

С.Н.САУТИН
А.Е.ПУНИН

МИР КОМПЬЮТЕРОВ И ХИМИЧЕСКАЯ ТЕХНОЛОГИЯ



ЛЕНИНГРАД «ХИМИЯ»
ЛЕНИНГРАДСКОЕ ОТДЕЛЕНИЕ
1991

ББК 6П7.1
С216
УДК 54.04:681

Рецензент — д-р хим. наук профессор *В. Н. Пак*

Саутин С. Н., Пунин А. Е.

С216 Мир компьютеров и химическая технология.— Л.: Химия, 1991.— 144 с.: ил.
ISBN 5—7245—0174—0

Рассказывается о роли вычислительной техники в интенсификации химических процессов, ускорении научно-технического прогресса. В увлекательной форме изложены принципы подготовки задач для решения на ЭВМ, описаны устройство и принцип действия компьютеров. Приведены примеры расчетов с помощью калькуляторов и основы программирования на алгоритмическом языке БЕЙСИК. Показаны возможности персональных компьютеров. Отмечены важнейшие направления использования ЭВМ в химии и химической технологии.

Для учащихся средних школ, техникумов и профессионально-технических училищ, изучающих основы информатики и вычислительной техники. Может быть полезна студентам химико-технологических специальностей вузов.

С $\frac{2802000000-073}{050(01)-91}$ 73—91

ББК 6П7.1

ISBN 5—7245—0174—0

© С. Н. Саутин, А. Е. Пунин, 1991

ПРЕДИСЛОВИЕ

С появлением вычислительных машин начался новый этап в развитии химии и технологии. Открылись возможности широкого использования математических методов для построения математических моделей химических реакций, обработки результатов экспериментальных исследований, поиска оптимальных условий проведения технологических процессов. В связи с этим перед химиками встала задача овладения вычислительной техникой. От современного специалиста в области химии и химической технологии требуется знание возможностей вычислительных машин и умение их использовать в своей профессиональной деятельности.

Благодаря использованию ЭВМ в научных исследованиях, деятельность специалистов становится более производительной и целеустремленной. Использование ЭВМ при проектировании химических заводов позволяет сократить сроки выполнения работ и одновременно повысить качество проектов. ЭВМ на производстве дают возможность создавать системы автоматического управления, контроля и автоматической защиты. Наконец, применение ЭВМ в процессе обучения химиков-технологов позволяет повысить их познавательную активность и расширить круг решаемых профессиональных задач.

В наши дни компьютеры становятся незаменимыми помощниками химиков. Книга, которую Вы держите в руках, рассказывает об этом. Она адресована широкому кругу читателей, желающих узнать, где и как используются микрокалькуляторы и персональные компьютеры в области химии и химической технологии. Для чтения книги не требуются специальные знания в области математики и программирования. Книга может быть полезна также учащимся старших классов средних школ, техникумов и профтехучилищ.

В книге читатель найдет достаточное количество программ для компьютера и подробно разобранные примеры вычислений. Это обусловлено стремлением авторов не только поставить задачи, но и показать пути их решения.

Авторы считают своим долгом выразить глубокую благодарность рецензенту д-ру хим. наук профессору В. Н. Паку за ценные замечания и предложения, учтенные при окончательной доработке рукописи.

Итак, мы начинаем разговор о применении компьютеров, которые образно называют катализаторами научно-технического прогресса.

ПЕРВОЕ ЗНАКОМСТВО С КОМПЬЮТЕРОМ**МАШИНА ОБРАБАТЫВАЕТ ИНФОРМАЦИЮ**

Первые построенные человеком машины предназначались для *обработки материалов*: с помощью этих машин материалам придавалась новая форма или они превращались в другие материалы. Примерами такого рода машин могут служить гончарный круг и ткацкий станок. С появлением паровой машины на сцену выступают *энергетические машины*, которые служат для преобразования одной формы энергии в другую. Сегодняшние представители энергетических машин — гигантские турбины и генераторы — действуют на современных электростанциях. Машины нового типа, которым посвящена эта книга, — электронные вычислительные (ЭВМ, компьютеры). Их называют также *информационными машинами*, поскольку они предназначены для обработки информации.

С информацией всегда приходится иметь дело, когда необходимо получить новое вещество и переработать его в изделия. Отсюда ясно, что важной областью применения ЭВМ является обработка информации в области химии и химической технологии.

Однако ЭВМ обнаруживают еще одно существенное отличие от привычных нам орудий производства. Все прежние машины увеличивали физические возможности человека. Колесо повысило скорость передвижения человека. Токарный станок во много раз увеличил способность человеческой руки придавать предметам нужную форму. Наконец, электрогенераторы позволили в огромной степени увеличить мускульную силу человека. С этой точки зрения особенность ЭВМ состоит в том, что они расширяют не физические, а *умственные* способности человека. Только с появлением компьютеров стало возможным создание систем *искусственного интеллекта*. Судя по той роли, которую ЭВМ уже сыграли в развитии цивилизации, можно ожидать, что их дальнейшее совершенствование внесет еще больший вклад в научно-технический прогресс.

Что же представляет собой современная ЭВМ, из каких основных устройств она состоит?

КАК УСТРОЕНА ЭВМ

Электронные вычислительные машины справедливо называют чудом XX века. Несмотря на большое разнообразие ЭВМ, можно уловить определенное сходство их структур и указать типовые устройства, которые входят в каждую из них. Упрощенная структура ЭВМ представлена на рис. 1. Для простоты рисунка опущены линии связи устройства управления с внешним запоминающим устройством, с устройствами ввода и вывода информации, не показан источник питания.

Итак, мы переходим к рассмотрению основных блоков электронной вычислительной машины.

Исходная информация в виде программы вычислений и данных для расчетов поступает на устройство ввода. Часто эта информация бывает нанесена на перфокарты, перфоленты или различные магнитные носители в виде лент или дисков. В последнее время широкое распространение получили гибкие магнитные диски (флôппи-диски, или дискеты).

Удобно вводить информацию в ЭВМ с помощью *дисплéя*. Внешне дисплей напоминает обычный телевизор, перед экраном которого расположена клавиатура, как у пишущей машинки. На экране дисплея появляется текст, который набирается на клавиатуре.

Вводимая информация поступает в оперативное запоминающее устройство, которое предназначено для хранения программы вычислений, исходных данных, промежуточных и конечных результатов расчетов. Элементарной базой оперативной памяти ЭВМ обычно служат магнитные сердечники или электронные схемы.

Внешнее запоминающее устройство ЭВМ, в отличие от оперативного, имеет несравненно бôльшую емкость, но существенно меньшее быстродействие. Во внешней памяти хранится стандартное программное обеспечение ЭВМ и иная разнообразная информация. Например, там могут содержаться обширные данные

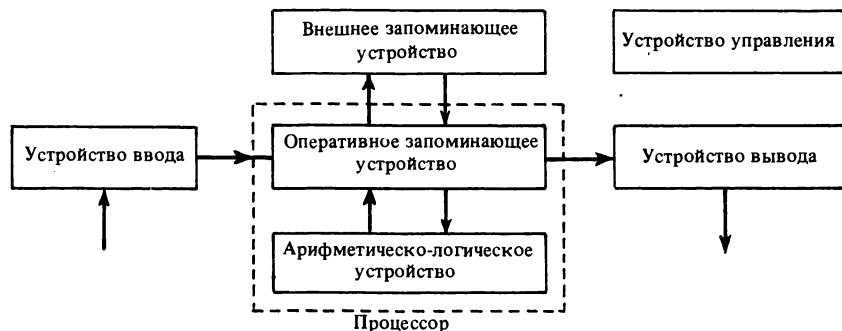


Рис. 1. Упрощенная структура электронной вычислительной машины

о физико-химических свойствах веществ, о способах их получения и промышленного производства.

Арифметико-логическое устройство осуществляет арифметические и логические операции над числами в процессе вычислений.

Устройство управления «руководит» работой ЭВМ в соответствии с введенной в нее программой.

Арифметико-логическое устройство и устройство управления образуют в совокупности *процессор* — главную составную часть электронной вычислительной машины. В состав процессора иногда включается и *оперативная память*.

Вывод результатов вычислений из ЭВМ осуществляется на печатающее устройство (принтер), дисплей или графопостроитель (плоттер).

Устройство управления ЭВМ предназначено также для контроля качества функционирования компьютера, осуществляемого обслуживающим персоналом.

Рассмотренное нами краткое описание вычислительной машины дает лишь самое общее представление об основных ее устройствах и взаимодействиях между ними в процессе работы. Теперь мы можем более подробно познакомиться с некоторыми из них.

ДИСПЛЕЙ

Экранным устройством отображения информации является дисплей (рис. 2).

Дисплеи бывают разных типов. Простейшие показывают только несколько строк текста с одним видом шрифта, более сложные — текст с прописными и строчными буквами. Отдельные части текста могут выделяться повышенной яркостью. Некоторые современные модели дисплеев высвечивают тексты различными шрифтами, отдельные элементы текста выделяются несколькими градациями яркости и многими оттенками *цвета*.

К большой ЭВМ можно присоединить до сотни дисплеев, которые могут быть удалены от нее на значительное расстояние, измеряемое километрами.

Когда нажимаешь на пульте дисплея клавишу с изображением, скажем, буквы А, то эта буква высвечивается на экране. В принципе она может появиться в любом месте экрана. Чтобы поместить символ в нужную часть экрана, пользуются *курсором* (маркером). Курсор — это метка на экране, указывающая место, где появится очередной символ. После нажатия клавиши введенная буква высвечивается именно там, куда указывал курсор, а сам курсор передвигается на одну позицию вправо или, если строка кончилась, к началу следующей строки. Специальные клавиши на пульте дисплея позволяют перемещать курсор вправо и влево, вверх и вниз, к началу следующей строки, к левому верхнему углу экрана.

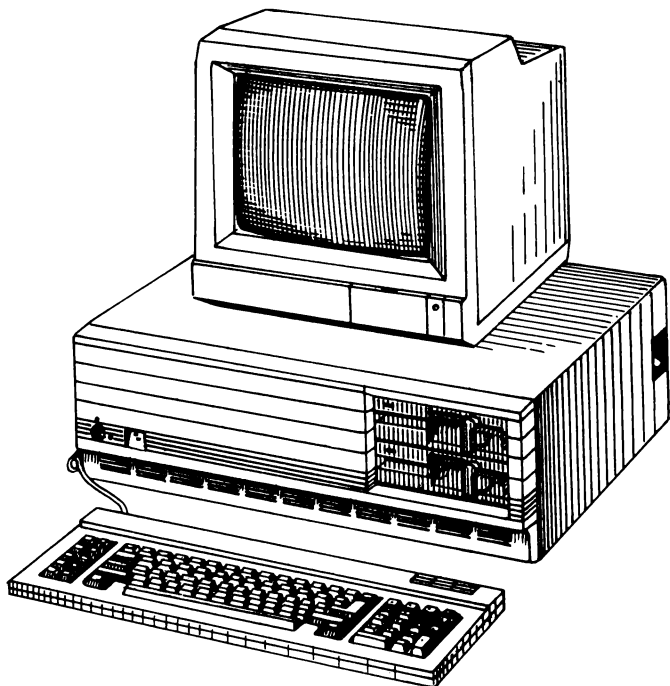


Рис. 2. Алфавитно-цифровой дисплей
в составе персональной ЭВМ

Предположим, что мы по ошибке нажали не ту клавишу пульта дисплея. Тогда можно вернуть курсор на одну позицию влево, к только что введенной неверной букве. Если теперь нажать клавишу с нужной буквой, она и появится на экране, а прежний символ исчезнет.

На многих дисплеях можно сдвигать и раздвигать слова и строки, поэтому из-за случайно пропущенной буквы или слова не приходится «переписывать» строку, абзац или страницу. Достаточно подвести курсор к нужному месту экрана, включить режим «вставка» и нажать клавишу с пропущенным символом.

Приведем пример такой корректировки текста на экране дисплея. Пусть был ошибочно набран текст

РАЛОТАЕМ НА ДСПЛЕЕ

Вы, вероятно, заметили две ошибки, которые мы подчеркнули. На многих дисплеях курсор имеет вид такой черточкой, располагающейся под указываемым местом.

Для исправления ошибок подводим курсор к месту, куда нужно вставить букву, включаем режим «вставка» и нажимаем клавишу с буквой И. Затем подводим курсор к букве Л и нажимаем

клавишу с буквой Б. Теперь на дисплее должны быть слова без ошибок:

РАБОТАЕМ НА ДИСПЛЕЕ

Некоторые дисплеи оборудованы дополнительными приспособлениями, например устройством типа «мышь», т. е. коробочкой на колесиках, соединенной с дисплеем гибким проводом. Если ее катать по столу или по другой поверхности, то в том же направлении по экрану перемещается и курсор. Изменив направление движения «мыши», можно направить курсор в нужную часть экрана.

Вместо «мыши» или дополнительно к ней в некоторых современных дисплеях используется устройство *джойстик* (joystick — термин, не имеющий пока русского аналога). Джойстик имеет вид рычажка, вертикально торчащего из пульта, причем рычажок этот можно наклонять во всех направлениях. Вместе с движением рычажка по экрану движется курсор или иной объект, в зависимости от заложенной в компьютер программы.

В ряде дисплеев используется *световое перо* — стержень на гибком проводе. Этим пером можно указать на отдельный участок изображения на экране. Пусть, например, на экране высвечено «меню», т. е. список предлагаемых услуг ЭВМ. Указав световым пером нужную строку, можно сообщить компьютеру, что выбрано.

В последнее время появились *сенсорные дисплеи*, снабженные тачскрином (английские слова touch screen означают «касание экрана»). Достаточно коснуться пальцем какой-либо части изображения на экране такого дисплея — и ЭВМ воспримет определенную команду пользователя.

КОМПЬЮТЕРНАЯ ПАМЯТЬ

Объем компьютерной памяти обычно измеряют в *байтах*. Байт — это такой участок памяти, в котором размещается один символ — буква, цифра или знак препинания. В одном байте можно хранить восемь *битов* — элементарных единиц информации, каждая из которых эквивалентна одному ответу на вопрос типа «да — нет». Тысячу байтов (точнее 1024 байта) обозначают понятием *килобайт* (кбайт), 1024 кбайт равны одному *мегабайту* (Мбайт), 1024 Мбайт — одному *гигабайту* (Гбайт).

Возможности запоминания и хранения информации в современных ЭВМ настолько велики, что поражают воображение. Например, на одной бобине (катушке) магнитной ленты диаметром менее 30 см могут уместиться сотни миллионов букв или цифр (сотни мегабайт). ЭВМ крупных вычислительных центров хранят в своей памяти десятки миллиардов символов (десятки гигабайт).

Японская фирма «Тосиба» приступила к выпуску энциклопедий и справочников на магнитных носителях. На компактном магнитном диске диаметром 13 см записывается более 300 Мбайт, или, другими словами, текст 7500 учетно-издательских листов, т. е. содержание примерно 100 томов формата Большой советской энциклопедии. При помощи компьютерной приставки-преобразователя записанные материалы можно воспроизвести на экране дисплея.

В СССР на базе ЭВМ созданы мощные информационно-поисковые системы. Они эффективно используются, в частности, в химии и химической технологии, помогая специалистам находить нужную информацию.

МАКРО, МИНИ, МИКРО

Сколько вокруг нас вычислительной техники! ЭВМ можно встретить на уроке информатики в школе, в дисплейном зале вуза, на заводе, в исследовательской лаборатории, на борту самолета и космического корабля. Компьютеры решают различные задачи, отличаются друг от друга конструкцией, внешним видом и размерами. Можно ли упорядочить это множество ЭВМ, т. е. разделить его на отдельные группы? Оказывается, можно.

По принципу действия ЭВМ делятся на *аналоговые* (АВМ), *цифровые* (ЦВМ) и *гибридные* (ГВМ).

Аналоговые вычислительные машины называют также *моделирующими* установками. Они отличаются от ЦВМ тем, что оперируют непрерывными величинами. Такими величинами чаще всего являются напряжения на различных участках электрической схемы, причем структура схемы строго соответствует условию решаемой задачи. Измерения осуществляются стрелочными или цифровыми вольтметрами. В состав АВМ входят типовые решающие блоки, такие как сумматоры, интеграторы, инвертеры. Из них, как из детских кубиков, строится схема для решения задачи.

В основе использования АВМ лежит *аналогия* между математической формулировкой решаемой задачи и системой уравнений, описывающей электрические процессы, протекающие в вычислительной машине.

Наибольшее применение в химической промышленности АВМ нашли при исследовании систем автоматического управления технологическими процессами.

ЦВМ иногда называют машинами *дискретного* действия. В процессе вычислений они оперируют цифрами. ЦВМ — это наиболее представительная и многочисленная группа вычислительных машин.

ГВМ объединяют в себе достоинства аналоговых и цифровых машин.

По области применения ЭВМ можно разделить на *универсальные* и *управляющие*. Универсальные применяются для выполнения любых инженерных, научных и экономических расчетов. Например, с помощью универсальных ЭВМ химики рассчитывают структуры сложных молекул, планируют условия проведения эксперимента, ведут обработку экспериментальных данных, проектируют технологическое оборудование заводов.

Управляющие ЭВМ используются для решения задач оптимального управления различными объектами в лабораторных или производственных условиях. Например, бортовая ЭВМ входит в состав автопилота воздушного лайнера. На заводах ЭВМ управляют процессами выплавки и разливки стали, процессами получения и переработки пластмасс.

По масштабам решаемых задач ЭВМ делятся на большие (макро), малые (мини) и сверхмалые (микро).

Большие ЭВМ используются на вычислительных центрах и в составе информационно-поисковых систем. Такие машины можно встретить в вузах, научно-исследовательских и проектных институтах, на производственных объединениях.

Мини-ЭВМ используются для выполнения расчетов или для управления производственными процессами в масштабе цеха или нескольких цехов. Они могут подключаться к большим ЭВМ в качестве *терминалов*. Это значит, что в случае необходимости задача передается по кабельным линиям связи на большую ЭВМ, обладающую большей скоростью счета и объемом памяти. Естественно, что там задача решается значительно быстрее. Результаты вычислений по тем же линиям связи снова поступают в мини-ЭВМ, где и выдаются на печать.

Микро-ЭВМ — продукт микроминиатюризации электронных схем. Это машины, собранные на одном или нескольких кристаллах (чипах). Благодаря успехам микроэлектроники налажен массовый выпуск больших (БИС) и сверхбольших (СБИС) интегральных схем. Основным элементом микро-ЭВМ является *микропроцессор*. В большинстве современных микро-ЭВМ в качестве внешних запоминающих устройств применяются *гибкие* и *жесткие магнитные диски*. Возможности микро-ЭВМ достаточно велики, несмотря на их малые размеры. С помощью этих машин химики выполняют инженерные и научные расчеты.

ЧИСЛА ИЗ НУЛЕЙ И ЕДИНИЦ

Человечество издавна пользуется десятичной системой счисления. Мы ее используем и даже не задумываемся о других существующих системах счисления. Еще в глубокой древности счет осуществлялся с помощью пальцев правой и левой руки. Поэтому за основание системы счисления было взято число 10. Но то, что удобно для человека, к сожалению, не всегда «удобно» для вычисли-

тельной машины. Например, для ЭВМ более удобной оказалась двоичная система счисления. Основанием ее служит число 2. В двоичной системе существуют только две цифры — 0 и 1. Следовательно, любое число в двоичной системе счисления представляет собой определенную последовательность нулей и единиц.

Выбор двоичной системы обусловлен тем, что в процессе вычислений на ЦВМ все ее основные элементы в любой момент времени находятся в одном из двух состояний. Магнитные сердечники запоминающего устройства могут быть намагничены либо в одном, либо в другом направлении. Естественно, удобно одному из названных состояний поставить в соответствие число 1, а другому — 0. Тогда открываются широкие возможности для выполнения вычислений.

Представим себе, что кто-то решил использовать в ЭВМ десятичную систему счисления. Это значит, что каждый элемент в процессе счета должен последовательно переходить в каждое из десяти различных состояний!. Может быть и можно создать такую ЭВМ, но, несомненно, она окажется чрезвычайно сложной. Одновременно с увеличением сложности обычно резко снижается надежность. Поэтому конструкторы цифровых вычислительных машин безоговорочно отдали предпочтение двоичной системе счисления.

Рассмотрим теперь принцип построения чисел в различных системах. Пусть, например, за основание системы принято некоторое число g . Тогда любое число в этой системе будет состоять из цифр: 0, 1, ..., $(g-1)$. Любое n -разрядное число записывается в этом случае в форме

$$a_n, a_{n-1}, \dots, a_2, a_1, a_0,$$

причем различные a_i в такой записи представляют собой отдельные цифры от нуля до $(g-1)$. Значение этого числа равно

$$a_n g^n + a_{n-1} g^{n-1} + \dots + a_2 g^2 + a_1 g^1 + a_0 g^0.$$

Десятичная система счисления является частным случаем такой схемы при $g=10$. Отсюда следует, что цифры этой системы — это числа от 0 до 9. Например, число 127 состоит из цифр: 1, 2 и 7. Его значение равно

$$1 \cdot 10^2 + 2 \cdot 10^1 + 7 \cdot 10^0 = 127.$$

В двоичной системе счисления, очевидно, существуют только цифры 0 и 1. Пусть, например, с помощью них записано число 101011101011. Тогда значение этого числа равно

$$1 \cdot 2^{11} + 0 \cdot 2^{10} + 1 \cdot 2^9 + 0 \cdot 2^8 + 1 \cdot 2^7 + 1 \cdot 2^6 + \\ + 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 2795.$$

Таким образом, имея число, записанное в двоичной системе счисления, очень просто перевести его в десятичную систему.

Чтобы перейти от десятичной системы счисления к двоичной, можно применить следующий способ. Пусть, например, требуется перевести в двоичную систему число 215. Делим это число на 2, но не доводим эту операцию до конца, а обрываем ее, как только получится остаток, меньший 2:

$$\begin{array}{r} 215 \mid 2 \\ \underline{2} \\ 15 \\ \underline{14} \\ 1 \\ \underline{=} \end{array}$$

Запомним этот остаток. Теперь берем целую часть 107, получившуюся при делении, и делим ее на 2, причем опять-таки заканчиваем эту операцию, как только получаем остаток, меньший 2:

$$\begin{array}{r} 107 \mid 2 \\ \underline{10} \\ 7 \\ \underline{6} \\ 1 \\ \underline{=} \end{array}$$

Запоминаем и этот остаток. С числом 53 поступаем точно так же: делим его на 2 и записываем остаток; продолжаем действовать таким же точно образом, пока при делении не получится целочисленная часть, меньшая 2:

$$\begin{array}{r} 53 \mid 2 \\ \underline{4} \\ 13 \\ \underline{12} \\ 1 \\ \underline{=} \end{array}; \quad \begin{array}{r} 26 \mid 2 \\ \underline{2} \\ 13 \\ \underline{6} \\ 6 \\ \underline{6} \\ 0 \\ \underline{=} \end{array}; \quad \begin{array}{r} 13 \mid 2 \\ \underline{12} \\ 1 \\ \underline{=} \end{array}; \quad \begin{array}{r} 6 \mid 2 \\ \underline{6} \\ 0 \\ \underline{=} \end{array}; \quad \begin{array}{r} 3 \mid 2 \\ \underline{2} \\ 1 \\ \underline{=} \end{array}$$

Если теперь записать полученные остатки в *обратной* последовательности, т. е. начиная с последнего, и приписать в начале последнее целочисленное частное, меньшее 2, то получим представление десятичного числа 215 в двоичной системе:

11010111.

Чтобы не создавать пользователям ЭВМ лишних проблем с системами счисления, конструкторы вычислительных машин нашли простой выход — они заставили ЭВМ принять на себя весь труд по переводу чисел из одной системы счисления в другую. Таким образом, человек вводит в ЭВМ числа в привычной для него десятичной системе и получает результаты вычислений в той же системе, а все операции внутри ЭВМ осуществляются в двоичной системе.

Основной недостаток двоичной системы счисления по сравнению с десятичной заключается в большой длине двоичных чисел при записи.

ДЕЙСТВИЯ НАД МЫСЛЯМИ

Вычислительная машина выполняет не только арифметические, но и логические операции. Познакомимся с ними.

Одним из основных понятий формальной логики является *высказывание*. Им может быть любое замечание, сообщение или утверждение. Высказывание может совпадать или не совпадать с действительностью.

В первом случае высказывание называют *истинным* и присваивают ему значение 1, в других случаях высказывание называют *ложным* и присваивают ему значение 0.

В дальнейшем мы будем обозначать высказывания строчными (малыми) латинскими буквами.

Из одних высказываний с помощью логических операций можно получить другие высказывания. Простейший случай — это *отрицание* некоторого высказывания. Если дано высказывание a , то его отрицание записывается в виде $\neg a$ и читается «не a ». Так, если a означает высказывание «вещество растворимо в воде», то $\neg a$ — это высказывание «вещество нерастворимо в воде».

Отрицание и все остальные логические операции можно пояснить с помощью простых геометрических моделей. Пусть круг на рис. 3 означает высказывание a ; тогда отрицанию этого высказывания соответствует остальная площадь четырехугольника, т. е. все то, что не относится к a .

Если высказывание a истинно, т. е. имеет значение 1, то $\neg a$ — ложно, т. е. имеет значение 0, и наоборот:

a	$\neg a$
1	0
0	1

Когда даны два высказывания a и b , можно определить новое высказывание типа « a и b », которое записывается следующим образом:

$$a \wedge b.$$

Эта логическая операция называется *конъюнкцией*.

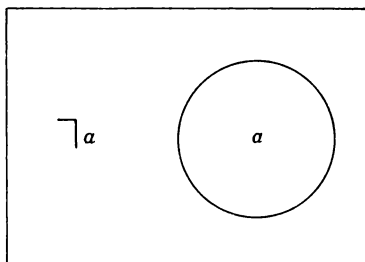


Рис. 3. Геометрическая интерпретация операции отрицания

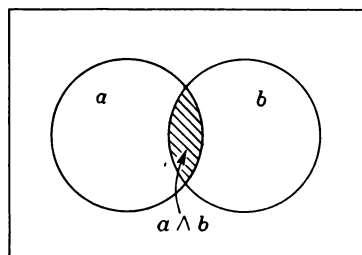


Рис. 4. Геометрическая интерпретация конъюнкции

Возьмем в качестве высказывания a утверждение «вещество является газом», а в качестве b — утверждение «вещество растворимо в воде». Тогда высказывание $a \wedge b$ будет иметь вид предложения «вещество является газом и оно растворимо в воде». Полученное новое высказывание совпадает с действительностью, т. е. истинно, только в том случае, когда истинны *одновременно* высказывания a и b .

На рис. 4 показаны два пересекающихся круга. Пусть один круг означает высказывание a , а другой — высказывание b . Тогда заштрихованная площадь, принадлежащая обоим кругам, соответствует высказыванию $a \wedge b$.

Ниже в таблице приведены результаты вычисления истинности высказывания $a \wedge b$ при различных значениях высказываний a и b :

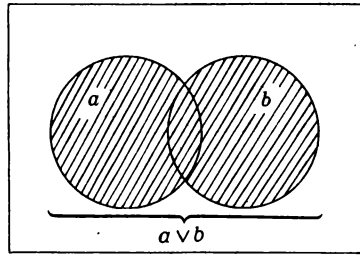
a	b	$a \wedge b$
1	1	1
1	0	0
0	1	0
0	0	0

Очевидно, что здесь представлена таблица умножения a и b . По этой причине конъюнкцию называют также *логическим произведением*.

Высказывания a и b могут быть связаны между собой словом «или», что в результате дает новое высказывание « a или b », которое записывается в виде $a \vee b$. Эта операция носит название *дизъюнкции*. Если в качестве a взять высказывание «вещество находится в твердом состоянии», а в качестве b высказывание «вещество находится в жидком состоянии», то $a \vee b$ характеризует высказывание «вещество находится в твердом или жидком состоянии».

Например, вода при температурах от -10 до $+10$ °С, как известно, бывает в виде льда или жидкости, что соответствует названной дизъюнкции.

Рис. 5. Геометрическая интерпретация дизъюнкции



Высказывание $a \vee b$ ложно только тогда, когда оба входящих в него высказывания ложны. Во всех других случаях дизъюнкция истинна, как это следует из таблицы

a	b	$a \vee b$
1	1	1
1	0	1
0	1	1
0	0	0

Пусть на геометрической схеме (рис. 5) два пересекающихся круга представляют собой высказывания a и b . Тогда дизъюнкция ($a \vee b$) изображается площадью, заштрихованной на рис. 5.

Два высказывания a и b можно с помощью слов «если» и «то» связать в высказывание «если a , то b » или «из a следует b », что символически записывается $a \subset b$. Эта операция называется *логической импликацией*.

Пусть a означает высказывание «к концам проводника приложено электрическое напряжение», а b - высказывание «по проводнику идет электрический ток». Тогда для высказывания $a \subset b$ получим словесную формулировку «если к концам проводника приложено напряжение, то по нему идет электрический ток».

Истинность импликации можно вычислить, пользуясь следующей таблицей:

a	b	$a \subset b$
1	1	1
1	0	0
0	1	0
0	0	1

Итак, мы познакомились с некоторыми элементами *алгебры логики* — науки об исчислении истинности различных высказываний.

Основателем этой алгебры, называемой часто *булевой*, по праву считается английский математик Дж. Буль (1815—1864 гг.).

Булева алгебра оперирует с логическими переменными, принимающими лишь два значения,— истинно и ложно, постулируя положение о том, что высказывание может быть либо истинным, либо ложным — и третьего не дано. Операции над переменными и функциями алгебры логики, принимающими два значения, хорошо согласуются с операциями над двоичными числами, которыми оперирует ЭВМ. Все это вместе взятое предопределило широкое использование алгебры логики в вычислительной технике.

РОДОСЛОВНАЯ КОМПЬЮТЕРОВ

Теперь совершим краткий экскурс в историю становления и развития средств вычислительной техники.

История возникновения и развития электронных вычислительных машин не насчитывает еще и полувека, однако начало так называемого инструментального счета, т. е. использования различного рода вычислительных средств, уходит в глубины веков. Современное состояние средств вычислительной техники есть результат кропотливых поисков и находок многих ученых и изобретателей предшествующих поколений, стремившихся создать такие механизмы и устройства, которые ускоряли бы счет и получение результатов расчетов.

Начнем с того периода, когда английский ученый, инженер и изобретатель профессор математики Кембриджского университета Ч. Беббидж (1791—1871 гг.) высказал в 1883 году основополагающие идеи относительно возможности создания программно-управляемого автомата. Предложенная им машина с программным управлением, предвосхитившая многие достижения в области вычислительной техники, предназначалась для составления таблиц различных функций. По замыслу Беббиджа, его машина должна была выполнять несколько арифметических операций с запоминанием промежуточных и конечных результатов.

К сожалению, по ряду причин, связанных главным образом с уровнем развития техники того времени, замыслам ученого при его жизни не суждено было воплотиться в действительность. Однако Беббидж высказал множество блестящих идей, которые и в наши дни не утратили своей актуальности.

В 1884—1889 гг. американский инженер Г. Холлерит (1860—1929 гг.) получил несколько патентов на электромеханическую вычислительную машину и табулятор. Свою разработку Холлерит назвал счетно-аналитической машиной. Она ускорила обработку данных переписи населения 1890 года в три раза по сравнению с предыдущей переписью.

Первый проект электронной цифровой вычислительной машины разработал в 1937—1939 гг. в г. Эймс (штат Айова, США) профес-

сор физики и математики Дж. В. Атанасов, болгарин по национальности. Согласно проекту, машина должна была решать системы алгебраических уравнений с 30 неизвестными. Операции должны были производиться над 50-разрядными двоичными числами, то есть содержащими 50 цифр в составе двоичного числа.

Более совершенную вычислительную машину создал в 1944 году профессор Гарвардского университета (США) Х. Айкен. Машина имела название «Марк-1» и была изготовлена фирмой ИВМ. Она оперировала с 23-разрядными числами со скоростью сложения 0,3 с, умножения — 6 с и деления — 11,4 с. Это была первая цифровая автоматическая вычислительная машина на электро-механических элементах (реле).

В начале 1940-х годов советский инженер Н. И. Бессонов сконструировал релейную вычислительную машину РВМ-1. Она содержала 5500 реле и оперировала двоичными числами со скоростью сложения — 0,02 с, умножения — 0,05 с.

Следует однако заметить, что машины Холлерита, Атанасова, Айкена и Бессонова не были прототипами современных вычислительных машин в полном смысле слова. Эра ЭВМ сформировалась с появлением *электронных ламп*, на основе которых началось изготовление быстродействующих вычислительных устройств.

Первая ЭВМ на электронных лампах ЭНИАК (ENIAC — Electronic Numerical Integrator and Calculator, т. е. электронный цифровой интегратор и вычислитель) была сконструирована Д. Эккертом и Д. Моучли (США) в 1946 году. Она содержала 18 000 электронных ламп и около 5000 электро-механических реле. Общая масса всего оборудования этой ЭВМ составляла 30 т! С позиций сегодняшнего дня микроэлектроники можно однако заметить, что машина, более мощная чем ЭНИАК, может быть реализована на одном кристалле интегральной микросхемы.

В 1945 году американский математик Дж. фон Нейман предложил хранить программу вычислений непосредственно в памяти ЭВМ. На основе этой идеи в 1948 году была создана электронная машина ЭДВАК, положившая начало *первому поколению* ЭВМ (1950—1960 гг.). Элементной базой машин первого поколения были *электронные лампы*.

Работы по созданию ЭВМ в Советском Союзе были начаты в 1948 году под руководством С. А. Лебедева. В 1950 году была изготовлена первая советская машина МЭСМ (малая электронно-счетная машина). В 1951—1952 гг. была завершена разработка БЭСМ — быстродействующей электронно-счетной машины.

Коллективами ученых и инженеров, возглавляемыми И. С. Бруком, Ю. И. Базилевским и Б. И. Рамеевым, в 1952—1954 гг. были созданы на электронных лампах вычислительные машины М-2, «Стрела» и «Урал». «Стрела» была первой советской машиной, пущенной в серию.

Начиная с 1955—1956 гг. средства вычислительной техники практически полностью обновлялись с периодичностью в 7—8 лет. Стали говорить о поколениях ЭВМ, их особенностях и характерных признаках.

При переходе от поколения к поколению скорость счета и объем памяти ЭВМ возрастали в несколько раз. Одновременно совершенствовалась элементная и конструкторско-технологическая база вычислительных машин. Машины *второго поколения* создавались на основе *полупроводниковых приборов*. Возрастали возможности ЭВМ, а вместе с ними расширялась и область применения компьютеров.

Существенно изменилось соотношение стоимости ЭВМ и ее программного обеспечения. Так, для машин первого поколения стоимость программного обеспечения составляла 10—15 % от общей стоимости ЭВМ, для второго поколения — 30—40 %, а в третьем — уже 50—60 %. Тенденция удорожания программного обеспечения сохраняется и в последующих поколениях. Некоторые специалисты в области вычислительной техники в шутку говорят, что если так дело пойдет и дальше, то скоро ЭВМ будет продавать в виде... приложения к программному обеспечению!

В 1970-х годах началось серийное производство электронных вычислительных машин третьего поколения, элементной базой которых были интегральные и гибридные микросхемы.

Одна из главных отличительных особенностей машин этого поколения — наличие развитых *операционных систем*. Операционная система представляет собой комплекс программных средств, обеспечивающих автоматизацию процессов подготовки и прохождения через ЭВМ программ вычислений, оптимизацию функционирования технических средств машины и способствующих повышению производительности труда обслуживающего персонала.

Совершенствование конструкций ЭВМ сопровождалось увеличением скорости вычислений. Однако на этом пути встретилось серьезное препятствие. Дело в том, что скорость счета на ЭВМ лимитируется предельной скоростью движения электронов, которая хотя и очень велика, но все же ограничена. Возник вопрос, как выйти из создавшегося тупика?

Давно известно, что все вычислительные операции в ЭВМ выполняются строго *последовательно*, одна за другой. Например, пока не закончено одно арифметическое действие, машина не приступает к следующему. Причина этого проста: результат предыдущей операции служит исходной информацией для следующей.

Таким образом, чтобы преодолеть *физический барьер* на пути увеличения скорости вычислений необходимо было отказаться от привычной схемы последовательного счета и перейти к *параллельным* вычислениям. Математики доказали, что при решении многих сложных задач ряд вычислений можно выполнять *одно-*

временно на разных вычислительных устройствах. Так появились ЭВМ с *параллельно* работающими процессорами. К их числу следует отнести «Эльбрус-1» и «Эльбрус-2» (СССР), «Крей» (США). Это ЭВМ *четвертого поколения*, их элементная база — микросхемы среднего уровня интеграции. С появлением *многопроцессорных* комплексов скорость вычислений и объем памяти существенно увеличились.

С целью более эффективного использования ЭВМ их стали объединять в *сети*. Ресурсы вычислительных машин, объединенных в сеть, являются общими. Связь между такими ЭВМ осуществляется по телефонным проводам. Наиболее длинная сеть ЭВМ создана в нашей стране между вычислительными центрами Академии Наук СССР. Она протянулась от Ленинграда до Владивостока. При создании сверхбольших сетей ЭВМ связь между ними будет осуществляться через спутники.

Большие перспективы открывает разработка ЭВМ пятого поколения. Проект создания таких машин был опубликован в Японии и вызвал оживленное обсуждение во всем мире. Элементная база новых машин — (*большие интегральные схемы*) — будет характеризоваться сверхвысокой степенью интеграции электронных микросхем. В программном обеспечении найдут широкое применение методы «искусственного интеллекта» и *экспертные системы*. Память таких ЭВМ будет содержать наряду с числовой информацией также опыт и знания крупнейших специалистов в различных областях химии, физики, математики, техники, биологии, медицины. С помощью так называемых механизмов *логического вывода* ЭВМ сможет вырабатывать *новые знания* на основе существующих знаний, т. е. автоматически генерировать *новую информацию* в виде изобретений, научных гипотез, а может быть ... и открытий.

Итак, стремительное развитие вычислительной техники явилось результатом научно-технического прогресса. В то же время применение ЭВМ, как катализатор, ускоряет этот прогресс.

ЭВМ НА СТОЛЕ

Среди большого разнообразия вычислительных машин выделяется группа *персональных компьютеров*. Интерес к ним не случаен. Они компактны, занимают мало места, относительно дешевы и в то же время обладают большими вычислительными возможностями.

Персональные компьютеры появились благодаря огромным успехам в области технологии *микроэлектронных схем*. Чтобы нагляднее представить, что сделано конструкторами и технологами в этом направлении, прибегнем к такому сравнению: если бы последние 30 лет самолеты «шли в ногу» с ЭВМ, то межконтинентальный авиалайнер стоил бы сегодня не дороже мопеда, облетал бы земной шар за полчаса, затрачивая на это канистру топлива, и требовал бы лишь мелкого ремонта раз в 50 лет.

Основа такого успеха — переход от сборных электронных схем к *интегральным* микросхемам. Вместо десятков тысяч радиодеталей — одна плоская крошечная коробочка с ножками-выводами; вместо длительного и трудоемкого ручного монтажа — полностью автоматизированное производство; вместо несметного множества паяных соединений и контактов — электрическая схема, созданная прямо *на кристалле* полупроводника. Отсюда и малые габариты, и надежность и низкая стоимость.

Развитие микроэлектроники, повышение уровня интеграции элементов на одном кристалле, создание больших и сверхбольших интегральных схем высокой плотности привело к появлению *микропроцессоров*. Микропроцессор — это одно из основных устройств персональных ЭВМ.

В больших вычислительных машинах к процессору принято относить устройство управления потоком информации, арифметическо-логическое устройство, а иногда и запоминающее устройство. В микропроцессоре все эти устройства размещаются на одном кристалле площадью около 1 см^2 (рис. 6).

Каковы же возможности персонального компьютера? Он способен выполнять сотни тысяч команд в секунду. Объем его оперативной памяти — многие тысячи машинных слов. На экране дисплея можно видеть не только текст программы и результаты вычислений, но и различные графические изображения — картинки, графики, рисунки. Если в компьютере имеется цветной дисплей, то для вывода на экран графической информации используются несколько цветов: черный, белый, красный, зеленый, синий и др., причем для каждого цвета допускается несколько градаций яркости, — палитра богатая! Большие возможности открывает использование *гибких и жестких магнитных дисков*. На них могут быть записаны многочисленные программы для вычислений, результаты расчетов, пояснительный текст и т. п.

