

- Автопривязка 62
- Алгоритм 27
- Аналитические методы 26
- Возмущающие параметры 20, 22, 30
- Входные параметры 20, 22, 30
- Выходные параметры 20, 22, 30
- Главное меню AutoCAD 43
- Графическое окно AutoCAD 44
- Деструктуризация 21
- Детерминированное описание 24
- Динамические модели 25
- Зависимые размеры 12, 121
- Задачи идентификации 23
- Интерполяция 24
- Идентификация 23
- Интерпретаторы 17
- Информационное обеспечение 9
- Команды AutoCAD 46
  - 1D 77
  - ADCENTER 85
  - APPLOAD 109
  - ARC 56
  - AREA 77
  - ARRAY 72
  - BASE 84
  - BHATCH 82
  - BHATCH 82
  - BLOCK 83
  - BREAK 75
  - CHAMFER 76
  - CHANGE 54, 74
  - CIRCLE 56
  - COLOR 55
  - COLOR 55
  - COPY 71
  - DBLIST 77
  - DDIM 80, 81
  - DIM 78
  - DIMALIGNED 79
  - DIMANGULAR 80
  - DIMBASELINE 80
  - DIMCONTINUE 80
  - DIMDIAMETER 80
  - DIMLINEAR 79
  - DIMORDINATE 79
  - DIMRADIUS 79
  - DIST 77
  - DTEXT 65
  - ERASE 73
  - EXPLODE 76, 84
  - EXTEND 76
  - FILLET 76
  - GRID 50
  - HATCH 82
  - INSERT 83
  - INSERT 84
  - LAYER 53
  - LEADER 80
  - LIMITS 49
  - LINE 54
  - LINETYPE 61
  - LINETYPE 61
  - LIST 77
  - LTSCALE 62
  - MINSERT 84
  - MIRROR 73
  - MOVE 71
  - MTEXT 65
  - OOPS 49
  - ORTHO 51
  - PEDIT 75
  - PLINE 59
  - PURGE 54
  - REDRAW 52
  - REDRAWALL 52
  - REGEN 52
  - REGENALL 52
  - ROTATE 73
  - SCALE 74
  - SCRIPT 16
  - SELECT 71
  - SETVAR 42, 51
  - SNAP 50
  - STRECH 74
  - TEXT 65
  - TOLERANCE 80
  - TRIM 76
  - U 48
  - UNDO 48
  - UNITS 49, 80
  - VLIDE 106
  - ZOOM 51
  - БАЗА 84
  - БДСПISOK 77
  - БЛОК 83

ВСЕОСВЕЖИ 52  
ВСЕРЕГЕН 52  
ВСТАВЬ 83  
-ВСТАВЬ 84  
ВЫБЕРИ 71  
ВЫНОСКА 80  
ДИАЛРАЗМ 81  
ДИСТ 77  
ДОПУСК 80  
ДТЕКСТ 65  
ДУГА 56  
ЕДИНИЦЫ 49, 80  
ЗЕРКАЛО 73  
ИЗМЕНИ 54, 74  
КООРД 77  
КОПИРУЙ 71  
КРУГ 56  
КШТРИХ 82  
-КШТРИХ 82  
ЛИМИТЫ 49  
ЛМАСШТАБ 62  
МАССИВ 72  
МАСШТАБ 74  
МВСТАВЬ 84  
МТЕКСТ 65  
О 48  
ОБРЕЖЬ 76  
ОЙ 49  
ОРТО 51  
ОСВЕЖИ 52  
ОТМЕНИ 48  
ОТРЕЗОК 54  
ПАКЕТ 16  
ПЕРЕНЕСИ 71  
ПЛИНИЯ 59  
ПЛОЩАДЬ 77  
ПОВЕРНИ 73  
ПОКАЖИ 51  
ПОЛРЕД 75  
РАЗМЕР 78  
РАЗОРВИ 75  
РАСТЯНИ 74  
РАСЧЛЕНИ 76, 84  
РЕГЕН 52  
РЗМБАЗОВЫЙ 80  
РЗМДИАМЕТР 80  
РЗМЛИНЕЙНЫЙ 79  
РЗМОРДИНАТА 79  
РЗМПАРАЛ 79  
РЗМРАДИУС 79  
РЗМУГЛОВОЙ 80  
РЗМЦЕПЬ 80

СЕТКА 50  
-СЛОЙ 53  
СОПРЯГИ 76  
СОТРИ 73  
СПИСОК 77  
ТЕКСТ 65  
ТИПЛИН 61  
-ТИПЛИН 61  
УДАЛИ 54  
УДЛИНИ 76  
УСТПЕРЕМ 42, 51  
ФАСКА 76  
ЦВЕТ 55  
-ЦВЕТ 55  
ШАГ 50  
ШТРИХ 82