Создатели персональных компьютеров разработали для них несколько языков программирования. Некоторые из этих языков рассчитаны на неподготовленного пользователя. Например, «Робик» — это простейший язык, содержащий только русские слова и цифры. Программа вычислений на «Робике» читается так же легко и естественно, как обычный текст.

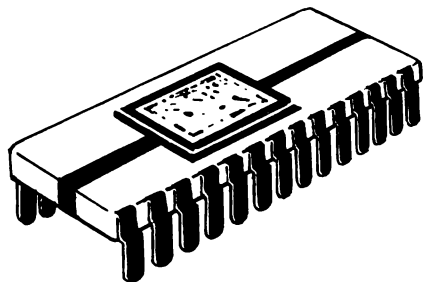


Рис. 6. Микропроцессор

Для персональных компьютеров разработано много *игровых* программ. Если, например, ввести программу «Шахматист», то на экране дисплея тотчас появится знакомое клетчатое поле с фигурами и часы. Сделаем первый ход: нажмем на пульте клавиши «е» и «4». Белая пешка на экране продвинется вперед на две клетки. А вот и ответный ход компьютера — навстречу переместилась черная пешка. Вряд ли вы отличите стиль игры электронного партнера от человеческого, хотя машина с помощью заложенной в нее программы способна только перебирать варианты на несколько ходов вперед, выискивая среди них наиболее выгодный. У машины нет ни опыта, ни фантазии и все же... она скорее всего обыграет вас, если вы не имеете хотя бы первого спортивного разряда!

Но дело не только в шахматах. Многие *учебные* программы для персональных компьютеров облечены в увлекательную игровую форму. «Обучайся, играя!» — таков один из принципов обучения с помощью ЭВМ.

ХИМИЯ ДЛЯ КОМПЬЮТЕРОВ

Современная вычислительная машина — это продукт творческого труда ученых, инженеров и рабочих различных специальностей. В создании ЭВМ принимают участие математики, специалисты по электронике, механики, химики. Роль химии в производстве компьютеров достаточно велика. Они участвуют в получении материалов для элементной базы ЭВМ, разрабатывают и реализуют технологию нанесения покрытий на изделия микроэлектроники. Большое значение в производстве компьютеров принадлежит электрохимическим процессам. Все это образует традиционную сферу применения химии и химической технологии в области вычислительной техники. Однако есть и нетрадиционные сферы применения.

Глядя на окружающий нас мир, мы редко задумываемся о механизме зрения. Между тем оценить красоту природы или прочесть книгу мы можем лишь потому, что сетчатка нашего глаза преобразует свет в электрические сигналы. Основную роль в преобразовании зрительной информации в электрические импульсы играет родопсин — особый белок, входящий в состав сетчатки.

Очень похожий на родопсин белок был обнаружен у бактерий, живущих в соленых озерах. Ученые установили, что с помощью этого белка бактерии превращают свет солнца в необходимую для жизни химическую энергию.

Ученые института биофизики АН СССР сумели выделить этот белок в чистом виде. Его изучение показало, что сходство белка бактерий и родопсина почти полное. Как и родопсин, белок разлагается в темноте. Разлагаясь — меняет цвет, а восстанавли-

ваясь — возвращает свою исходную окраску. За сходство с родопсином новый белок так и назвали — бактериородопсином.

В дальнейшем химики научились получать из обезвоженного бактериородопсина тонкие пленки, напоминающие светочувствительный слой фотобумаги. Оказалось, что изображение на бактериородопсине можно стирать, записывать заново, снова стирать и т. д. Качество изображения, зафиксированного на бактериородопсине, во много раз выше, чем на обычной фотобумаге: каждая его частичка в пленке в сотни раз мельче, чем частицы фотоэмульсии.

Изображение — это информация, и в принципе не важно, как ее записывать — на ферритовые сердечники, магнитные диски или фотопластинки. Главное, чтобы плотность записи была как можно выше, а в этом с бактериородопсином не могут соперничать магнитные носители современных ЭВМ. Важно отметить, что запись информации на диск с белковой фотоэмульсией можно осуществлять почти столь же быстро, как и на магнитный. Информация эта надежно хранится в течение нескольких месяцев. Так исследования физико-химических и оптических свойств бактериородопсина помогли отыскать новый перспективный материал для памяти вычислительных машин.

Второй главный узел ЭВМ — процессор, который непосредственно оперирует числами. Могут ли в нем работать органические «детали»?

Недавно был синтезирован ряд белковых соединений, способных аналогично нервным клеткам (нейронам) передавать импульсы возбуждения. Представим себе, что на небольшой подложке нанесены миллионы микроскопически малых белковых частиц, словно транзисторы в микросхеме. Так выглядит «живой» процессор, созданный в институте биофизики АН СССР.

Если возбудить одну из частичек на подложке, то сигнал возбуждения распространится на соседние. Управление у белкового процессора *химическое*: воздействуя на него различными веществами, можно регулировать скорость и форму распространения сигналов.

В настоящее время обнаружено более 50 химических соединений, на основе которых могут быть построены процессоры. Каждая частица белка на подложке процессора имеет очень маленький диаметр и занимает площадь, в тысячу раз меньшую, чем транзистор на подложке интегральной микросхемы.

Биологическим компьютерам сегодня еще необходима оптика, нельзя обойтись и без электроники. Но уже можно говорить о новом поколении вычислительных устройств — гибридах электронной техники и биологии, живого и неживого. В основе работы таких устройств большую роль играют физико-химические процессы. И не будем забывать: работы по созданию «живых» вычислительных машин сегодня только начинаются.

АЛГОРИТМЫ ВОКРУГ НАС**ШАГ ЗА ШАГОМ**

Выдающийся узбекский ученый Мухаммед бен Муса ал-Хорезми, живший в IX веке, был математиком, астрономом и географом. Его «Арифметический трактат» в XII веке был переведен с арабского языка на латинский и появился в средневековой Европе.

Четыре арифметических действия стали доступны не только ученым Европы, но и обыкновенным людям. А сами правила выполнения арифметических действий долго еще связывали с именем ал-Хорезми (по латыни Algorithmi), называл их *алгоритмами*.

Термин «алгоритм», получивший широкое распространение, постепенно расширил свое первоначальное значение. Ныне алгоритмом называют *любую точно определенную последовательность действий* (необязательно математических), необходимых для выполнения какой-либо работы или решения задачи.

С алгоритмами приходится постоянно встречаться в повседневной жизни. Например, переходя в городе проезжую улицу, мы руководствуемся правилом: «Сначала посмотри налево, а, дойдя до середины улицы, — направо».

Думается, среди читателей этой книги немало фотолюбителей. Они с удовольствием делают снимки, проявляют фотопленки, печатают фотографии. При этом, разумеется, никто и не задумывается об алгоритмах. А между тем без алгоритмов здесь никак не обойтись.

Прочтем, например, инструкцию на пакетике проявителя, который мы купили в магазине. Она гласит:

«Содержимое малого пакета растворить в 350 мл кипяченой воды при температуре 40—50 °С.

Затем растворить содержимое большого пакета.

Объем раствора довести до 500 мл.

Раствор профильтровать.

Проявлять при температуре 18—20 °С.

Время проявления фотобумаг — 2—3 мин.»

Ну чем не алгоритм! Строго выполнив эту последовательность операций (рис. 7) шаг за шагом, мы получим отпечаток на фотобумаге. Можно с уверенностью сказать, что отступление от ал-

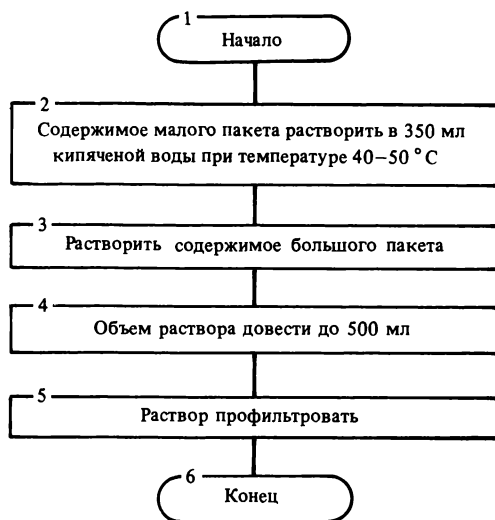


Рис. 7. Блок-схема алгоритма приготовления раствора

горитма может здесь привести лишь к ухудшению качества отпечатка.

Часто алгоритмы сопровождаются различными пояснениями, *комментариями*. В комментариях может разъясняться смысл алгоритма в целом и отдельных его частей.

Например, комментарием к алгоритму можно считать следующую надпись на пакетике с проявителем:

«В свежеприготовленном растворе можно проявить до 60 листов фотобумаги размером 9×12 см.»

Итак, мы познакомились с простейшими алгоритмами, операции в которых выполняются строго *последовательно*, одна за другой, приводя в конечном счете к желаемому результату. Однако большинство алгоритмов имеет более сложную структуру.

ПОСЛЕДОВАТЕЛЬНОСТИ ОПЕРАЦИЙ

Вы решили позвонить своему знакомому с вашего домашнего телефона. Попробуем составить алгоритм набора нужного номера.

Прежде всего, очевидно, Вы снимаете телефонную трубку. Услышав непрерывный гудок, набираете знакомый номер. После этого Вы сталкиваетесь с одной из двух ситуаций. В первом случае Вам повезло — Ваш знакомый в это время оказался у телефона и снял трубку, разговор состоялся. Во втором случае Вас ждала неудача: либо телефон был занят, либо абонента не было у телефона. Конечно, в этом случае Вы положите трубку на телефонный аппарат.

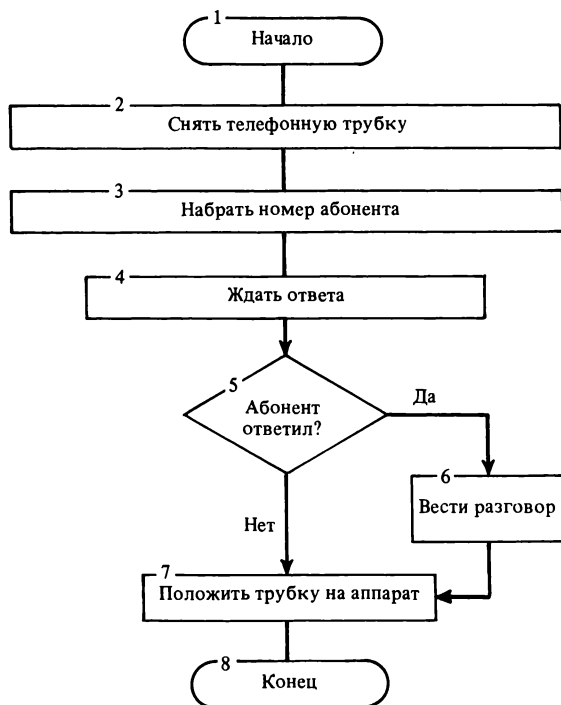


Рис. 8. Блок-схема алгоритма разветвляющейся структуры

Схема рассмотренного алгоритма приведена на рис. 8.

Как Вы уже заметили, алгоритм имеет *разветвляющуюся* структуру. В зависимости от складывающихся ситуаций при наборе телефонного номера приходится выполнять одну из возможных операций.

Бывают алгоритмы и с *циклической* структурой. В их составе имеются многократно повторяющиеся последовательности операций — так называемые *циклы* операций, или просто — циклы.

Например, в процессе проведения опытов химии часто пользуются аналитическими весами. Рассмотрим алгоритм взятия навески, т. е. отвешивания заданного количества вещества.

Прежде всего необходимо подготовить весы, взять вещество, разновес и пинцет. Затем надо положить на одну из чашек весов гирьки, соответствующие заданной массе вещества. После этого следует взять ложечку и насыпать немного вещества на другую чашку весов. Если масса насыпанного вещества строго соответствует заданной и весы оказались в равновесии, то взвешивание полагается закончить. Если же масса насыпанного вещества еще мала, то следует его добавить. Эту операцию необходимо продол-

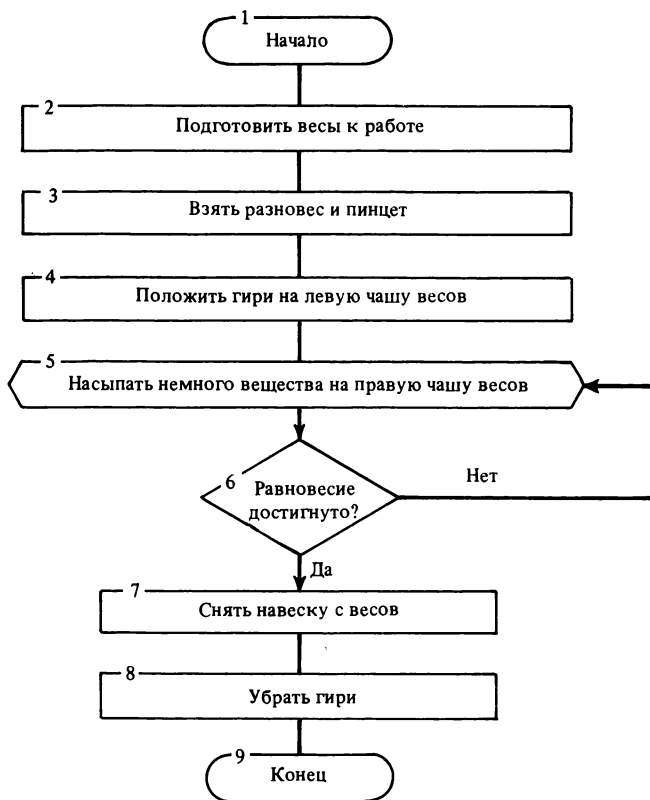


Рис. 9. Блок-схема алгоритма циклического процесса

жать циклом до тех пор, пока не наступит равновесие. Алгоритм взвешивания показан на рис. 9.

Алгоритмы сложных вычислений имеют, как правило, сложные разветвляющиеся и циклические структуры. Именно благодаря циклам удастся в краткой форме записывать схемы решения многих трудоемких задач химии и химической технологии.

Всякий алгоритм имеет следующие свойства:

1. **Определенность.** Каждое правило алгоритма должно быть четким и однозначным, чтобы при одних и тех же начальных данных получался один и тот же результат.

2. **Результативность.** Решение задачи с помощью алгоритма является *конечным* процессом, который завершается получением определенного результата.

3. **Дискретность.** Процесс преобразования исходных данных в результате вычислений осуществляется дискретно, так что только после окончания одного действия можно приступить к выполнению следующего.

4. Массовость. Алгоритм решения задачи разрабатывается в общем виде так, чтобы его можно было применять для *класса* задач, различающихся лишь исходными данными. При этом исходные данные могут выбираться из области, называемой *областью применимости алгоритма*.

Когда говорят о решении задачи на ЭВМ, то часто одновременно употребляют термины *алгоритм* и *программа*. Следует отметить, что эти понятия не очень четко разграничены. Обычно программа — это окончательный вариант алгоритма решения задачи, записанный на каком-нибудь конкретном *алгоритмическом языке* программирования.

СХЕМА АЛГОРИТМА

Мы уже познакомились с двумя различными способами представления алгоритмов — словесным описанием и структурной схемой (блок-схемой).

Словесное описание раскрывает смысл каждой отдельной операции алгоритма, но оно не обладает той наглядностью, которую имеет блок-схема. На схеме мы видим движение потоков информации и связи между всеми операциями.

Чтобы обеспечить *унификацию* изображений блок-схем алгоритмов, изображения отдельных элементов стандартизованы (табл. 1).

Определенный этап вычислений на схеме алгоритма обозначается прямоугольником, внутри которого записывается содержание этого этапа.

Проверка условия изображается *ромбом*, внутри которого записывается это условие. В результате проверки условия выбирается один из двух возможных путей вычислительного процесса. Если условие выполняется, то следующим реализуется этап по стрелке «Да». Если условие не выполняется, то осуществляется переход по стрелке «Нет».

Начало и конец вычислительного процесса изображаются фигурой, напоминающей овал. Внутри нее записываются слова «Начало» или «Конец».

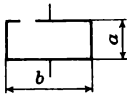
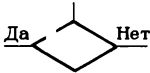
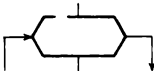





Ввод исходных данных и вывод результатов вычислений изображаются в соответствии с табл. 1. Внутри пишется слово «Ввод» или «Печать», а также перечисляются переменные, подлежащие вводу и выводу.

Комментарий располагается рядом с блоком, к которому он относится.

Составление блок-схем алгоритмов — работа творческая. Иногда приходится много раз переделывать исходный вариант.

В заключение составим в качестве примера блок-схему вычисления среднего арифметического значения нескольких чисел. Такая задача встает, например,

Т а б л и ц а 1. Наиболее часто применяемые обозначения в блок-схемах алгоритмов

Символ, название	Обозначение	Пояснение
Процесс		Вычислительные действия, или последовательность вычисл. действий
Решение		Проверка условий
Модификация		Начало цикла
Предопределенный процесс		Вычисления по подпрограмме, стандартной подпрограмме
Документ		Вывод, печать результатов на бумаге
Перфокарта		Ввод данных с перфокарт
Соединитель		Разрыв линий потока информации
Пуск, останов		Начало, конец, останов, вход и выход в подпрограммах
Комментарий	Текст комментария	Пояснения, содержание подпрограмм, формулы

при нахождении среднего состава природного сырья, используемого для проведения некоторых химико-технологических процессов.

Допустим, что на стекольный завод поступило сырье с различным содержанием диоксида кремния SiO_2 . Было взято n проб сырья и определено содержание SiO_2 в каждой из них. Эти значения обозначим x_i . Расчетная формула для нахождения среднего арифметического значения $x_{\text{ср}}$:

$$x_{\text{ср}} = \frac{1}{n} (x_1 + x_2 + \dots + x_n).$$

Ее удобно переписать в следующем виде:

$$x_{\text{ср}} = \frac{1}{n} \sum_{i=1}^n x_i.$$

Не правда ли, запись стала короче? Знак \sum избавил нас от необходимости перечислять все слагаемые суммы. Это тем более полезно, когда неизвестно заранее общее число слагаемых. Такую символическую запись суммирования мы будем применять в дальнейшем.

В приведенной формуле нижний индекс i в процессе суммирования слагаемых x_i принимает последовательно значения от 1 до n .

Сначала приведем словесное описание алгоритма:

1. Начало.
2. Начальное значение суммы слагаемых \sum положим равным нулю: $\sum = 0$.
3. Введем в ЭВМ число n , указывающее количество слагаемых.
4. Начальное значение числа введенных слагаемых R положим равным нулю.
5. Введем очередное слагаемое.
6. Проверим условие $R = n$, т. е. ввели ли мы все заданные слагаемые? Если условие выполнено, то результат суммирования \sum делим на число слагаемых n и выдаем на печать значение среднего арифметического $x_{\text{ср}}$ (пункт 7). Иначе переходим к пункту 5.
7. Прекращаем реализацию алгоритма (конец алгоритма).

Блок-схема этого алгоритма показана на рис. 10. Она наглядно иллюстрирует процесс вычислений. На ней видны не только все этапы расчетов, но и циклический процесс перехода от пункта 6 к пункту 5.

Вы наверняка заметили номера, которые стоят на отдельных блоках алгоритма. Они помогают, особенно в сложных и громоздких алгоритмах, разбираться в последовательности выполнения операций.

Итак, теперь мы умеем не только составлять алгоритмы, но и изображать их.

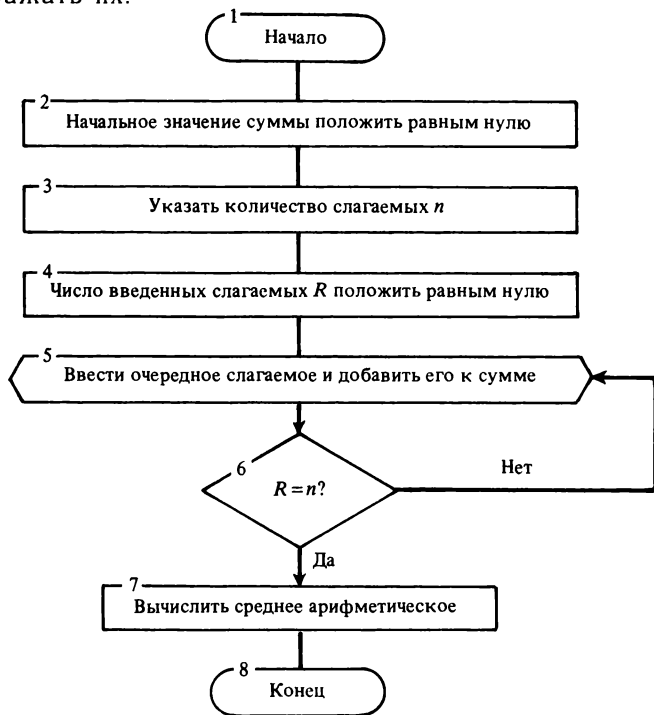


Рис. 10. Блок-схема алгоритма вычисления среднего арифметического значения

САМАЯ МАЛЕНЬКАЯ ЭВМ**ПРОГРАММИРУЕМЫЙ МИКРОКАЛЬКУЛЯТОР (ПМК)**

В наши дни программируемый микрокалькулятор доступен практически каждому. Кто не видел эту маленькую изящную коробочку с красивыми кнопками и маленьким экраном! Микрокалькулятор может быть очень полезен школьнику и студенту, инженеру и ученому.

ПМК — это самая малая из всех существующих ЭВМ. Хотя возможности его меньше, чем у больших ЭВМ и персональных компьютеров, основное достоинство вычислительных машин — возможность работать по введенной *программе* — есть и у него. А так как микрокалькулятор портативен и, главное, недорог, то *доступность* его является даже преимуществом перед стационарными ЭВМ. Ни с чем не сравнить ощущение комфорта при работе на своей собственной вычислительной машине, сидеть за которой можно когда угодно и сколько угодно, постоянно экспериментируя и открывая для себя все новые ее тайны.

В нашей стране выпускается несколько типов ПМК. Один из самых распространенных среди них — это «Электроника БЗ-34». Поэтому он и был выбран нами в качестве эталона для рассказа о ПМК. Естественно, что все сказанное о нем в значительной мере относится и к другим разновидностям программируемых микрокалькуляторов.

Химики-технологи давно почувствовали интерес к ПМК и широко применяют их в своей исследовательской работе.

Итак, микрокалькулятор перед нами. Тридцать разноцветных кнопок-клавиш выстроились стройными рядами по пять в линию. На каждой кнопке — свое обозначение. Имеются также надписи рядом с кнопками. Некоторые обозначения понятны нам сразу. Это, например, цифры, знаки арифметических операций, названия элементарных функций.

Над рядами кнопок расположены два выключателя. Левый служит для включения ПМК в работу, а правый устанавливает режим работы с тригонометрическими функциями: «Р» означает радианы, «Г» — градусы.

Передвинем движок левого переключателя вправо. На индикаторе зеленым светом загорится цифра 0. Микрокалькулятор готов к работе.

Заметим, что сигналы от нажатия кнопок по-разному воспринимаются ПМК в зависимости от того, в каком режиме он работает. Таких режима два. В первом выполняются *команды*, отдаваемые непосредственно нажатием на кнопки, во втором решение задачи осуществляется *автоматически* по программе, введенной в память ПМК.

Включая микрокалькулятор, мы автоматически устанавливаем первый (непрограммный) режим его работы.

Работу на ПМК начнем с освоения его клавиатуры. Будем нажимать на кнопки с изображением цифр. На экране загораются те же цифры, что и обозначенные на кнопках: 1, 2, ..., 7, 8. Нажимаем на кнопку с цифрой 9. Девятая цифра на экране не появляется. Это значит, что ПМК рассчитан на работу не более чем с восьмиразрядными числами, и, если восемь цифр числа уже введены, дальнейший ввод невозможен, нажатие на цифровые кнопки больше не воспринимается калькулятором.

Чтобы занести в память ПМК дробное число, нажимаем после цифр целой части кнопку с изображением «,» и затем вводим дробную часть. Ради тренировки введите в калькулятор число $\pi = 3,1415926$.

Очевидно, возможности такой записи чисел ограничены. Нельзя, например, ввести в микрокалькулятор значение заряда электрона, выраженное в кулонах: $1,6021892 \cdot 10^{-19}$. Аналогичная проблема возникает и при вводе числа Авогадро ($6,02 \cdot 10^{23}$), которое характеризует число молекул в одном моле вещества.

Подобные затруднения легко преодолим, если записывать числа в так называемой *экспоненциальной* форме $a \cdot 10^b$, где число a начинается с ненулевой цифры, после которой стоит запятая. При такой записи число a называется *мантиссой*, а b — *порядком*.

Чтобы ввести в ПМК число, записанное в экспоненциальной форме, надо сначала набрать мантиссу. Если число отрицательное, вслед за этим следует нажать клавишу «/—/». Затем, чтобы ввести порядок числа, необходимо нажать кнопку «ВП» (Ввод Порядка). В правой части экрана тотчас появятся два нуля. Теперь нажатие на любую цифровую кнопку вызывает появление соответствующей цифры в правом углу экрана. Если порядок числа отрицательный, нужно нажать на кнопку «/—/».

Введем в ПМК значение заряда электрона. Для этого нажмем на кнопки «1», «,», «6», «0», «2», «1», «8», «9», «2», «ВП», «1», «9», «/—/». На экране увидим число $1,6021892^{-19}$.

Экспоненциальная форма записи чисел расширяет возможности ПМК. Диапазон чисел, доступных микрокалькулятору огромен. Наибольшее из них $9,9999999 \cdot 10^{99}$, а наименьшее по абсолютной величине равно 10^{-99} .

Если вы ошиблись при вводе числа — не беда. Нажмите клавишу «Сх». Число исчезнет с экрана. Заметим, что любое число, введенное в калькулятор с клавиатуры, помещается в ячейку памяти, называемую *регистром X*. Содержимое этого регистра отображается на индикаторе. Действие кнопки «Сх» заключается в том, что освобождается регистр X от записанного в нем числа (отсюда и ее название: Стереть X). На индикаторе ПМК отражается лишь результат стирания, т. е. цифра 0. После этого ввод можно повторить или ввести новое число. Теперь перейдем к вычислениям

РАСЧЕТЫ НА ПМК

Записывая любое арифметическое выражение, мы всегда имеем в виду порядок выполнения входящих в него арифметических операций. Сначала — действия в скобках, потом — умножение и деление, далее — сложение и вычитание. К примеру, вычисляя значение выражения $a \times b + c \times d$, мы сначала выполняем два умножения, а затем складываем полученные произведения. Числа, над которыми выполняются операции, называются *операндами*. Арифметическая операция над двумя числами именуется *двуместной*.

Для размещения операндов двуместной операции требуются два различных регистра памяти. Регистр X, куда вводятся числа с клавиатуры ПМК, — это лишь один из четырех взаимосвязанных регистров, обозначаемых буквами X, Y, Z, T, и образующих вместе так называемый *стек*. В регистрах X и Y должны быть числа, участвующие в любой двуместной операции.

Делается это так. Сначала один из операндов набирается на клавиатуре и тем самым заносится в регистр X. Потом нажимается клавиша «↑». Содержимое регистра X при этом копируется в регистр Y. Если теперь набрать на клавиатуре новое число, то оно запишется опять-таки в регистр X, стирая его содержимое. После этого можно нажимать клавишу требуемой двуместной операции. Результат всякой выполненной операции поступает в регистр X, поэтому он и виден на индикаторе.

В дальнейшем, для краткости, мы иногда будем писать не «регистр X», «регистр Y, а RX, RY. Такую же запись будем использовать и в отношении регистров Z и T.

Порядок расположения операндов в регистрах X и Y безразличен, если числа складываются или перемножаются. Расположение операндов при возведении числа x в степень y задается самой записью операции: в RX — основание, в RY — показатель степени. А вот на случай вычитания и деления порядок расположения операндов следует запомнить.

При вычитании: в регистре X — вычитаемое, а в регистре Y — уменьшаемое. *При делении:* в регистре X — делитель, в регистре Y — делимое.

При вычислениях на микрокалькуляторах получила широкое распространение *обратная бесскобочная*, или *польская*, запись, предложенная польским математиком Я. Лукасевичем. Все операции при такой записи имеют *одинаковый* приоритет и знаки их ставятся не между операндами, а *после* них, т. е. например, не $a \times b$, а $ab \times$. Это избавляет от необходимости использовать скобки и проверять приоритеты.

Рассмотрим выражение $a \times (b + c \times d)$. В польской записи оно имеет вид: $cd \times b + a \times$. Все здесь просто и ясно.

В МИРЕ АТОМОВ

В химических превращениях постоянно участвуют атомы и молекулы. Как атомы, так и молекулы очень малы, а число их весьма велико. Поэтому при расчетах, связанных с числом атомов, обычно возникают очень большие и очень малые числа. Оперировать такими числами особенно удобно, если воспользоваться микрокалькулятором.

Мы уже упоминали о числе Авогадро, которое равно $6,02 \cdot 10^{23}$ и характеризует число молекул в одном моле вещества. Атомная масса меди равна 64. Следовательно, 64 г меди содержат $6,02 \cdot 10^{23}$ атомов. Отсюда можно вычислить массу одного атома:

$$64 : (6,02 \cdot 10^{23}) = 10,63 \cdot 10^{-23} \text{ г.}$$

Ее не надо, разумеется, путать с атомной массой, выраженной в кислородных единицах.

Плотность железа составляет $7,9 \text{ г} \cdot \text{см}^{-3}$, а его атомная масса равна 56. Отсюда нетрудно найти объем 56 г железа:

$$56 : 7,9 = 7,1 \text{ см}^3.$$

Очевидно, в этом объеме опять-таки содержатся $6,02 \cdot 10^{23}$ атомов. Тогда каждый атом железа занимает объем

$$7,1 : (6,02 \cdot 10^{23}) = 1,12 \cdot 10^{-23} \text{ см}^3.$$

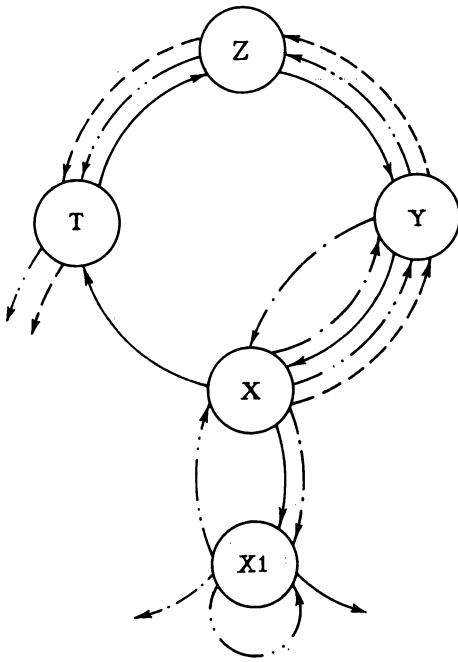
Вот с какими числами легко оперирует наш микрокалькулятор!

ПУТЕШЕСТВИЕ ПО СТЕКУ

Рассматривая арифметические операции, выполняемые на микрокалькуляторе, мы пока упоминали только регистры памяти X и Y. Между тем при вводе чисел для выполнения любой операции и при ее совершении происходит движение чисел по *всему* стеку. Это движение затрагивает еще один регистр, примыкающий к стеку и обозначаемый X1. На рис. 11 показаны потоки информации в стеке.

После ввода в регистр X некоторого числа мы нажимаем кнопку «↑». При этом содержимое регистра X копируется в реги-

Рис. 11. Поток информации в стеке



старе Y. Прeжнее содержимое RY смещается в RZ, содержимое RZ — в RT, а старое содержимое RT при этом исчезает.

Когда над числами, находящимися в регистрах X и Y, совершается какая-либо арифметическая операция и ее результат записывается в регистр X, прежнее содержимое регистра X пересылается в регистр X1, содержимое регистра Z смещается в RY, содержимое RT — в RZ, оставляя после себя в регистре T свою копию. То, что было в RY, при этом пропадает.

Когда содержимое регистра X используется для вычисления какой-либо функции, результат записывается в RX. Прeжнее его содержимое поступает в регистр X1. Содержимое других регистров не меняется.

Названия всех *функций*, вычисляемых на микрокалькуляторе, обычно написаны над клавишами. Чтобы совершить любую «надклавишную» операцию, нужно предварительно нажать кнопку «F», а уже потом соответствующую клавишу. Заметим также, что вычисление любой функции называется *одноместной* операцией, поскольку требует для своего выполнения только одного исходного числа.

Если число, находящееся в регистре X, не введено туда с клавиатуры, а является результатом какой-либо операции, то ввод нового числа в RX смещает его содержимое в RY, содержимое RY — в RZ, содержимое RZ — в RT; а содержимое RT стирается.

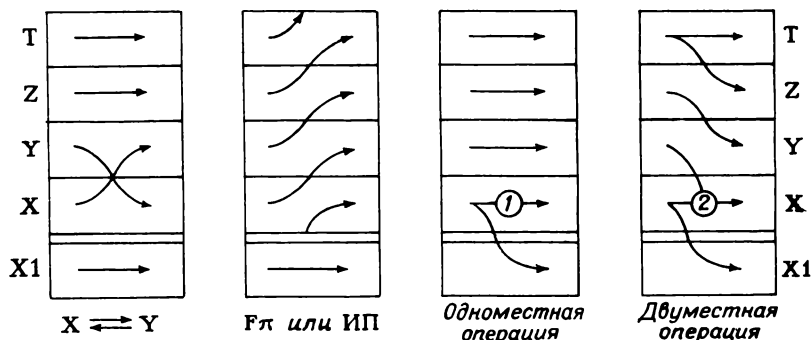


Рис. 12. Движение чисел по стеку

Однако это еще не все возможные перемещения чисел по стеку. Допустим, необходимо использовать содержимое регистра X1. Нажмем для этого клавиши «F» и «Vx» (Вверх). Число из регистра X1 переместится в RX, а числа, находившиеся до этого в регистрах стека, сместятся точно так же, как при вводе нового числа.

Можно заставить числа двигаться по стеку и в обратном направлении, нажав клавиши «F» и «O». На клавиатуре ПМК этот символ образован двумя полукруглыми стрелками, замкнутыми одна на другую. Тогда число из RT перейдет в RZ, из RZ — в RY, из RY — в RX, а из RX — в RT и RX1.

Можно переставить числа, находящиеся в RX и RY, нажав клавишу «XY». Содержимое RX при этом перейдет не только в RY, но и в RX1.

Диаграммы, иллюстрирующие движение чисел по стеку, приведена на рис. 12.

Кроме стека в микрокалькуляторе есть еще одна область для хранения чисел. Это — *адресуемые регистры*. Их четырнадцать. Обозначаются они так: R0, R1, ..., R9, RA, RB, RC, и RD. В эти регистры можно переносить и хранить там содержимое регистра X. Делается это с помощью двух клавиш: «П» (Память) и «N» (где N — цифра или буква, т. е. «имя» регистра, в который пересылается число из регистра X).

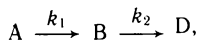
Извлекается информация из нужного регистра нажатием опять-таки двух клавиш: «ИП» (Из Памяти) и «N». При переписи информации из регистра X в адресуемый регистр содержимое стека не меняется, а при записи из адресуемого регистра в RX информация движется так, как при вводе числа с клавиатуры.

СКОРОСТЬ ХИМИЧЕСКОЙ РЕАКЦИИ

Скоростью химической реакции называется величина, характеризующая массу вещества, образующегося (или исчезающего)

в ходе этой реакции в единицу времени в единице объема реакционной массы.

Пусть, например, в колбе протекает последовательная двухстадийная реакция



где A — исходное вещество; B — промежуточный компонент; D — конечный продукт реакции; k_1 и k_2 — константы скоростей первой и второй стадии реакции.

Значения k_1 и k_2 зависят от температуры по закону Аррениуса:

$$k_1 = k_{01} e^{-\frac{E_1}{RT}}; \quad k_2 = k_{02} e^{-\frac{E_2}{RT}},$$

Здесь k_{01} , k_{02} — некоторые коэффициенты, называемые предэкспоненциальными множителями; $e = 2,71828$ — основание натуральных логарифмов; E_1 , E_2 — энергии активации первой и второй стадий химической реакции; $R = 8,31434$ Дж/(К·моль) — универсальная газовая постоянная; T — температура, при которой проводится процесс, К.

Скорость реакции по компоненту B , которую мы обозначим через r_B , можно рассчитать по формуле

$$r_B = k_1 C_A - k_2 C_B,$$

где C_A и C_B — концентрации веществ в зоне реакции.

Расчет r_B можно выполнить на микрокалькуляторе, если заданы значения всех входящих в формулу величин. Последовательность операций следующая:

1. Ввести с клавиатуры значение E_1 .
2. Ввести значение R . При этом значение E_1 поступит в RY .
3. Нажать клавишу «÷». Произойдет деление E_1 на R .
4. Ввести значение температуры T .
5. Нажать клавишу «÷».
6. Нажать клавишу «/—/». Произойдет изменение знака.
7. Нажать клавиши «F» и «e^x».
8. Ввести значение концентрации C_A .
9. Нажать клавишу «×».
10. Ввести значение k_{01} .
11. Нажать клавишу «×». В RX поступит первый член расчетной формулы.
12. Нажать клавиши «П» и «1». Результат поступит в RI .
13. Ввести с клавиатуры значение энергии активации E_2 .
14. Ввести значение R . При этом E_2 поступит в RY .
15. Нажать клавишу «÷». Произойдет деление E_2 на R .
16. Ввести значение температуры T .
17. Нажать клавишу «÷».
18. Нажать клавишу «/—/». Произойдет изменение знака.
19. Нажать клавиши «F» и «e^x».
20. Ввести значение концентрации C_B .
21. Нажать клавишу «×».

22. Ввести значение k_{02} .
23. Нажать клавишу «X». В RX поступит второй член расчетной формулы.
24. Нажать клавиши «ИП», «1».
25. Нажать клавиши «F», « \odot ».
26. Нажать клавишу «/—/».
27. Прочсть результат вычислений на индикаторе ПМК.

Последовательность вычислений получилась достаточно длинной. Если нужно провести расчеты при других концентрациях и температурах, то всю работу с начала до конца придется повторить несколько раз. В этом заключается недостаток работы на ПМК — в *непосредственном* режиме. *Программный* режим имеет то преимущество, что многократно повторяющаяся последовательность операций оформляется в виде программы и вносится в память ПМК. Для выполнения очередного варианта расчетов достаточно ввести лишь новые исходные данные, а затем реализовать все операции, включенные в программу. При этом экономится время и исключаются возможные ошибки.

БЫСТРО ЛИ ДВИЖУТСЯ МОЛЕКУЛЫ!

ПМК обладает замечательным свойством. Он может решать задачи с помощью введенной в него программы. Отметим, во-первых, что вычисления по программе осуществляются значительно быстрее, чем вручную. Во-вторых, с помощью одной и той же программы можно решить множество однотипных задач, используя различные исходные данные.

Для ввода программы в ПМК его переводят в *режим программирования*. Это делают путем последовательного нажатия клавиш «F» и «ПРГ». Заметим, что клавиша «F» самостоятельной роли не играет, а позволяет лишь выбирать функции, указанные *над* другими клавишами. Для краткости будем в дальнейшем вместо «F», «ПРГ» писать просто ФПРГ. Аналогичные сокращенные записи будем применять также к наборам, начинающимся со «служебных» клавиш «K», «П» и «ИП».

Чтобы после ввода программы перейти в режим вычислений, следует нажать клавиши FАВТ.

Попробуем составить простейшую программу для умножения двух чисел a и b . Перед началом вычислений число a будем записывать в R1, а число b — в RX.

Итак, включим микрокалькулятор и нажмем клавиши ФПРГ. В правом углу экрана появятся цифры 00. ПМК готов к приему команд.

Программа для решения нашей задачи показана в табл. 2. Она написана в трех колонках. В первой колонке пишется *адрес* команды, во второй — сама *команда* (т. е. указаны клавиши, нажимаемые при ее вводе), в третьей — *код* команды. Стоящие ря-

Т а б л и ц а 2. Программа для вычисления произведения двух чисел

Адрес	Команда	Код
00	ИП1	61
01	×	12
02	С/П	50
03	БП	51
04	00	00

дом обозначения во второй и третьей колонках по существу выражают одно и то же, но с той лишь разницей, что во второй колонке стоит *символ*, встречающийся на клавиатуре ПМК, а в третьей колонке — своеобразный *перевод* того же символа на язык, «понятный» микрокалькулятору.

Если сравнить программу с последовательностью клавиш, которые мы нажимаем, решая задачу вручную, то нетрудно убедиться, что программа почти полностью повторяет предыдущий набор клавиш.

Команда С/П (Стоп/Пуск) используется в нашей программе для прекращения процесса вычислений. После этого можно прочесть результат на экране ПМК. Команда С/П используется также для *начала счета* по программе.

Команда БПпт (в нашем случае — это БП00), где *пт* — любое двузначное число от 00 до 97, читается: Безусловный Переход на адрес *пт*. Она прерывает последовательное выполнение команд, записанных в программе. После нее выполняется команда, записанная по адресу *пт*.

В нашем случае команда БП00 введена для того, чтобы после одного расчета автоматически подготовиться к началу другого.

Правильность работы любой программы обычно проверяют, решая какой-нибудь простой *тестовый* пример. Мы не можем отказать себе в удовольствии вычислить на ПМК произведение 2×2 . Для этого сначала переведем микрокалькулятор в режим вычислений, нажав клавиши FАВТ. Теперь направим число 2 в регистр R1. С этой целью нажмем клавиши 2 и П1. Заметим, что при этом число 2 попадет также и в регистр R1. Нажатием клавиши В/О (Возврат, Очистка) мы передаем управление команде, записанной по нулевому адресу, т. е. в самое начало программы. Итак, все готово к вычислениям по программе. Нажмем клавишу С/П. На экране появится число 4. Это значит, что программа была набрана правильно. Теперь с ее помощью мы можем вычислять произведения самых различных чисел.

Вы уже, наверное, заметили, что команда БПпт размещается на *двух* строчках программы, а именно: в первой из них находится БП, во второй — адрес команды, которой передается управление.

Т а б л и ц а 3. Последовательность появления цифр на экране ПМК

Команда		Изображение на экране		
				00
ИП1	61			01
×	12	61		02
С/П	50	12	61	03
БП	51	50	12	04
00	00	51	50	05

Для тренировки еще раз введем программу в ПМК и внимательно проследим за экраном. После нажатия клавиш ФПРГ в правой части экрана зажгутся цифры 00. Дальнейшая последовательность появляющихся цифр представлена в табл. 3.

Первые две нулевые цифры означают не только готовность к вводу программы, то также и то, что первая из введенных нами команд записывается по адресу 00. После нажатия последующих клавиш в правом углу экрана последовательно появляются наборы цифр: 01, 02 и т. д. В левом же углу мы видим появление наборов цифр: 61, 12, ... При этом каждая новая комбинация из двух цифр смещает вправо появившиеся ранее комбинации. В дальнейшем на экране будут постоянно видны четыре двузначных набора цифр. Самый левый — это код последней введенной команды, далее вправо — коды двух предыдущих команд. Наконец, последняя пара цифр — адрес команды, которую предстоит еще вводить.

Коды позволяют контролировать, правильно ли вводится программа. Если вы случайно ошиблись при вводе команды, не отчаивайтесь. Нажмите клавишу «ШГ» (Шаг назад) и повторите ввод нужной команды.

Вот мы и научились набирать простейшие программы. Попробуем применить наши навыки.

Известно, что средние скорости движения молекул v (в м/с) газообразных веществ рассчитываются по формуле .

$$v = \sqrt{3RT/M},$$

где $R=8,31434$ Дж/(К·моль) — универсальная газовая постоянная; T — температура, К; M — масса одного моля вещества, кг/моль.

В качестве газообразных веществ выберем водород и азот. Напомним, что молекулярная масса водорода равна 2, а азота — 28.

Расчеты будем выполнять для различных температур в диапазоне от 140 до 500 К с интервалом 50 К. Конечно, для этой цели полезно составить программу вычислений и занести ее в память ПМК (табл. 4).

Для выполнения расчетов полезно представить универсальную газовую постоянную в виде $R=8314,34$ г·м²/(К·моль·с²).

Т а б л и ц а 4. Программа расчета средних скоростей движения молекул

Адрес	Команда	Код	Адрес	Команда	Код
00	ИП1	61	05	÷	13
01	×	12	06	$\sqrt{\quad}$	21
02	3	03	07	С/П	50
03	×	12	08	БП	51
04	ИП2	62	09	00	00

Переключим ПМК в режим вычислений, нажав клавиши FАВТ. Введем значение 8314,34 в регистр R1, молекулярную массу азота, равную 28,— в регистр R2. Нажмем кнопку В/О. Занесем значение температуры $T = 150$ К в регистр RХ. Теперь все готово к вычислениям по программе.

Нажимаем клавишу С/П. Начинается мелькание цифр на экране ПМК. Через несколько секунд на индикаторе высвечивается результат — число 365,54523. Это — средняя скорость движения молекул азота при температуре 150 К. Как следует из табл. 5, скорости движения более легких молекул водорода еще больше.

Для сравнения можно напомнить, что современный пассажирский авиалайнер летит со скоростью около 250 м/с.

Т а б л и ц а 5. Средние скорости движения молекул при различных температурах

Температура, К	Скорость движения молекул, м/с		Температура, К	Скорость движения молекул, м/с	
	водорода	азота		водорода	азота
150	1367,7450	365,54523	350	2089,2650	558,37957
200	1579,3359	422,09527	400	2233,5183	596,93286
250	1765,7512	471,91686	450	2369,0039	633,14292
300	1934,2835	516,95903	500	2497,1493	667,39124

КАК РАБОТАЕТ ПМК

Когда мы смотрим телепередачу, то, конечно, не думаем о принципах устройства телевизора. Владельцу магнитофона не обязательны знания физических основ записи и воспроизведения звука. Однако всегда полезно знать, хотя бы в общих чертах, как работает устройство, с которым имеешь дело. Вот мы и решили кратко рассказать о работе ПМК.

Как и любая вычислительная машина, ПМК имеет устройства ввода и вывода информации, процессор, запоминающее устройство и устройство управления. Клавиатура сочетает функции

устройства ввода и пульта управления. Информация выводится на экран (индикатор).

Команда, введенная с клавиатуры, направляется в запоминающее устройство, состоящее из нескольких частей: программной памяти, регистров данных, постоянного запоминающего устройства, программного указателя и адресного стека.

Программная память содержит ячейки с номерами от 00 до 97. Вводимые команды поступают в последовательно расположенные ячейки.

Адресный стек состоит из пяти ячеек и используется для запоминания адреса команды, на которую передается управление после окончания работы какой-либо *подпрограммы*.

Регистры данных служат для записи и хранения чисел. Таких регистров 14.

Постоянное запоминающее устройство содержит программы, с помощью которых реализуется процесс вычислений в ПМК. Эти программы раз и навсегда «защиты» в память ЭВМ. Именно они обеспечивают выполнение арифметических операций и вычисляют значения функций, указанных на клавиатуре.

Сердцем ПМК является *процессор*, точнее, арифметическо-логическое устройство. Оно работает совместно с *операционным стекком*, состоящим из пяти регистров: X, Y, Z, T и XI. Регистры X и Y особенно важны, так как из них берутся числа для выполнения двуместных операций: сложения, вычитания, умножения, деления и возведения в степень. Одноместные операции, такие как извлечение квадратного корня, возведение в квадрат, вычисление тригонометрических функций, выполняются с содержимым регистра X.

В соответствии с кодом команды процессор выполняет операцию, а ее результат помещает в регистр X. Напомним, что содержимое лишь этого регистра отображается на экране. Теперь Вам будет понятно, что на индикаторе ПМК во время его работы мелькают промежуточные результаты вычислений, появляющиеся в регистре X.

Проследим полный цикл работы микрокалькулятора при выполнении программы. Предположим, что она уже введена в память, установлен режим вычислений FАВТ и все необходимые числа внесены в адресуемые регистры. Нажатием клавиши В/О мы очищаем программный указатель. Клавиша С/П запускает программу.

Устройство управления считывает команду, адрес которой находится в *программном указателе*. После ее анализа и определения типа операции команда пересылается в процессор. По сигналам, идущим из устройства управления, процессор вырабатывает результат операции. Затем анализируется содержимое программного указателя и выясняется, какая команда должна выполняться следующей. После этого весь цикл повторяется.

Несколько иначе выполняются команды, отдаваемые микрокалькулятору в непрограммном (ручном) режиме работы. Такая команда не записывается в программную память, а поступает непосредственно в устройство управления, и при ее выполнении программный счетчик не меняет своего состояния. Благодаря этому появляется возможность решать на микрокалькуляторе задачи, сочетая *автоматическую обработку* информации по программе, хранящейся в его памяти, с *непрограммными вычислениями*, выполняемыми нажатием на клавиши. Такое сочетание оказывается полезным при необходимости решать на ПМК сложные задачи, для которых не удастся составить единую программу, уместящуюся в памяти микрокалькулятора.

Теперь несколько слов о самих программах. Вы уже заметили, что даже элементарные программы требуют для их разработки определенных затрат времени. Поэтому программы, полезные для будущих расчетов, следует сохранять и накапливать.

Перед тем, как начать составлять новую программу, полезно ознакомиться с литературой, в которой уже имеется значительное количество готовых программ. Список литературы приведен в конце книги.

РАДИОАКТИВНЫЙ РАСПАД УРАНА

Декабрьским днем 1938 года в одной из берлинских лабораторий произошло событие, вошедшее в историю науки как расщепление атомного ядра. О. Ган и его ассистент химик Ф. Штрасман подвергли бомбардировке нейтронами уран и провели после этого тщательный химический анализ. С помощью специального прибора они регистрировали импульсы, испускаемые облученным препаратом. Это позволяло им идентифицировать образовавшиеся элементы не только химически, но и физически — по *периоду полураспада*. Напомним, что периодом полураспада называется время, в течение которого радиоактивность препарата уменьшается вдвое.

В табл. 6 представлены результаты измерений, произведенных О. Ганом. Последний столбец получен из предпоследнего вычитанием постоянной, равной 15,9 распадов/мин. Именно такой фон измерял регистрирующий прибор в отсутствие радиоактивного препарата.

Закон, по которому изменяется радиоактивность источника, в 1938 году был уже известен:

$$y = y_0 e^{-at},$$

где y — число распадов в минуту в момент времени t ; y_0 — число распадов в минуту в начальный момент времени $t=0$, t — время, мин.

Располагая экспериментальными данными, можно рассчитать коэффициент a по формуле:

$$a = \frac{\sum_1^n (t \ln y) - \frac{\sum_1^n t \sum_1^n \ln y}{n}}{\sum_1^n t^2 - \frac{\left(\sum_1^n t\right)^2}{n}},$$

где n — число выполненных измерений.

Имея под рукой микрокалькулятор, нетрудно проверить, что $a = 0,00454$.

Чтобы вычислить период полураспада t_n , необходимо в исходное выражение для y подставить $y_0/2$, т. е.

$$\frac{y_0}{2} = y_0 e^{-at_n}.$$

Отсюда получается $t_n = 152,665$ мин.

Достигнутый в то время уровень знаний позволял с уверенностью утверждать, что такое значение не может быть периодом полураспада радия. Это означало, что ядро урана претерпело *расщепление*. Так было открыто деление атомного ядра.

Таблица 6. Измерения в процессе радиоактивного распада урана

Время t , мин	Показания счетчика числа распадов y	Число распадов в минуту	Интенсивность распадов $1/\Delta t$	Приведенная радиоактив- ность (без учета фона)
0	71 398	—	—	—
2	71 562	164	82,0	66,1
6	71 885	323	80,8	64,9
10	72 170	285	71,3	55,4
16	72 610	440	73,3	57,4
28	73 511	901	75,1	59,2
48	74 653	1 142	57,1	41,2
58	75 311	658	65,8	49,9
89	76 854	1 543	49,8	33,9
95	77 200	346	57,7	41,8
101	77 576	376	62,7	46,8
115	78 202	626	44,7	28,8
121	78 557	355	59,2	43,3
127	78 875	318	53,0	37,1

Примечание. Здесь Δt — интервал по времени.

НИТРУЮЩИЕ СМЕСИ

ПМК можно с успехом использовать для расчета концентраций веществ, входящих в состав различных смесей (например, нитрующих).

Нитрующей называется смесь концентрированной азотной кислоты или оксидов азота с неорганическими или органическими веществами.

При приготовлении нитрующих смесей известны следующие величины: количество смеси, которое следует приготовить, ее состав и состав исходных компонентов. Искомыми величинами являются количества компонентов.

Предположим, что нитрующая смесь должна быть приготовлена из n компонентов, в состав которых входят m веществ, и требуется найти количество каждого компонента в смеси.

Попробуем сначала поставить задачу в общем виде. Для этого введем обозначения: M — количество приготавливаемой смеси, кг; x_j ($j = \overline{1, n}$) — количество j -го компонента в смеси, кг; b_i ($i = \overline{1, m}$) — содержание i -го вещества в смеси, %; a_{ij} — содержание i -го вещества в j -м компоненте, %.

Тогда количество i -го вещества в смеси, с одной стороны, будет равно Mb_i , а с другой, — сумме количеств этого вещества в каждом из n компонентов:

$$\sum_{j=1}^n a_{ij}x_j \quad (i = \overline{1, m}).$$

Значит,

$$\sum_{j=1}^n a_{ij}x_j = Mb_i \quad (i = \overline{1, m}),$$

или

$$Ax = Mb;$$

$$A = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \cdot & & \cdot \\ a_{m1} & \dots & a_{mn} \end{pmatrix}; \quad x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}; \quad b = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}.$$

По смыслу задачи понятно, что количество приготавливаемой нитрующей смеси должно быть равно сумме количеств всех входящих в нее компонентов:

$$M = \sum_{j=1}^n x_j.$$

Итак, в процессе нахождения количества каждого вещества в смеси получена система линейных алгебраических уравнений, которую нам необходимо решить.

Пусть требуется приготовить нитрующую смесь из трех компонентов, содержащих воду, азотную и серную кислоты. Необходимо

рассчитать, какое количество каждого компонента необходимо взять, чтобы получить M кг смеси, содержащей b_1 , b_2 и b_3 % H_2O , HNO_3 и H_2SO_4 , соответственно, если содержание воды, азотной и серной кислот в каждом компоненте задано матрицей

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}.$$

Поскольку смесь должна быть приготовлена из трех компонентов, каждый из которых состоит из трех веществ, то система линейных уравнений имеет вид:

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = Mb_1;$$

$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = Mb_2;$$

$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = Mb_3.$$

Чтобы найти значения x_i , можно воспользоваться правилом Крамера. Для этого найдем определитель матрицы A , т. е.

$$\det A = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}.$$

Если определитель системы $\det A \neq 0$, то можно найти все неизвестные x_i . Например,

$$x_2 = \frac{M}{\det A} \begin{vmatrix} a_{11} & b_1 & a_{13} \\ a_{21} & b_2 & a_{23} \\ a_{31} & b_3 & a_{33} \end{vmatrix}.$$

Допустим, требуется приготовить 4250 кг нитрующей смеси следующего состава: воды (b_1) — 22 %, азотной кислоты (b_2) — 16 %, серной кислоты (b_3) — 62 %. Исходными компонентами являются:

1) меланж, т. е. смесь состава $\text{H}_2\text{O}(a_{11})$ — 5 %, $\text{HNO}_3(a_{21})$ — 85 %, $\text{H}_2\text{SO}_4(a_{31})$ — 10 %;

2) олеум, т. е. дымящая серная кислота [$\text{H}_2\text{O}(a_{12})$ — 0 %, $\text{HNO}_3(a_{22})$ — 0 %, $\text{H}_2\text{SO}_4(a_{32})$ — 104,5 %];

3) отработанная серная кислота [$\text{H}_2\text{O}(a_{13})$ — 30 %, $\text{HNO}_3(a_{23})$ — 0 %, $\text{H}_2\text{SO}_4(a_{33})$ — 70 %].

Найдем расход кислот, идущих на приготовление заданной смеси.

Составим определитель системы и вычислим его

$$\det A = \begin{vmatrix} 5 & 0 & 30 \\ 85 & 0 & 0 \\ 10 & 104,5 & 70 \end{vmatrix} = 85 \cdot 104,5 \cdot 30.$$

Затем найдем

$$\det A_1 = 4250 \begin{vmatrix} 22 & 0 & 30 \\ 16 & 0 & 0 \\ 62 & 104,5 & 70 \end{vmatrix} = 4250 \cdot 16 \cdot 104,5 \cdot 30.$$

Теперь рассчитаем расход меланжа:

$$x_1 = \frac{4250 \cdot 16 \cdot 104,5 \cdot 30}{85 \cdot 104,5 \cdot 30} = 800 \text{ (кг)}.$$

Вычислив

$$\det A_2 = 4250 \begin{vmatrix} 5 & 22 & 30 \\ 85 & 16 & 0 \\ 10 & 62 & 70 \end{vmatrix} = 4250 \cdot 100 \cdot 280,$$

найдем расход олеума:

$$x_2 = \frac{4250 \cdot 100 \cdot 280}{85 \cdot 104,5 \cdot 30} = 475 \text{ (кг)}.$$

Найдем теперь

$$\det A_3 = 4250 \begin{vmatrix} 5 & 0 & 22 \\ 85 & 0 & 16 \\ 10 & 104,5 & 62 \end{vmatrix} = 4250 \cdot 104,5 \cdot 10 \cdot 179$$

и вычислим расход отработанной серной кислоты:

$$x_3 = \frac{4250 \cdot 104,5 \cdot 10 \cdot 179}{85 \cdot 104,5 \cdot 30} = 2975 \text{ (кг)}.$$

Чтобы проверить правильность решения, найдем сумму

$$x_1 + x_2 + x_3 = 800 + 475 + 2975 = 4250 \text{ (кг)}.$$

Следовательно, задача решена правильно.

**АЛГОРИТМИЧЕСКИЕ ЯЗЫКИ
ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА**

Микрокалькулятор, с которым Вы уже успели познакомиться, безусловно, полезен для многих расчетов. Однако его вычислительные возможности невелики. Значительно большие перспективы перед химиками открывает применение персональных компьютеров. Эти ЭВМ имеют значительный объем памяти и обладают хорошим быстродействием. Программы для персональных компьютеров обычно хранятся на гибких магнитных дисках (флôппи-дисках). Такие ЭВМ имеют дисплеи, печатающие устройства, а некоторые из них — еще и устройства машинной графики.

Информация вводится в персональный компьютер на различных алгоритмических языках: БЕЙСИК, ФОРТРАН, ПАСКАЛЬ и др. В настоящее время одним из самых распространенных языков является БЕЙСИК.

Чтобы общаться с персональной ЭВМ, нужно освоить основы алгоритмического языка и научиться писать на нем программы.

Название языка БЕЙСИК (BASIC) возникло от сокращения английских слов Beginner's All-purpose Symbolic Instruction Code, что в переводе означает «многоцелевой язык символических инструкций для начинающих».

ПЕРВАЯ ПРОГРАММА

Рассмотрим задачу об истечении жидкости из цилиндрического сосуда через отверстие, находящееся в его дне. Пусть уровень жидкости равен h и поддерживается постоянным, площадь поперечного сечения сосуда равна S , а площадь отверстия в дне — F . Скорость истечения жидкости рассчитывается по формуле

$$w = \frac{kF}{S} \sqrt{2gh},$$

где k — коэффициент истечения, который зависит от свойств жидкости и формы отверстия, через которое она вытекает; g — ускорение свободного падения.

Программа для расчета скорости истечения жидкости имеет следующий вид:

```
10 REM РАСЧЕТ СКОРОСТИ ИСТЕЧЕНИЯ ЖИДКОСТИ
20 LET G=9.81
30 INPUT K,F,S,H
40 LET W=K*F/S*SQR(2*G*H)
50 PRINT W
60 STOP
70 END
```

Программа на БЕЙСИКЕ состоит из последовательности пронумерованных *строк*. Номера строк обычно изменяются от 1 до нескольких десятков тысяч (это зависит от конструкции ЭВМ). Удобно вести нумерацию строк, начиная с 10, с интервалом в 10. Вводиться в ЭВМ строки могут в любом порядке, а выполняться — только в порядке возрастания их номеров. Номера строк используются в операторах передачи управления.

Нумерация строк с *интервалами* очень удобна; благодаря ей имеется возможность, в случае необходимости, вводить в программу новые строки, не затрагивая остального текста.

В нашей программе в строке с номером 10 записан *комментарий*, указывающий на назначение программы. В строке 20 переменной G, которая означает ускорение свободного падения g , присваивается значение 9,81. В следующей строке записан *оператор ввода* значений k , F , S и h (в программе соответственно, K , F , S , H). Далее идет оператор, осуществляющий вычисление скорости истечения жидкости. В строке 50 записан *оператор печати* результатов вычислений. Оператор STOP, который мы видим в строке 60, прекращает работу программы. Наконец, оператор END символизирует *конец текста* программы.

Существует несколько *версий* алгоритмического языка БЕЙСИК, поэтому, приступая к программированию, следует сначала ознакомиться с описанием соответствующей версии.

ЭЛЕМЕНТЫ ЯЗЫКА

Операторы БЕЙСИКА строятся из элементов, которыми являются числа, переменные, стандартные функции, арифметические выражения.

Запись чисел близка к естественной, только для отделения целой части от дробной вместо запятой используется *точка*. Очень большие и очень маленькие числа записываются в экспоненциальной форме с указанием порядка, но вместо основания степени 10 употребляется буква E. Например, число Авогадро $6,02 \cdot 10^{23}$ записывается в виде 6.02E23.

Порядок числа, т. е. цифры после буквы E, может быть только целым числом, содержащим одну или две цифры.

Запись *целых чисел* заканчивается знаком % (например, 1935%).

Т а б л и ц а 7. Стандартные функции

Запись на языке БЕЙСИК	Математическое определение	Запись на языке БЕЙСИК	Математическое определение
ABS (X)	$ x $	LOG (X)	$\ln x$
ATN (X)	$\arctg x$	SGN (X)	$\begin{cases} 1, & \text{если } x > 0 \\ 0, & \text{если } x = 0 \\ -1, & \text{если } x < 0 \end{cases}$
COS (X)	$\cos x$		
SIN (X)	$\sin x$		
EXP (X)	e^x	SQR (X)	\sqrt{x}
INT (X)	$[x]$ — наибольшее целое число не превосходящее x	RND (X)	Генератор случайных чисел

Для обозначения *вещественных* переменных в языке БЕЙСИК используются имена (идентификаторы), состоящие либо из одной буквы, либо из буквы и цифры (например, X, A2, S). В качестве букв используются заглавные (прописные) буквы *латинского* алфавита. К именам переменных целого типа добавляется знак % (например, M%, N%).

В языке БЕЙСИК имеются также переменные, значениями которых являются строки символов, — это *символьные переменные*. Они обозначаются так же, как и числовые, т. е. одной буквой или буквой и цифрой, только после символьной переменной обязательно ставится специальный знак $\text{\textcircled{X}}$. Например, A $\text{\textcircled{X}}$ (читается: A символьное), X1 $\text{\textcircled{X}}$ и т. д. Длина символьной переменной, т. е. число входящих в нее символов, не должно превышать 255. В состав символьных переменных можно включать любые знаки, имеющиеся на клавиатуре персонального компьютера, например буквы *русского* алфавита.

При работе на персональном компьютере имеется возможность использовать стандартные функции, приведенные в табл. 7. Аргументы функций заключаются в круглые скобки.

В *арифметическом выражении* могут входить числа, переменные и функции, соединенные знаками арифметических операций. Если в арифметическом выражении имеется несколько различных арифметических операций, то порядок их выполнения задается правилами *приоритета*: \wedge (возведение в степень); $*$, $/$ (умножение и деление); $+$, $-$ (сложение, вычитание). Если в арифметическом выражении встречаются круглые скобки, то операции в скобках выполняются в первую очередь. Порядок действий здесь не отличается от обычного, принятого в математике.

ОПЕРАТОРЫ

Программа для компьютера, как известно, состоит из отдельных операторов.

Оператор присваивания начинается словом LET (в переводе с английского — «пусть»), за которым стоит имя переменной*. Далее следует знак операции присваивания «=», только внешне напоминающий знак равенства. В правой части оператора стоит арифметическое выражение, например

LET Y=A*X+B

При переводе значения температуры T из градусов Цельсия в кельвины мы пользуемся оператором

LET T=T+273

Смысл такого оператора очевиден: нужно взять исходное значение T и, прибавив к нему число 273, присвоить полученный результат той же переменной T .

Условный оператор служит для передачи управления к тому или другому оператору в зависимости от проверяемого условия. Общий вид условного оператора:

IF условие THEN N
или
IF условие GOTO N

Здесь условие имеет вид

$$\left(\text{арифметическое выражение} \right)_1 \Theta \left(\text{арифметическое выражение} \right)_2,$$

где Θ — одно из отношений:

- < — меньше;
- <= — меньше или равно (не больше);
- > — больше;
- >= — больше или равно (не меньше);
- = — равно;
- < > — не равно.

В операторе используются слова: IF (если), THEN (то), GOTO (идти к ...).

Если условие удовлетворяется, то осуществляется переход к строке с номером N . Если же условие не удовлетворяется, то осуществляется переход к оператору, который следует за условным.

Встречается также условный оператор вида

IF условие THEN оператор

Здесь после слова THEN может следовать любой оператор языка БЕЙСИК. В частности, им может быть и любой условный оператор.

* В некоторых версиях БЕЙСИКа слово LET в операторе присваивания отсутствует.

Рассмотрим пример. Температура охлаждающегося раствора t зависит от времени τ и рассчитывается по формуле

$$t = t_2 + (t_1 - t_2) e^{-\frac{\alpha \tau}{mc}}$$

где t_1 — температура раствора в момент времени $\tau=0$; t_2 — температура среды, окружающей сосуд с раствором; m — масса раствора; c — удельная теплоемкость; α — коэффициент теплоотдачи.

Требуется рассчитать температуру раствора в период времени от 0 до 120 мин с интервалом в 2 мин.

Для выполнения расчетов можно воспользоваться следующей программой:

```

10 REM ОХЛАЖДЕНИЕ РАСТВОРА
20 INPUT K, M, C, T1, T2
30 LET X=0
40 LET T=T2+(T1-T2)*EXP(-K*X/M/C)
50 PRINT X, T : X=X+2
60 IF X <= 3 GOTO 40
70 STOP
80 END

```

В БЕЙСИКЕ нет греческих букв, поэтому вместо τ мы использовали букву X, а вместо α — букву K (ведь от выбора буквы результат не зависит).

В одной строке программы разрешается записывать несколько операторов. Так, в строке с номером 50 имеются два оператора, разделенные знаком «:».

Оператор безусловного перехода имеет вид GOTO N, где N — номер строки. Этот оператор передает управление первому оператору в строке с номером N.

Цикл является наиболее типичной структурой алгоритмов, реализуемых на ЭВМ. Для организации циклов в языке БЕЙСИК служит пара операторов FOR — NEXT. Общий вид операторов цикла:

```
FOR V = A1 TO A2 STEP A3
```

тело цикла

```
NEXT V
```

Здесь v — переменная, управляющая работой цикла; A_1 , A_2 , A_3 — начальное и конечное значения и шаг изменения переменной цикла. Если $A_3=1$, то конструкцию STEP1 можно опустить, не писать.

Операторы, расположенные между FOR и NEXT, образуют так называемое тело цикла и выполняются многократно.

Цикл, образованный операторами FOR — NEXT реализуется в такой последовательности: переменной v присваивается начальное значение A_1 , и она сравнивается с конечным значением A_2 . Если при положительном шаге A_3 значение $A_1 \leq A_2$ или при отрицательном шаге A_3 значение $A_1 \geq A_2$, то выполняются операторы тела цикла, и по оператору NEXT осуществляется возврат к началу

цикла. Значение v изменяется на шаг $v = v + \Delta_3$ и сравнивается с конечным значением. Если конечное значение не достигнуто, цикл повторяется. Иначе цикл заканчивается и происходит переход к оператору, следующему за NEXT.

Рассмотрим пример. Известно, что удельная теплоемкость вещества зависит от температуры. Эта зависимость выражается формулой

$$c = a + bT + dT^2,$$

где a , b и d — некоторые постоянные коэффициенты; T — температура.

Требуется рассчитать удельную теплоемкость в интервале температур от 100 до 800 К с шагом 50 К. Для выполнения расчетов воспользуемся программой, содержащей цикл:

```
10 INPUT A,B,D
20 FOR T=100 TO 800 STEP 50
30 LET C=A+B*T+D*T^2
40 PRINT C
50 NEXT T
60 STOP
70 END
```

При выполнении программы все нужные нам значения удельных теплоемкостей будут выданы на печать.

Вы уже, наверное, заметили, что в некоторых программах используется *оператор-комментарий*. Он служит для большей наглядности программ, указывает их назначение и смысл отдельных фрагментов расчетов. В комментариях после слова REM могут быть записаны любые символы языка БЕЙСИК, в том числе и *русские* буквы. Важно помнить, что все написанное за словом REM до конца строки воспринимается вычислительной машиной как комментарий и при выполнении программы игнорируется.

Оператор STOP служит для прекращения вычислений по программе. Допускается использование нескольких операторов STOP, расположенных в различных местах программы.

Выполнение программы может быть продолжено, если с пульта управления ЭВМ подать команду GOTO N. Здесь N обозначает номер строки, на которую передается управление.

Использование оператора STOP не является обязательным в языке БЕЙСИК. Если он отсутствует, то выполнение программы продолжается до ее естественного завершения.

КАК ВВЕСТИ И ВЫВЕСТИ ИНФОРМАЦИЮ

Оператор INPUT позволяет ввести данные с клавиатуры дисплея и занести их в память ЭВМ. После слова INPUT указываются названия переменных, для которых будут вводиться значения. При выполнении оператора ввода ЭВМ делает паузу и на экране высвечивается знак вопроса. Машина как бы спрашивает: «Какое число вы хотите ввести?». В ответ следует набрать нужное число на клавиатуре ЭВМ и после этого нажать кнопку с надписью BK

(Возврат Каретки). Если вводятся несколько чисел, то они отделяются одна от другой *запятыми*.

Порядок ввода чисел должен всегда *строго* соответствовать списку переменных в операторе INPUT, их ни в коем случае нельзя менять местами. Для облегчения этой задачи в состав оператора ввода можно включать *подсказку*, которая представляет собой текст, заключенный в кавычки и предшествующий списку ввода. Например, пусть требуется ввести значения температуры T и концентрации вещества C . Это можно сделать с помощью оператора

```
10 INPUT " ВВЕДИТЕ ТЕМПЕРАТУРУ И КОНЦЕНТРАЦИЮ " T, C
```

При выполнении такого оператора на экране дисплея сначала появится текст, записанный в кавычках, а потом — знак вопроса. Конечно, подсказка облегчает работу при вводе информации в компьютер.

Ввод информации можно осуществить также с помощью операторов READ и DATA. В программе они всегда присутствуют одновременно. Оператор READ имеет список вводимых переменных, например

```
READ A, X, B1
```

Оператор DATA содержит числа, которые присваиваются в качестве значений переменным, входящим в список ввода оператора READ. Рассмотрим операторы

```
10 DATA 8.4, 2.5, 14.9  
20 READ X, Y, Z
```

После выполнения оператора READ

```
X = 8.4      Y = 2.5      Z = 14.9
```

Нужно иметь в виду, что числа, вводимые оператором READ бесследно исчезают из оператора DATA. Чтобы вернуть их обратно, служит оператор RESTORE.

Допустим, что в программе имеются операторы

```
10 DATA 5, 6, 7, 8  
20 READ A, B, C, D : RESTORE  
30 READ X, Y
```

После выполнения этой группы операторов переменные будут иметь следующие значения:

```
A = 5      B = 6      C = 7      D = 8      X = 5      Y = 6
```

Оператор PRINT служит для вывода информации на экран дисплея. Вслед за словом PRINT идет обычно *список вывода*, который может содержать имена переменных, числа, арифметические выражения и тексты. Текст применяют для печати всевозможных

заголовков и пояснений, например PRINT «ПЕРИОДИЧЕСКАЯ СИСТЕМА ЭЛЕМЕНТОВ».

Текст заключается в кавычки и может содержать любые символы языка БЕЙСИК, в том числе буквы русского алфавита.

Если в списке вывода содержатся арифметические выражения, то они сначала вычисляются.

Сопровождая результаты расчетов на ЭВМ текстовой информацией, мы существенно облегчаем их восприятие.

РАБОТА НА ДИСПЛЕЕ

После освоения основ программирования возникает желание приступить к решению задач на персональном компьютере. Включив ЭВМ в работу, мы вводим с магнитного диска *операционную систему* и *интерпретатор* языка БЕЙСИК. Интерпретатор — это вспомогательная программа, осуществляющая перевод текста с БЕЙСИКА на *внутренний* язык машины. Когда компьютер готов к работе, на экране высвечивается слово READY. Теперь можно начинать ввод программы.

Программа вводится по строкам. Для ввода строки нужно сначала набрать на клавиатуре ее номер и далее — содержащиеся в ней операторы, отделяя один оператор от другого, если их несколько, символом «: ». Следует иметь в виду, что перенос оператора с одной строки на другую *не допускается*, а общее число символов в строке не должно превышать 72.

Ввод набранной строки в оперативную память ЭВМ осуществляется нажатием клавиши ВК (Возврат Каретки). Если при наборе строки допущена ошибка, то, используя клавишу ЗБ (забой) можно стереть ошибочные символы в конце строки и набрать правильные.

После ввода всей программы ее можно запустить на счет командой RUN.

В процессе набора программы возможны ошибки, которые выявляются только во время счета.

В случае необходимости можно внести в программу исправления и изменения. Если нужно вставить новую строку между соседними строками с номерами N_1 и N_2 , то для этого набирается строка с соответствующим номером и нажимается клавиша ВК. Для стирания какой-либо строки следует набрать ее номер и нажать клавишу ВК.

Программу можно вывести на экран дисплея командой LIST. При этом строки программы располагаются в порядке возрастания номеров, независимо от последовательности их ввода.

Запись программы из оперативной памяти на диск выполняется командой

SAVE имя

Имя — это набор от 1 до 6 символов, начинающийся с буквы. Программе, находящейся в оперативной памяти ЭВМ, можно присвоить новое имя командой

```
RENAME ИМЯ
```

При этом старое имя «забывается».

Загрузка программы с диска в оперативную память осуществляется командой *

```
OLD ИМЯ
```

Приобретенные навыки можно закрепить только постоянной тренировкой, поэтому почаще общайтесь с компьютером.

СРЕДНЯЯ ТЕМПЕРАТУРА

В ходе химической реакции может происходить выделение или поглощение теплоты, поэтому температура реакционной массы изменяется во времени. Если нужно найти среднюю температуру, при которой протекала реакция, то следует сложить все измеренные температуры и разделить полученную сумму на количество измерений, т. е. вычислить среднее арифметическое значение. Составим для этой цели программу на языке БЕЙСИК.

Сумму значений температур обозначим буквой S, каждое отдельное значение температуры — буквой T, общее количество измерений — буквой N, номер выполненного измерения — буквой I. Программу вычислений запишем в следующем виде:

```
10 REM ВЫЧИСЛЕНИЕ СРЕДНЕЙ ТЕМПЕРАТУРЫ
20 INPUT "ВВЕДИТЕ ОБЩЕЕ ЧИСЛО ИЗМЕРЕНИЙ " ,N
30 IF N<1 GOTO 110
40 LET S=0
50 FOR I=1 TO N
60 INPUT "ВВЕДИТЕ ЗНАЧЕНИЕ ТЕМПЕРАТУРЫ " ,T
70 LET S=S+T
80 NEXT I
90 LET S=S/N
100 PRINT "СРЕДНЯЯ ТЕМПЕРАТУРА = ";S
110 STOP
```

Чтобы рассчитать среднюю температуру химической реакции, нужно с пульта ЭВМ последовательно ввести все результаты измерений. Если же измерения не проводились, т. е. $N=0$, компьютер откажется выполнять какие-либо вычисления по данной программе. Это предусмотрено оператором, стоящим в строке с номером 30. С этой простой программы можно начать практическую работу у пульта персонального компьютера.

* В некоторых версиях БЕЙСИКа загрузка программы осуществляется оператором

```
LOAD ИМЯ
```

ЗАЧЕМ НУЖНЫ ИНДЕКСЫ!

Переменные величины, которые мы до сих пор применяли для составления программ, называются *простыми* (неиндексированными) переменными. Каждая такая переменная имеет свое имя; и под каждую из них выделяется ячейка памяти ЭВМ. Использование только одних простых переменных затрудняет, а подчас делает даже невозможным решение многих важных научных и прикладных задач. Для устранения этих трудностей используются *массивы*.

Массивом в языке БЕЙСИК называется упорядоченная последовательность величин, обозначаемая *одним именем*. Упорядоченность заключается в том, что элементы массива располагаются в последовательных ячейках памяти. Номер *элемента* массива указывается в круглых скобках после его имени, например, A(1) — первый элемент массива A, X(15) — пятнадцатый элемент массива X. Числа, стоящие в скобках, называются *индексами*. Элементы массивов принято называть *индексированными переменными*, или переменными с индексами.

Пусть, например, в процессе расчетов технологического процесса используется массив концентраций определенного вещества, содержащий 5 элементов. Обозначим концентрацию буквой C, тогда массив можно представить в следующем виде:

C(1) , C(2) , C(3) , C(4) , C(5)

Для описания массивов в языке БЕЙСИК служит оператор DIM (от слова DIMENSION — размерность), который должен стоять в программе до любого оператора, в котором используется соответствующая индексированная переменная. Вслед за словом DIM указывается имя массива и в скобках — максимальный номер элемента, например DIM S(100) указывает, что в программе используется массив S, содержащий 100 элементов.

В языке БЕЙСИК допускаются переменные с одним и двумя индексами. Они являются элементами соответствующих *одномерных* и *двумерных* массивов. Двумерный массив — это таблица, причем первый индекс указывает строку, а второй индекс — столбец, где расположен соответствующий элемент. Например, оператор DIM A(15, 20) описывает двумерный массив, элементы которого размещены в памяти ЭВМ в виде таблицы, содержащей 15 строк и 20 столбцов. Элемент A(7, 9) находится в 7 строке и 9 столбце.

Допустим, группа из 10 человек сдала 4 экзамена и получила следующие оценки:

за 1-й экзамен: 4, 5, 5, 3, 4, 3, 5, 4, 4, 4;
за 2-й экзамен: 4, 4, 3, 4, 4, 3, 3, 5, 4, 3;
за 3-й экзамен: 3, 5, 5, 4, 5, 4, 4, 4, 5, 5;
за 4-й экзамен: 4, 5, 4, 3, 4, 3, 3, 3, 4, 3.

Требуется составить программу для вычисления среднего балла на каждом экзамене. Такая программа на языке БЕЙСИК имеет вид:

```
10 REM РАСЧЕТ СРЕДНИХ БАЛЛОВ
20 DIM B(4,10),C(4)
30 DATA 4,5,5,3,4,3,5,4,4,4
40 DATA 4,4,3,4,4,3,3,5,4,3
50 DATA 3,5,5,4,5,4,4,4,5,5
60 DATA 4,5,4,3,4,3,3,3,4,3
70 FOR I=1 TO 4 : FOR J=1 TO 10
80 READ B(I,J) : NEXT J : NEXT I
90 FOR I=1 TO 4 : LET S=0
100 FOR J=1 TO 10 : LET S=S+B(I,J)
110 NEXT J : C(I)=S/10 : NEXT I
120 PRINT " СРЕДНИЕ БАЛЛЫ НА ЭКЗАМЕНАХ "
130 FOR I=1 TO 4 : PRINT C(I) ; : NEXT I
140 STOP : END
```

В этой программе исходные данные для наглядности заданы с помощью операторов DATA.

ФОРМА ПЕЧАТИ РЕЗУЛЬТАТОВ

После того, как мы познакомились с массивами и индексированными переменными, следует снова вернуться к операторам ввода и вывода информации, чтобы научиться печатать результаты в наглядной и красивой форме.

Для удобства восприятия выводимых результатов экран дисплея условно разделен на 5 зон по 14 позиций в каждой. Печать каждого следующего числа осуществляется в следующую *зону*. Такая форма вывода обеспечивается использованием знака «,» (запятой) в качестве разделителя между элементами списка вывода. Появление запятой всегда указывает на переход к следующей зоне. Такой способ вывода информации называется выводом в *зонном формате*.