Критерий оптимизации 29  
    обобщенный 32  
Компиляторы 17  
Комплексная деталь 11  
Конструктивные размеры 13  
Конструктивный расчет 5  
Координаты 45, 47  
Лингвистическое обеспечение 9  
Математическая модель 22  
Математическое моделирование 22  
Математическое обеспечение 8  
Математическое описание 23  
Материальная модель 22  
Макропроцесс 21  
Методическое обеспечение 9  
Моделирование 22  
Моделирующий алгоритм 23  
Независимые размеры 13  
Непараметризованные размеры 13  
Нестационарные процессы 25  
Нормирование 32  
Обратные задачи 23  
Объектная привязка 62  
Ограничения 26, 29  
Окно команд AutoCAD 44  
Определяющие размеры 12, 121  
Оптимизация 29  
Организационное обеспечение 9  
Отклик 21, 24  
Отладка программ 18

Пакетные файлы	16	
Панели инструментов AutoCAD	43	
Параметризация	10, 11	
Параметры, классификация	20	
Параметризованные размеры	12	
План эксперимента	24	
Поверочный расчет	5, 34	
Полузависимые размеры	13	
Приглашение AutoCAD	46	
Программирование	10, 15, 17	
Программное обеспечение	8	
Проект	5	
Проектирование	5	
автоматизированное	7	
автоматическое	8	
неавтоматизированное	7	
частично автоматизированное	7	
Проектные операции	7	
Проектные процедуры	6	
Проектные решения	6	
Прямые задачи	23	
Псевдоимена команд AutoCAD	48	
Рабочий проект	6	
Размерные параметры	13	
Распределенные параметры	25	
Свободные параметры	13	
Связи	19	
Синтез	21	
структурный	5	
Система	19	
Системы координат	45	
Системный анализ	21	
Системные переменные	42, 51	
AREA	77	
AUNITS	81	
CELTSCALE	62	
CELTYPE	61	
COORDS	45	
DIMASZ	81	
DIMLFAC	81	
DIMRND	81	
DIMSCALE	81	
DIMITAD	81	
DIMITIN	74	
DIMITIN	81	
DIMITIX	81	
DIMITOFL	81	
DIMTSZ	81	
DIMTXT	81	
FILEDIA	16	
LTSCALE	62	
LUNITS	81	
MIRRTEXT	73	
Perimeter	77	
SPLINETYPE	75	
UCSICON	46	
P3MVCIN	81	
P3MVL3AC	81	
P3MVL3TR	81	
P3MP3MB	81	
P3MTEKCT	81	
P3MTMB	81	
P3MTMEЖГ	81	
P3MTHPL	81	
Слой	53	
Сосредоточенные параметры	25	
Стадии проектирования	6	
Структурное программирование	27	
Структурный синтез	5	
Статистическое описание	24	
Статические модели	25	
Стационарные процессы	25	
Сценарии	16	
Технический проект	6	
Техническое задание	5, 6	
Техническое обеспечение	9	
Типовые процессы	21	
Трансляторы	17	
Управляющие параметры	20, 22, 30	
Файлы, расширения		
dwg	41	
dwt	42	
scr	16	
Фактор	21, 24	
Физическая модель	22	
Физическое описание	23	
Формализованное описание	23	
Формат бумаги	49	
Функции AutoLISP	86	
-	88	
*	88	
/	88	
/=	100	
+	87, 88	
<	100	

<=	100	If	102
=	100	InitGet	97
>	100	Inters	88
>=	100	Itoa	97
1-	88	Last	99
1+	88	Length	99
Abs	88	List	90, 98
Alert	95	Load	110
And	101	Log	88
Angle	88	Max	88
Angtos	97	Member	99
Appned	99	Min	88
Ascii	98	MinusP	100
Atan	88	Not	101
AtoF	98	Nth	99
AtoI	97	Null	101
Atom	101	NumberP	101, 118
BoundP	101	Or	101
Car	99	Polar	88
Cdr	99	PrinI	95
Chr	97	PrinC	95
Command	110	Print	95
Cond	102	Progn	102
Cons	99	Prompt	95
Cos	88	Quote	91
DeFun	116	Rem	88
Distance	88	Repeat	103
EndLast	113	Reverse	99
EndNext	113	Rtos	97
EntDel	115	SetQ	90
EntGet	113	Sin	88
EntMake	115	Sqrt	87, 88
EntMod	115	StrCat	98
EntSel	113	StrLen	98
EntUpd	115	SubStr	98
Equal	100	Terpri	95
Eval	91, 117	While	103
Exp	88	Write-Char	95
Expt	88	ZeroP	100
Fix	88		
Float	88	Целевая функция	30
Gcd	88	Шаблон	42
GetAngle	93	Численные методы	26
GetCorner	93	Экспериментальная точка	24
GetDist	93	Экстраполяция	24
GetInt	93, 94	Элементарные процессы	21
GetKWord	93, 94	Элементы системы	19
GetOrient	93	Эскизный проект	6
GetPoint	93, 94	Язык программирования	17
GetReal	93		
GetStrings	93		
GrText	95		