После последнего элемента списка вывода также может стоять запятая. Тогда при выполнении следующего оператора PRINT печать элементов его списка осуществляется в следующей зоне, без переноса на следующую строку.

Пусть например, $X = 12,3$; $Y = 7,5$; $Z = 147$.

Тогда при выполнении операторов

```
60 PRINT X , Y ,
70 PRINT Z
```

на одной строке будут отпечатаны три числа:

```
12.3          7.5          147
```

Если же в строке 60 после Y не будет стоять запятая, то последнее число (147) будет отпечатано с новой строки.

Когда в качестве разделителя используется знак «;» (точка с запятой) вывод информации осуществляется в так называемом *плотном формате*, т. е. каждое следующее число печатается после предыдущего через пробел.

Вернемся к предыдущему примеру и воспользуемся оператором

```
50 PRINT X ; Y ; Z
```

Тогда будут отпечатаны числа в следующем виде:

```
12.3 7.5 147
```

Так поступают при выводе большого количества данных.

Оператор PRINT USING позволяет задать точный вид печати отдельных элементов списка вывода. Он имеет форму

```
PRINT USING "формат" , список
```

Формат указывает, как должен осуществляться вывод результатов. При печати числовых значений для каждой выводимой цифры и знака числа в формате предусматривается символ #. Остальные символы, включая пробелы, печатаются без всяких изменений.

Для вывода числа, содержащего дробную часть, в формате задается столько символов #, сколько всего цифр требуется напечатать, и точка для отделения цифр целой части от дробной, т. е. использование формата, позволяет исключить ненужные цифры. Например, с помощью оператора

```
20 PRINT USING "#.## ##.#" , 7.421 , -2.60
```

печатаются числа

```
7.42 -2.6
```

Если в формате печати указан только один образ представления числа, то он может использоваться многократно — для каждого элемента в списке вывода. При этом печать каждого следующего элемента будет осуществляться с новой строки.

Используя оператор PRINT USING, мы можем печатать таблицы в удобной для восприятия форме. Например, с помощью программы

```
10 REM ПЕЧАТЬ ТАБЛИЦЫ
20 LET K=0.25
30 FOR I=1 TO 4
40 FOR J=1 TO 4
50 LET K=K+2.25
60 PRINT USING "##.##" ; K;
70 NEXT J
80 PRINT
90 NEXT I
100 STOP : END
```

печатается следующая таблица чисел

2.50	4.75	7.00	9.25
11.50	13.75	16.00	18.25
20.50	22.75	25.00	27.25
29.50	31.75	34.00	36.25

Оператор PRINT в строке 80 служит только для возврата каретки перед печатью новой строки.

В языке БЕЙСИК имеется *функция TAB*, которая обычно используется в составе списка вывода оператора PRINT. Она обеспечивает вывод следующего за ней элемента в позицию, номер которой задан как аргумент функции TAB. Аргументом может быть любое арифметическое выражение.

Если значение аргумента функции TAB содержит дробную часть, то при нахождении позиции печати дробная часть отбрасывается. Аргумент должен быть в пределах от 0 до 71 (по числу позиций строки экрана дисплея). Когда аргумент функции TAB меньше текущей позиции строки, эта функция игнорируется компьютером, а печать осуществляется через пробел после предыдущего числа.

Например, оператор

```
20 PRINT TAB(25) ; 7
```

выводит число 7 в 25-ю позицию строки экрана.

При выполнении операторов

```
30 LET X=15  
40 PRINT TAB(X) ; "*"
```

в 15-ю позицию экрана выводится символ «*». Функция TAB используется также для вывода различной графической информации.

ПОДПРОГРАММЫ

Часто в нескольких частях программы встречается одна и та же совокупность операций. Чтобы избежать многократной записи операторов, реализующих эти действия, можно составить так называемую *подпрограмму* и обращаться к ней всякий раз, когда в этом возникает необходимость.

Использование подпрограммы уменьшает общее число операторов и, следовательно, требует меньше памяти для размещения программы.

Обращение к подпрограмме осуществляется оператором GOSUB N, где N — номер строки, с которой она начинается. Последним оператором подпрограммы является оператор RETURN, по которому осуществляется возврат в то место программы, из которого произошло обращение к подпрограмме, т. е. к оператору, следующему за GOSUB N.

Подпрограмм может быть несколько. Одна подпрограмма может содержать передачу управления к другой подпрограмме. Пусть, например, требуется рассчитать по формуле Аррениуса

$$k = k_0 e^{-\frac{E}{RT}}$$

константы скоростей трех химических реакций в интервале температур от T1 до T2 с шагом T3. Эту задачу можно решить с помощью следующей программы:

```
10 INPUT T1,T2,T3,R
20 INPUT K0,E : GOSUB 500
.....
90 INPUT K0,E : GOSUB 500
.....
140 INPUT K0,E : GOSUB 500
150 STOP
500 FOR T=T1 TO T2 STEP T3
510 LET K=K0*EXP(-E/R/T)
520 PRINT T,K
530 NEXT T : PRINT
540 RETURN
```

Подпрограмма начинается с 500-й строки. Обращение к ней осуществляется из трех различных мест программы, а именно из строк с номерами 20, 90 и 140. Благодаря подпрограмме, удается избежать лишних записей, однако количество вычислительных операций при этом не уменьшается.

КОМПЬЮТЕР ЧЕРТИТ И РИСУЕТ

Развитие вычислительной техники привело к расширению области применения компьютеров. Со временем они стали использоваться не только для вычислений и хранения информации, но и для выполнения множества различных графических работ. В связи с этим в составе алгоритмических языков программирования появились операторы *машинной графики*.

Простейший способ создания различных изображений с помощью ЭВМ основан на использовании оператора вывода информации PRINT, который позволяет объединить буквы, числа и другие символы, имеющиеся на клавиатуре дисплея, в единый рисунок. В современных версиях алгоритмического языка БЕЙСИК и других языков имеются специальные операторы для построения изображений в виде точек и линий.

Простые изображения можно построить, располагая некоторые символы в соответствии с очертаниями предмета. Например, с помощью оператора

```
10 PRINT "*****"
```

можно напечатать последовательность пяти звездочек. Прямоугольный контур легко построить с помощью операторов

```
10 PRINT "*****"  
20 PRINT "*      *"  
30 PRINT "*          *"  
40 PRINT "*****"
```

С помощью операторов PRINT можно, например, напечатать крупными буквами слово ХИМИЯ:

```
10 PRINT " * * * * * * * * * *"  
20 PRINT "  * * * ** ** ** * ** * *"  
30 PRINT "   * * * * * * * * * *"  
40 PRINT "    * * ** * * * * ** * * *"  
50 PRINT "     * * * * * * * * * *"
```

Для программирования сложного изображения его сначала следует нарисовать на бумаге в клетку. С помощью этого рисунка можно определить, в каких позициях следует печатать символы, чтобы получилось изображение контура или силуэта предмета.

Число операторов PRINT, необходимых для вывода изображения, можно сократить, если информацию о позициях символа закодировать в операторе DATA. Существует несколько способов кодирования печатаемой строки символов. Например, каждую строку изображения можно представить парой чисел. Первое число указывает начальную позицию печати цепочки символов в строке, а второе число определяет длину этой цепочки.

Рассмотрим теперь некоторые специальные операторы, используемые в языке БЕЙСИК для построения различных рисунков и графиков.

Перед выводом изображения экран дисплея полезно очистить от «мусора», т. е. от различных записей, которые всегда появляются на экране при вводе программы в компьютер. Очистку экрана осуществляют с помощью оператора CLS.

Для проведения на экране прямой линии используется оператор

```
LINE ( X1 , Y1 ) - ( X2 , Y2 )
```

Здесь X1, Y1 — координаты начальной точки отрезка, X2, Y2 — координаты конечной точки.

За начало координат во многих персональных компьютерах принят верхний левый угол экрана. Ось X направлена горизонтально вправо, а ось Y — вертикально вниз.

Рассмотрим программу

```
10 REM ПУЧОК ЛИНИЙ  
20 CLS  
30 LINE (60,100)-(80,150)  
40 LINE (60,100)-(40,120)  
50 LINE (60,100)-(250,25)
```

Оператор, стоящий в строке 20, очищает экран. Последующие три оператора строят пучок прямых линий, исходящих из одной и той же точки с координатами (60, 100).

На экране дисплея можно построить окружность. Для этого служит оператор

```
CIRCLE ( X , Y ) , R
```

Здесь X, Y — координаты центра окружности; R — радиус.

С помощью программы

```
10 REM ОКРУЖНОСТИ
20 CLS
30 FOR R=10 TO 70 STEP 20
40 CIRCLE (150,120),R
50 NEXT R
```

будут построены на экране дисплея концентрические окружности с радиусами 10, 30, 50 и 70.

Заметим, что X, Y и R ограничены размерами экрана. Например, значение X обычно лежит в интервале (0,255), а Y — в интервале (0,240).

С помощью оператора

```
LINE ( X1 , Y1 ) - ( X2 , Y2 ) , , B
```

вычерчивается на экране контур прямоугольника, причем (X1, Y1) и (X2, Y2) являются координатами концов одной из его диагоналей.

Оператор

```
LINE ( X1 , Y1 ) - ( X2 , Y2 ) , , BF
```

вычерчивает на экране *закрашенный* прямоугольник.

Мы рассмотрели только некоторые наиболее простые и часто применяемые операторы машинной графики языка БЕЙСИК, но и они показывают значительные возможности персонального компьютера в области отображения на дисплее графической информации.

МНОЖЕСТВО АЛГОРИТМИЧЕСКИХ ЯЗЫКОВ

Сегодня существует свыше 500 (!) различных алгоритмических языков для записи программ. Еще больше вариантов (версий) этих языков. Однако только десяток языков послужил основой для разработки почти всего программного обеспечения ЭВМ.

В 1954—1957 гг. был разработан первый алгоритмический язык высокого уровня — ФОРТРАН. Его название происходит от сокращения английских слов Formula Translation — перевод формул. До сих пор этот язык не утратил своей популярности. На нем написано очень большое количество программ для выпол-

нения научных и инженерных расчетов. Благодаря этому Фортран еще долгое время будет активно использоваться.

Вслед за Фортраном широкое распространение получил АЛГОЛ-60 (от английских слов Algorithmic Language — алгоритмический язык). В настоящее время он практически не используется, однако его влияние на развитие языков программирования не вызывает сомнения. Сегодня существует большое семейство алголоподобных языков. Столь значителен был успех разработчиков АЛГОЛА-60.

Язык БЕЙСИК, разработанный в 1965 году для учебных целей, базируется на использовании *интерпретатора*, т. е. специальной программы, переводящей операторы в машинные коды. Как только ЭВМ встречает законченный оператор, она тут же переводит его в коды, выполняет и выдает результат. БЕЙСИК является одним из основных алгоритмических языков персональных ЭВМ.

Программирование в области экономики, где главная задача состоит в обработке данных с разнообразными структурами, обусловило появление в 1960 г. языка КОБОЛ. Разработка и успешное использование КОБОЛа продемонстрировали возможности языков программирования имитировать естественный язык профессиональной деятельности экономистов.

Одним из распространенных языков программирования является ПЛ/1, аккумулировавший в себя возможности ФОРТРАНа, АЛГОЛа и КОБОЛа. ПЛ/1 используется в основном для средних и больших ЭВМ, установленных на вычислительных центрах.

В 1970 г. для обучения информатике был разработан алгоритмический язык ПАСКАЛЬ. Он необычайно быстро получил признание программистов разных стран. Распространению ПАСКАЛЯ способствовала относительная простота конструкций языка и, пожалуй, в наибольшей степени — возможность организации составных типов данных на основе простых. На базе ПАСКАЛЯ было разработано несколько более сложных языков, в том числе алгоритмический язык АДА.

Особое место среди языков программирования занимает ЛИСП, разработанный в конце 1950-х годов. Он до сих пор остается одним из самых эффективных языков для обработки данных. Название LISP происходит от двух английских слов List Processing — обработка списков. Для решения вычислительных задач ЛИСП не очень удобен.

В последние годы получил распространение алгоритмический язык ПРОЛОГ, используемый для решения задач *искусственного интеллекта* на ЭВМ, оперирующих с банками данных и базами знаний.

Итак, имеется много разных языков программирования, и их число постоянно увеличивается. Однако не существует самого лучшего алгоритмического языка.

ОПЕРАЦИОННЫЕ СИСТЕМЫ

Программа для ЭВМ реализуется не сама по себе, а в определенной вычислительной среде, основную часть которой образует *операционная система*. Первые операционные системы появились в конце 1940-х годов и представляли собой наборы простых процедур, обеспечивающих ввод и вывод информации. В то время их даже не отличали от простых библиотечных программ.

В середине 1950-х годов одновременно с повышением быстродействия ЭВМ возникла необходимость сократить интервалы времени между окончанием работы одной программы и началом работы другой. Операционные системы того времени обеспечивали исполнение программ одну за другой — как говорят, в *пакетном режиме*. Кроме того, они обеспечивали доступ к внешней памяти ЭВМ.

В 1960-х годах появились операционные системы с *разделением времени*. В основу этого режима был положен принцип *прерывания* вычислительных процессов. Процессор переключался с выполнения одной программы на другую. При этом у пользователей создавалось впечатление одновременности работы на ЭВМ.

В те же годы началось широкое внедрение ЭВМ в *управление* технологическими процессами. Для решения этих задач потребовались новые качества операционных систем, в частности, способность реагировать на внешние *прерывания* при наличии жестких ограничений, накладываемых на время выполнения вычислений. Эти системы получили название операционных систем *реального времени*.

Современные операционные системы — это организованные совокупности программ, которые действуют как *интерфейсы* (средства общения) между аппаратурой ЭВМ и пользователями. Существенно расширились функции операционных систем. Они пополнились набором средств для облегчения программирования, компоновки, отладки и сопровождения программ, а также для распределения ресурсов ЭВМ при выполнении вычислений.

В 1980-х годах наступил период, когда в развитии операционных систем на одно из первых мест выдвинулась проблема эффективности интерфейса пользователя. Все, что выполняет ЭВМ, человек воспринимает через интерфейс пользователя, который выполняет роль посредника между человеком и реализуемой программой.

Для персональных ЭВМ наиболее удобными оказываются *диалоговые системы* поддержки решений, в которых имеется богатый набор программ как вычислительного, так и вспомогательного характера (например, для организации диалога).

Сейчас наиболее популярны такие приемы в построении интерфейса, как *окна, меню и таблицы*. Разделив экран дисплея на отдельные участки, называемые окнами, можно выводить на экран

информацию о ряде одновременно выполняемых процессов. Меню — это выдача на экран всех возможных продолжений работы компьютера, из которых пользователь выбирает на каждом шаге единственное.

Удобно представление информации в виде таблиц. Оно зачастую более наглядно, чем обычный текстовый диалог, поскольку открывается возможность охватить несколько объектов в их взаимосвязи. К тому же при поиске информации в памяти ЭВМ отпадает необходимость набирать на клавиатуре дисплея подробный запрос, что существенно снижает вероятность ошибок.

Вообще говоря, способ организации общения человека с компьютером в наибольшей степени влияет на эффективность использования вычислительной техники. Чем больше услуг предоставляет интерфейс, тем вероятнее, что любой человек — как любитель, так и профессионал — сможет выбрать наиболее удобную для себя форму общения с ЭВМ.

КОМПЬЮТЕР УЧАСТВУЕТ В ЭКСПЕРИМЕНТЕ**ЭВМ УПРАВЛЯЕТ ЭКСПЕРИМЕНТОМ**

Эксперимент всегда служил основой научных достижений в области химии и химической технологии. Идет время, усложняются эксперименты, возрастает объем информации, который получается в результате их проведения. Исследователям все труднее становится анализировать данные в ходе эксперимента и управлять сложными экспериментальными установками. Эти обстоятельства обусловили необходимость создания *автоматизированных систем научных исследований* (АСНИ). В основе работы каждой такой системы лежит применение ЭВМ, однако принципы использования компьютера могут быть различными.

В наиболее простых случаях во время проведения эксперимента осуществляется только сбор информации путем измерения и записи значений различных физических величин. Обработка и анализ данных осуществляются после завершения эксперимента. Такие системы могут быть использованы только для не слишком быстро протекающих процессов. Кроме того, управление экспериментом исследователь обычно осуществляет при этом вручную.

Встречаются и более сложные условия проведения эксперимента. Во-первых, часто химические процессы протекают быстро. Во-вторых, обработку экспериментальных данных требуется проводить в темпе проведения опытов. Наконец, необходимо автоматически управлять ходом эксперимента. Для решения этих задач требуется автоматизированная система, управляемая компьютером.

Если расчеты предшествуют эксперименту или осуществляются после его проведения с целью обработки полученных данных, то мощность современных ЭВМ позволяет надеяться, что с этими расчетами можно справиться, используя традиционные методы организации и управления вычислительными процессами. Однако при проведении вычислений с помощью ЭВМ в темпе эксперимента появляются специфические проблемы в управлении вычислениями и построении соответствующих программ. Возникающие здесь задачи составляют содержание большого раздела современного программирования — программирования в *реальном времени*.



Рис. 13. Схема обработки данных и управления экспериментом

Наибольшей эффективности эксперимента можно достигнуть, если обрабатывать данные в темпе их поступления при проведении опытов. Высокая производительность ЭВМ позволяет надеяться, что все особенности изучаемого явления не останутся незамеченными, а в случае отклонения процесса от заранее запланированного режима удастся вовремя делать нужные коррективы. Управляющие сигналы вырабатывает одна из программ, записанных в память ЭВМ. Таким образом, компьютер оказывается включенным в *контур управления* экспериментом.

В общем виде схема обработки данных и управления экспериментом приведена на рис. 13. От датчиков экспериментальной установки сигналы различной физической природы в определенные моменты времени, задаваемые *устройством опроса* датчиков, поступают на вход *преобразователя* сигнал — код. Такой преобразователь называют аналого-цифровым. На его выходных регистрах формируются численные значения соответствующих физических величин в виде цифровых записей, пригодных для обработки на ЭВМ. С помощью каналов связи вычислительной машины эти данные поступают на обработку в оперативную память. Измерения, относящиеся к одному и тому же моменту опроса датчиков, группируются в *один кадр*. Каналы связи ЭВМ обладают особенностью, в силу которой выгодно как можно большее число кадров объединить в *единый блок* для передачи в оперативную память. Это объясняется тем, что непосредственно пересылка данных осуществляется за весьма короткое время, необходимое для прохождения электрическими сигналами расстояния от выходных регистров аналого-цифровых преобразователей до регистров оператив-

ной памяти. Значительно большее время необходимо на размещение данных в памяти ЭВМ.

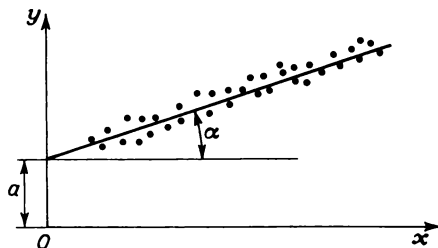
Если центральный процессор ЭВМ в момент поступления блока данных выполнял команды некоторой программы, то осуществляется *прерывание* этих вычислений, производится запоминание состояния прерванной программы для последующего продолжения ее работы. Процессор предоставляется в распоряжение программы, контролирующей прием новой порции данных. Чтобы уменьшить затраты машинного времени на связь ЭВМ с объектом исследований, сеансы связи выгодно делать реже, а для этого следует укрупнять блоки данных, собирая *кадры*, относящиеся к разным моментам времени в единую упорядоченную группу.

Таким образом, даже просто прием данных эксперимента требует *согласования* работы ЭВМ с процессами, протекающими в исследуемом объекте. Действия вычислительной системы должны быть строго согласованы с процессом выработки данных экспериментальной установкой. Поступившие в оперативную память ЭВМ данные должны быть обработаны программами, вычисляющими различные характеристики изучаемого в эксперименте явления. Некоторые результаты этих вычислений должны оперативно выводиться на устройства отображения — алфавитно-цифровые и графические дисплеи, печатающие устройства; другие — запоминаться для последующего, более тщательного анализа; третьи — использоваться в качестве входных значений для иных программ, предназначенных для вычисления новых характеристик. Следовательно, действия вычислительной системы должны быть *синхронизированы* с внешними процессами, т. е. ЭВМ должна работать в режиме реального времени. Даже незначительная задержка в выдаче некоторой команды управления будет равносильна *отказу* системы в работе, поэтому *надежность* — главное требование при создании систем реального времени. Она обеспечивается за счет повышения надежности аппаратных средств, а также соответствующих алгоритмов и программ.

ЛИНЕЙНАЯ РЕГРЕССИЯ

Термин *регрессия* ввел в употребление Ф. Гальтон — один из создателей математической статистики. Сопоставляя рост детей и их родителей, он обнаружил, что соответствие между ростом отцов и детей выражено слабо. Оно оказалось меньшим, чем он ожидал. Гальтон объяснил это явление наследственностью не только от родителей, но и от более отдаленных предков. По его предположениям, рост определяется наполовину родителями, на четверть — дедушкой и бабушкой, на одну восьмую — прадедом и прабабушкой и т. д. Мы не беремся судить, насколько был прав Гальтон. Главное в том, что он обратил внимание на движение *назад* по генеалогическому дереву и назвал его регрессией, заимствовав поня-

Рис. 14. Зависимость между измеряемыми величинами



тие движения назад, противоположное прогрессу — движению вперед. Впоследствии слово «регрессия» заняло в статистике заметное место, хотя, как это часто бывает в любом языке, в том числе и в языке науки, в него теперь вкладывают другой смысл — оно означает *статистическую связь* между случайными величинами.

При обработке экспериментальных данных химики часто строят различные графики, пользуясь декартовой (прямоугольной) системой координат (рис. 14). Иногда зависимости между измеряемыми величинами x и y носят *линейный* характер, поэтому экспериментальные точки группируются около некоторой прямой линии. Уравнение прямой линии имеет вид

$$y = a + vx.$$

Здесь a — длина отрезка от начала координат до точки пересечения прямой с осью y ; b — тангенс угла наклона α прямой к оси x ($b = \operatorname{tg} \alpha$).

Прямую линию стараются провести так, чтобы сумма квадратов отклонений расчетных значений y_p от экспериментальных значений y_s была минимальной для всех n рассматриваемых опытов:

$$s = \sum_{i=1}^n (y_p - y_s)^2 \rightarrow \min.$$

Математически строго доказано, что этому условию удовлетворяют коэффициенты уравнения, вычисляемые по следующим формулам:

$$a = \frac{s_3 s_4 - s_5 s_2}{s_1 s_4 - s_2 s_2}; \quad b = \frac{s_1 s_5 - s_2 s_3}{s_1 s_4 - s_2 s_2}.$$

$$s_2 = \sum_{i=1}^n x_i; \quad s_3 = \sum_{i=1}^n y_i;$$

Здесь использованы обозначения: $S_1 = n$;

$$s_4 = \sum_{i=1}^n x_i^2; \quad s_5 = \sum_{i=1}^n y_i x_i.$$

Другими словами, n — это общее число опытов, в которых измерялись значения величин x и y ; s_2 характеризует сумму всех измеренных значений x , а s_3 — сумму всех значений величины y . Остальные обозначения понятны из формул.

Уравнение $y = a + bx$, в которое подставлены найденные значения коэффициентов, принято называть *уравнением линейной регрессии*. Пользуясь этим уравнением, можно, не проводя дальнейших опытов, *рассчитать* для заданного x соответствующее значение y . Таким образом, с помощью уравнения регрессии можно *прогнозировать* величину y . Разумеется, такой прогноз не будет абсолютно точным. Близость прогнозируемого значения y к фактическому значению зависит в основном от точности, с которой выполняется эксперимент, и от того, насколько существующая зависимость между y и x близка в действительности к линейной.

Расчеты коэффициентов линейной регрессии всегда выполняются по одним и тем же формулам и требуют при ручном счете значительных затрат времени. Чтобы избавить исследователей от этой рутинной работы, применяют персональный компьютер. Вот текст программы на языке БЕЙСИК, выполняющей все необходимые операции:

```

10 CLS : REM ЛИНЕЙНАЯ РЕГРЕССИЯ
20 DIM X(50), Y(50)
30 INPUT " ВВЕДИТЕ ЧИСЛО ЭКСПЕРИМЕНТОВ "; N
40 FOR I=1 TO N
50 PRINT " ВВЕДИТЕ "; I, " ПАРУ ЗНАЧЕНИЙ X И Y "; INPUT X(I), Y(I)
60 NEXT I
70 PRINT " УРАВНЕНИЕ РЕГРЕССИИ"
80 LET S1=N : LET S2=0 : LET S3=0
90 LET S4=0 : LET S5=0
100 FOR I=1 TO N
110 LET S2=S2+X(I) : LET S3=S3+Y(I)
120 LET S4=S4+X(I)*X(I)
130 LET S5=S5+Y(I)*X(I)
140 NEXT I
150 LET D1=S1*S4-S2*S2
160 LET D2=S3*S4-S5*S2
170 LET D3=S1*S5-S2*S3
180 LET A=D2/D1 : LET B=D3/D1
190 PRINT " Y = "; A, " + "; B, " * X"
200 LET S=0
210 FOR I=1 TO N
220 LET S=S+(Y(I)-(A+B*X(I)))^2
230 NEXT I
240 LET S=SQR(S/(N-1))
250 PRINT " СРЕДНЕЕ КВАДРАТИЧНОЕ ОТКЛОНЕНИЕ "; USING "###.###"; S
260 STOP : END

```

По этой программе вычисляются не только коэффициенты уравнения регрессии, но и среднее квадратичное отклонение расчетных значений y от экспериментальных. Если сумму квадратов отклонений обозначить через s , то среднее квадратичное отклонение s_y вычисляется по формуле

$$s_y = \sqrt{\frac{s}{n-1}}.$$

Рассмотрим применение программы для выполнения практических расчетов.

Известно, что с повышением температуры растворимость солей в воде увеличивается. Напомним, что растворимостью называется концентрация соли в насыщенном растворе.

Экспериментальные данные по растворимости соли при различных температурах:

Температура (x), °C	10	20	30	40	50	60	70	80
Растворимость (y), г/л	42	50	62	67	75	86	92	103

Обработав экспериментальные данные с помощью компьютера, получаем уравнение регрессии $y = 33,71 + 0,85x$ и среднее квадратичное отклонение $s_y = 1,46$.

Программа оказалась достаточно простой и удобной для использования, но, к сожалению, область ее применения ограничена только линейными зависимостями. Однако нужно иметь в виду, что некоторые нелинейные зависимости можно преобразовать в линейные.

Например, гиперболическая связь $y = a_0 + a_1/x$ линеаризуется заменой переменной $z = 1/x$, тогда $y = a_0 + a_1z$.

Показательная связь $y = a_0 e^{a_1 x}$ линеаризуется путем логарифмирования: $v = \ln y = \ln a_0 + a_1 x$.

Степенная связь $y = a_0 x^{a_1}$ после логарифмирования линеаризуется заменой $\ln y = p$; $\ln a_0 = c$; $\ln x = z$; тогда $p = c + a_1 z$.

Логарифмическая связь $y = a_0 + a_1 \ln x$ линеаризуется заменой $z = \ln x$; тогда $y = a_0 + a_1 z$.

Комбинированная связь $y = 1/(a_0 + a_1 e^{-x})$ линеаризуется заменой $v = 1/y$ и $z = e^{-x}$; тогда $v = a_0 + a_1 z$.

Широкое использование линейных зависимостей и связей, легко приводимых к линейным, объясняется следующим. Линейные связи просты и требуют относительно малого объема вычислений, а методика их установления более глубоко разработана.

Рассмотрим пример линеаризации. Как известно, зависимость константы скорости химической реакции от температуры имеет экспоненциальный характер

$$k = k_0 e^{-\frac{E}{RT}}$$

Прологарифмировав это выражение, получим:

$$\ln k = \ln k_0 - \frac{E}{R} \frac{1}{T}$$

Введем следующие обозначения:

$$y = \ln k; \quad a = \ln k_0; \quad b = -\frac{E}{R}; \quad x = \frac{1}{T}$$

С учетом этих обозначений получаем линеаризованное уравнение регрессии $y = a + bx$. Теперь мы снова можем воспользоваться нашей программой. Вычислив коэффициенты a и b , можно затем рассчитать предэкспоненциальный множитель k_0 и энергию активации E .

Таким образом, благодаря линеаризации область применения программы значительно расширяется.

КОРРЕЛЯЦИЯ

Желая установить связь между двумя величинами, исследователь наносит их значения на график. Каждой паре значений величин x и y на графике соответствует точка. Обычно такие точки не ложатся на одну линию, а занимают на плоскости чертежа некоторую область, образуя так называемую *диаграмму рассеивания* (рис. 15). Причины рассеивания могут быть самыми различными. Среди них можно назвать погрешности измерений, влияние неучтенных факторов и т. д. Даже если один и тот же опыт в химической лаборатории проводится несколько раз, результаты измерений обычно отличаются друг от друга.

Мы знаем, что связь между двумя величинами, характеризующими химико-технологический процесс, можно выразить уравнением регрессии. Однако оно не позволяет с *абсолютной* точностью предсказать значение y , если известно значение x . Прогнозирование y осуществляется только с некоторой вероятностью. Такая связь между двумя величинами называется *корреляционной*. Она существенно отличается от связи *детерминистической*, описываемой строгими физико-химическими формулами.

Благодаря рассеянию, точки на графике располагаются в некоторой конечной области, обычно имеющей форму эллипса. Чем теснее связь между рассматриваемыми величинами, тем более вытянут эллипс вдоль одной из своих осей (рис. 15, *а* и *б*). Наоборот, если связь между x и y отсутствует, то разброс точек на диаграмме рассеивания по форме приближается к кругу (рис. 15, *в*).

Корреляционную связь между величинами называют *положительной*, если при увеличении одной из них возрастает другая величина. *Отрицательной* называют такую корреляционную связь, когда при увеличении значений одной из величин другая уменьшается.

Видя на графике корреляционную связь между величинами, мы можем дать ей *качественную* оценку. Для *количественной* оценки линейной корреляции используется *коэффициент парной корреляции*

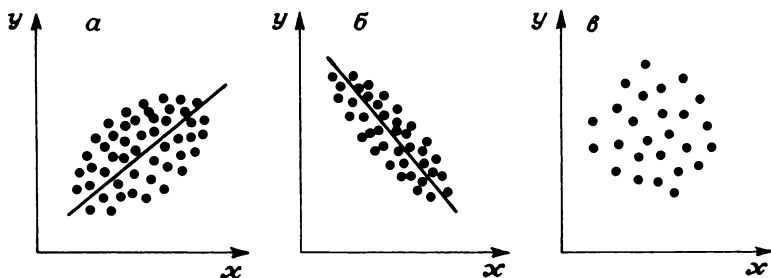


Рис. 15. Диаграммы рассеивания

ции r_{xy} , который рассчитывается по формуле

$$r_{xy} = \frac{s_1 s_5 - s_2 s_3}{\sqrt{(s_1 s_4 - s_2^2) (s_1 s_6 - s_3^2)}}$$

где $s_1 = n$; $s_2 = \sum_{i=1}^n x_i$; $s_3 = \sum_{i=1}^n y_i$; $s_4 = \sum_{i=1}^n x_i^2$; $s_5 = \sum_{i=1}^n x_i y_i$; $s_6 = \sum_{i=1}^n y_i^2$.

Этот коэффициент может принимать следующие значения:

1) $r_{xy} = 0$; это свидетельствует об отсутствии корреляционной связи между x и y ;

2) $r_{xy} = 1$; в данном случае существует строгая положительная детерминистическая связь;

3) $r_{xy} = -1$; между x и y существует строгая отрицательная детерминистическая связь;

4) $-1 < r_{xy} < +1$; это наиболее распространенный случай: корреляционная связь может быть как положительной, так и отрицательной и характеризоваться различной *степенью тесноты связи*.

Расчеты коэффициентов парной корреляции полезно выполнять на персональном компьютере, пользуясь следующей программой:

```

10 CLS : REM КОЭФФИЦИЕНТ ПАРНОЙ КОРРЕЛЯЦИИ
20 DIM X(50),Y(50)
30 INPUT "ВВЕДИТЕ КОЛИЧЕСТВО ПАР X И Y ";N
40 FOR I=1 TO N
50 PRINT "ВВЕДИТЕ ";I;" ПАРУ ЗНАЧЕНИЙ X И Y "; : INPUT X(I),Y(I)
60 NEXT I
70 PRINT "КОЭФФИЦИЕНТ ПАРНОЙ КОРРЕЛЯЦИИ = ";
80 LET S1=N : LET S2=0 : LET S3=0
90 LET S4=0 : LET S5=0 : LET S6=0
100 FOR I=1 TO N
110 LET S2=S2+X(I) : LET S3=S3+Y(I)
120 LET S4=S4+X(I)*X(I)
130 LET S5=S5+X(I)*Y(I)
140 LET S6=S6+Y(I)*Y(I)
150 NEXT I
160 LET A=S1*S5-S2*S3
170 LET B=(S1*S4-S2*S2)*(S1*S6-S3*S3)
180 LET R=A/SQR(B) : PRINT USING "###.####" ;R
190 STOP

```

Чем ближе абсолютное значение коэффициента корреляции $|r_{xy}|$ к единице, тем сильнее линейная связь между x и y . Следует отметить, что r_{xy} одновременно отражает степень случайности и криволинейности связи между величинами x и y . Например, зависимость y от x может быть близкой к функциональной, но *существенно нелинейной*. В этом случае коэффициент корреляции будет значительно меньше единицы.

Таким образом, тесноту связи между двумя величинами можно объективно оценить только после рассмотрения графика «корреляционного поля» и вычисления коэффициента парной корреляции.

КАКАЯ ФОРМУЛА ЛУЧШЕ!

Многие зависимости между величинами, характеризующими химико-технологические процессы, носят *нелинейный* характер. Достаточно напомнить, что константа скорости реакции экспоненциально возрастает с увеличением температуры. С целью определения вида зависимости между экспериментально изучаемыми величинами следует на основе результатов опыта построить *эмпирическую линию регрессии* (рис. 16). Для этого весь диапазон изменения величины x разбивают на k равных интервалов длиной Δx . Все точки, попавшие в интервал с номером j , относят к его середине x_j .

Для каждого интервала j вычисляют среднее арифметическое значение y_j

$$\bar{y}_j = \frac{1}{n_j} \sum_{i=1}^{n_j} y_{j,i},$$

где n_j — число точек в интервале с номером j .

Затем последовательно соединяют точки с координатами (x_j, \bar{y}_j) отрезками прямых линий. Полученная ломаная линия — это эмпирическая линия регрессии. По виду этой линии можно подобрать соответствующую формулу, например одну из тех, которые приведены в табл. 8.

Задачу определения коэффициентов, входящих в уравнение регрессии, решают обычно с помощью ЭВМ, применяя *метод наименьших квадратов*. Сущность метода заключается в том, что путем варьирования коэффициентов подбираются такие их значения, которые *минимизируют* сумму квадратов отклонений расчетных значений y от экспериментальных.

Качество всякой проделанной работы принято оценивать по конечному результату. Так и качество уравнения регрессии оценивают по его *прогнозирующей способности*. Уравнение хорошо прогнозирует экспериментальные данные, т. е. уравнение адекватно

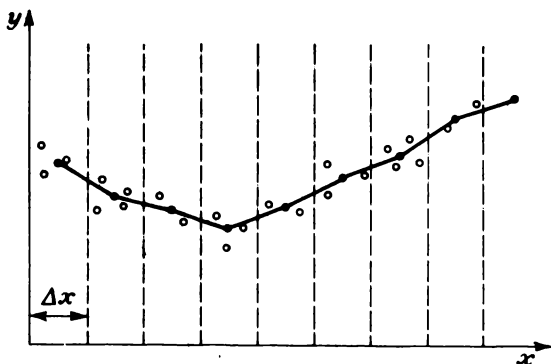


Рис. 16. Эмпирическая линия регрессии

Т а б л и ц а 8. Коэффициенты расчетных зависимостей, полученные методом наименьших квадратов

Вид зависимости	Уравнение регрессии	Выражение для определения коэффициентов
Линейная	$y = a_0 + a_1 x$	$a_1 = \frac{\sum x_i \sum y_i - n \sum x_i y_i}{(\sum x_i)^2 - n \sum x_i^2};$ $a_0 = \frac{1}{n} (\sum y_i - a_1 \sum x_i)$
Гиперболическая	$y = a_0 + \frac{a_1}{x}$	$a_1 = \frac{n \sum \frac{y_i}{x_i} - \sum \frac{1}{x_i} \sum y_i}{n \sum \frac{1}{x_i^2} - (\sum \frac{1}{x_i})^2};$ $a_0 = \frac{1}{n} (\sum y_i - a_1 \sum \frac{1}{x_i})$
Степенная	$y = a_0 x^{a_1}$	$a_1 = \frac{\sum \ln x_i \sum \ln y_i - n \sum \ln x_i \ln y_i}{(\sum \ln x_i)^2 - n (\ln x_i)^2};$ $a_0 = \exp \left[\frac{1}{n} (\sum \ln y_i - a_1 \sum \ln x_i) \right]$
Показательная	$y = 10^{a_0 + a_1 x}$	$a_1 = \frac{n \sum x_i \lg y_i - \sum x_i \sum \lg y_i}{n \sum x_i^2 - (\sum x_i)^2};$ $a_0 = \frac{1}{n} (\sum \lg y_i - a_1 \sum x_i)$
	$y = \exp (a'_0 + a'_1 x)$	Для поиска a'_0 и a'_1 в формулах заменить \lg на \ln
Экспоненциальная	$y = a_0 e^{a_1 x}$	$a_1 = \frac{\sum x_i \sum \ln y_i - n \sum x_i \ln y_i}{(\sum x_i)^2 - n \sum x_i^2};$ $a_0 = \exp \left[\frac{1}{n} (\sum \ln y_i - a_1 \sum x_i) \right]$
Показательная	$y = a_0 a_1^x$	$\lg a_1 = \frac{n \sum x_i \lg y_i - \sum x_i \sum \lg y_i}{n \sum x_i^2 - (\sum x_i)^2};$ $\lg a_0 = \frac{1}{n} (\sum \lg y_i - \lg a_1 \sum x_i)$

Примечание. Во всех формулах знак \sum эквивалентен $\sum_{i=1}^n$.

описывает эти данные, если расхождения между расчетными и экспериментальными значениями находятся в допустимых пределах.

Существует несколько способов проверки *адекватности* уравнения регрессии. Например, для этих целей можно использовать корреляционное отношение η и среднюю относительную ошибку прогнозирования ϵ (в %), которые рассчитываются по следующим формулам:

$$\eta = \sqrt{1 - \frac{\sum_{i=1}^n (y_{s,i} - y_{p,i})^2}{\sum_{i=1}^n (y_{s,i} - \bar{y})^2}}; \quad \epsilon = \frac{100}{n} \sum_{i=1}^n \left| \frac{y_{s,i} - y_{p,i}}{y_{s,i}} \right|,$$

где y_s , y_p и \bar{y} — экспериментальное, расчетное и среднее значение величины y ; n — количество экспериментальных значений y , использованных для проверки адекватности.

МНОГОФАКТОРНЫЕ ЗАВИСИМОСТИ

Производительность аппарата, в котором протекает химическая реакция, зависит от нескольких влияющих факторов, таких как температура, концентрации реагирующих веществ, давление и т. д. Ситуация, при которой одна величина зависит от нескольких величин, довольно типична для химической технологии. Поиск эмпирической формулы, которая связывает величину y с влияющими факторами x_1, \dots, x_n , в общем случае представляет собой достаточно сложную задачу. Разработано несколько методов ее решения и среди них можно назвать *метод Брандона*.

Сущность этого метода заключается в том, что нелинейную функцию y предлагается искать в виде произведения

$$y = \bar{y} f_1(x_1) f_2(x_2) \dots f_n(x_n),$$

где \bar{y} — среднее арифметическое среди измеренных значений функции y ; каждая из функций $f_i(x_i)$ считается функцией только *одного* аргумента с номером i .

Функция $f_i(x_i)$ подбирается в зависимости от характера влияния x_i на y . Она, например, может быть возрастающей или убывающей, линейной или нелинейной (степенной, показательной, экспоненциальной и т. д.).

Реализацию метода начинают с вычисления среднего арифметического значения y по всем m опытам:

$$\bar{y} = \frac{1}{m} \sum_{j=1}^m y_{s,j}.$$

Затем все экспериментальные значения делят на \bar{y} :

$$y_{1,j} = y_{s,j} / \bar{y}.$$

Таким образом, исходными данными для нахождения $f_1(x_1)$ служат $y_{1,j}$ и соответствующие $x_{1,j}$. Здесь j характеризует номер

опыта. Методику подбора эмпирических формул функций одной независимой переменной мы уже рассматривали.

Найдя $f_1(x_1)$, вычисляют *остаточный член* $y_{2,j}$ для результатов каждого опыта:

$$y_{2,j} = \frac{y_{1,j}}{f_1(x_{1,j})}.$$

Теперь предполагают, что $y_{2,j}$ уже не зависит от фактора x_1 , а зависит от x_2, x_3, \dots, x_n .

Исходными данными для нахождения функции $f_2(x_2)$ являются остаточный член $y_{2,j}$ и значения $x_{2,j}$ в каждом опыте. Получив формулу $f_2(x_2)$, вычисляют следующий остаточный член:

$$y_{3,j} = \frac{y_{2,j}}{f_2(x_{2,j})}.$$

Так постепенно, шаг за шагом, с помощью компьютера находят все нужные функции: $f_1(x_1), \dots, f_n(x_n)$. Остается только подставить их в конечную формулу. После этого полагается проверить адекватность формулы экспериментальным данным.

Покажем применение метода Брандона для математического описания процесса газификации угля водяным паром в среде расплава хлорида натрия. Факторами, влияющими на этот технологический процесс, являются: T — температура, К; d — средний размер гранул угля, мм; H — высота слоя расплава, мм; C — объемная концентрация водяного пара в дутье, %. Выходные параметры технологического процесса: y_1 — выход газа при газификации, м³/кг; y_2 — объемное содержание CO + H₂ в отходящем газе, %.

Исходные экспериментальные данные приведены в табл. 9.

Математическое описание процесса газификации угля будем строить в виде двух эмпирических уравнений:

$$y_1 = \bar{y}_1 f_1(T) f_2(d) f_3(H) f_4(C);$$

$$y_2 = \bar{y}_2 f_5(T) f_6(d) f_7(H) f_8(C).$$

Расчет средних значений ведется по формулам

$$\bar{y}_1 = \frac{1}{m} \sum_{j=1}^m y_{1,j}; \quad \bar{y}_2 = \frac{1}{m} \sum_{j=1}^m y_{2,j}.$$

В табл. 9 отражены результаты 12 опытов, поэтому $m=12$.

Выполнив вычисления, находим: $\bar{y}_1 = 1,686$; $\bar{y}_2 = 73,87$.

После завершения всех вычислений получаем два уравнения:

$$y_1 = 1,686 (-2,449 + 0,299C - 0,495 \cdot 10^{-2}C^2 + 0,33 \cdot 10^{-4}C^3) \cdot 0,02694 \exp(0,0034T) \times (0,272 + 0,7728 \cdot 10^{-2}H) (1,148 - 0,1342d);$$

$$y_2 = 73,87 (-13,46 + 0,02716T - 0,127 \cdot 10^{-4}T^2) (1,006 - 0,1168 \cdot 10^{-3}C).$$

В последнее уравнение не вошли сомножители, учитывающие влияние высоты слоя расплава H и среднего размера гранул угля d . Это сделано потому, что влияние этих факторов на y_2 оказалось незначительным.

Расчетные значения y_1 и y_2 , найденные с помощью полученных уравнений регрессии, представлены в табл. 9.

Т а б л и ц а 9. Исходные экспериментальные данные и результаты расчетов по методу Брандона

Номер опыта	Влияющие факторы				Выходные параметры			
	Т, К	d, мм	H, мм	C, %	экспериментальные		расчетные	
					y_1 , м ³ /кг	y_2 , %	y_1 , м ³ /кг	y_2 , %
1	1073	1,06	50	36,7	1,13	81,3	1,101	78,795
2	1073	1,06	100	23,7	1,09	80,6	1,208	78,614
3	1073	1,06	130	36,7	2,14	78,8	2,135	78,496
4	1073	1,06	100	82,5	2,69	78,1	2,506	78,07
5	1073	2,45	100	23,7	0,95	78,1	0,983	78,614
6	1073	1,06	100	36,7	1,81	77,7	1,748	78,495
7	1003	1,06	100	82,5	1,88	76,7	1,977	73,984
8	1073	0,16	100	23,7	1,30	76,6	1,353	78,014
9	1073	1,06	75	36,7	1,37	76,1	1,423	78,495
10	1123	1,06	100	82,5	3,27	73,7	2,972	76,39
11	1003	1,06	100	66,7	1,13	68,6	0,956	72,121
12	943	1,06	100	82,5	1,47	64,2	1,612	63,179

Для количественной оценки адекватности уравнений были рассчитаны корреляционное отношение η и средняя относительная ошибка ϵ :

	η , доли	ϵ , %
Выход газа при газификации \bar{y}_1 , м ³ /кг	0,9813	0,8213
Объемное содержание смеси CO + H ₂ в отходящем газе y_2 , %	0,8721	2,4385

Рассмотрение оценок адекватности свидетельствует о достаточно высоком качестве математического описания. Так, средняя относительная погрешность расчета выхода газа составляет 0,82 %, а объемное содержание смеси CO + H₂ в отходящем газе — примерно 2,44 %. Следовательно, наши уравнения можно уверенно использовать для расчета параметров процесса газификации. Однако следует иметь в виду, что область адекватности математического описания лежит только в диапазонах изменения влияющих факторов, представленных в таблице исходных экспериментальных данных:

$$943 \leq T \leq 1073; \quad 0,16 \leq d \leq 2,45; \\ 50 \leq H \leq 130; \quad 23,7 \leq C \leq 82,5.$$

За пределами этих диапазонов прогнозирование параметров процесса возможно, но чем дальше мы будем отходить от границ указанных параметров, тем больше может быть ошибка прогноза.

Метод Брандона полезно применять в тех случаях, когда нужно быстро и достаточно точно описать сложный химико-технологический процесс на основе полученных экспериментальных данных. При этом не ставится задача отразить химические реакции, лежащие в основе процесса. Необходимо лишь связать

в виде уравнений регрессии влияющие факторы с выходными показателями.

Использование метода Брандона требует значительного объема вычислений, который может быть выполнен в разумные сроки только с применением ЭВМ.

СГЛАЖИВАНИЕ

При проведении некоторых экспериментальных исследований наблюдаются частые и резкие изменения значений измеряемых величин. Это может создавать определенные трудности в изучении химико-технологических процессов и управлении ими. Борются с ними путем *сглаживания*.

При расчете сглаженного значения используют значения измеряемой величины в соседних точках. Пусть, например, имеется массив значений температуры T , которые измерены через равные промежутки времени:

70,2;	70,7;	71,8;	72,3;	73,4;	74,3;
75,3;	75,8;	76,7;	78,1;	78,7;	80,1

Сглаженное значение $Y(1)$ в первой точке принимается равным $T(1)$, т. е. в нашем случае $Y(1) = T(1) = 70,2$ °С. Аналогично поступают с последней точкой, полагая, что $Y(N) = T(N)$.

Сглаженное значение $Y(2)$ рассчитывают по формуле

$$Y(2) = \frac{1}{5} (T(1) + 3T(2) + T(3)).$$

Здесь множитель 3 называется *весовым коэффициентом*. Он как бы усиливает роль значения $T(2)$ в процессе сглаживания. По аналогичной формуле рассчитывается значение $Y(N-1)$:

$$Y(N-1) = \frac{1}{5} (T(N-2) + 3T(N-1) + T(N)).$$

Во всех остальных точках расчет сглаженных значений осуществляется по формуле

$$Y(I) = \frac{1}{15} (T(I-2) + 3T(I-1) + 7T(I) + 3T(I+1) + T(I+2)).$$

Весовые коэффициенты 3 и 7 характеризуют вклад отдельных слагаемых при вычислении сглаженного значения функции. Текст программы приводится ниже:

```
10 CLS : REM СГЛАЖИВАНИЕ
15 DIM T(12), Y(12)
20 DATA 70.2, 70.7, 71.8, 72.3, 73.4, 74.3
25 DATA 75.3, 75.8, 76.7, 78.1, 78.7, 80.1
30 LET N=12
35 FOR I=1 TO N : READ T(I) : NEXT I
```

```

40 LET Y(1)=T(1) ; LET Y(N)=T(N)
45 LET Y(2)=(T(1)+3*T(2)+T(3))/5
50 LET Y(N-1)=(T(N-2)+3*T(N-1)+T(N))/5
55 FOR I=3 TO (N-2)
60 LET A=3*T(I-1)+7*T(I)+3*T(I+1)
65 LET Y(I)=(T(I-2)+A+T(I+2))/15
70 NEXT I
75 FOR I=1 TO N
80 PRINT USING "#####.###" ; I,T(I),Y(I)
85 NEXT I
90 STOP

```

Метод сглаживания получил довольно широкое распространение в экспериментальных исследованиях.

В рассмотренном нами примере учитываются по два соседних значения с каждой стороны с весовыми коэффициентами 1, 3, 7, 3 и 1. В случае необходимости количество учитываемых соседних точек и весовые коэффициенты можно легко изменить.

Отметим, однако, что применять сглаживание нужно осмотрительно. В некоторых особых случаях резко выделяющиеся точки могут характеризовать существенные *качественные изменения*, происходящие в исследуемом объекте. Сглаживая экспериментальные данные, мы можем легко утратить информацию о таких явлениях.

ГРАФИК И ГИСТОГРАММА

Компьютер может не только осуществить обработку результатов эксперимента, но и вывести их на печать в виде *графика*. В качестве примера составим программу для вычисления функции $y = x^2 + 5$ и построения ее графика. Значения аргумента x будем изменять от -5 до $+5$ с шагом 1. Расчетные точки на графике будем печатать символом «*», используя функцию TAB. Вот текст программы:

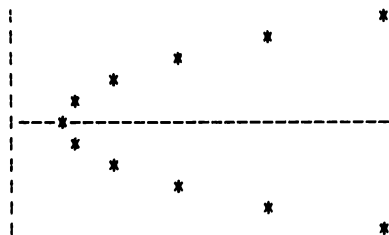
```

10 CLS ; REM ГРАФИК ФУНКЦИИ Y=X^2+5
20 LET X=-5
30 IF X<>0 GOTO 80
40 PRINT "I";
50 FOR I=1 TO 3 ; PRINT "-"; ; NEXT I ; PRINT "*";
60 FOR I=1 TO 25 ; PRINT "-"; ; NEXT I ; PRINT
70 GOTO 90
80 PRINT "I"; TAB(X*X+5); "*"
90 LET X=X+1
100 IF X<=5 GOTO 30
110 STOP

```

Для печати оси абсцисс мы использовали символ «!». Ось ординат печатается символами «—». При составлении программы мы имели в виду, что аргумент функции TAB может изменяться только в диапазоне от 0 до 71. График функции представлен

Рис. 17. График функции $y = x^2 + 5$



на рис. 17. Если значения функции выходят за пределы указанного диапазона, то необходимо вводить масштаб. Он выбирается с таким расчетом, чтобы минимальное и максимальное значения функции изображались точками, не выходящими за пределы экрана дисплея.

Гистограмма — это один из способов наглядного представления числовых значений величин, дающий возможность их визуального сравнения. С помощью персонального компьютера каждое числовое значение можно изобразить в определенном масштабе в виде отрезка горизонтальной линии, построенной из каких-либо символов, например звездочек. Последовательные значения выводятся на последовательные строки, начиная с одной и той же позиции.

В качестве примера рассмотрим программу для построения гистограммы выпуска продукции цехом химического завода по месяцам в течение года, а также среднемесячной производительности. Ниже приведен текст программы:

```

10 CLS : REM ГИСТОГРАММА
15 LET S=0 : LET N=0
20 DATA 1,28,2,32,3,25,4,37,5,40
30 DATA 6,35,7,26,8,37,9,41,10,32
40 DATA 11,36,12,40,0
50 READ A
60 IF A=0 THEN GOTO 110
70 READ Y
75 IF A<10 THEN PRINT " ";
80 PRINT A; " ***";
90 GOSUB 2000
100 LET S=S+Y : LET N=N+1 : GOTO 50
110 LET S=S/N
120 PRINT " CP. ***";
130 LET Y=S : GOSUB 2000
140 STOP
2000 LET C=20
2010 IF C>=Y GOTO 2050
2020 LET C=C+1
2030 PRINT "*";
2040 GOTO 2010
2050 PRINT TAB(35) : PRINT USING "##.##"; Y
2060 RETURN
2070 END

```

Построенная гистограмма представлена на рис. 18.

1	*****	28.00
2	*****	32.00
3	*****	25.00
4	*****	37.00
5	*****	40.00
6	*****	35.00
7	*****	26.00
8	*****	37.00
9	*****	41.00
10	*****	32.00
11	*****	36.00
12	*****	40.00
СР.	*****	34.00

Рис. 18. Гистограмма выпуска продукции

ЗАЧЕМ ПЛАНИРУЮТ ЭКСПЕРИМЕНТ!

Среди 1093 патентов, выданных Бюро патентов США изобретателю Т. А. Эдисону, был патент № 223898 от 27 января 1880 года на лампу с угольной нитью накаливания. Три тысячи человек прибыли посмотреть на специальных поездах, заказанных Эдисоном, на сотни электрических лампочек, развешанных в его мастерской и над окрестными дорогами в Менло Парк (штат Нью-Джерси). Это было в последний день 1879 года. Но прежде чем была проведена эта победная демонстрация, Эдисон перепробовал шесть тысяч веществ, содержащих углерод, — от обыкновенных швейных ниток, покрытых углем, до продуктов питания и смол.

Исследования шести тысяч веществ — это десятки тысяч опытов, и на эту титаническую работу ушло около двух лет. Думается, если бы Эдисон владел математической теорией эксперимента, то количество опытов можно было бы сократить, по-видимому, в несколько раз. Но во времена Эдисона еще не было такой теории.

Приступая к исследованиям в лаборатории, химики всегда намечают определенную программу действий, иначе говоря, *планируют эксперимент*. Цель всякого исследования заключается в получении определенного объема *новой информации* об изучаемом веществе или о процессе его получения. В ходе экспериментов используется дорогостоящее оборудование, реактивы, затрачивается труд ученых. Чтобы эксперимент шел оптимально, требуется свести к минимуму все затраты на его осуществление. Поэтому, составляя программу экспериментальных исследований, химики заранее планируют количество опытов и условия их проведения.

В 1950-х годах проблема оптимального планирования эксперимента привлекла внимание математиков. В результате их работ появились весьма экономичные планы экспериментов для нахождения уравнений регрессии, описывающих сложные многофакторные химико-технологические процессы. Постепенно развилась и

сформировалась *математическая теория планирования эксперимента*.

Применение компьютеров на различных стадиях планирования эксперимента и обработки экспериментальных данных позволяет избавить химиков от утомительных расчетов и исключить возможные ошибки вычислений.

Уравнения регрессии, полученные на основе планирования эксперимента, используются в дальнейшем для оптимизации технологических процессов или свойств получаемых материалов.

Нужно сказать, что первые попытки применить математические методы для оптимального планирования эксперимента были сделаны английским математиком Р. Фишером еще в начале 1920-х годов. Однако особенно быстрыми темпами теория планирования эксперимента стала развиваться после 1951 г. в связи с появлением работ Д. Бокса и К. Уилсона. Значительный вклад в развитие методов оптимального планирования эксперимента внесли советские ученые В. В. Налимов, Ю. П. Адлер, Г. К. Круг и др.

ПОВЕРХНОСТЬ ОТКЛИКА

Независимые переменные величины, влияющие на протекание химического процесса, принято называть *факторами*. Так, факторами могут быть температура, давление, состав исходной реакционной смеси и т. п. Эти величины при планировании эксперимента обычно обозначают малыми буквами x_1, \dots, x_n .

Протекание химико-технологического процесса количественно характеризуется одной или несколькими величинами, например производительностью оборудования, себестоимостью выпускаемой продукции и т. п. Такие величины в теории планирования эксперимента называют *функциями отклика* и обозначают буквами y_1, \dots, y_m . Функции отклика зависят от влияющих факторов.

Геометрический образ, соответствующий функции отклика, называют *поверхностью отклика* (рис. 19), а координатное пространство, по осям которого отложены факторы, — *факторным пространством*.

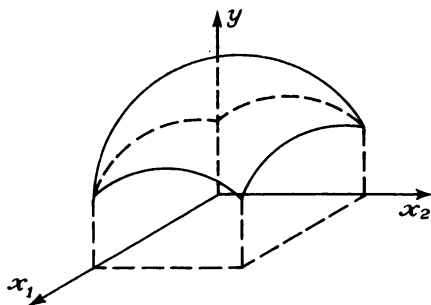


Рис. 19. Поверхность отклика

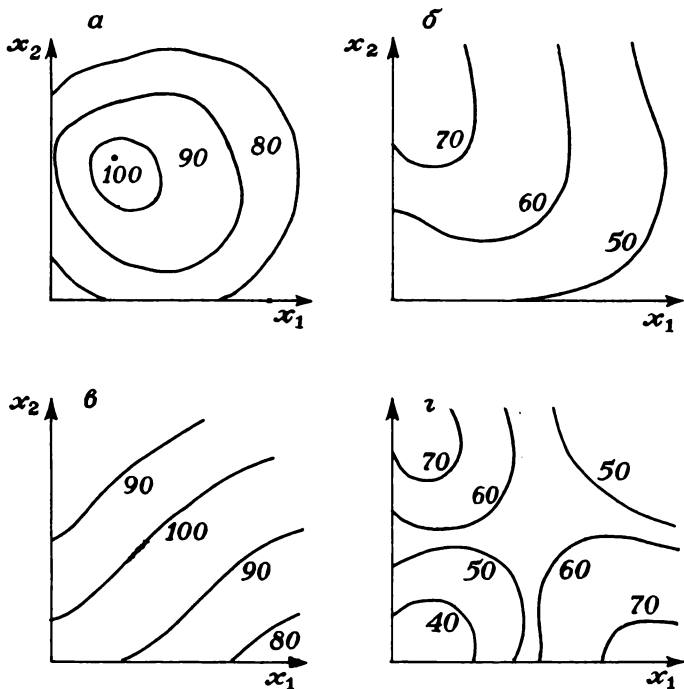


Рис. 20. Типы поверхностей отклика

Для удобства рассмотрения поверхность отклика может быть изображена на плоскости в координатах x_1 и x_2 линиями, соответствующими постоянным значениям функции отклика. Это делается аналогично изображению рельефа местности на географических картах.

На рис. 20 изображены некоторые типы поверхности отклика. Здесь в качестве примера функции отклика взята степень чистоты продукта химической реакции, выраженная в процентах. На рис. 20, а поверхность отклика имеет вид «вершины» и соответствует области значений факторов, где расположен максимум величины y . Очевидно, аналогичный вид имеют *линии постоянного уровня* и в случае минимума функции y .

Поверхность, изображенная на рис. 20, б, характеризует плавное возрастание функции отклика с уменьшением значений фактора x_1 и увеличением x_2 . Такую поверхность принято называть «стационарным возвышением».

Поверхность, показанная на рис. 20, в, называется «хребтом». Его вершина соответствует наибольшим значениям функции отклика. Аналогично располагаются линии постоянных значений y и в случае «оврага», дно которого соответствует минимальным значениям функции отклика.

Наконец, на рис. 20, г изображена поверхность, называемая «седлом». На двух участках этой поверхности наблюдается возрастание функции отклика, а на двух других — убывание.

Следует отметить, что на практике встречаются поверхности отклика и более сложной формы.

ПРОВЕРИМ ВОСПРОИЗВОДИМОСТЬ ОПЫТОВ

Прежде чем приступить к планированию эксперимента, необходимо убедиться в том, что опыты *воспроизводимы*, т. е. результаты опытов, проведенных в одинаковых условиях, близки друг к другу. Для этой цели проводят несколько серий *параллельных* опытов. Условия реализации опытов каждой серии — одинаковы, а разных серий — отличаются друг от друга. Однако все опыты проводятся в рассматриваемой области изменения влияющих факторов. Результаты этих опытов сводят в таблицу (табл. 10). Количество опытов во всех сериях должно быть одинаковым.

Для каждой серии параллельных опытов вычисляют среднее арифметическое значение функции отклика

$$\bar{y}_j = \frac{1}{k} \sum_{i=1}^k y_{ji},$$

где j — номер серии; k — число параллельных опытов, проведенных при одинаковых условиях.

Затем вычисляют для каждой серии параллельных опытов величину, называемую оценкой дисперсии:

$$s_j^2 = \frac{1}{k-1} \sum_{i=1}^k (y_{ji} - \bar{y}_j)^2.$$

Среди всех оценок дисперсий находят наибольшую. Мы обозначим ее через $\max s_j^2$. Затем находят отношение наибольшей из

Т а б л и ц а 10. Эксперимент для проверки воспроизводимости опытов

Номер серии опытов	Результаты параллельных опытов			Средние значения	Оценки дисперсии
1	y_{11}	y_{12}	y_{1k}	\bar{y}_1	s_1^2
2	y_{21}	y_{22}	y_{2k}	\bar{y}_2	s_2^2
3	y_{31}	y_{32}	y_{3k}	\bar{y}_3	s_3^2
...
j	y_{j1}	y_{j2}	y_{jk}	\bar{y}_j	s_j^2
...
N	y_{N1}	y_{N2}	y_{Nk}	\bar{y}_N	s_N^2

Таблица 11. Критические значения критерия Кохрена

Число серий опытов (число оценок дисперсий)	$f = k - 1$			
	1	2	3	4
2	0,999	0,975	0,939	0,906
3	0,967	0,871	0,798	0,746
4	0,907	0,768	0,684	0,629
5	0,841	0,684	0,598	0,544

оценок дисперсий к сумме всех оценок дисперсий:

$$G_p = \frac{\max s_j^2}{\sum_{j=1}^N s_j^2}.$$

Величина G_p называется расчетным значением критерия Кохрена. Критические, т. е. предельно допустимые значения критерия Кохрена G , приведены в табл. 11.

Для нахождения G необходимо знать общее число N оценок дисперсий и так называемое число степеней свободы f , связанных с каждой из них, причем $f = k - 1$.

Опыты считаются воспроизводимыми, когда выполняется условие $G_p \leq G$.

Если опыты невоспроизводимы, то можно попытаться достигнуть воспроизводимости путем выявления и устранения источников нестабильности эксперимента, а также за счет использования более точных измерительных приборов.

Наконец, если никакими способами невозможно обеспечить воспроизводимость, то математические методы планирования к такому эксперименту применять нельзя.

В качестве примера рассмотрим эксперимент, в котором измерялся выход продукта химической реакции y (%), зависящий от двух факторов — температуры x_1 (°C) и концентрации вещества x_2 (%). Условия проведения опытов и результаты измерений приведены в табл. 12. Следовательно,

$$G_p = \frac{\max s_i^2}{\sum_j S_j^2} = \frac{1,28}{0,50 + 0,72 + 1,28} = 0,51.$$

Критическое значение критерия Кохрена $G = 0,967$. Оно найдено по табл. 11 при $N = 3$ и $f = k - 1 = 2 - 1 = 1$.

Условие $G_p \leq G$ выполнено; следовательно, опыты можно считать воспроизводимыми.

Оценки дисперсий нескольких серий параллельных опытов можно усреднить и найти величину

$$s_y^2 = \frac{1}{N} \sum_{j=1}^N s_j^2,$$

Т а б л и ц а 12. Условия проведения опытов и результаты измерений

Номер серии опытов	Условия опытов		Результаты измерений		$\bar{y}_i, \%$	s_i^2
	x_1	x_2	y_{i1}	y_{i2}		
1	24	45	35,0	36,0	35,5	0,50
2	24	55	39,3	38,1	38,7	0,72
3	26	45	31,8	33,4	32,6	1,28

называемую *оценкой дисперсии воспроизводимости*. С ней связано число степеней свободы $f = N(k - 1)$. На основании табл. 12 находим:

$$s_y^2 = \frac{1}{3} (0,50 + 0,72 + 1,28) = 0,83;$$

$$f = N(k - 1) = 3(2 - 1) = 3.$$

Оценку дисперсии среднего значения рассчитывают по формуле

$$s_y^2 = s_y^2/k$$

С ней также связано число степеней свободы $f = N(k - 1)$. В рассматриваемом примере

$$s_y^2 = 0,83/2 \approx 0,42.$$

Если при проведении эксперимента опыты дублируют и пользуются средними значениями функции отклика y , то при обработке экспериментальных данных следует использовать s_y^2 . В тех случаях, когда из-за недостатка времени, большой трудоемкости или высокой стоимости эксперимента опыты не дублируют, при обработке экспериментальных данных используют s_y^2 .

Таким образом, вычисления, связанные с проверкой воспроизводимости опытов, достаточно просты. Для их проведения достаточно использовать микрокалькулятор.

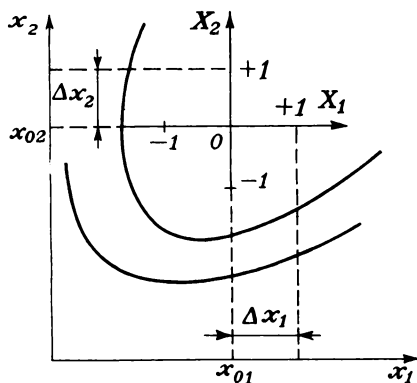
ПОЛНЫЙ ФАКТОРНЫЙ ЭКСПЕРИМЕНТ

Под *математическим описанием* химико-технологического процесса обычно понимают систему уравнений, связывающих функции отклика с влияющими факторами. В простейшем случае это может быть одно уравнение. Часто математическое описание называют *математической моделью*.

С помощью математических методов планирования эксперимента можно получить математическую модель химико-технологического процесса даже при отсутствии сведений о механизме его протекания. Это в ряде случаев бывает очень полезно.

Ценность математического описания заключается в том, что оно: 1) дает качественную и количественную информацию о влиянии каждого фактора; 2) позволяет рассчитать значение функции отклика при заданном режиме ведения технологического процесса; 3) может служить основой для оптимизации.

Рис. 21. Введение кодированных переменных



Кроме того, на основе методов планирования эксперимента можно количественно описать свойства таких сложных продуктов, как сплавы, пластмассы, резины, керамика, ситаллы, бетоны и т. п.

Математические модели, получаемые с помощью методов планирования эксперимента, принято называть *экспериментально-статистическими*.

Метод полного факторного эксперимента дает возможность получить математическое описание химико-технологического процесса в некоторой области факторного пространства, лежащей в окрестности выбранной точки с координатами $(x_{01}, x_{02}, \dots, x_{0i}, x_{0n}; i=1, n$, где n — число факторов).

Перенесем начало координат факторного пространства в данную точку (рис. 21). С этой целью введем новые переменные величины

$$X_i = \frac{x_i - x_{0i}}{\Delta x_i},$$

где Δx_i — выбранный нами масштаб по оси X_i .

Величины X_i не имеют размерностей и называются *кодированными переменными*.

С помощью полного факторного эксперимента ищут математическое описание химико-технологического процесса в виде уравнения

$$y = b_0 + b_1 X_1 + b_2 X_2 + \dots + b_n X_n + b_{12} X_1 X_2 + \dots + b_{(n-1)n} X_{n-1} X_n.$$

В него входит свободный член b_0 , члены в виде произведений коэффициентов регрессии b_i на X_i и члены, содержащие парные произведения кодированных переменных. Таким образом, это — неполное квадратное уравнение.

Все факторы в ходе полного факторного эксперимента варьируют на двух уровнях, соответствующих значениям кодированных переменных $+1$ и -1 . В табл. 13 приведены условия опытов

Т а б л и ц а 13. **Полный двухфакторный эксперимент**

Номер опыта	Факторы		Функция отклика
	X_1	X_2	
1	-1	-1	y_1
2	+1	-1	y_2
3	-1	+1	y_3
4	+1	+1	y_4

полного двухфакторного эксперимента. Часть таблицы, обведенная штриховыми линиями, называется *матрицей планирования*.

Матрица содержит полный набор всех возможных комбинаций уровней варьирования факторов. Отсюда *полный факторный эксперимент* получил свое название.

Как следует из рис. 22, результаты опытов, приведенные в табл. 13, соответствуют на факторной плоскости вершинам квадрата с центром в начале координат.

В табл. 14 приведены условия опытов полного трехфакторного эксперимента. Эти опыты соответствуют в факторном пространстве вершинам куба с центром в начале координат.

Основные принципы построения матриц планирования полного факторного эксперимента: 1) уровни варьирования первого фактора чередуются от опыта к опыту; 2) частота смены уровней варьирования каждого последующего фактора вдвое меньше, чем у предыдущего.

Общее число опытов полного факторного эксперимента:

$$N = 2^n,$$

где n — число факторов.

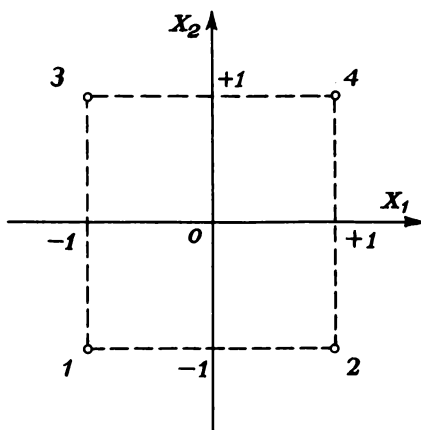


Рис. 22. Опыты полного двухфакторного эксперимента

Т а б л и ц а 14. **Полный трехфакторный эксперимент**

Номер опыта	Факторы			Функция отклика	Номер опыта	Факторы			Функция отклика
	X_1	X_2	X_3			X_1	X_2	X_3	
1	-1	-1	-1	y_1	5	-1	-1	+1	y_5
2	+1	-1	-1	y_2	6	+1	-1	+1	y_6
3	-1	+1	-1	y_3	7	-1	+1	+1	y_7
4	+1	+1	-1	y_4	8	+1	+1	+1	y_8

На основании результатов полного факторного эксперимента вычисляют коэффициенты регрессии, пользуясь следующими формулами:

$$b_0 = \frac{1}{N} \sum_{j=1}^N y_j; \quad b_i = \frac{1}{N} \sum_{j=1}^N X_{ji} y_j;$$

$$b_{lm} = \frac{1}{N} \sum_{j=1}^N X_{jl} X_{jm} y_j \quad (l \neq m).$$

Некоторые из коэффициентов регрессии могут оказаться пренебрежимо малыми — *незначимыми*. Чтобы установить, значим коэффициент или нет, необходимо прежде всего вычислить оценку дисперсии, с которой он определяется:

$$s_b^2 = s_y^2 / N.$$

Следует отметить, что по результатам полного факторного эксперимента все коэффициенты определяются с одинаковой погрешностью.

Принято считать, что коэффициент регрессии значим, если выполнено условие

$$|b| \geq s_b t.$$

Здесь t — значение *критерия Стьюдента*, взятое из табл. 15 (Стьюдент — это псевдоним английского математика и химика Госсета).

Для пользования табл. 15 необходимо знать число степеней свободы f , связанное с оценкой дисперсии s_y^2 .

Т а б л и ц а 15. **Значения критерия Стьюдента**

f	t	f	t	f	t
1	12,71	5	2,57	8	2,31
2	4,30	6	2,45	9	2,26
3	3,18	7	2,36	10	2,23
4	2,78				

Если проверка показала, что коэффициент регрессии незначим, то соответствующий член можно исключить из уравнения.

Получив уравнение регрессии, следует проверить его адекватность, т. е. способность достаточно хорошо описывать поверхность отклика и прогнозировать результаты опытов. Для проверки адекватности вычисляют оценку дисперсии адекватности по формуле

$$s_{ад}^2 = \frac{1}{N-B} \sum_{j=1}^N (y_j^i - y_j^p)^2.$$

Здесь B — число значимых коэффициентов регрессии; y_j^i , y_j^p — экспериментальное и расчетное значение функции отклика в j -м опыте; N — число опытов полного факторного эксперимента.

С оценкой дисперсии адекватности связано число степеней свободы $f_{ад} = N - B$.

Затем находят расчетное значение критерия Фишера:

$$F_p = s_{ад}^2 / s_y^2.$$

Уравнение регрессии считается адекватным, если выполняется условие

$$F_p \leq F,$$

где F — критическое значение критерия Фишера (табл. 16).

Для пользования табл. 19 необходимо знать числа степеней свободы, связанных с числителем и знаменателем отношения F_p .

Рассмотрим применение полного факторного эксперимента для получения математического описания химико-технологического процесса. Выход целевого продукта химической реакции обозначим через y (%). На ход процесса влияют следующие факторы: температура реакционной смеси в аппарате $-x_1$ ($^{\circ}\text{C}$), объемная скорость подачи реагента $-x_2$ ($\text{м}^3/\text{с}$), концентрация реагента на

Т а б л и ц а 16. Значения критерия Фишера

Число степеней свободы f_i (для знаменателя)	Число степеней свободы f_i (для числителя)						
	1	2	3	4	5	6	8
1	161,40	199,50	215,70	224,60	230,20	234,00	238,90
2	18,51	19,00	19,16	19,25	19,30	19,33	19,37
3	10,13	9,55	9,28	9,12	9,01	8,94	8,84
4	7,71	6,94	6,59	6,39	6,26	6,16	6,04
5	6,61	5,79	5,41	5,19	5,05	4,95	4,82
6	5,99	5,14	4,76	4,53	4,39	4,28	4,15
7	5,59	4,74	4,35	4,12	3,97	3,87	3,73
8	5,32	4,46	4,07	3,84	3,69	3,58	3,44
9	5,12	4,26	3,86	3,63	3,48	3,37	3,23
10	4,96	4,10	3,71	3,48	3,33	3,22	3,07

Т а б л и ц а 17. Условия проведения полного трехфакторного эксперимента

Характеристика плана	x_1	x_2	x_3
Основной уровень	250	0,3	0,5
Интервал варьирования	15	0,05	0,1
Верхний уровень	265	0,35	0,6
Нижний уровень	235	0,25	0,4

Т а б л и ц а 18. Матрица планирования и результаты полного трехфакторного эксперимента

Номер опыта	Факторы			Функция отклика	
	x_1	x_2	x_3	y^3	y^p
1	-1	-1	-1	14,55	13,47
2	+1	-1	-1	45,30	47,71
3	-1	+1	-1	12,40	13,47
4	+1	+1	-1	50,12	47,71
5	-1	-1	+1	7,38	7,75
6	+1	-1	+1	27,52	26,80
7	-1	+1	+1	8,12	7,75
8	+1	+1	+1	26,20	26,80

входе в аппарат x_3 (кмоль/м³). Условия проведения опытов, матрица планирования и результаты экспериментов приведены в табл. 17 и 18, соответственно.

Обработаем результаты эксперимента. Сначала рассчитаем коэффициенты регрессии:

1) коэффициенты при влияющих факторах

$$b_1 = \frac{1}{8} \sum_{j=1}^8 X_{j1} y_j^3 = \frac{1}{8} (-14,55 + 45,30 - 12,40 + 50,12 - 7,38 + 27,52 - 8,12 + 26,20) = 13,34;$$

$$b_2 = \frac{1}{8} \sum_{j=1}^8 X_{j2} y_j^3 = 0,26; \quad b_3 = \frac{1}{8} \sum_{j=1}^8 X_{j3} y_j^3 = -6,64;$$

2) коэффициенты при парных произведениях факторов

$$b_{12} = \frac{1}{8} \sum_{j=1}^8 X_{j1} X_{j2} y_j^3 = \frac{1}{8} (14,55 - 45,30 - 12,40 + 50,12 + 7,38 - 27,52 - 8,12 + 26,20) = 0,61;$$

$$b_{13} = \frac{1}{8} \sum_{j=1}^8 X_{j1} X_{j3} y_j^3 = -3,78; \quad b_{23} = \frac{1}{8} \sum_{j=1}^8 X_{j2} X_{j3} y_j^3 = -0,41;$$

3) свободный член уравнения регрессии

$$b_0 = \frac{1}{8} \sum_{j=1}^8 y_j^* = \frac{1}{8} (14,55 + 45,30 + 12,40 + \\ + 50,12 + 7,38 + 27,52 + 8,12 + 26,20) = 23,95.$$

Рассчитаем оценку дисперсии, с которой определяются коэффициенты уравнения регрессии. Оценка дисперсии воспроизводимости $s_y^2 = 1,1$ найдена по результатам предварительного эксперимента и с нею связано число степеней свободы $f_y = 2$. Тогда оценка дисперсии, с которой вычисляются коэффициенты регрессии, равна:

$$s_b^2 = s_y^2 / N = 1,1 / 8 = 0,1375.$$

Проверим значимость коэффициентов регрессии. Табличное значение критерия Стьюдента $t = 4,30$. Коэффициент значим, если выполняется условие $|b| > s_b t$. После исключения незначимых коэффициентов уравнение регрессии принимает вид

$$y = 23,95 + 13,34X_1 - 6,64X_3 - 3,78X_1X_3.$$

Проверим адекватность полученного уравнения регрессии. Найдем расчетное значение функции отклика для первого опыта, подставив в уравнение регрессии значения кодированных переменных:

$$y^p = 23,95 + 13,34(-1) - 6,64(-1) - 3,78(-1)(-1) = 13,46.$$

Расчетные значения функции отклика для остальных опытов приведены в табл. 21.

Рассчитаем оценку дисперсии адекватности:

$$s_{ад}^2 = \frac{1}{8-4} \sum_{j=1}^8 (y_j^* - y_j^p)^2 = \frac{1}{8-4} (14,55 - 13,46)^2 + \\ + (45,30 - 47,71)^2 + \dots + (26,20 - 26,80)^2 = 3,76.$$

Число степеней свободы, связанное с оценкой дисперсии адекватности, равно $f_{ад} = 8 - 4 = 4$.

Вычислим расчетное значение критерия Фишера:

$$F_p = 3,76 / 1,1 = 3,42.$$

Критическое значение критерия Фишера в данном случае равно 19,25. Следовательно, найденное нами уравнение регрессии, адекватно описывает экспериментальные данные.

Заметим, что уравнение регрессии записано применительно к кодированным переменным, которые связаны с рассматриваемыми физическими величинами следующими соотношениями:

$$X_1 = \frac{x_1 - 250}{15}; \quad X_2 = \frac{x_2 - 0,3}{0,05}; \quad X_3 = \frac{x_3 - 0,5}{0,1}.$$

Подставив эти выражения в уравнение регрессии и приведя затем подобные члены, получим уравнение регрессии в физических переменных:

$$y = -257,91 + 2,16x_1 + 563,72x_3 - 2,53x_1x_3.$$

Программа для обработки результатов полного трехфакторного эксперимента приведена ниже:

```

10 REM ПРОГРАММА ДЛЯ ОБРАБОТКИ РЕЗУЛЬТАТОВ
20 REM ПОЛНОГО ТРЕХФАКТОРНОГО ЭКСПЕРИМЕНТА
30 DIM X(8,3),Y(8),Y2(8),B(6),S(6),T1(6)
40 DATA -1,1,-1,1,-1,1,-1,1
50 DATA -1,-1,1,1,-1,-1,1,1
60 DATA -1,-1,-1,-1,1,1,1,1
70 FOR J=1 TO 3 : FOR I=1 TO 8
80 READ X(I,J) : NEXT I : NEXT J
90 FOR I=1 TO 8
100 INPUT "ЭКСПЕРИМЕНТАЛЬНЫЕ ЗНАЧЕНИЯ Y",Y(I)
110 NEXT I
120 INPUT "ОЦЕНКА ДИСПЕРСИИ ВОСПРОИЗВОДИМОСТИ ";S1
130 INPUT "КРИТЕРИЙ СТЬЮДЕНТА ";T
140 FOR I=0 TO 6 : LET S(I)=0 : NEXT I
150 FOR I=1 TO 8 : LET S(0)=S(0)+Y(I)
160 LET S(1)=S(1)+X(I,1)*Y(I)
170 LET S(2)=S(2)+X(I,2)*Y(I)
180 LET S(3)=S(3)+X(I,3)*Y(I)
190 LET S(4)=S(4)+X(I,1)*X(I,2)*Y(I)
200 LET S(5)=S(5)+X(I,1)*X(I,3)*Y(I)
210 LET S(6)=S(6)+X(I,2)*X(I,3)*Y(I)
220 NEXT I
230 LET S2=SQR(S1/8)
240 PRINT "КОЭФФИЦИЕНТЫ РЕГРЕССИИ "
250 FOR I=0 TO 6
260 LET B(I)=S(I)/8 : LET T1(I)=ABS(B(I)/S2)
270 PRINT B(I) : NEXT I : LET Z=0
280 PRINT " ЗНАЧИМЫЕ КОЭФФИЦИЕНТЫ "
290 FOR I=0 TO 6
300 IF T1(I)>T THEN GOTO 320
310 LET B(I)=0 : GOTO 330
320 LET Z=Z+1
330 NEXT I
340 FOR I=0 TO 6 : PRINT B(I) : NEXT I
350 FOR J=1 TO 8 : LET Y2(J)=B(0)
360 FOR I=1 TO 3
370 LET Y2(J)=Y2(J)+B(I)*X(J,I)
380 NEXT I
390 LET Y2(J)=Y2(J)+X(J,1)*X(J,2)*B(4)
400 LET Y2(J)=Y2(J)+X(J,1)*X(J,3)*B(5)
410 LET Y2(J)=Y2(J)+X(J,2)*X(J,3)*B(6)
420 NEXT J : LET S3=0
430 PRINT "РАСЧЕТНЫЕ ЗНАЧЕНИЯ Y"
440 FOR I=1 TO 8 : PRINT Y2(I)
450 LET S3=S3+(Y(I)-Y2(I))^2
460 NEXT I
470 LET S4=S3/(8-Z)
480 PRINT " ОЦЕНКА ДИСПЕРСИИ АДЕКВАТНОСТИ "; S4
490 F=S4/S1 : PRINT "ЗНАЧЕНИЕ КРИТЕРИЯ ФИШЕРА "; F
500 STOP : END

```

Приведенная программа существенно ускоряет и упрощает обработку экспериментальных данных. С ее помощью рассчитываются коэффициенты регрессии и выдается на дисплей вся необходимая информация для проверки адекватности полученного уравнения.