## Оглавление

<b>Введение</b> .....	3
<b>1. Основы автоматизированного проектирования</b> .....	5
1.1. Этапы проектирования .....	5
1.2. Автоматизация проектирования .....	7
1.3. Состав САПР .....	8
1.4. Автоматизация чертежных работ .....	10
1.5. Параметризация чертежей и комплексные детали .....	11
Классификация параметризованных размеров (12). Средства создания комплексных деталей (14)	
1.6. Программирование чертежных работ .....	15
Сценарии и пакетные файлы (16). Язык программирования (17)	
1.7. Моделирование высокотемпературных и теплоэнергетических установок .....	19
Системный анализ технических объектов (19). Понятие о моделях и моделировании (22). Оптимизация технических систем (29). Особенности моделирования парогенерирующих установок (34). Особенности моделирования и проектирования кожухотрубных теплообменников (36)	
<b>2. Автоматизация чертежных работ в системе AutoCAD</b> .....	41
2.1. Общие принципы работы в системе .....	41
Запуск системы AutoCAD и создание нового чертежа (41). Вид экрана при работе в AutoCAD (43). Координаты курсора и системы координат (45). Вызов команд (46). Отмена действий (48)	
2.2. Установка параметров и масштаба чертежа .....	49
Выбор масштаба чертежа (51). Работа со слоями (53)	
2.3. Построение графических объектов .....	54
2.4. Объектная привязка .....	62
2.5. Ввод текста .....	65
2.6. Редактирование объектов чертежа .....	69
Выбор объектов (69). Перемещение, копирование и удаление объектов (71). Изменения размеров и положения объектов (73). Редактирование объектов (74). Сопряжение нескольких объектов (76). Получение информации о чертеже (77)	
2.7. Простановка размеров .....	77
2.8. Штрихование .....	81
2.9. Блоки .....	83
<b>3. Язык программирования AutoLISP</b> .....	86
3.1. Основные понятия языка AutoLISP .....	86
Списки и функции (86). Комментарии (89). Запись координат точек (90). Типы и инициализация переменных (90). Ссылки на переменные (91). Ввод значений переменных с клавиатуры (92). Вывод информации на экран (95). Запись сложных выражений (96). Функции для работы с текстом (97). Функции для работы со списками (98). Логические функции и функции управления порядком выполнения программы (100)	
3.2. Работа в среде VisualLISP .....	105
Окна VisualLISP (105). Отладка программ (107). Загрузка в AutoCAD программ AutoLISP (109).	

3.3. Использование языка AutoLISP в среде AutoCAD .....	110
Использование выражений AutoLISP при вводе команд AutoCAD (110). Функция Command (110). Отладка функции Command (112). Доступ к примитивам чертежа (113)	
3.4. Создание новых функций и команд .....	116
Описание новых функций(116). Описание новых команд (118). Создание библиотеки функций (120).	
<b>4. Автоматизированное проектирование элементов теплоэнергетического оборудования.....</b>	<b>121</b>
4.1. Создание программы построения параметризованного чертежа....	121
4.2. Проектирование горелки .....	124
4.3. Проектирование паро-парового теплообменника.....	130
Назначение паро-парового теплообменника (130). Особенности расчета паро-парового теплообменника (130). Программа конструктивного расчета паро-парового теплообменника (131). Программа AutoLISP для построения чертежа паро-парового теплообменника (135)	
<b>Заключение .....</b>	<b>139</b>
<b>Библиографический список .....</b>	<b>139</b>
<b>Предметный указатель .....</b>	<b>140</b>

Учебное издание

**ТРУБАЕВ Павел Алексеевич**

**АВТОМАТИЗИРОВАННОЕ ПРОЕКТИРОВАНИЕ  
ЭНЕРГОТЕХНОЛОГИЧЕСКОГО ОБОРУДОВАНИЯ**

Учебное пособие

Редактор Г. Н. Афолина

Подписано в печать 29.07.05. Формат 60×84<sup>1</sup>/<sub>16</sub>. Усл. печ. л. 8,49. Уч.-изд. л. 9,13.  
Тираж 100 экз. Заказ . Цена р. к.

Отпечатано в Белгородском государственном технологическом университете  
им. В. Г. Шухова. 308012, г. Белгород, ул. Костюкова, 46