КОЛИЧЕСТВО ОПЫТОВ МОЖНО УМЕНЬШИТЬ

С увеличением числа учитываемых факторов резко возрастает число опытов полного факторного эксперимента. Это следует из

Т а б л и ц а 19. Полный трехфакторный эксперимент и его дробные реплики

Номер опыта	Факторы			Функция отклика	Дробные реплики	
	X_1	X_2	X_3			
1	-1	-1	-1	y_1	} 1/4	} 1/2
2	+1	-1	-1	y_2		
3	-1	+1	-1	y_3	} 1/4	
4	+1	+1	-1	y_4		
5	-1	-1	+1	y_5	} 1/4	} 1/2
6	+1	-1	+1	y_6		
7	-1	+1	+1	y_7	} 1/4	
8	+1	+1	+1	y_8		

формулы $N=2^n$. Достаточно ввести в рассмотрение один новый фактор и количество опытов сразу увеличится вдвое. Однако для нахождения коэффициентов регрессии не всегда требуется много опытов. В таких случаях можно уменьшить объем экспериментальных работ, воспользовавшись *методом дробного факторного эксперимента*, который известен также под названием *метода дробных реплик*. Этот метод заключается в том, что для нахождения уравнения регрессии используется определенная часть полного факторного эксперимента: 1/2; 1/4 и т. д. Такие системы опытов называются *дробными репликами* (см. табл. 19).

Расчет коэффициентов регрессии, проверка значимости коэффициентов и адекватности математического описания в данном случае осуществляются так же, как и при полном факторном эксперименте.

Пусть требуется найти коэффициенты уравнения регрессии

$$y = b_0 + b_1X_1 + b_2X_2 + b_3X_3.$$

Если для этой цели воспользоваться полным трехфакторным экспериментом, то необходимо провести 8 опытов. Однако эту задачу можно решить и с помощью меньшего числа опытов. Например, возьмем матрицу полного двухфакторного эксперимента (табл. 20) и приравняем произведение X_1X_2 к фактору X_3 .

Рассчитаем коэффициенты регрессии:

$$b_0 = \frac{1}{4} (y_1 + y_2 + y_3 + y_4);$$

$$b_1 = \frac{1}{4} (-y_1 + y_2 - y_3 + y_4);$$

$$b_2 = \frac{1}{4} (-y_1 - y_2 + y_3 + y_4).$$

Обратим внимание на то, что в табл. 20 столбцы для произведения X_1X_2 и фактора X_3 полностью совпадают. Поэтому коэффициенты регрессии b_{12} и b_3 не могут быть определены раздельно.

Т а б л и ц а 20. Дробный факторный эксперимент типа 2^{3-1}

Номер опыта	X_1	X_2	X_1X_2	X_3	Функция отклика
1	-1	-1	+1	+1	y_1
2	+1	-1	-1	-1	y_2
3	-1	+1	-1	-1	y_3
4	+1	+1	+1	+1	y_4

Может быть найдена только их сумма:

$$b_{12} + b_3 = \frac{1}{4} (y_1 - y_2 - y_3 + y_4).$$

Этот недостаток рассматриваемого плана является своеобразной «платой» за уменьшение общего числа опытов с восьми до четырех.

Такое планирование эксперимента, когда некоторые из факторов приравнивают к произведениям нескольких факторов, называется *планированием со смешиванием*. Его обозначают символом 2^{n-p} , где n — общее число факторов, а p — число факторов, приравненных к произведениям. С этой точки зрения в табл. 20 приведена матрица планирования типа 2^{3-1} .

Существует правило, позволяющее определить, какие коэффициенты регрессии определяются совместно при планировании со смешиванием. Рассмотрим это правило на примере.

Методом дробных реплик будем искать математическое описание процесса в виде уравнения регрессии

$$y = b_0 + b_1X_1 + b_2X_2 + b_3X_3 + b_4X_4 + b_5X_5.$$

Воспользуемся планированием типа 2^{5-2} и примем

$$X_4 = -X_1X_2; \quad X_5 = X_1X_2X_3.$$

Такие равенства в методе дробных реплик называются *генерирующими соотношениями*.

Следует отметить, что выбор генерирующих соотношений в общем случае произволен. Однако он существенно влияет на характер совместных оценок коэффициентов регрессии.

Правило определения совместных оценок коэффициентов заключается в следующем:

1. Примем во внимание, что $X_i^2 = 1$, $X_i \cdot 1 = X_i$.

2. Умножив обе части генерирующих соотношений соответственно на X_4 и X_5 , получим:

$$1 = -X_1X_2X_4; \quad 1 = X_1X_2X_3X_5.$$

Эти равенства называются *определяющими контрастами*. Перемножив их почленно, получим новые определяющие контрасты.

В данном случае

$$1 = -X_3X_4X_5.$$

3. Составим алгебраическую сумму из единицы и правых частей всех полученных определяющих контрастов:

$$s = 1 - X_1X_2X_4 + X_1X_2X_3X_5 - X_3X_4X_5.$$

4. Умножив каждый из факторов на s и заменив факторы соответствующими коэффициентами регрессии β , получим искомые *совместные оценки*:

$$b_1 \rightarrow \beta_1 - \beta_{24} + \beta_{235} - \beta_{1345};$$

$$b_2 \rightarrow \beta_2 - \beta_{14} + \beta_{135} - \beta_{2345};$$

$$b_3 \rightarrow \beta_3 - \beta_{1234} + \beta_{125} - \beta_{45};$$

$$b_4 \rightarrow \beta_4 - \beta_{12} + \beta_{12345} - \beta_{35}.$$

где β_i — истинные коэффициенты регрессии.

В рассмотренном примере удалось значительно сократить общее количество опытов. Так, полный пятифакторный эксперимент состоит из 32 опытов. План типа 2^{5-2} содержит 8 опытов, т. е. в четыре раза меньше.

Отметим еще одну интересную особенность дробного факторного эксперимента. Планирование со смешиванием иногда применяют в тех случаях, когда необходимо уменьшить влияние неуправляемых временных изменений некоторых влияющих факторов, т. е. влияние так называемого *временного дрейфа*. Примером временного дрейфа может служить уменьшение со временем активности катализатора вследствие отравления его каталитическими ядами.

При постановке большой серии опытов, требующих длительного времени, всегда приходится опасаться нежелательных изменений исходных свойств реагентов, некоторых характеристик технологического оборудования и т. п. Влияние этого временного дрейфа на параметры математического описания процесса можно практически устранить, разбивая всю серию опытов на отдельные *блоки* так, чтобы эффект от временного дрейфа оказался смешанным с произведениями факторов, для которых коэффициенты регрессии достаточно малы.

Допустим, что необходимо устранить влияние временного дрейфа на параметры уравнения регрессии, получаемого в результате полного трехфакторного эксперимента. С этой целью разобьем эксперимент на два блока и введем новую независимую переменную X_d , характеризующую дрейф. Положим $X_d = X_1X_2X_3$.

Исходя из матрицы планирования, представленной в табл. 21, будем считать, что в первом блоке все результаты опытов вследствие временного дрейфа завышены на β_d , а во втором — занижены на ту же величину.

Т а б л и ц а 21. Планирование в условиях временного дрейфа

Номер блока	X_1	X_2	X_3	$X_{12} = X_1 X_2 X_3$	Функция отклика
1	-1	-1	+1	+1	$y_1 + \beta_d$
	+1	-1	-1	+1	$y_2 + \beta_d$
	-1	+1	-1	+1	$y_3 + \beta_d$
	+1	+1	+1	+1	$y_4 + \beta_d$
2	-1	-1	-1	-1	$y_5 - \beta_d$
	+1	-1	+1	-1	$y_6 - \beta_d$
	-1	+1	+1	-1	$y_7 - \beta_d$
	+1	+1	-1	-1	$y_8 - \beta_d$

Если уравнение регрессии отыскивается в виде

$$y = b_0 + b_1 X_1 + b_2 X_2 + b_3 X_3 + b_{12} X_1 X_2 + b_{13} X_1 X_3 + b_{23} X_2 X_3 + b_{123} X_1 X_2 X_3,$$

то коэффициенты регрессии будут являться следующими оценками:

$$\begin{aligned} b_0 &\rightarrow \beta_0; & b_2 &\rightarrow \beta_2; \\ b_1 &\rightarrow \beta_1; & b_3 &\rightarrow \beta_3; \\ b_{12} &\rightarrow \beta_{12}; & b_{23} &\rightarrow \beta_{23}; \\ b_{13} &\rightarrow \beta_{13}; & b_{123} &\rightarrow \beta_{123} + \beta_d. \end{aligned}$$

Рассчитаем, например, коэффициенты b_1 и b_{123} :

$$\begin{aligned} b_1 &= \frac{1}{8} [-(y_1 + \beta_d) + (y_2 + \beta_d) - (y_3 + \beta_d) + (y_4 + \beta_d) - \\ &\quad - (y_5 - \beta_d) + (y_6 - \beta_d) - (y_7 - \beta_d) + (y_8 - \beta_d)] = \\ &= \frac{1}{8} (-y_1 + y_2 - y_3 + y_4 - y_5 + y_6 - y_7 + y_8); \end{aligned}$$

$$\begin{aligned} b_{123} &= \frac{1}{8} [(y_1 + \beta_d) + (y_2 + \beta_d) + (y_3 + \beta_d) + (y_4 + \beta_d) - \\ &\quad - (y_5 - \beta_d) - (y_6 - \beta_d) - (y_7 - \beta_d) - (y_8 - \beta_d)] = \\ &= \frac{1}{8} (y_1 + y_2 + y_3 + y_4 - y_5 - y_6 - y_7 - y_8) + \beta_d. \end{aligned}$$

Следовательно, все коэффициенты регрессии, кроме b_{123} , не содержат погрешностей, обусловленных временным дрейфом.

КРУТОЕ ВОСХОЖДЕНИЕ

Полученное с помощью полного или дробного факторного эксперимента уравнение регрессии служит не только математической моделью химико-технологического процесса, но используется и для его оптимизации. *Оптимизацией* процесса называют целенаправленный поиск наилучших в определенном смысле условий

его проведения. Задачи оптимизации приходится решать при разработке новых технологий и реконструкции действующих производств.

Величина, характеризующая уровень оптимизации процесса, называется *критерием оптимальности* или *целевой функцией*. Например, критерием оптимальности может быть выход целевого продукта химической реакции в расчете на израсходованное сырье.

Качество процесса химической технологии обычно выражается не одним, а несколькими показателями. Безусловно, хочется, чтобы все показатели достигали одновременно своих наилучших значений. К сожалению, это невозможно. Например, максимальная производительность оборудования и минимальная себестоимость продукции обычно имеют место при различных технологических режимах.

Чтобы осуществлять оптимизацию, необходимо располагать определенными *ресурсами*. Например, для увеличения выхода целевого продукта химической реакции, мы должны иметь возможность варьировать в определенных пределах значениями таких величин, как температура реакционной смеси, исходные концентрации реагентов и т. д.

Оптимизацию проводят, как правило, в условиях ограничений на ресурсы оптимизации и некоторые выходные параметры процесса.

Задачу оптимизации можно сформулировать следующим образом. Задан критерий оптимальности

$$y_1 = f_1(x_1, x_2, \dots, x_n)$$

и ограничения на остальные выходные параметры процесса:

$$m_2 \leq y_2 \leq M_2;$$

$$\dots$$

$$m_k \leq y_k \leq M_k.$$

Заданы ограничения на ресурсы оптимизации:

$$x_{1 \min} \leq x_1 \leq x_{1 \max};$$

$$\dots$$

$$x_{n \min} \leq x_n \leq x_{n \max}.$$

Требуется найти такие значения ресурсов оптимизации (x_1^* , ..., x_n^*), при которых критерий оптимальности достигает своего *экстремального* (т. е. наибольшего или наименьшего) значения.

Д. Бокс и К. Уилсон предложили использовать для оптимизации результаты полного или дробного факторного эксперимента. Сущность этого метода состоит в следующем.

Среди всех имеющихся функций отклика, описывающих объект оптимизации, выбирают одну наиболее важную и принимают ее в качестве критерия оптимальности y_1 . Затем указывают ограни-

чения, накладываемые на остальные функции отклика и на влияющие факторы.

Один из наиболее сильно влияющих факторов принимают за базовый и для него вычисляют произведение соответствующего коэффициента регрессии на шаг варьирования. Например, для первого фактора это произведение имеет вид $b_1 \Delta x_1$. Затем для базового фактора выбирают шаг Δx_1^* , с которым будет осуществляться оптимизация. С целью более точного нахождения координат оптимума обычно принимают $\Delta x_1^* \leq \Delta x_1$.

После этого вычисляют отношение

$$\gamma = \frac{\Delta x_i^*}{b_i \Delta x_1}.$$

Для всех остальных факторов шаги движения к оптимуму рассчитывают по формуле

$$\Delta x_i^* = \gamma b_i \Delta x_1.$$

Движение к оптимуму начинают из *центра* плана, который был использован для получения уравнения регрессии. Значения факторов x_1, \dots, x_n на каждом новом шаге оптимизации находят путем прибавления Δx_i^* к соответствующим предыдущим значениям. Так осуществляется оптимизация по методу Бокса и Уилсона, получившему название *метода крутого восхождения*.

Отметим некоторые особенности этого метода.

Движение из центра плана начинается в сторону наиболее быстрого увеличения критерия оптимальности. Это происходит благодаря тому, что шаги Δx_i^* пропорциональны коэффициентам регрессии b_i .

Если ищется минимум критерия y_1 , то новые значения факторов x_1, \dots, x_n находят из предыдущих путем вычитания Δx_i^* . Такой способ оптимизации назван *методом наискорейшего спуска*.

Движение к экстремуму прекращают при выполнении следующих условий: 1) значения факторов x_1, \dots, x_n (одного или нескольких) или функций отклика y_2, \dots, y_k (одной или нескольких) вышли на допустимые границы; 2) достигнут экстремум критерия оптимальности.

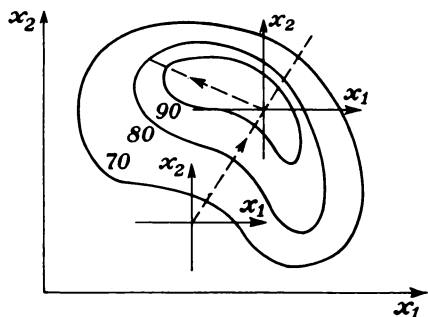


Рис. 23. Оптимизация по методу крутого восхождения

Как следует из рис. 23, в ходе крутого восхождения по поверхности отклика сначала может быть достигнут *локальный экстремум* y_1 на направлении движения. В таком случае необходимо в окрестности этого экстремума поставить новый полный факторный эксперимент и реализовать новое крутое восхождение.

Отметим, что в реальных условиях оптимум обычно находится на границах области допустимых значений x_1, \dots, x_n .

Рассмотрим пример реализации метода крутого восхождения.

Пусть в результате полного факторного эксперимента получено адекватное уравнение регрессии

$$y_1 = 35,6 + 1,95X_1 + 1,3X_2,$$

где y_1 — выход целевого продукта реакции, %; X_1 — безразмерная температура; X_2 — безразмерная концентрация реагента.

Введем также в рассмотрение функцию отклика y_2 , характеризующую скорость химической реакции [кмоль/(м³·ч)]. Будем считать, что на скорость реакции наложено ограничение вида $y_2 \geq 2,5$.

Допустим, что ограничения на ресурсы оптимизации имеют вид:

$$30^\circ \leq x_1 \leq 120^\circ;$$

$$10\% \leq x_2 \leq 70\%.$$

Будем оптимизировать выход целевого продукта химической реакции методом крутого восхождения. В качестве базового фактора примем температуру. Считая, что в ходе полного факторного эксперимента интервал варьирования $\Delta x_1 = 5^\circ$, примем шаг движения на крутом восхождении $\Delta x_1^* = 4^\circ$, тогда

$$\gamma = \frac{\Delta x_1^*}{b_1 \Delta x_1} = \frac{4}{1,95 \cdot 5} = 0,41.$$

Т а б л и ц а 22. Результаты опытов по методу крутого восхождения

Характеристика	x_1	x_2	X_1	X_2	y_1^r	y_1^e	y_2^e
Центр плана	50	25	0	0	35,6	35,1	2,9
Интервал варьирования	5	1	1	1	—	—	—
Шаг движения	4	0,5	0,8	0,5	—	—	—

Номер опыта	Крутое восхождение						
1	54	25,5	0,8	0,5	37,8	36,9	3,2
2	58	26,0	1,6	1,0	40,0	37,2	3,7
3	62	26,5	2,4	1,5	42,2	38,5	2,8
4	66	27,0	3,2	2,0	44,4	40,7	2,3
5	70	27,5	4,0	2,5	46,6	38,1	1,9
6	74	28,0	4,8	3,0	48,8	37,2	1,6

Примечание: y_1^r и y_1^e — расчетные и экспериментальные значения выхода целевого продукта химической реакции, соответственно; y_2^e — экспериментально найденные значения скорости реакции.

Шаг изменения концентрации на крутом восхождении равен:

$$\Delta x_2 = \gamma b_2 \Delta x_2 = 0,41 \cdot 1,3 \cdot 1 \approx 0,5 \%$$

Здесь $\Delta x_2 = 1$ взят по условиям полного факторного эксперимента.

Результаты опытов, выполненных по методу крутого восхождения, приведены в табл. 29.

Расчетные значения критерия оптимальности непрерывно возрастают на крутом восхождении. В отличие от этого экспериментальные значения y_j^* сначала возрастают, а затем, пройдя через максимум, начинают убывать. Такое расхождение между расчетными и экспериментальными данными объясняется тем, что в процессе крутого восхождения мы выходим за пределы области адекватности уравнения регрессии.

В опыте 4 достигнут максимальный выход продукта реакции, однако скорость реакции здесь меньше допустимого значения. По-видимому, предпочтение следует отдать опыту 3.

Ограничения на величины x_1 и x_2 в ходе оптимизации не были нарушены.

СИМПЛЕКС ВЕДЕТ К ОПТИМУМУ

Для оптимизации химико-технологических процессов широко используется симплексный метод. Свое название метод получил от слова симплекс. *Симплексом* называется правильный многогранник, имеющий $n + 1$ вершину, где n — число факторов, являющихся ресурсами оптимизации. Так, если факторов два, то симплексом является правильный треугольник. Сущность симплексного метода оптимизации иллюстрирует рис. 24.

Начальная серия опытов соответствует вершинам исходного симплекса (точки 1, 2 и 3). Сравнивая между собой результаты опытов в точках 1, 2 и 3, находят среди них самый «плохой», с точки зрения выбранного критерия оптимальности. Пусть, например, самым «неудачным» оказался опыт в точке 1. Этот опыт исключают из симплекса, а вместо него вводят опыт в точке 4, которая симметрична точке 1 относительно противоположной стороны треугольника, соединяющей точки 2 и 3.

Далее сравнивают между собой результаты опытов в вершинах нового симплекса, отбрасывают самый «плохой» из них и перено-

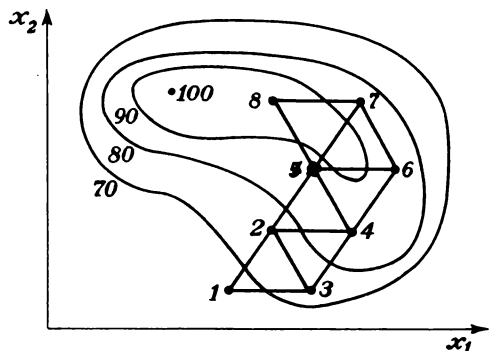


Рис. 24. Оптимизация по симплексному методу

сят соответствующую вершину симплекса в точку 5. Затем рассмотренная процедура повторяется в течение всего процесса оптимизации.

Если достигнут экстремум критерия оптимальности, то дальнейшее движение симплекса прекращается. Это значит, что новый шаг возвращает исследователя в предыдущую точку факторного пространства.

Важно иметь в виду, что симплексный метод является *локальным* методом поиска экстремума. Это значит, что если в рассматриваемой области факторного пространства существует несколько экстремумов критерия оптимальности, то данный метод позволяет найти только тот из них, который расположен ближе к точкам исходного симплекса. Поэтому, если есть подозрение на существование нескольких экстремумов критерия оптимальности, нужно осуществить их поиск, каждый раз начиная оптимизацию из новой области факторного пространства. Затем следует сравнить между собой найденные оптимальные условия и из всех вариантов выбрать наилучший.

При оптимизации необходимо принимать во внимание ограничения, наложенные на влияющие факторы и некоторые функции отклика.

Следует отметить, что при использовании симплексного метода не обязательно дублировать опыты. Дело в том, что ошибка в отдельном опыте может только несколько замедлить оптимизацию. Если же последующие опыты выполняются безупречно, то движение к оптимуму продолжается.

В табл. 23 приведена матрица опытов исходного симплекса в кодированных переменных. Символом «0» обозначены координаты центра симплекса. Величины, входящие в эту таблицу, рассчитываются по формулам

$$K_i = \frac{1}{\sqrt{2i(i+1)}}; \quad R_i = iK_i,$$

где i — номер фактора в матрице планирования.

Т а б л и ц а 23. Матрица исходного симплекса

Номер опыта	X_1	X_2		X_{n-1}	X_n	Функция отклика
1	K_1	K_2		K_{n-1}	K_n	y_1
2	$-R_1$	K_2		K_{n-1}	K_n	y_2
3	0	$-R_2$		K_{n-1}	K_n	y_3
...
$n-1$	0	0		K_{n-1}	K_n	y_{n-1}
n	0	0		$-R_{n-1}$	K_n	y_n
$n+1$	0	0		0	$-R_n$	y_{n+1}

Т а б л и ц а 24. Условья проведения начальной серии опытов

Номер опыта	X_1	X_2	X_3	X_4
1	0,5	0,289	0,204	0,158
2	-0,5	0,289	0,204	0,158
3	0	-0,578	0,204	0,158
4	0	0	-0,612	0,158
5	0	0	0	-0,632

Результаты опытов, представленные в табл. 23, соответствуют вершинам симплекса, сторона которого равна единице, а центр совпадает с началом координат (в кодированных переменных).

Матрица планирования исходного симплекса для четырех факторов приведена в табл. 24.

Аналогично можно рассчитать условия исходной серии опытов для большего числа факторов.

Очевидно, наибольшее количество опытов придется ставить в начале эксперимента. Затем на каждом шаге оптимизации выполняется только один опыт.

Приступая к оптимизации, необходимо рассчитать матрицу исходной серии опытов в физических переменных, пользуясь формулой

$$x_i = x_{0i} + \Delta x_i X_i,$$

где x_{0i} — координата центра плана в физических переменных; Δx_i — интервал варьирования фактора x_i , определяющий размер симплекса; X_i — значение кодированной переменной.

Условия каждого нового опыта рассчитываются по формуле

$$x_i = \frac{2}{n} \left(\sum_{j=1}^{n+1} x_{ji} - x_i^* \right) - x_i^*.$$

Здесь n — число факторов в матрице планирования; j — номер опыта; i — номер фактора; x_i^* — значение i -го фактора в самом «плохом» опыте симплекса.

На любом шаге оптимизации, осуществляемой симплексным методом, можно при желании включить в программу исследований новый фактор, который был зафиксирован на некотором постоянном уровне. При этом значения всех ранее рассматриваемых факторов рассчитываются по формуле

$$x_i = \frac{1}{n+1} \sum_{j=1}^{n+1} x_{ji},$$

где $i = 1, 2, \dots, n$, т. е. являются средними арифметическими значениями соответствующих координат предыдущего симплекса.

Т а б л и ц а 25. Значения основных уровней факторов и интервалов их варьирования

Фактор	Основной уровень	Интервал варьирования
x_1 , кмоль/м ³	1,0	0,1
x_2 , кмоль/м ³	1,5	0,2
x_3 , °С	60,0	5,0

Значение вновь вводимого фактора определяется по формуле

$$x_{n+1} = x_{0(n+1)} + \Delta x_{n+1} (R_{n+1} + K_{n+1}),$$

где $x_{0(n+1)}$ — основной уровень значений вводимого фактора; Δx_{n+1} — интервал варьирования, определяющий размер симплекса по вводимому фактору; R_{n+1} , K_{n+1} — кодированные значения, для которых были указаны расчетные формулы.

Отметим, что добавление нового фактора в состав полного факторного эксперимента требует увеличения числа опытов вдвое. В этом смысле симплексный метод имеет очевидное преимущество.

Рассмотрим теперь пример оптимизации химико-технологического процесса. Пусть требуется оптимизировать выход целевого продукта химической реакции y (%), который получается при взаимодействии двух реагентов с концентрациями x_1 и x_2 (кмоль/м³) при температуре x_3 (°С).

Выберем основные уровни факторов и интервалы их варьирования (табл. 25).

Рассчитаем условия проведения первых четырех опытов, образующих исходный симплекс, и полученные результаты сведем в табл. 26. Так, для третьего опыта

$$x_{31} = 1 + 0,1 \cdot 0 = 1;$$

$$x_{32} = 1,5 + 0,2(-0,578) = 1,38$$

$$x_{33} = 60 + 5 \cdot 0,204 = 61.$$

Здесь первый индекс обозначает номер опыта, а второй — номер фактора.

Сравнивая между собой результаты первых четырех опытов, видим, что самый низкий выход целевого продукта химической реакции получился в третьем опыте. Этот опыт следует исключить из дальнейшего рассмотрения.

Заменим его опытом 5, условия проведения которого рассчитаем по формулам

$$x_{51} = \frac{2}{3} (1,05 + 0,95 + 1,00) - 1,00 = 1,00;$$

$$x_{52} = \frac{2}{3} (2 \cdot 1,56 + 1,50) - 1,38 = 1,70; \quad x_{53} = \frac{2}{3} (2 \cdot 61 + 57) - 61 = 58.$$

Т а б л и ц а 26. Планирование эксперимента по симплексному методу

Номер опыта	x_1	x_2	x_3	Функция отклика	Номер опыта	x_1	x_2	x_3	Функция отклика
1	1,05	1,56	61	72,3	4	1,00	1,50	57	68,2
2	0,95	1,56	61	70,1	5	1,00	1,70	58	73,9
3	1,00	1,38	61	65,4	6	1,00	1,72	63	76,5

Т а б л и ц а 27. Симплексный план эксперимента для четырех факторов

Номер опыта	x_1	x_2	x_3	x_4	Функция отклика
1	1,05	1,56	61	300	72,3
2	0,95	1,56	61	300	70,1
5	1,00	1,70	58	300	73,9
6	1,00	1,72	63	300	76,5
7	1,00	1,64	61	380	78,1

В новом симплексе, образованном опытами 1, 2, 4 и 5, самым «плохим» оказывается опыт 4. Заменяем его опытом 6, условия проведения которого рассчитаем по известным формулам.

Дальнейшая процедура оптимизации аналогична.

Рассмотрим теперь вопрос о том, как включить в программу исследований новый фактор, например частоту вращения мешалки в аппарате. Пусть до этих пор она была постоянной и равной 300 об/мин. Будем считать эту величину фактором x_4 и примем для нее шаг варьирования $\Delta x_4 = 100$ об/мин.

Предыдущий симплекс для трех факторов (см. табл. 26) состоял из опытов 1, 2, 5 и 6. Чтобы из него получить новый симплекс для четырех факторов, введем опыт 7 (табл. 27).

Условия проведения опыта рассчитаем по следующим формулам:

$$x_{71} = \frac{1}{4} (1,05 + 0,95 + 2 \cdot 1,00) = 1,00;$$

$$x_{72} = \frac{1}{4} (2 \cdot 1,56 + 1,70 + 1,72) = 1,64;$$

$$x_{73} = \frac{1}{4} (2 \cdot 61 + 58 + 63) = 61;$$

$$x_{74} = 300 + 100(0,632 + 0,158) \approx 380.$$

Далее можно продолжить оптимизацию, пользуясь рассмотренной выше процедурой.

ПЛАН СТАНОВИТСЯ СЛОЖНЕЕ

Процесс оптимизации часто приводит в область факторного пространства, где кривизна поверхности отклика велика и вследствие этого поверхность не может быть описана многочленом, получаемым с помощью полного или дробного факторного эксперимента. Для адекватного математического описания здесь требуется многочлен более высокой степени (например, второй). С этой целью используют *центральное композиционное планирование эксперимента* (ЦКП). Различают два вида такого планирования — *ортогональное* и *ротатабельное*.

Число опытов при ортогональном планировании определяется по формуле

$$N = 2^n + 2n + 1,$$

где 2^n — число опытов, образующих полный факторный эксперимент (*ядро плана*); $2n$ — число так называемых *звездных точек* в факторном пространстве, имеющих

Т а б л и ц а 28. Ортогональное центральное композиционное планирование для двух факторов

Системы опытов	Номер опыта	X_1	X_2	X_1X_2	X_1^*	X_2^*
Полный факторный эксперимент	1	-1	-1	+1	+0,33	+0,33
	2	+1	-1	-1	+0,33	+0,33
	3	-1	+1	-1	+0,33	+0,33
	4	+1	+1	+1	+0,33	+0,33
Опыты в звездных точках	5	+1	0	0	+0,33	-0,67
	6	-1	0	0	+0,33	-0,67
	7	0	+1	0	-0,67	+0,33
	8	0	-1	0	-0,67	+0,33
Опыт в центре плана	9	0	0	0	-0,67	-0,67

координаты $(\pm\alpha, 0, 0, \dots, 0)$; $(0, \pm\alpha, 0, \dots, 0)$, ..., $(0, 0, \dots, \pm\alpha)$ *; 1 — опыт в центре планирования, т. е. в точке факторного пространства с координатами $(0, 0, \dots, 0)$.

Если с помощью полного факторного эксперимента не удастся получить адекватного математического описания, то к опытам этого эксперимента добавляются опыты в звездных точках и в центре плана, а полученную при этом композицию используют для получения математического описания химического процесса в виде *многочлена второй степени*. Отсюда и произошло название метода — центральное композиционное планирование.

Значения звездного плеча α для ЦКП с различным числом факторов n следующие:

n	2	3	4	5
α	1,000	1,215	1,414	1,547

Уравнение регрессии при ортогональном центральном композиционном планировании ищут в следующем виде:

$$y = b_0 + b_1X_1 + b_2X_2 + \dots + b_nX_n + b_{12}X_1X_2 + \dots + b_{(n-1)n}X_{n-1}X_n + b_{11}X_1^* + \dots + b_{nn}X_n^*.$$

Входящие в это уравнение вспомогательные переменные рассчитываются по формулам

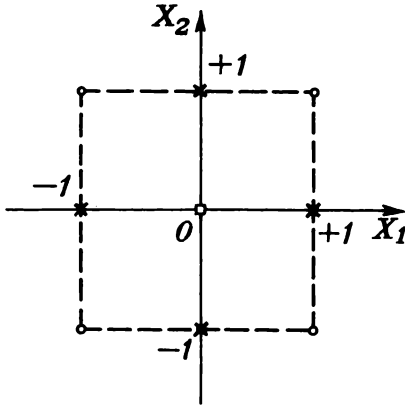
$$X_{ji}^* = X_{ji}^2 - \frac{1}{N} \sum_{j=1}^N X_{ji}^2,$$

где j — номер опыта; i — номер фактора.

Для того чтобы получить уравнение регрессии в обычной форме

$$y = b_0 + b_1X_1 + b_2X_2 + \dots + b_nX_n + b_{12}X_1X_2 + \dots + b_{(n-1)n}X_{n-1}X_n + b_{11}X_1^2 + b_{22}X_2^2 + \dots + b_{nn}X_n^2,$$

* Величина α называется *звездным плечом*.



находят величину

$$b_0 = b_0^* - \frac{b_{11}}{N} \sum_{j=1}^N X_{j1}^2 - \dots - \frac{b_{nn}}{N} \sum_{j=1}^N X_{jn}^2.$$

В табл. 28 в качестве примера приведена матрица ортогонального центрального композиционного планирования (ОЦКП) для двух факторов, а на рис. 25 изображена схема проведения этих опытов.

Коэффициенты регрессии рассчитываются по следующим формулам:

$$b_0^* = \frac{1}{N} \sum_{j=1}^N y_j; \quad b_i = \frac{\sum_{j=1}^N X_{ji} y_j}{\sum_{j=1}^N (X_{ji})^2}, \quad i \neq 0;$$

$$b_{ik} = \frac{\sum_{j=1}^N X_{ji} X_{jk} y_j}{\sum_{j=1}^N (X_{ji} X_{jk})^2}, \quad i \neq k; \quad b_{ii} = \frac{\sum_{j=1}^N X_{ji}^* y_j}{\sum_{j=1}^N (X_{ji}^*)^2}.$$

Для расчета оценок дисперсий в определении коэффициентов регрессии используются следующие выражения:

$$s_{b_0^*}^2 = \frac{s_y^2}{N}; \quad s_{b_0}^2 = s_{b_0^*}^2 + \frac{n s_{b_{ii}}^2}{N} \sum_{j=1}^N X_{ji}^2;$$

$$s_{b_i}^2 = \frac{s_y^2}{\sum_{j=1}^N (X_{ji})^2}, \quad i \neq 0; \quad s_{b_{ii}}^2 = \frac{s_y^2}{\sum_{j=1}^N (X_{ji}^*)^2}; \quad s_{b_{ik}}^2 = \frac{s_y^2}{\sum_{j=1}^N (X_{ji} X_{jk})^2}, \quad i \neq k.$$

Коэффициент b_i считается значимым, если выполняется условие $|b_i| > s_{b_i} t$. Здесь t — значение критерия Стьюдента. Аналогично проверяется значимость остальных коэффициентов регрессии. Проверка адекватности уравнения регрессии осуществляется с помощью критерия Фишера, как и в случае полного факторного эксперимента.

КОНТУРНО-ГРАФИЧЕСКИЙ АНАЛИЗ

При изучении формы поверхности отклика в ряде случаев можно обойтись без составления математического описания функции отклика, используя приемы *контурно-графического анализа*. Сущность его состоит в определенном расположении опытов в факторном пространстве, получении дополнительной информации путем *линейной интерполяции* экспериментальных данных и построения на факторной плоскости (или на двумерных сечениях) *линий постоянного уровня* функции отклика.

Экспериментальные точки располагают таким образом, чтобы они охватывали всю область факторного пространства, представляющую интерес для исследователя. Схема, предложенная П. Берчем, предполагает постановку шести опытов (рис. 26). Один из них проводится в центре исследуемой области, а остальные — в вершинах пятиугольника. Схема В. Клеймана (рис. 27) требует постановки четырех опытов в вершинах прямоугольника и двух опытов на его оси симметрии.

Разумеется, схемы П. Берча и В. Клеймана не исчерпывают всех возможных вариантов расположения экспериментальных точек в факторном пространстве. Следует также иметь в виду, что расстояния между этими точками не должны быть слишком велики. В противном случае при большой кривизне и сложной форме поверхности отклика погрешности линейной интерполяции наложат заметный отпечаток на результаты исследований.

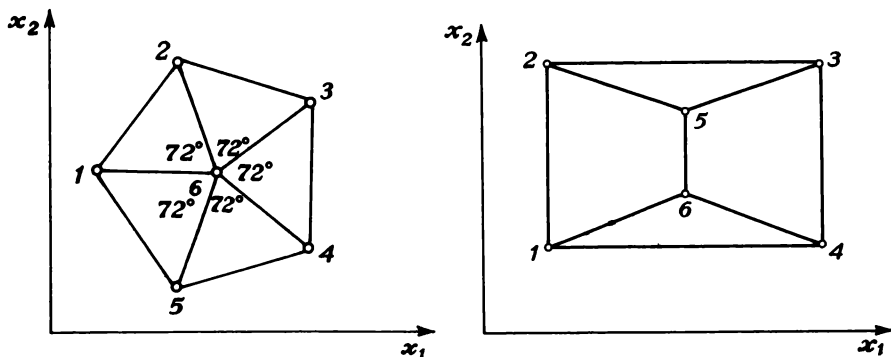


Рис. 26. Схема опытов для контурно-графического анализа, предложенная П. Берчем

Рис. 27. Схема расположения опытов, предложенная В. Клейманом

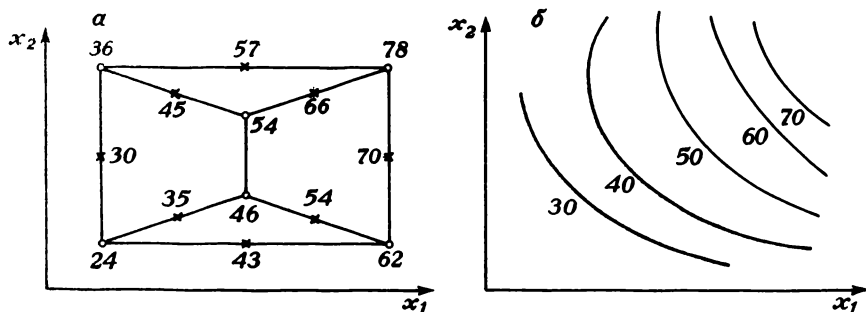


Рис. 28. Проведение контурно-графического анализа по схеме В. Клеймана

На рис. 28 показано, как осуществляется контурно-графический анализ по схеме В. Клеймана. Сначала проводят намеченную серию опытов. Затем соединяют линейными отрезками соседние точки и находят методом линейной интерполяции значения функции отклика в серединах этих отрезков (как средние арифметические значения результатов опытов в соединяемых экспериментальных точках). Наконец, проводят линии через точки с одинаковыми значениями функции отклика.

Контурно-графический анализ поверхности отклика по схеме П. Берча проводится аналогично.

Рассмотренной методикой можно пользоваться и при построении *двумерных сечений* поверхности отклика. Для этого выбирают два фактора, а значения всех остальных факторов поддерживают постоянными при проведении опытов.

Методом контурно-графического анализа можно строить на кальке линии постоянного уровня различных функций отклика, характеризующих технологический процесс. Совмещая координатные оси этих графиков и просматривая кальки на просвет, можно достаточно быстро выбрать оптимальные условия ведения процесса.

ЛАТИНСКИЕ КВАДРАТЫ И ПРЯМОУГОЛЬНИКИ

В химико-технологических исследованиях нашел применение метод латинских квадратов и прямоугольников. Его математическую основу составляет предложенная Р. Фишером схема *ортогональных латинских квадратов*. При таком планировании число исследуемых влияющих факторов должно быть равно числу уровней каждого фактора или меньше его.

Для случая, когда число факторов больше числа уровней каждого фактора, разработаны схемы *ортогональных латинских прямоугольников*.

Одним из основных признаков ортогональных планов является сочетание каждого уровня любого фактора *одинаковое* количество

раз со *всеми* уровнями остальных факторов. Схему ортогонального плана условно обозначают: $m \times n$, где m — число факторов, а n — число уровней, на которых они варьируются.

В табл. 29—34 приведены наиболее удобные для применения на практике схемы планирования 3×3 , 4×4 , 4×3 , 5×5 , 8×4 , 9×3 .

Количество опытов для реализации плана латинского прямоугольника равно произведению числа факторов на число уровней

Т а б л и ц а 29. Схема планирования для трех факторов (*A, B, C*) на трех уровнях

Номер опыта	<i>A</i>	<i>B</i>	<i>C</i>	Номер опыта	<i>A</i>	<i>B</i>	<i>C</i>
1	1	1	1	6	2	1	3
2	1	2	3	7	3	3	3
3	1	3	2	8	3	1	2
4	2	2	2	9	3	2	1
5	2	3	1				

Т а б л и ц а 30. Схема планирования для четырех факторов (*A, B, C, D*) на четырех уровнях

Номер опыта	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	Номер опыта	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
1	1	1	1	1	9	3	3	3	3
2	1	2	3	4	10	3	4	1	2
3	1	3	4	2	11	3	1	2	4
4	1	4	2	3	12	3	2	4	1
5	2	2	2	2	13	4	4	4	4
6	2	1	4	3	14	4	3	2	1
7	2	4	3	1	15	4	2	1	3
8	2	3	1	4	16	4	1	3	2

Т а б л и ц а 31. Схема планирования для четырех факторов (*A, B, C, D*) на трех уровнях

Номер опыта	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	Номер опыта	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
1	1	1	1	1	10	2	3	1	3
2	1	2	3	3	11	2	2	2	1
3	1	3	2	2	12	2	1	3	3
4	1	2	3	1	13	3	3	3	2
5	1	3	2	3	14	3	1	2	1
6	1	1	1	2	15	3	2	1	3
7	2	2	2	2	16	3	3	3	1
8	2	1	3	3	17	3	2	1	2
9	2	3	1	1	18	3	1	2	3

варьирования, за исключением плана 4×3 , где число опытов 18, а не 12.

Цифры на схемах планов обозначают номера уровней факторов. Нумерация уровней в общем случае произвольная.

Если требуется использовать меньшее число факторов, чем указано в схеме, то нужно просто отбросить «лишние» столбцы, оставив в остальном схему планирования без изменений.

Один из лишних столбцов в схеме планирования можно использовать для разделения серии опытов на более мелкие *блоки*, что позволит, например, проводить опыты с различными партиями

Таблица 32. Схема планирования для пяти факторов (*A, B, C, D, E*) на пяти уровнях

Номер опыта						Номер опыта					
	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>		<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>
1	1	1	1	1	1	14	3	1	4	2	5
2	1	2	3	4	5	15	3	2	1	5	4
3	1	3	5	2	4	16	4	4	4	4	4
4	1	4	2	5	3	17	4	5	1	2	3
5	1	5	4	3	2	18	4	1	3	5	2
6	2	2	2	2	2	19	4	2	5	3	1
7	2	3	4	5	1	20	4	3	2	1	5
8	2	4	1	3	5	21	5	5	5	5	5
9	2	5	3	1	4	22	5	1	2	3	4
10	2	1	5	4	3	23	5	2	4	1	3
11	3	3	3	3	3	24	5	3	1	4	2
12	3	4	5	1	2	25	5	4	3	2	1
13	3	5	2	4	1						

Таблица 33. Схема планирования для восьми факторов (*A, B, C, D, E, F, G, H*) на четырех уровнях

Номер опыта									Номер опыта								
	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>	<i>G</i>	<i>H</i>		<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>	<i>G</i>	<i>H</i>
1	1	1	1	1	1	1	1	1	17	3	3	3	3	3	3	3	3
2	1	2	1	3	4	4	2	3	18	3	4	3	1	2	2	4	1
3	1	3	2	1	3	4	4	2	19	3	1	4	3	1	2	2	4
4	1	4	2	3	2	1	3	4	20	3	2	4	1	4	3	1	2
5	1	1	3	4	4	2	3	2	21	3	3	1	2	2	4	1	4
6	1	2	3	2	1	3	4	4	22	3	4	1	4	3	1	2	2
7	1	3	4	4	2	3	2	1	23	3	1	2	2	4	1	4	3
8	1	4	4	2	3	2	1	3	24	3	2	2	4	1	4	3	1
9	2	2	2	2	2	2	2	2	25	4	4	4	4	4	4	4	4
10	2	1	2	4	3	3	1	4	26	4	3	4	2	1	1	3	2
11	2	4	1	2	4	3	3	1	27	4	2	3	4	2	1	1	3
12	2	1	1	4	1	2	4	3	28	4	1	3	2	3	4	2	1
13	2	2	4	3	3	1	4	1	29	4	4	2	1	1	3	2	3
14	2	1	4	1	2	4	3	3	30	4	3	2	3	4	2	1	1
15	2	4	3	3	1	4	1	2	31	4	2	1	1	3	2	3	4
16	2	3	3	1	4	1	2	4	32	4	1	1	3	2	3	4	2

Таблица 34. Схема планирования для девяти факторов (A, B, C, D, E, F, G, H, K) на трех уровнях

Номер опыта	A	B	C	D	E	F	G	H	K	Номер опыта	A	B	C	D	E	F	G	H	K
1	1	1	1	1	1	1	1	1	1	15	2	1	3	2	3	3	1	2	1
2	1	2	1	3	2	3	1	2	3	16	2	2	1	3	3	2	3	1	1
3	1	3	1	2	3	2	1	3	2	17	2	3	1	2	1	1	3	2	3
4	1	1	2	3	3	1	3	2	2	18	2	1	1	1	2	3	3	3	2
5	1	2	2	2	1	3	3	3	1	19	3	3	3	3	3	3	3	3	3
6	1	3	2	1	2	2	3	1	3	20	3	1	3	2	1	2	3	1	2
7	1	1	3	2	2	1	2	3	3	21	3	2	3	1	2	1	3	2	1
8	1	2	3	1	3	3	2	1	2	22	3	3	1	2	2	3	2	1	1
9	1	3	3	3	1	2	2	2	1	23	3	1	1	1	3	2	2	2	3
10	2	2	2	2	2	2	2	2	2	24	3	2	1	3	1	1	2	3	2
11	2	3	2	1	3	1	2	3	1	25	3	3	2	1	1	3	1	2	2
12	2	1	2	3	1	3	2	1	3	26	3	1	2	3	2	2	1	3	1
13	2	2	3	1	1	2	1	3	3	27	3	2	2	2	3	1	1	1	3
14	2	3	3	3	2	1	1	1	2										

сырья. Разделение серии опытов на отдельные блоки часто бывает необходимо, так как большое число экспериментов трудно поставить одновременно. Разделение на блоки осуществляется так, чтобы в каждом из них содержалось одинаковое число уровней каждого фактора.

Эффект определенного уровня фактора равен разности двух величин: среднего значения выходного параметра (\bar{y}_j) во всех случаях, когда фактор находился на уровне j , и среднего значения выходного параметра для всей серии опытов (\bar{y}).

Значение эффекта может получиться со знаком (+) или со знаком (-) в зависимости от того, увеличивает или уменьшает рассматриваемый уровень фактора выходной параметр (y) по сравнению со средним значением \bar{y} . Число рассчитываемых эффектов равно произведению числа факторов на число уровней.

По рассчитанным эффектам можно построить графики зависимостей выходного параметра (y) от каждого из рассматриваемых влияющих факторов. Эти зависимости несут определенную смысловую нагрузку и, кроме непосредственного интереса, позволяют определить *оптимальный уровень* каждого фактора на фоне средних значений других факторов. Сочетание таких оптимумов позволяет оценить область оптимальных режимов протекания технологического процесса.

Может оказаться, что оптимальный режим уже входил в состав проведенной системы опытов. Однако может быть, что такой опыт еще и не проводился.

Рассмотрим пример применения планов на основе латинских прямоугольников для подбора оптимального соотношения компонентов питательной среды для процесса ферментации.

Т а б л и ц а 35. Подбор оптимальных концентраций компонентов питательной среды

Номер опыта	Номер блока	Соевая мука	Кукурузная мука	Кукурузный экстракт	Мел	Сернокислый аммоний	Кашалотовый жир	Минеральная добавка	Выход <i>y</i>
1	1	1,5	2,0	0	0,4	0,2	4,0	0,5	1256
2	1	4,5	4,0	1,0	0,4	0,4	4,5	1,5	1849
3	1	4,5	3,0	0	0,6	4,0	4,0	1,2	1752
4	1	1,5	6,0	4,0	0,6	0,7	3,0	0,8	2113
5	1	2,0	3,0	2,0	0,9	0,7	1,0	1,5	1264
6	1	3,0	6,0	0,5	0,9	1,0	1,5	0,5	1560
7	1	3,0	3,0	2,0	1,2	0,4	4,0	0,8	1930
8	1	2,0	4,0	0,5	1,2	0,2	3,0	1,2	2473
9	2	2,0	3,0	0,5	0,6	0,4	1,5	0,8	2490
10	2	3,0	6,0	2,0	0,6	0,2	1,0	1,2	1011
11	2	3,0	2,0	0,5	0,4	0,7	3,0	1,5	1155
12	2	2,0	4,0	2,0	0,4	1,0	2,0	0,5	740
13	2	1,5	2,0	1,0	1,2	4,0	4,5	1,2	1503
14	2	4,5	4,0	0	1,2	0,7	1,0	0,8	2330
15	2	4,5	3,0	4,0	0,9	0,2	3,0	0,5	2040
16	2	1,5	6,0	0	0,9	0,4	2,0	1,5	2550
17	3	3,0	4,0	1,0	0,9	0,7	2,0	1,2	2100
18	3	2,0	2,0	0	0,9	1,0	3,0	0,8	1443
19	3	4,0	6,0	4,0	1,2	0,4	4,0	0,5	1327
20	3	3,0	3,0	2,0	1,2	0,2	1,5	1,5	1447
21	3	4,5	6,0	0,5	0,4	0,2	4,0	0,8	980
22	3	1,5	4,0	0,5	0,6	1,0	1,0	1,5	1050
23	3	4,5	2,0	2,0	0,6	0,7	1,5	0,5	926
24	3	4,5	6,0	2,0	1,5	1,0	3,0	1,5	572
25	4	1,5	4,0	2,0	0,9	0,2	1,5	0,8	890
26	4	4,5	2,0	0,5	0,9	0,4	1,0	1,2	1193
27	4	3,0	4,0	0	0,6	0,4	3,0	0,5	1397
28	4	2,0	4,0	2,0	0,6	0,7	1,5	0,5	928
29	4	2,0	4,0	1,0	0,6	0,2	4,0	1,5	1528
30	4	2,0	6,0	0	0,4	0,7	1,5	1,2	500
31	4	1,5	3,0	0,5	1,5	0,7	2,0	0,5	3240
32	4	3,0	3,0	1,0	0,4	1,0	1,0	0,8	1160

В опытах варьировались 7 компонентов среды, каждый из которых — на четырех уровнях. Опыты были разделены на 4 блока по 8 вариантов среды в каждом блоке (табл. 35). На основании полученных данных рассчитаны эффекты всех уровней факторов. Результаты приведены в табл. 36. По данным этой таблицы легко построить графики зависимостей эффекта от концентрации каждого из варьируемых компонентов и определить их оптимальные значения.

Если для какого-либо из компонентов оптимальным оказался самый высокий или самый низкий уровень фактора (например, в данном случае — для соевой муки и мела), то целесообразно поставить новую серию опытов, в которой диапазоны изменения концентраций этих компонентов расширятся в соответствующую сторону. При этом, конечно, можно использовать схемы с меньшим числом уровней каждого фактора.

В рассматриваемом эксперименте средний выход антибиотика составил 1522 ед./мл.

Из табл. 35 следует, что в оптимальных соотношениях взяты компоненты среды в опыте 31, и эта среда дает выход (*y*) более чем в два раза выше среднего.

Т а б л и ц а 36. Влияние различных концентраций компонентов среды на выход антибиотика

Соевая мука	%	<u>1,5</u>	2,0	3	4,5
	эффект	<u>168</u>	-51	-52	-67
Кукурузная мука	%	2	<u>3</u>	4	6
	эффект	-155	<u>267</u>	92	-195
Кукурузный экстракт	%	0	<u>0,5</u>	1	2
	эффект	62	<u>246</u>	181	-490
Мел	%	0,4	0,6	0,9	<u>1,2</u>
	эффект	-452	11	108	<u>331</u>
Серноокислый аммоний	%	0,2	<u>0,4</u>	<u>0,7</u>	1
	эффект	-69	<u>185</u>	<u>182</u>	-299
Кашалотовый жир	%	1	1,5	<u>2</u>	3
	эффект	-198	-126	<u>331</u>	-8
Минеральная добавка	%	0,5	<u>0,8</u>	1,2	1,5
	эффект	39	<u>145</u>	-90	-95

Если опыты выполняются тщательно, то с помощью планов ортогональных латинских прямоугольников можно быстро получить ответ на вопрос об оптимальных условиях ведения процесса. Если же кривые «эффект — концентрация» имеют «скачущий» характер, то это свидетельствует о том, что результаты опытов недостаточно достоверны. Обычно при этом и абсолютные величины эффектов невелики.

При проведении опытов на одной среде оценка дисперсии вычисляется по формуле

$$s_y^2 = \frac{\sum_{i=1}^m (y_i - \bar{y})^2}{k-1},$$

а на нескольких средах

$$s_y^2 = \frac{\sum_{i=1}^n \sum_{j=1}^m (y_{ij} - \bar{y})^2}{n(k-1)}.$$

Здесь k — число повторений опыта; n — количество вариантов сред, на которых ставились эксперименты.

Доверительная оценка ϵ для любого эффекта определяется по формуле

$$\epsilon = t s_y \sqrt{1/p},$$

где p — константа, равная числу опытов, деленному на число уровней каждого фактора; t — критерий Стьюдента при числе степеней свободы $(k-1)$ — при первом способе расчета дисперсии и $n(k-1)$ — при втором.

Если абсолютная величина эффекта превышает ϵ , то он *значим*, в противном случае — *незначим*.

Незначимость всех эффектов для какого-либо фактора может быть вызвана разными причинами, например:

- 1) взяты малые интервалы между уровнями фактора;
- 2) проведено малое число параллельных опытов при большой дисперсии;
- 3) рассматриваемый фактор по существу не влияет на процесс;
- 4) взятый диапазон уровней фактора является оптимальным.

При всех обстоятельствах опыты целесообразно дублировать.

Планирование эксперимента с помощью латинских прямоугольников дает возможность изучить влияние выбранных факторов как на один, так и на несколько выходных параметров объекта исследований. Важно отметить, что при планировании эксперимента можно использовать как *качественные*, так и *количественные* факторы.

СОВМЕЩЕННЫЙ ПЛАН

В практике химических исследований возникает необходимость получать уравнение регрессии, когда наряду с количественными используются также и качественные факторы. Для этой цели удобно использовать планы эксперимента, построенные на основе *совмещения* полного или дробного факторного эксперимента с латинскими квадратами. Как мы уже знаем, латинским квадратом называется квадратная таблица из $n \times n$ элементов, причем каждый элемент встречается один раз в каждой строке и один раз в каждом столбце. Например, латинский квадрат 4×4 имеет вид:

<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
<i>B</i>	<i>D</i>	<i>A</i>	<i>C</i>
<i>C</i>	<i>A</i>	<i>D</i>	<i>B</i>
<i>D</i>	<i>C</i>	<i>B</i>	<i>A</i>

Латинский квадрат можно совмещать с факторным планом, когда *один* из параметров не имеет количественной оценки.

Рассмотрим пример такого планирования эксперимента. Пусть требуется найти оптимальный режим упрочнения спеченного ситалла (кристаллизованное стекло), если его прочность зависит от следующих факторов: x_1 — пористости стекла, %; x_2 — времени обработки в соли, мин; x_3 — концентрации соли, %; x_4 — температуры обжига, °С; x_5 — времени обжига, ч; x_6 — вида соли.

Для решения этой задачи воспользуемся планом полного факторного эксперимента типа 2^4 и совместим его с латинским квадратом 4×4 (табл. 37). Принимая во внимание, что латинский квадрат 4×4 имеет вид:

		$X_1 (+1)$		$X_1 (-1)$	
		$X_2 (+1)$	$X_2 (-1)$	$X_2 (+1)$	$X_2 (-1)$
$X_4 (+1)$	$X_3 (+1)$	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
	$X_3 (-1)$	<i>C</i>	<i>D</i>	<i>A</i>	<i>B</i>
$X_4 (-1)$	$X_3 (+1)$	<i>B</i>	<i>A</i>	<i>D</i>	<i>C</i>
	$X_3 (-1)$	<i>D</i>	<i>C</i>	<i>B</i>	<i>A</i>

Т а б л и ц а 38. Реплика типа 2^{5-1} , совмещенная с латинским квадратом 4×4

Номер опыта	X_1	X_2	X_3	X_4	X_5	X_6	Номер опыта	X_1	X_2	X_3	X_4	X_5	X_6
	1	-1	-1	-1	-1	-1		A	9	-1	+1	-1	-1
2	+1	+1	+1	-1	-1	A	10	+1	+1	+1	-1	+1	B
3	-1	-1	+1	+1	+1	D	11	-1	+1	+1	+1	-1	C
4	-1	+1	-1	+1	+1	A	12	-1	-1	-1	+1	-1	B
5	+1	+1	-1	-1	-1	D	13	+1	-1	-1	-1	+1	C
6	+1	-1	-1	+1	+1	D	14	+1	+1	-1	+1	-1	C
7	-1	+1	+1	-1	-1	D	15	-1	-1	+1	-1	+1	C
8	+1	+1	+1	+1	+1	A	16	+1	-1	+1	+1	-1	B

мая $X_5 = X_1 X_2 X_3$, возьмем $1/2$ реплики от полного факторного эксперимента типа 2^5 и также совместим ее с латинским квадратом 4×4 (табл. 38).

Основные характеристики плана экспериментов представлены в табл. 39.

В результате обработки экспериментальных данных получено уравнение регрессии

$$y = 1218,75 + 42,88X_1 + 37,01X_2 + 145,25X_3 + 126,53X_4 + 125,75X_5,$$

которое адекватно описывает процесс упрочнения ситалла. С его помощью можно осуществлять оптимизацию режима упрочнения, пользуясь методом крутого восхождения.

Т а б л и ц а 39. Характеристики плана экспериментов

Характеристика	Факторы				
	x_1	x_2	x_3	x_4	x_5
Верхний уровень	30	61	31	1250	10
Нижний уровень	12	1	1	1150	1
Основной уровень	21	31	16	1200	5,5
Интервал варьирования	9	30	15	50	4,5

КОМПЬЮТЕР-ТРУЖЕНИК

ПОМОЩНИК УЧИТЕЛЯ

Персональные компьютеры не только стали объектами изучения в учебных заведениях, но и нашли применение в качестве средств активизации учебно-познавательного процесса. В память ЭВМ можно заложить большое количество информации из различных разделов химии и химической технологии. По желанию обучаемого эта информация выдаетея на экран дисплея или печатающее устройство. К настоящему времени разработано значительное количество автоматизированных обучающих и контролирующих учебных курсов, ориентированных на персональные компьютеры.

Здесь мы рассмотрим программу на языке БЕЙСИК для контроля знаний по номенклатуре неорганических веществ. В ней широко используются символьные переменные. Вот текст этой программы:

```

10 REM ПРОГРАММА КОНТРОЛЯ ЗНАНИЙ ПО ХИМИИ
20 DIM A$(20),B$(20)
30 REM ВВОД ИНФОРМАЦИИ
40 FOR I=1 TO 20
50 READ A$(I),B$(I)
60 NEXT I
70 LET S=0
80 REM КОМПЬЮТЕР ЗАДАЕТ ВОПРОСЫ
90 FOR I=1 TO 20
100 PRINT "ФОРМУЛА ХИМИЧЕСКОГО СОЕДИНЕНИЯ :";A$(I)
110 INPUT "ВВЕДИТЕ НАЗВАНИЕ ЭТОГО ВЕЩЕСТВА :";D$
120 IF D$=B$(I) THEN GOTO 140
130 PRINT "ВАШ ОТВЕТ ОШИБОЧНЫЙ " ; GOTO 150
140 PRINT "ВАШ ОТВЕТ ПРАВИЛЬНЫЙ " ; LET S=S+1
150 NEXT I
160 REM КОМПЬЮТЕР ВЫСТАВЛЯЕТ ОЦЕНКУ ЗНАНИЙ
170 LET P=INT(S/20*5+0.5)
180 IF P>=3 THEN GOTO 210
190 PRINT "ВАША ОЦЕНКА - 2 "
200 STOP
210 PRINT "ВАША ОЦЕНКА ";P
220 STOP
230 REM ВОПРОСЫ И ОТВЕТЫ
240 DATA "CA(НСО3)2"," ГИДРОКАРБОНАТ КАЛЬЦИЯ"
250 DATA "H2SO4"," СЕРНАЯ КИСЛОТА"

```

В начале работы программы в память ЭВМ вводится исходная информация: А \times — массив из 20 вопросов и В \times — массив из

20 правильных ответов. В строках 240 и 250 показаны только два первых элемента каждого из этих массивов. Остальные элементы Вам следует расположить в последующих строках.

Вопросы один за другим высвечиваются на экране дисплея. Сначала появляется формула химического соединения $\text{Ca}(\text{HCO}_3)_2$, и компьютер предлагает назвать это вещество. Если Вы верно назвали соединение, т. е. набрали на клавиатуре текст «гидрокарбонат кальция» и нажали клавишу ВК, то на экране появляются слова «ответ правильный». В противном случае будет написано: «ответ ошибочный».

После того как весь список вопросов будет исчерпан, компьютер подсчитает количество правильных ответов и выставит оценку.

Программу нетрудно видоизменить и приспособить для проверки знания формул химических соединений. Вот ее текст:

```
10 REM ПРОГРАММА ПРОВЕРКИ ЗНАНИЯ ФОРМУЛ ВЕЩЕСТВ
20 DIM A*(20),B*(20)
30 REM ВВОД ИНФОРМАЦИИ
40 FOR I=1 TO 20
50 READ A*(I),B*(I)
60 NEXT I
70 LET S=0
80 REM КОМПЬЮТЕР ЗАДАЕТ ВОПРОСЫ
90 PRINT "ФОРМУЛЫ ПИШИТЕ БЕЗ ПРОБЕЛОВ , БОЛЬШИМИ БУКВАМИ"
100 FOR I=1 TO 20
110 PRINT "ВЕЩЕСТВО " ; A*(I)
120 INPUT "ВВЕДИТЕ ФОРМУЛУ ЭТОГО ВЕЩЕСТВА " ; D*
130 IF D*=B*(I) THEN GOTO 150
140 PRINT "ВАШ ОТВЕТ ОШИБОЧНЫЙ " ; GOTO 160
150 PRINT "ВАШ ОТВЕТ ПРАВИЛЬНЫЙ " ; LET S=S+1
160 NEXT I
170 REM КОМПЬЮТЕР ВЫСТАВЛЯЕТ ОЦЕНКУ ЗНАНИЙ
180 LET P=INT(S/20*5+0.5)
190 IF P>=3 THEN GOTO 220
200 PRINT "ВАША ОЦЕНКА - 2 "
210 STOP
220 PRINT "ВАША ОЦЕНКА " ; P
230 STOP
240 REM ВОПРОСЫ И ОТВЕТЫ
250 DATA "СЕРНАЯ КИСЛОТА " ,"H2SO4"
260 DATA "СУЛЬФАТ МЕДИ (2) " ,"CUSO4"
```

Эта программа работает аналогично предыдущей. В строках 250 и 260 записаны только первые два вопроса и правильных ответа. Остальную входную информацию следует поместить в 270-й и последующих строках.

Можно построить многие другие программы для контроля знаний по различным разделам химии.

ИТОГИ ОЛИМПИАДЫ

При решении различных задач приходится упорядочивать элементы массивов в порядке возрастания или убывания. Для этого воспользуемся следующим алгоритмом. Вначале найдем наиболь-

ший элемент массива и поставим его на первое место. Затем найдем наибольший из оставшихся элементов и поменяем его местами со вторым элементом. После нахождения наибольшего из последних двух элементов и размещения его на предпоследнем месте упорядочивание заканчивается.

С помощью этого алгоритма можно, например, упорядочить результаты (баллы), полученные победителями и призерами олимпиады по химии, проводимой в школе. Одновременно с этим будут определены места, занятые лидерами соревнований.

С помощью компьютера результаты олимпиады можно вывести на печать в виде таблицы. Для этого нужно уметь печатать (или выводить на экран дисплея) горизонтальные и вертикальные линии, печатать содержание граф.

Вывод горизонтальной линии в позиции с I по N осуществляется операторами

```
FOR I=1 TO N : PRINT "-" ; : NEXT I : PRINT
```

Здесь последний оператор PRINT обеспечивает перевод каретки на следующую строку.

Печать вертикальных линий производится, благодаря выводам в одни и те же позиции каждой строки символа «I» или «!».

Ниже приводится программа для подведения итогов олимпиады и распределения первых десяти мест:

```
10 REM ИТОГИ ОЛИМПИАДЫ ПО ХИМИИ
20 DATA "ГЕРАСИМОВ",9.0,"ЛУКИН",10.3,"ПЕТРОВ",9.6
30 DATA "СМИРНОВ",12.5,"ИВАНОВ",11.7,"АЛЕКСЕЕВ",12.1
40 DATA "ЕФИМОВА",11.2,"СЕМЕНОВ",10.9,"КРАСНОВ",8.8
50 DATA "КУЗНЕЦОВА",9.4
60 DIM A*(10),R(10)
70 REM ВВОД ФАМИЛИИ И РЕЗУЛЬТАТОВ (В БАЛЛАХ)
80 FOR I=1 TO 10 : READ A*(I),R(I) : NEXT I
90 GOSUB 1000
100 PRINT TAB(10);"РЕЗУЛЬТАТЫ ОЛИМПИАДЫ ПО ХИМИИ"
110 GOSUB 1100
120 PRINT TAB(5);"I МЕСТО I"; TAB(20);"ФАМИЛИЯ";
130 PRINT TAB(35);"I";TAB(38); "РЕЗУЛЬТАТ";
140 PRINT TAB(50);"I"
150 GOSUB 1100
160 FOR I=1 TO 10
170 PRINT TAB(5);"I";TAB(8);I;TAB(13);"I";
180 PRINT TAB(20);A*(I);TAB(35);"I";
190 PRINT USING "#####.#";R(I);
195 PRINT TAB(50);"I"
200 NEXT I
210 GOSUB 1100
220 STOP
1000 REM УПОРЯДОЧЕНИЕ МАССИВА БАЛЛОВ ПО УБЫВАНИЮ
1010 FOR I=1 TO 9 : LET P=R(I) : LET K=I
1020 FOR J=I+1 TO 10
1030 IF R(J)<=P GOTO 1050
1040 LET P=R(J) : LET K=J
1050 NEXT J : LET R(K)=R(I) : LET R(I)=P
1060 LET Q#=A*(I) : LET A*(I)=A*(K)
1070 LET A*(K)=Q#
```



```

1080 NEXT I : RETURN
1100 PRINT TAB(5); "-";
1110 FOR K=1 TO 45 : PRINT "-"; : NEXT K
1120 PRINT : RETURN

```

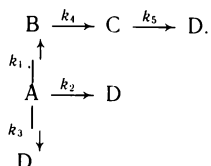
В результате выполнения программы будет напечатана следующая таблица:

РЕЗУЛЬТАТЫ ОЛИМПИАДЫ ПО ХИМИИ		
МЕСТО I	ФАМИЛИЯ	I РЕЗУЛЬТАТ
1	СМИРНОВ	12.5
2	АЛЕКСЕЕВ	12.1
3	ИВАНОВ	11.7
4	ЕФИМОВА	11.2
5	СЕМЕНОВ	10.9
6	ЛУКИН	10.3
7	ПЕТРОВ	9.6
8	КУЗНЕЦОВА	9.4
9	ГЕРАСИМОВ	9.0
10	КРАСНОВ	8.8

Первый опыт печати таблиц и упорядочения массивов может в дальнейшем оказаться полезным при обработке на ЭВМ результатов различных химических экспериментов.

МОДЕЛЬ ХИМИЧЕСКОЙ РЕАКЦИИ

Рассмотрим схему химической реакции, состоящей из нескольких стадий:



Механизм этой реакции включает последовательные и параллельные стадии. Здесь А — это исходное вещество, В — промежуточный продукт, С — требуемый (целевой) продукт, а D — нежелательный побочный продукт. Предполагается, что константы скоростей отдельных стадий зависят от температуры по закону Аррениуса:

$$k_i = k_{0i} e^{-\frac{E_i}{RT}}$$

где k_{0i} — предэкспоненциальный множитель; E_i — энергия активации; R — универсальная газовая постоянная; T — температура, К.

Будем считать, что рассматриваемая реакция протекает в трубчатом химическом реакторе с идеальным вытеснением потока реакционной массы.

Заметим, что целевой продукт С сначала образуется из вещества В, а затем превращается в вещество D. Следовательно, если время пребывания реакционной массы в химическом реакторе мало, то концентрация вещества С в выходном потоке тоже мала. С увеличением времени пребывания концентрация целевого продукта увеличивается. Однако начиная с некоторого момента, концентрация вещества С на выходе реактора начинает уменьшаться, несмотря на увеличение времени пребывания. Отсюда следует, что существует определенное значение времени пребывания реакционной смеси в аппарате, при котором концентрация вещества С в выходном потоке максимальна. Это значение времени пребывания принято называть оптимальным.

Температура также существенно влияет на производительность реактора по целевому продукту. Поэтому задача оптимизации состоит в том, чтобы найти распределение температур по длине трубчатого реактора, которое максимизирует концентрацию целевого продукта в выходном потоке.

Для оптимизации технологического процесса построим математическую модель трубчатого реактора, пользуясь законом действия масс и принципом независимости протекания отдельных стадий химической реакции. Модель имеет вид системы обыкновенных дифференциальных уравнений:

$$\frac{dx_1}{dt} = -(k_1 + k_2 + k_3) x_1; \quad \frac{dx_2}{dt} = k_1 x_1 - k_4 x_2; \quad \frac{dx_3}{dt} = k_4 x_2 - k_5 x_3$$

с начальными условиями $x_1(0) = x_{01}$, $x_2(0) = 0$, $x_3(0) = 0$, где k_i определяются уравнением Аррениуса. Концентрации продуктов А, В и С обозначены через x_1 , x_2 и x_3 , соответственно.

Задача оптимизации заключается в нахождении распределения температур $T(t)$ и значения времени пребывания t_j , при которых концентрация $x_3(t_j)$ максимальна. Верхний предел температуры T_{\max} обычно ограничивается технологическими соображениями.

Решить дифференциальные уравнения математической модели можно на персональном компьютере с помощью программы

```

10 REM МОДЕЛИРОВАНИЕ ХИМИЧЕСКОЙ РЕАКЦИИ
20 INPUT " ВВЕДИТЕ КОЭФФИЦИЕНТЫ A1 , A2"; A1,A2
30 LET C1=2.11E5 ; LET C2=4.17E4 ; LET C3=3.72E4
40 LET C4=6.89E3 ; LET C5=8.09E3
50 LET T1=1 ; LET D=0.02
60 LET E1=1600 ; LET E2=14000 ; LET E3=15000
70 LET E4=10000 ; LET E5=15000
80 LET T2=0 ; LET T=A1 ; LET X1=1 ; LET X2=0
90 LET X3=0 ; LET R=1.9865
100 LET K1=C1*EXP(-E1/R/T)
110 LET K2=C2*EXP(-E2/R/T)
120 LET K3=C3*EXP(-E3/R/T)
130 LET K4=C4*EXP(-E4/R/T)
140 LET K5=C5*EXP(-E5/R/T)
150 LET X1=X1-(K1+K2+K3)*X1*D
160 LET X2=X2+(K1*X1-K4*X2)*X1*D

```

```

170 LET X3=X3+(K4*X2-K5*X3)*D
180 LET T2=T2+D : LET T=A1-A2*T2
190 IF T2>T1 THEN 220
200 PRINT T2,T,X3
210 GOTO 100
220 STOP : END

```

Интегрирование дифференциальных уравнений осуществляется *методом Эйлера*.

Программа предназначена для расчета профиля температуры по длине трубчатого химического реактора, описываемого линейным уравнением

$$T = a_1 - a_2 t,$$

где a_1 — температура потока на входе в реактор; a_2 — скорость понижения температуры; t — время пребывания элемента потока в аппарате.

Результаты расчетов при $a_1 = 760$ К и $a_2 = 60$ К/мин приведены в табл. 40.

Из табл. 40 следует, что концентрация целевого продукта x_3 сначала увеличивается, а затем, достигнув наибольшего значения 0,33, через некоторое время начинает уменьшаться. Очевидно, для рассматриваемых условий ведения процесса оптимальное время пребывания в реакторе $t_f = 0,5$ мин.

Т а б л и ц а 40. Зависимость температуры реакционной массы (T) и концентрации целевого продукта (x_3) от времени пребывания

t , мин	T , К	x_3	t , мин	T , К	x_3
0,02	758,8	0,015	0,42	734,8	0,324
0,06	756,4	0,066	0,46	732,4	0,328
0,10	754,0	0,124	0,50	730,0	0,330
0,14	751,6	0,176	0,54	727,6	0,330
0,18	749,2	0,219	0,58	725,2	0,330
0,22	746,8	0,252	0,62	722,8	0,330
0,26	744,4	0,277	0,66	720,4	0,328
0,30	742,0	0,296	0,70	718,0	0,327
0,34	739,6	0,309	0,72	716,8	0,326
0,38	737,2	0,318	0,74	715,6	0,325

ЭКОЛОГИЧЕСКИЕ РАСЧЕТЫ

Проблема обеспечения чистоты среды обитания приобрела глобальное значение. К сожалению, химические заводы являются одними из наиболее интенсивных источников загрязнения воздушного и водного бассейнов.

Атмосфера, окружающая Землю, подразделяется на три зоны: нижнюю — тропосферу, среднюю — стратосферу и верхнюю — ионосферу. В тропосфере сосредоточено примерно 80 % массы воздуха.

В нашей стране большое внимание уделяется проблемам *экологии*, разрабатываются научно обоснованные меры, обеспечивающие условия оптимального взаимодействия человека с окружающей средой. При проектировании химических заводов решаются вопросы минимизации выбросов вредных веществ. В СССР создана государственная служба контроля за загрязнением атмосферы. Предельно допустимая концентрация (ПДК) является критерием, ограничивающим содержание вредных веществ в воздухе.

Одна из методик расчета концентраций вредных веществ в *приземном слое атмосферы* разработана Главной Геофизической обсерваторией им. А. И. Воейкова. Эта методика позволяет оценить максимальную разовую концентрацию при *опасной скорости ветра*, т. е. такой скорости, при которой возникает наибольшее значение приземной концентрации вещества, поступающего из одиночного источника. Если имеется несколько источников выбросов вредного вещества, то рассчитывается средняя для рассматриваемой совокупности источников опасная скорость ветра. Расчет рассеивания вредных веществ, содержащихся в выбросах из труб химических заводов, обычно ведут на электронных вычислительных машинах. В результате расчетов находят условия, при которых приземная концентрация вредных веществ, обусловленная выбросами, была бы меньше ПДК.

Расчеты по указанной методике осуществляются с помощью приведенных ниже формул.

Максимальная приземная концентрация вредных веществ C_{M_i} (мг/м³) в выбросе газозвушной смеси из одиночного источника с круглым устьем при неблагоприятных метеорологических условиях и опасной скорости ветра равна

$$C_{M_i} = \frac{AM_i F m_i}{H_i^2 \sqrt[3]{V_i \Delta T_i}}.$$

Здесь A — климатологический параметр; M_i — количество вредного вещества, выбрасываемого в атмосферу, г/с; F — параметр, характеризующий оседание примесей; H_i — высота трубы источника выброса вредных веществ в атмосферу, м; V_i — объем газозвушной смеси, выбрасываемой в атмосферу из источника загрязнения в единицу времени, м³/с; $\Delta T_i = T_i - T_b$ — разность между температурой выбрасываемой газозвушной смеси T_i и температурой воздуха T_b .

Коэффициент m_i , учитывающий условия выхода газозвушной смеси из устья источника загрязнения, рассчитывается по формуле

$$m_i = \frac{1}{0,67 + 0,1 \sqrt{\bar{f}_i} + 0,34 \sqrt[3]{\bar{f}_i}};$$

$$\bar{f}_i = 10^3 \frac{\omega_i^2 D_i}{H_i^2 \Delta T_i}; \quad \omega_i = \frac{4V_i}{\pi D_i^2},$$

где ω_i — средняя скорость газозвушной смеси на выходе из устья (отверстия) трубы, м/с; D_i — диаметр отверстия трубы, м.

Максимальная приземная концентрация вредного вещества $C_{м u_i}$ (мг/м³) в выбросе газозвушной смеси из одиночного источника с круглым отверстием при неблагоприятных метеорологических условиях и средней для данной совокупности источников опасной скорости ветра равна

$$C_{м u_i} = r_i C_{M_i};$$

$$r_i = 0,67 \frac{u_{м.с}}{u_{M_i}} + 1,67 \left(\frac{u_{м.с}}{u_{M_i}} \right)^2 - 1,34 \left(\frac{u_{м.с}}{u_{M_i}} \right)^3$$

при условии, что $u_{м.с}/u_{M_i} \leq 1$. Если же $u_{м.с}/u_{M_i} > 1$, то

$$r_i = \frac{3u_{м.с}/u_{M_i}}{2 \left(\frac{u_{м.с}}{u_{M_i}} \right)^2 - \frac{u_{м.с}}{u_{M_i}} + 2}.$$

Средняя для рассматриваемой совокупности источников опасная скорость ветра $u_{м.с}$ (м/с) рассчитывается по формуле:

$$u_{м.с} = \frac{\sum_{i=1}^n u_{M_i} C_{M_i}}{\sum_{i=1}^n C_{M_i}},$$

где $u_{м.с}$ — опасная скорость ветра, м/с;

При расчетах принимают:

- 1) $u_{M_i} = 0,5$ при условии, что $u_{M_i} \leq 0,5$;
- 2) $u_{M_i} = v_{M_i}$ при условии, что $0,5 < v_{M_i} \leq 2$;
- 3) $u_{M_i} = v_{M_i} (1 + 0,12\sqrt{f_i})$ при условии, что $v_{M_i} > 2$.

Здесь $v_{M_i} = 0,65^3 \sqrt{V_i \Delta T_i / H_i}$.

Суммарная концентрация $C_{м u}$ (мг/м³) примеси при неблагоприятных метеорологических условиях и средней для данной совокупности источников опасной скорости ветра равна

$$C_{м u} = \sum_{i=1}^n C_{м u_i}.$$

Расстояние от *одиночного* источника выброса x_m (м), на котором при опасной скорости ветра достигается максимальная приземная концентрация вредных веществ, оценивается величиной

$$x_m = 20H,$$

где H — высота трубы.

Расстояние от источника выброса, на котором при средней для данной совокупности источников опасной скорости ветра при-

земная концентрация вредных веществ достигает максимального значения, равно

$$x_{m i} = p_i x_m.$$

Здесь

$$p_i = 8,43 \left(1 - \frac{u_{m.c}}{u_{M_i}} \right)^5 + 1,$$

если $0,25 < u_{m.c}/u_{M_i} \leq 1$, и

$$p_i = 0,32 \frac{u_{m.c}}{u_{M_i}} + 0,68,$$

если $u_{m.c}/u_{M_i} > 1$.

Таким образом, для расчета максимальной приземной концентрации вредного вещества, обусловленной выбросами загрязнений из труб химического завода, необходимо иметь данные о количестве и типе промышленных установок, числе труб, из которых осуществляется выброс нагретой смеси, содержащей вредные примеси, об объеме выбрасываемой газовой смеси и количестве выбрасываемых в атмосферу примесей.

Для расчета величины $C_{m i}$ удобно пользоваться приведенной ниже программой, которая написана на языке БЕЙСИК. Величины, используемые для вычислений, представлены в табл. 41.

```

10 REM ЭКОЛОГИЧЕСКИЕ РАСЧЕТЫ
20 DIM A1(10), A2(10), A3(10), A4(10), B1(10)
30 DIM V1(10), V2(10), D1(10), C1(10), R(10)
40 DIM U1(10), U2(10), U3(10), P(10), X1(10)
50 DIM W(10)
60 INPUT "КЛИМАТОЛОГИЧЕСКИЙ ПАРАМЕТР = "; A
70 INPUT "НАИМЕНЬШАЯ ВЫСОТА ТРУБЫ = "; H1
80 INPUT "НАИБОЛЬШАЯ ВЫСОТА ТРУБЫ = "; H2
90 INPUT "КОЭФФИЦИЕНТ, УЧИТЫВАЮЩИЙ ОСЕДАНИЕ ПРИМЕСЕЙ = "; F
100 INPUT "ДИАМЕТР УСТЯ ТРУБЫ = "; D2
110 INPUT "КОЛИЧЕСТВО ИСТОЧНИКОВ ВЫБРОСА = "; N1
120 INPUT "КОЛИЧЕСТВО ЗНАЧЕНИЙ ВЫСОТЫ ТРУБЫ = "; N2
130 FOR I=1 TO N1
140 INPUT "КОЛИЧЕСТВО ВЫБРОСОВ УСТАНОВОК I-ГО ТИПА = "; A1(I)
150 INPUT "КОЛИЧЕСТВО УСТАНОВОК I-ГО ТИПА = "; A2(I)
160 INPUT "ОБЪЕМ ВЫБРОСОВ ИЗ УСТАНОВКИ I-ГО ТИПА = "; V1(I)
170 INPUT "РАЗНОСТЬ ТЕМПЕРАТУР ВЫБРОСА И ВОЗДУХА = "; D1(I)
180 NEXT I
190 LET D3=(H2-H1)/N2
200 FOR J=1 TO N2
210 LET H3=H1+(J-1)*D3 : LET S=0 : LET S1=0
220 FOR I=1 TO N1
230 LET W(I)=4*V(I)/(3.14*D2^2*A2(I))
240 LET AP(I)=1000*W(I)^2*D2/(H3^2*D1(I))
250 LET B1(I)=1/(0.67+0.1*SQR(A4(I))+0.34*A4(I)^(1/3))
260 LET Z=(A2(I)/(V1(I)*D1(I)))^(1/3)*A1(I)
270 LET C1(I)=(A/H3^2)*F*B1(I)*Z
280 LET V2(I)=0.65*(V1(I)*D1(I)/A2(I)/H3)^(1/3)
290 LET U3(I)=V2(I)
300 IF V2(I)<0.5 THEN LET U3(I)=0.5
310 IF V2(I)>2 THEN LET U3(I)=(1+0.12*SQR(A4(I)))

```

```

320 LET B1=B1+C1(I)
330 LET B=B+U3(I)*C1(I)
340 NEXT I : NEXT J
350 LET U4=B/S1
360 LET B2=0
370 FOR I=1 TO N1
380 LET A3(I)=U4/U3(I)
390 IF A3(I)>1 THEN GOTO 420
400 LET R(I)=3*A3(I)/(2*(A3(I)^2)-A3(I)+2)
410 GOTO 430
420 LET R(I)=0.67*A3(I)+1.67*(A3(I)^2)-1.34*(A3(I)^3)
430 LET U2(I)=R(I)*C1(I)
440 LET B2=B2+U2(I)
450 NEXT I
460 FOR I=1 TO N1
470 IF A3(I)>1 THEN GOTO 500
480 LET P(I)=0.32*A3(I)+0.68
490 GOTO 510
500 LET P(I)=0.43*((1-A3(I))^5+1)
510 LET X1(I)=20*N3*P(I)
520 NEXT I
530 PRINT "ВЫСОТА ТРУБЫ =" ; N3
540 PRINT " РАССТОЯНИЕ "
550 FOR I=1 TO N1
560 PRINT X1(I)
570 NEXT I
580 PRINT "ПРИЗЕМНАЯ КОНЦЕНТРАЦИЯ =" ; B2
590 STOP
600 END

```

Т а б л и ц а 41. Величины, используемые для вычислений

Параметры	Обозначения в программе	Единицы измерения
Климатологический параметр	A	$c^{2/3} \cdot \text{мг} \cdot (\text{°C})^{1/3} / \text{г}$
Наименьшая из высот труб	H1	м
Наибольшая из высот труб	H2	м
Коэффициент, учитывающий оседание примесей	F	—
Диаметр устья трубы	D2	м
Количество источников выброса	N1	—
Количество значений высоты трубы, для которых проводится расчет	N2	—
Количество выбросов установок <i>i</i> -го типа	A1 (10)	г/с
Число установок <i>i</i> -го типа	A2 (10)	—
Разность температур газовой смеси и окружающего воздуха в выбросах установки <i>i</i> -го типа	D1 (10)	°C
Объем газовой смеси, выбрасываемой из трубы установки <i>i</i> -го типа	V1 (10)	м ³ /с

В процессе счета печатаются следующие значения: суммарная приземная концентрация $C_{ми}$; расстояние $x_{ми}$ от источников выброса до точки, в которой в зависимости от высоты трубы H источника загрязнения при неблагоприятных метеорологических условиях приземная концентрация вредного вещества достигает максимального значения.

Эти данные позволяют проектировщикам химического завода выбрать высоту трубы, при которой приземная концентрация вредных веществ в атмосфере не будет превышать значения ПДК.

В качестве примера рассмотрим расчет загрязнения атмосферы диоксидом серы от установок нефтеперерабатывающего завода, расположенного в Средней Азии.

Источниками загрязнений являются: 1) установка каталитического риформинга; 2) установка термического крекинга; 3) установка гидроочистки; 4) атмосферно-вакуумная трубчатка; 5) установка по производству серы из сероводорода.

Фооновую концентрацию диоксида серы, которая имеет место в отсутствие установок, примем равной нулю. Температура воздуха 40 °С. Расстояние между источниками загрязнения не превышает 360 м. Трубы всех пяти установок (источников загрязнений) имеют равную высоту. Исходные данные для расчета приведены в табл. 42. Результаты расчетов сведены в табл. 43 и 44. Исходя из них, можно выбрать необходимую высоту трубы. В нашем примере высота труб должна быть не менее 150 м. При такой высоте труб на любом расстоянии от источника загряз-

Таблица 42. Исходные данные для расчета загрязнений атмосферы диоксидом серы

Параметры	Единицы измерения	Значение
Климатический параметр	$c^{2/3} \cdot \text{мг} \cdot (\text{°C})^{1/3} / \text{г}$	200
Наименьшая высота трубы	м	90
Наибольшая высота трубы	м	190
Коэффициент, учитывающий оседание примесей	—	1
Диаметр устья трубы	м	2
Количество различных типов источников выброса	—	5
Количество рассматриваемых значений высоты трубы	—	11
Число установок одного типа	—	1
Количество выбросов в атмосферу из установок:	г/с	
1		200
2		250
3		150
4		180
5		300
Разность температур газовой смеси и окружающего воздуха для установок:	°С	
1		360
2		410
3		210
4		460
5		460
Объем газовой смеси, выбрасываемой из труб установок:	м ³ /с	
1		50
2		100
3		30
4		80
5		35

нения максимальная приземная концентрация диоксида серы не будет превышать значения ПДК, равного 0,5 мг/м³.

Т а б л и ц а 43. Суммарная приземная концентрация диоксида серы

$H, \text{ м}$	90	100	110	120	130	140	150	160	170	180	190
$C_{\text{м.и.}}, \text{ мг/м}^3$	1,0	0,8	0,7	0,6	0,5	0,5	0,4	0,3	0,3	0,3	0,2

Т а б л и ц а 44. Расстояние $x_{\text{м.и.}}$ от источника загрязнения

$H, \text{ м}$	Номер источника загрязнения				
	1	2	3	4	5
90	1805	1810	2949	1805	1827
100	2006	2011	2277	2005	2031
110	2207	2212	2505	2206	2234
120	2408	2413	2734	2406	2438
130	2609	2614	2962	2607	2641
140	2811	2815	3191	2808	2845
150	3012	3016	3419	3008	3049
160	3213	3217	3648	3209	3252
170	3414	3418	3876	3409	3456
180	3615	3619	4105	3610	3660
190	3817	3820	4333	3810	3863

САПР ХИМИЧЕСКОЙ ТЕХНОЛОГИИ

Ни одна химическая технология не может быть внедрена в производство без разработки соответствующего проекта, составления смет и выполнения чертежей. *Проект* — это комплекс документации, необходимой и достаточной для строительства нового производства или реконструкции действующего промышленного предприятия. Все, начиная с пробырки и кончая химическим заводом, создается на основе проектов. Понятно, что от качества проектов и сроков их разработки зависит очень многое. В проект должны закладываться новейшие достижения науки и техники. Сроки проектирования необходимо сокращать до минимума.

Нелегко труд проектировщиков. Им приходится выполнять многочисленные расчеты, изготавливать чертежи, сопоставлять различные варианты технических решений в поисках наилучшего из них. А сколько времени идет на составление пояснительных записок к проектам! Разумеется, сложность и трудоемкость проектов непрерывно возрастают, а их количество увеличивается. Подсчитано, что в химической промышленности нашей страны ежегодный прирост объема проектных работ составляет около 5%. Если этот прирост обеспечивать только за счет притока новой рабочей силы, то необходимо ежегодно увеличивать численность проектировщиков более чем на 40 000 человек! Конечно, такой путь не ведет к решению проблемы. Где же выход?..

Выход — в создании и внедрении САПР. Так принято называть *системы автоматизированного проектирования*, базирующиеся на использовании современной вычислительной техники.

«Что же тут нового? — спросит читатель. — Неужели раньше в проектных организациях не использовалась вычислительная техника?» Конечно, использовалась, но это были обычно *разрозненные программы* для выполнения отдельных частных расчетов, не объединенные в единую систему. САПР — это качественно новые системы проектирования. Их основу составляет большой, постоянно пополняемый комплекс программ, реализующий на ЭВМ все виды расчетов, необходимых для разработки проекта. Сюда входят расчеты отдельных машин и аппаратов, технологических режимов оборудования, параметров систем автоматизированного управления.

Вычислительные машины, на которых выполняются проектные расчеты, делают сотни тысяч и даже миллионы операций в секунду. Это ли не средство интенсификации проектных разработок!

Современные САПР избавляют инженеров от рутинной вычислительной работы, предоставляя им возможность больше времени уделять творческой деятельности, направленной на выработку новых технических решений.

В память вычислительной машины заносят информацию о наиболее перспективных отечественных и зарубежных разработках, образцах новой техники, новейших технологиях. Такой *банк знаний* заменяет проектировщику множество книг, журналов и справочников. Автоматизированные банки знаний аккумулируют в себе важнейшие достижения научной и инженерной мысли. Они служат источником новых идей и технических решений.

САПР успешно используются при разработке новых и совершенствовании действующих производств, например получения серной кислоты, аммиака, полиэтилена. Благодаря этому, существенно сокращаются затраты сырья и энергии на производство единицы продукции.

Но современные ЭВМ — это не только быстрые вычислители, но и прекрасные чертежники. Они могут с ювелирной точностью выполнять чертежные работы различной сложности — здесь и чертежи отдельных аппаратов химической технологии, и схемы химических производств, и изображения систем автоматического управления. Надписи на чертежах ЭВМ выполняют любым шрифтом на русском и иностранных языках. Графопостроитель, входящий в комплекс вычислительной машины и послушный воле проектировщика, быстро и безошибочно выполнит любой чертеж.

Итак, САПР во много раз ускорили разработку проектов, избавив многих проектировщиков от утомительного малопродуктивного ручного труда, связанного с выполнением расчетных и графических работ.

«И это всё?» — спросит читатель. Нет, не всё.

Достижения кибернетики породили новый класс систем автоматизированного проектирования — *интеллектуальные САПР*. Эти системы обладают замечательным свойством: автоматически формировать *новые знания* на базе уже имеющихся знаний. Другими словами, такие САПР способны создавать принципиально новые химические технологии, новые машины и аппараты.

Таким образом, с помощью САПР проектирование осуществляется значительно лучше и быстрее, чем традиционными методами.

КОМПЬЮТЕР И ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ

В последние годы компьютеры стали широко применяться в таких областях деятельности человека, как игра в шахматы, перевод с одного языка на другой, сочинение музыкальных произведений, выработка и принятие технических решений. Деятельность такого рода считается интеллектуальной. Поэтому компьютеры и соответствующие программы, используемые для машинной поддержки интеллектуальной деятельности человека, получили название систем *искусственного интеллекта*.

В литературе неоднократно высказывалось мнение о том, что название «искусственный интеллект» неудачно. Тем не менее оно обрело «права гражданства» и используется как в нашей стране, так и за рубежом.

Свое наиболее полное выражение искусственный интеллект находит в *экспертных системах*, которые основаны на использовании и переработке знаний. Первые системы искусственного интеллекта разрабатывались для решения таких задач, как автоматическое доказательство теорем, извлечение смысла из текстов на естественном языке, машинный перевод с одного языка на другой. Решение этих задач связано с нетривиальными логическими умозаключениями, а поиск результата, как правило, сводится к перебору и анализу огромного количества возможных вариантов. Однако перебор *всех* возможных вариантов одной шахматной партии даже на современных компьютерах занял бы годы. Человек же в таких ситуациях проявляет способность отбрасывать малоперспективные направления поиска, сокращая тем самым область перебора. В экспертных системах делается попытка моделировать именно эту способность человеческого мышления. Такой подход к решению задач получил название *эвристического*. Он используется наряду с методами математической логики.

Эвристические правила представляют собой отражение знаний специалиста-эксперта, которые он использует в своей профессиональной деятельности. Если в экспертной системе имеется необходимый объем знаний специалиста, как эвристических, так и фундаментальных, то ее поведение при решении задач в соответствующей проблемной области может достигать уровня эксперта. Таким образом, существенное отличие экспертных систем от обычных

программ состоит в том, что они оперируют не только с данными, но и со *знаниями*. Поэтому такие системы обладают развитыми средствами представления, хранения и обработки знаний.

Первые разработки в области создания систем искусственного интеллекта, основанных на знаниях, носили чисто исследовательский характер и были ориентированы на решение относительно простых задач. На них проверялись общие принципы построения таких систем. Для их реализации были предложены специальные *языки* представления информации и способы вывода решений. Однако в середине 1970-х годов стали появляться первые экспертные системы для решения сложных практических задач. Первенцами таких систем были DENDRAL и MYCIN. С появлением первых экспертных систем для решения задач в области химии и медицины системы искусственного интеллекта вышли за пределы лабораторий ученых и стали рассматриваться как перспективный коммерческий продукт.

СТРУКТУРА ЭКСПЕРТНЫХ СИСТЕМ

Экспертные системы имеют довольно развитую архитектуру. Надо сказать, что термин «архитектура» получил широкое распространение в вычислительной технике. Под ним понимают структуру внутреннего устройства различного рода вычислительных комплексов.

На рис. 29 представлена типичная схема организации экспертной системы. Современная экспертная система содержит по меньшей мере четыре основных элемента.

1. *База знаний* — это хранилище знаний эксперта о предметной области, которые записаны на определенном языке представления.

2. *Банк данных*, куда поступает информация, необходимая для решения задачи.

3. *Решатель* (механизм логического вывода) для извлечения соответствующих знаний из памяти ЭВМ и применения их к данным.

4. *Супервизор* — блок, обеспечивающий взаимодействие всех составных частей экспертной системы.

У любой экспертной системы обычно бывают два типа пользователей. Первый — это *эксперт*, осуществляющий наполнение базы знаний, проверку работоспособности системы и ее модификацию. Впрочем, он не столько пользователь, сколько создатель



Рис. 29. Схема организации экспертной системы

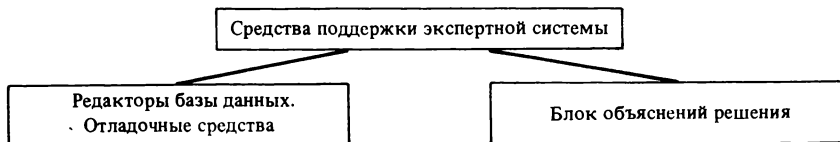


Рис. 30. Средства поддержки экспертных систем

системы. Второй тип пользователей — это те лица, для которых непосредственно разрабатывается экспертная система (их называют просто — *пользователи*). Эксперт и пользователь выполняют различные функции по отношению к экспертной системе, часть которых, естественно, перекрывается. Так, решение конкретных задач и требование от экспертной системы объяснений получаемых результатов характерны как для эксперта, так и для пользователя. В то же время ряд функций не пересекается и существует только на этапе разработки системы.

Средства поддержки экспертных систем (рис. 30), как правило, состоят из двух частей. Первая из них используется только на этапе разработки системы. Это — редакторы базы данных, отладочные средства и т. д. Вторая часть, обеспечивающая объяснительные возможности системы и реализующая интерфейс между человеком и ЭВМ, входит в готовую экспертную систему, как и блок накопления статистического материала о работе системы.

Отладочные средства (рис. 31) содержат *программы трассировки*, позволяющие конструктору экспертной системы следить за последовательностью выполнения операций при решении задач. С их помощью он может просматривать выполнение отдельных подпрограмм.

Средства прерывания дают возможность программисту останавливать в нужный момент исполнение программ, анализировать ситуации, изменять состояние переменных и снова запускать программу в работу. Эти средства особенно полезны на этапе отладки системы, так как помогают организовать останов в случае появления ошибки и обеспечить анализ характера ошибки.

Наиболее совершенные экспертные системы снабжаются специальными средствами *автоматической проверки* базы знаний по контрольным (тестовым) задачам с известным правильным решением. Результатом такой проверки могут быть списки задач, при



Рис. 31. Отладочные средства

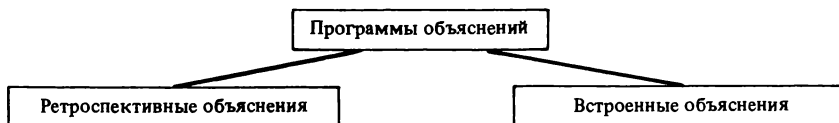


Рис. 32. Программы объяснений

решении которых были допущены ошибки. Этот список снабжается протоколом «рассуждений» экспертной системы, что позволяет выявлять причины, приведшие к ошибке.

Программы объяснений (рис. 32) позволяют человеку понимать «логику действий» экспертной системы. *Гипотетические объяснения* выдаются в тех случаях, когда нет полной гарантии, что результат решения задачи является вполне достоверным. Это, как говорится, информация к размышлению. *Ретроспективные объяснения* знакомят пользователя со всеми выполненными этапами поиска решения. *Встроенные объяснения* позволяют извлекать из памяти ЭВМ заранее подготовленные объяснительные тексты, которые подсоединены к соответствующим правилам и концепциям базы знаний.

Средства общения пользователя со всеми блоками экспертной системы образуют *интерфейс* (рис. 33). Поскольку экспертные системы функционируют в *интерактивном режиме* с пользователями, то они, как никакие другие системы, должны обладать развитыми интерфейсами.

Наиболее естественным и удобным способом общения человека с ЭВМ является его родной язык. Поэтому в области искусственного интеллекта быстро развиваются технические средства понимания естественного языка, точнее, *ограниченного* естественного языка. В экспертные системы, которые реализуют ввод и вывод информации на естественном языке, встраивают блоки *анализа* и *синтеза текстов* на ограниченном естественном языке.

Естественным для химика является оперирование информацией, представленной графически, — картинками, рисунками, схемами, графиками. Современные экспертные системы позволяют оперировать средствами машинной графики, поддерживают ввод и вывод графической информации. Для анализа, построения и корректировки графических изображений в них используются

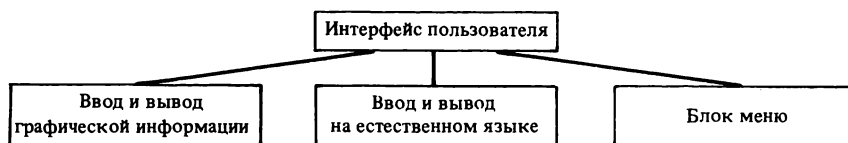


Рис. 33. Интерфейс экспертной системы

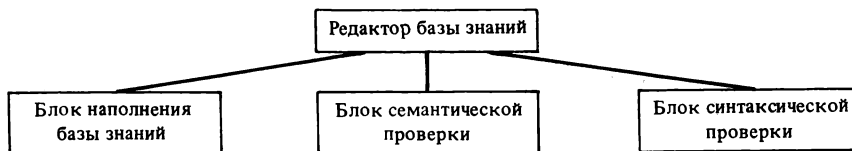


Рис. 34. Редактор базы знаний

специальные программы, как, например, редакторы изображений, программы динамической графики и т. п.

Большинство экспертных систем дает возможность пользователю вводить информацию в виде *меню*. Это значительно упрощает диалог человека с ЭВМ. Для поддержки такого режима работы используются *окна* на экране дисплея. При этом сообщения выводятся на дисплей в разные окна (окрашенные, например, в разные цвета) в зависимости от характера информации.

Поскольку экспертные системы являются сложными комплексами программ, для их функционирования необходима связь с операционной системой ЭВМ. Такая связь осуществляется через программные и технические средства. Они, например, позволяют осуществлять *стыковку* программ, написанных на разных алгоритмических языках.

На рис. 34 показаны блоки, входящие в состав *редактора* базы знаний. Средства редактирования необходимы только на этапе проектирования экспертной системы. Они обеспечивают ряд важнейших функций работы с базой знаний: ввод, формализацию и проверку знаний эксперта. С помощью редактора разработчик системы может вручную записывать в память ЭВМ высказывания эксперта на определенном языке представления знаний. Более современные редакторы позволяют до некоторой степени автоматизировать эту операцию.

При наполнении базы знаний новой «порцией» информации в ней могут возникать противоречия (конфликты) между уже содержащимися и вновь приобретенными знаниями. Ликвидация этих противоречий осуществляется с помощью средств синтаксической и семантической проверки.

Редакторы базы знаний служат и для выполнения более сложных функций — извлечения знаний из эксперта, автоматического обучения экспертной системы. Так, после первоначального наполнения базы знаний специальные программы осуществляют анализ имеющейся информации и генерируют различные возможные проблемные ситуации, которые представляются эксперту с целью определения заключений по этим ситуациям. Таким образом, они ставят эксперта перед необходимостью формулировать новые правила механизма логического вывода. Имея информацию о целях построения экспертной системы и оценивая текущий уровень

знаний, эти программы определяют «пробелы в знаниях», которые и предлагают заполнить эксперту.

Некоторые наиболее совершенные экспертные системы обладают возможностями автоматического обучения на имеющихся примерах решения задач. Поскольку наиболее «узким» местом в процессе создания экспертной системы является извлечение знаний из эксперта, то развитие исследований в области автоматического обучения систем обещает стать наиболее перспективным направлением работ по экспертным системам.

СПОСОБЫ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

Основа решений, получаемых с помощью экспертной системы,— это совокупность знаний, хранящаяся в памяти ЭВМ. Именно качество и количество знаний о предметной области определяют эффективность использования экспертной системы. Поэтому вопросы, связанные с представлением, хранением и использованием знаний являются важнейшими при создании и эксплуатации таких систем.

Для обеспечения высокого уровня работы экспертной системы ее база знаний должна содержать информацию различного рода. Это — *фундаментальные знания* о предметной области (например, о химических законах, о причинно-следственных взаимосвязях физико-химических процессов). В составе базы знаний должны быть и *эмпирические знания*. Там же должен храниться в формализованном виде опыт специалистов-экспертов.

Характерной чертой экспертных систем является способность оперировать со знаниями, представленными в символьной форме. Символ — это последовательность букв, которой поставлено в соответствие некоторое понятие проблемной области, например: кислота, основание, соль. Такие символы могут связываться смысловыми отношениями в выражения и образовывать символьные структуры: органическая кислота, растворимая соль и т. п. Такие символьные выражения являются основой представления знаний.

Проблема представления знаний включает в себя разработку языковых средств для описания различных объектов проблемной области, понятий, ситуаций, их иерархии и связей между ними. Известно, что трудоемкость представления знаний на таких языках программирования, как БЕЙСИК, ФОРТРАН, ЛИСП, ПАСКАЛЬ, существенно различна. Правильный выбор языка представления знаний может существенно облегчить моделирование процессов принятия решений. Языки представления знаний по своим возможностям должны быть достаточно богатыми, чтобы полно и точно описывать предметные области.

Другой важной проблемой, связанной с разработкой экспертных систем, является создание средств (механизма) *логического*

вывода решений. База знаний содержит информацию о проблемной области. Однако сами по себе эти знания «пассивны». Для того чтобы с помощью этих знаний находить решения, необходим набор определенных процедур манипулирования знаниями. Совокупность таких процедур называется *решателем*. Решатель, оперируя символьными структурами базы знаний, осуществляет символьный логический вывод решения. Экспертная система должна не только находить правильное решение, но и находить его быстро.

Рассмотрим два наиболее часто используемых способа представления знаний: *системы productions* и *фреймы*.

Термин «продукция» первоначально интерпретировался как процедура преобразования символьных строк. В настоящее время продукционные системы называют также системами, основанными на правилах.

Каждое правило состоит из двух частей: условие (левая) и действие (правая). Условие определяет возможность применения правила к базе данных, содержащихся в экспертной системе. Если условие удовлетворяется, то правило может быть применено. Действие правила заключается в изменении информации в базе данных.

Система управления экспертной системы выбирает, какое из нескольких применимых правил должно быть использовано. На рис. 35 представлена общая структура продукционной системы. Правила в системе productions выделены в отдельный блок — базу правил. Интерпретатор представляет собой программу, которая проверяет применимость правил и реализует их выполнение.

Базу правил можно рассматривать как совокупность знаний эксперта о предметной области, записанных в виде условных выражений типа «если..., то...». Продукции оказались удобным средством для моделирования процесса принятия решений экспертом.

К достоинствам продукционных систем можно отнести простоту и наглядность представления знаний, а также независимость базы правил от системы управления. Это позволяет легко осуществлять добавление новых и удаление или модификацию имеющихся правил, а также «настройку» экспертной системы на новые предметные области без изменения структуры управления. Продукционные системы, благодаря их простоте и относительной универсальности, лежат в основе большинства экспертных систем, коммерчески производимых в настоящее время.

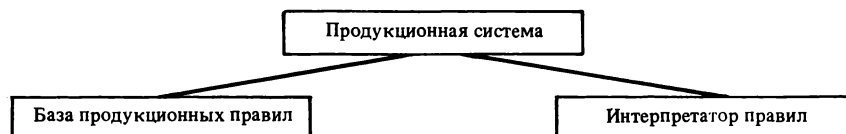


Рис. 35. Структура продукционной системы

Другой распространенной формой представления знаний, сочетающих в себе возможность описаний как отношений между понятиями предметной области, так и процедур выполнения некоторых действий, являются *сети фреймов*. Фреймы (от английского *frame* — рамка, каркас) — это информационные структуры для представления знаний о ситуациях, понятиях и объектах окружающего нас мира. Фреймы могут быть связаны между собой некоторыми отношениями и образовывать *семантическую сеть*. Основной особенностью таких сетей является то, что фреймы могут содержать процедуры, управляющие преобразованием информации как внутри самого фрейма, так и в других узлах сети.

Фрейм содержит структуры для хранения информации, называемые *слотами* (от английского *slot* — щель). Например, фрейм «хлор», используемый для описания химического элемента, может иметь слот с названием «представляет собой» и содержать в нем неизменную информацию в виде символьной структуры «галоид». Слоты могут содержать перечень возможных значений. В состав слотов могут входить и *присоединенные процедуры*, позволяющие осуществлять некоторые операции для получения значения этого слота.

Основной операцией вывода решения при использовании фреймового представления знаний является «настройка» фрейма, находящегося в базе знаний, на конкретный факт, объект или ситуацию. Использование фреймов для представления знаний позволяет получать описание предметной области в виде связанных, иногда иерархически упорядоченных, крупных информационных структур. Это приводит к более наглядному представлению знаний.

Средства поиска решения для фреймовых структур разнообразны и часто зависят от выбора языка представления знаний и самих фреймов. На некоторых этапах поиска могут применяться стратегии, аналогичные стратегиям в продукционных системах. Однако отсутствие общих эффективных формальных способов вывода решений на сети фреймов ограничивает в некоторой степени сферу их использования.

Современные экспертные системы могут «объяснять», каким образом они достигли определенного решения, однако полнота и характер объяснений могут быть различными. Одни системы обладают полным набором объяснительных средств, в то время как другие реализуют чаще всего *ретроспективные объяснения*.

Ретроспективный механизм позволяет получить цепочку «умозаключений» экспертной системы, приведшую ее к некоторому выводу. Например, пользователь может обратиться к экспертной системе с «предложением» объяснить, почему она задает тот или иной вопрос или как было получено интересующее его решение. В ответ он получит «дерево поиска» решений и пошаговую историю получения вывода.

В некоторых случаях экспертная система отвечает, почему она не смогла получить необходимое решение. Например, при проектировании химического реактора система может ответить, что в предложенных ей габаритных размерах реактор не может быть создан для обеспечения заданной производительности по целевому продукту, так как в базе данных нет катализатора, обладающего необходимой активностью.

Экспертные системы часто применяются для обучения пользователей, повышения их профессиональной компетенции. Поэтому блок объяснений приобретает смысл блока обучения, выполняя одну из основных функций *автоматизированных обучающих систем*.

ЭКСПЕРТНЫЕ СИСТЕМЫ СЛУЖАТ ХИМИКАМ

В настоящее время экспертные системы разработаны в основном для таких областей знаний, как электроника, медицина, геология. Что касается химии и химической технологии, то здесь экспертные системы встречаются не так часто. Однако несомненно, что экспертные системы — это весьма перспективная новая область химической кибернетики.

В химической технологии экспертные системы с успехом выполняют роль «консультантов» при решении задач, связанных с творческим поиском. Аналогичные задачи встречаются при автоматизированном проектировании. Располагая базой знаний и механизмом логического вывода, экспертная система «ведет» пользователя САПР по всему пути решения задачи.

Имеются сведения о системе KNOWTRAN, базирующейся на знаниях в области теплопередачи. По мнению разработчиков, эта система должна накапливать знания в процессе своего функционирования.

Экспертная система CONPHYDE служит для обоснованного выбора метода расчета равновесий в системах пар — жидкость.

Система PROSPECTOR содержит около 40 эвристических правил, предназначенных для выбора одного из 15 методов термодинамических расчетов. Правила выбора учитывают природу вещества, температуру и давление. Дальнейшее развитие этой системы идет в направлении выбора методов расчета фазовых равновесий как для чистых веществ, так и для смесей.

Для автоматизированной разработки вариантов схем технологических процессов используется экспертная система CASP, опирающаяся на базу правил. Однако она допускает и «выключение» искусственного интеллекта с тем, чтобы в нужный момент использовать знания, опыт и интуицию пользователя.

Экспертная система IDEA предназначена для синтеза вариантов процессов в установках, производящих один и тот же продукт из разного сырья по разной технологии. Одна из версий этой системы реализована на персональном компьютере.

Система HEATEX поддерживает проектирование сети теплообменников для рекуперации теплоты. В механизме переработки знаний используется 115 правил продукции.

Малая экспертная система CONSULTANT облегчает разработку систем автоматического регулирования ректификационных колонн.

В области химии лекарственных веществ разработана система СТРАК для анализа связей между структурой химического соединения и его биологической активностью. Экспертная система ОРАКУЛ используется для «конструирования» лекарств.

Вот и закончилась наша экскурсия в страну компьютеров. Как мы видим, компьютер стал надежным помощником химика и широко используется во всех сферах его деятельности. В этой популярной книге авторы стремились показать не только ЧТО можно делать с помощью компьютера, но и КАК это делать. Заинтересованные читатели, вероятно, захотят углубить свои знания в области применения вычислительной техники в химии и химической технологии. Это можно сделать, например, прочитав книги, приведенные ниже в библиографическом списке.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Александров В. В., Арсентьев В. Н., Арсентьева А. В. Что может ЭВМ? Л.: Машиностроение, 1988. 124 с.
2. Вильямс Р., Малкин К. Компьютеры в школе: Пер. с англ. М.: Прогресс, 1988. 336 с.
3. Гильде В., Альтрихтер Э. С микрокалькулятором повсюду: Пер. с нем. М.: Мир, 1988. 200 с.
4. Гринчишин Я. Т., Ефимов В. Н., Ломакович А. Н. Алгоритмы и программы на Бейсике. М.: Просвещение, 1988. 160 с.
5. Дьяконов В. П. Справочник по алгоритмам и программам на языке Бейсик для персональных ЭВМ. М.: Наука, 1987. 240 с.
6. Дьяконов В. П. Справочник по расчетам на микрокалькуляторах. М.: Наука, 1986. 224 с.
7. Закгейм А. Ю. Введение в моделирование химико-технологических процессов. М.: Химия, 1973. 224 с.
8. Паскалев Ж. Первые шаги в вычислительной технике: Пер. с болгарского. М.: Радио и связь, 1987. 152 с.
9. Перегудов М. А., Халамайзер А. Я. Бок о бок с компьютером. М.: Высшая школа, 1987. 192 с.
10. Розенброк Х., Стори С. Вычислительные методы для инженеров-химиков. М.: Мир, 1968. 443 с.
11. Скатецкий В. Г. Математическое моделирование физико-химических процессов. Минск: Вышейш. шк. 1981. 144 с.
12. Фокс А., Фокс Д. Бейсик для всех. М.: Энергоатомиздат, 1987. 183 с.
13. Цветков А. Н., Епанченков В. А. Прикладные программы для микро-ЭВМ «Электроника БЗ-34», «Электроника МК-56» и «Электроника МК-54». М.: Финансы и статистика, 1984. 175 с.
14. Эберт К., Эдерер Х. Компьютеры. Применение в химии: Пер. с нем. М.: Мир, 1988. 416 с.
15. Левин Р., Дранг Д., Эдельсон Б. Практическое введение в технологию искусственного интеллекта и экспертных систем с иллюстрациями на БЕЙСИКЕ: Пер. с англ. М.: Финансы и статистика, 1990. 239 с.

ОГЛАВЛЕНИЕ

Предисловие	3
Глава 1. Первое знакомство с компьютером .	4
Машина обрабатывает информацию .	4
Как устроена ЭВМ .	5
Дисплей	6
Компьютерная память .	8
Макро, мини, микро . .	9
Числа из нулей и единиц .	10
Действия над мыслями .	13
Родословная компьютеров	16
ЭВМ на столе	19
Химия для компьютеров .	21
Глава 2. Алгоритмы вокруг нас .	23
Шаг за шагом	23
Последовательности операций .	24
Схема алгоритма	27
Глава 3. Самая маленькая ЭВМ .	30
Программируемый микрокалькулятор (ПМК)	30
Расчеты на ПМК	32
В мире атомов . .	33
Путешествие по стеку	33
Скорость химической реакции .	35
Быстро ли движутся молекулы? .	37
Как работает ПМК	40
Радиоактивный распад урана .	42
Нитрующие смеси .	44
Глава 4. Алгоритмические языки персонального компьютера .	47
Первая программа	47
Элементы языка	48
Операторы	49
Как ввести и вывести информацию? .	52
Работа на дисплее .	54
Средняя температура .	55
Зачем нужны индексы? .	56

Форма печати результатов .	57
Подпрограммы	59
Компьютер чертит и рисует	60
Множество алгоритмических языков .	62
Операционные системы	64
Глава 5. Компьютер участвует в эксперименте .	66
ЭВМ управляет экспериментом .	66
Линейная регрессия	68
Корреляция	72
Какая формула лучше?	74
Многофакторные зависимости .	76
Сглаживание	79
График и гистограмма	80
Зачем планируют эксперимент? .	82
Поверхность отклика	83
Проверим воспроизводимость опытов .	85
Полный факторный эксперимент	87
Количество опытов можно уменьшить .	94
Крутое восхождение	98
Симплекс ведет к оптимуму .	102
План становится сложнее .	106
Контурно-графический анализ	109
Латинские квадраты и прямоугольники .	110
Совмещенный план	116
Глава 6. Компьютер-труженик	118
Помощник учителя .	118
Итоги олимпиады	119
Модель химической реакции .	121
Экологические расчеты	123
САПР химической технологии	129
Компьютер и искусственный интеллект .	131
Структура экспертных систем .	132
Способы представления знаний	136
Экспертные системы служат химикам .	139
Библиографический список	140

НАУЧНО-ПОПУЛЯРНОЕ ИЗДАНИЕ

Саутин Станислав Николаевич

Пунин Анатолий Евгеньевич

МИР КОМПЬЮТЕРОВ И ХИМИЧЕСКАЯ ТЕХНОЛОГИЯ

Редактор *Ю. К. Кузнецов*
Обложка художника *Ю. Б. Осенчакова*
Техн. редактор *Л. Ю. Линева*
Корректор *Л. С. Лазоренко*

ИБ № 2513

Сдано в набор 24.05.90. Подписано в печать 01.02.91. Формат бумаги 60×88¹/₁₆. Бумага офсетная № 2. Литературная гарнитура. Печать офсетная. Усл. печ. л. 8,82. Усл. кр.-отт. 9,32. Уч.-изд. л. 9,13. Тираж 55 000 экз. Зак. 570. Цена 1 р.

Ордена «Знак Почета» издательство «Химия», Ленинградское отделение. 191186, Ленинград, Д-186, Невский пр., 28.

Ленинградская типография № 2 головное предприятие ордена Трудового Красного Знамени Ленинградского объединения «Техническая книга» им. Евгении Соколовой Союзполиграфпрома при Государственном комитете СССР по печати. 198052, Ленинград, Л-52, Измайловский пр., 29.



КОММЕРЧЕСКИЙ ЦЕНТР

на кооперативных началах

Главснаба Ленгорисполкома

ЭКОС

МЫ РАБОТАЕМ В ПЕРСПЕКТИВНЫХ НАПРАВЛЕНИЯХ

ЭКОЛОГИЯ

- * разработка и внедрение современных экологически чистых и ресурсосберегающих технологий
- * моделирование и оптимизация технологических комплексов
- * комплексные экологические обследования и паспортизация промышленных предприятий и территорий

ЭКОНОМИКА

- * создание малых предприятий, хозяйственных товариществ с различными формами собственности
- * разработка образцов товаров народного потребления из отходов производств
- * консультационное обслуживание по различным экономико-правовым вопросам

ОБУЧЕНИЕ

- * методам и средствам оптимизации технологических комплексов
- * основам управления малыми предприятиями на основе комплексной интенсивной методики
- * современным методам управления качеством продукции

ИНФОРМАЦИОННО-КОММЕРЧЕСКИЕ УСЛУГИ

- * в приобретении и реализации отходов производства и некондиционной продукции
- * по маркетингу товаров народного потребления из отходов производств
- * по поиску партнеров (в том числе зарубежных) для реализации совместных проектов

ТЕЛЕФОНЫ В ЛЕНИНГРАДЕ: 355-91-51, 213-35-10

Программное обеспечение, представленное в данной книге и записанное на дискетах для профессиональных компьютеров класса IBM PC/XT/AT и „АГАТ-7”, можно приобрести в кооперативном коммерческом центре „ЭКОС” Главснаба Ленгорисполкома.

Стоимость одной дискеты – 58 рублей (IBM) и 39 рублей („АГАТ”).

Для получения дискеты необходимо перечислить эту сумму на расчетный счет № 27003461287 в Приморском отделении ЖСБ г. Ленинграда, МФО 171199 и выслать в адрес центра (199034, г. Ленинград, В. О., 13 линия, д. 12, кв. 1) письменную заявку на бланке-вкладыше. После получения заявки и поступления денег ККЦ „ЭКОС” в течение 10 дней направит в Ваш адрес дискету.

ЗАЯВКА

Просим выслать в наш адрес _____

дискету с программным обеспечением для:

– IBM PC/XT/AT

– „АГАТ-7”

(подчеркните нужный тип ЭВМ и укажите число дискет).

Оплата в сумме _____ рублей переведена на Ваш расчетный счет „_____” _____ 199 г., номер поручения (квитанции) _____

Подпись заказчика: