

ИНФОРМАТИК



Об истории создания и развитии
языка программирования Фортран
читайте на с. 32

Читайте в номере

Официальные документы 2–3

Материалы сборника. Оценка качества подготовки выпускников основной школы по информатике
Примерная программа курса информатики. 10–11-е классы (136 часов).

Уроки 4–12

В.М. Нечаев. Электронные таблицы и базы данных. Занятие 2. Книжная торговля

Учащимся предлагается помочь торговцу компьютерной литературой, который получает товар со склада по оптовым ценам и продает его в своем киоске по розничным, вести дела с помощью электронных таблиц.

Второе занятие цикла, посвященного популярным программным продуктам фирмы Microsoft — пакету для работы с электронными таблицами Excel 97 и системе управления базами данных Access 97.

Задачи 13–19

Л.Л. Акуленко-Босова. Задачи по системам счисления

Сумеете ли вы записать год своего рождения с помощью римских цифр? Каким наименьшим количеством гирь можно взвесить груз от 1 до 63 кг с точностью до 1 кг, помещая гири только на одну чашку весов? Когда $2 \times 3 = 11$, а $21 + 24 = 100$?

Семинар 20–26

А.А. Дуванов. Транслятор?.. Это очень просто!

“Транслятор — программа, которая преобразует коды одного языка программирования или формата данных в другой”. Однако не надо усложнять. Например, в Роботландском университете трансляторы пишут школьники...

Теоретические основы информатики 27–29

А.Л. Брудно. Что такое алгоритм

Заметка, которая должна представлять интерес для читателей хотя бы уже потому, что мнение ее автора, известного специалиста, одного из основателей российской информатики, не всегда совпадает с мнением авторов школьных учебников информатики.

Материалы сборника

Оценка качества подготовки выпускников основной школы по информатике



Сборник подготовили:

А.А. Кузнецов, Л.Е. Самовольнова, Н.Д. Угринович

Продолжение. Начало в № 38, 39/99

Примерная программа курса информатики 10—11-е классы (136 часов)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Цель курса — обеспечить прочное и сознательное овладение учащимися основами знаний о процессах получения, преобразования, хранения и использования информации и на этой основе раскрыть учащимся роль информатики в формировании современной научной картины мира, значение информационных технологий в развитии современного общества, привить учащимся навыки сознательного и рационального использования ЭВМ в своей учебной и последующей профессиональной деятельности.

Основными **задачами курса** являются:

- познакомить учащихся с понятиями *система, информация, модель, алгоритм* и их ролью в формировании современной картины мира;
- раскрыть общие закономерности информационных процессов в природе, обществе, технических системах;
- познакомить с принципами *структурирования, формализации* информации и выработать умение строить информационные модели для описания объектов и систем;
- развивать *алгоритмический и логический* стили мышления;
- сформировать умение организовать *поиск информации*, необходимой для решения поставленной задачи;
- сформировать умение *планировать* структуру действий, необходимых для достижения заданной цели, при помощи фиксированного набора средств;
- сформировать навыки поиска, обработки, хранения информации посредством современных *компьютерных технологий* для решения учебных задач, а в будущем и в профессиональной деятельности;
- выработать потребность обращаться к компьютеру при решении задач из любой предметной области, базирующуюся на осознанном владении *информационными технологиями* и *навыках* взаимодействия с компьютером.

СОДЕРЖАНИЕ ОБУЧЕНИЯ

1. Информация и информационные процессы (6 часов)

Вещество, энергия, информация — фундаментальные понятия современной науки. Различные подходы к определению информации.

Информационные процессы. Методы поиска информации. Хранение информации. Передача информации. Обработка информации. Использование информации.

Информационные процессы в живой природе, обществе, технике.

Информационные процессы как основа управления. Системы управления, их структура, роль обратной связи в управлении.

Информационная деятельность и информационная культура человека. Информационное общество, его особенности и основные черты. Защита информации, авторских прав на программное обеспечение.

2. Представление информации, системы счисления и основы логики (28 часов)

Язык как способ представления информации. Различные формы представления информации. Кодирование. Двоичная форма представления информации. Количество и единицы измерения информации.

Позиционные и непозиционные системы счисления. Двоичная система счисления. Представление чисел в двоичной системе счисления.

Системы счисления, используемые в ЭВМ: восьмеричная, шестнадцатеричная и др. Алгоритмы перевода чисел из двоичной системы счисления в восьмеричную, шестнадцатеричную системы счисления и обратно. Алгоритмы перевода целых чисел из p -ичной в q -ичную систему счисления. Двоичная арифметика. Особенности и преимущества использования в ЭВМ двоичной системы счисления. Представление чисел в

памяти ЭВМ: представление чисел с фиксированной и "плавающей" точкой, прямой, обратный и дополнительный коды.

Понятие об алгебре высказываний. Основные логические операции. Сложные высказывания. Построение таблиц истинности сложных высказываний. Основные законы преобразования алгебры логики. Решение логических задач с помощью алгебры логики.

Логические основы ЭВМ. Основные логические элементы, их назначение и обозначение на схемах. Функциональные схемы логических устройств. Регистры, их виды и назначение. Одноразрядный двоичный сумматор.

3. Компьютер (14 часов)

Основные устройства компьютера: устройства ввода информации, устройства вывода информации, устройства хранения информации (внутренняя и внешняя память), носители информации, устройства обработки информации, устройства передачи информации, устройства мультимедийной обработки информации.

Архитектура ЭВМ. Магистрально-модульный принцип построения компьютера.

Правила техники безопасности при работе на компьютере.

Программное обеспечение компьютера. Системное и прикладное программное обеспечение. Операционная система: назначение и основные функции. Файловая система. Путь к файлу. Графические пользовательские интерфейсы.

Компьютерные вирусы. Антивирусные программы и защита информации.

История развития ВТ. Поколения ЭВМ.

4. Моделирование и формализация (12 часов)

Моделирование. Формальная и неформальная постановка задачи. Основные принципы формализации. Основные типы информационных моделей: классификационные (табличные, иерархические), динамические (дискретные, непрерывные), логико-лингвистические (базы знаний).

Понятие об информационной технологии решения задач. Этапы решения задачи на компьютере: постановка задачи, построение модели, разработка алгоритма и программы, отладка и исполнение программы, анализ результатов. Компьютерный эксперимент.

5. Алгоритмизация и программирование (26 часов)

Понятие алгоритма. Исполнитель алгоритма. Свойства исполнителя алгоритма: система команд, среда, в которой он действует. Компьютер как формальный исполнитель.

Свойства алгоритма. Различные способы записи алгоритма. Основные типы алгоритмов: линейные, разветвляющиеся, циклические. Вспомогательные алгоритмы. Процедуры. Рекурсивные алгоритмы. Составление простейших алгоритмов и запись их различными способами.

Введение в язык программирования. Основные типы данных. Основные алгоритмические конструкции языка и соответствующие им операторы языка программирования. Подпрограммы: функции, процедуры, рекурсии. Структурированные типы данных: массивы, записи, файлы. Поиск и упорядочение элементов

массива. Работа с записями и файлами. Машинная графика. Приложения машинной графики: построение графиков функций, создание движущихся изображений, моделирование простейших физических процессов. Знакомство с численными методами. Создание диалоговых программ.

6. Информационные технологии (50 часов)

Технология обработки текстовой информации

Текстовый редактор: назначение и основные функции. Ввод и редактирование текста. Фрагмент текста, работа с фрагментом (выделение, перенос, копирование, удаление и т.д.). Абзац, операции с абзацами (форматирование, установление межстрочного интервала и т.д.). Оформление текста: шрифты, цвет символов, заполнение, обрамление и т.д. Ввод, заполнение и форматирование таблиц.

Технология обработки графической информации

Теоретические основы представления графической информации. Пиксель. Графические примитивы. Способы хранения графической информации и форматы графических файлов. Модели цветообразования и цветопередачи. Графический редактор: назначение, пользовательский интерфейс и основные функции. Работа с фрагментами изображения. Аниматоры.

Технология обработки числовой информации

Электронные таблицы: назначение и основные функции. Ячейка: абсолютная и относительная адресация. Форматы данных (числа, формулы, текст). Ввод и редактирование данных. Оформление таблиц. Решение расчетных задач. Решение уравнений. Решение задач методом подбора. Табулирование и построение графиков функций. Деловая графика (диаграммы различных видов).

Технология хранения, поиска и сортировки информации

Способы организации баз данных: иерархический, сетевой, реляционный. Системы управления базами данных (СУБД). Формы представления баз данных (таблица, картотека). Ввод и редактирование записей. Сортировка и поиск записей. Изменение структуры базы данных. Виды и способы организации запросов.

Мультимедийные технологии

Принципы и способы использования мультимедийных технологий. Основные требования к аппаратной части компьютера.

Компьютерные коммуникации

Передача информации. Линии связи, их основные компоненты и характеристики. Основные функциональные параметры модемов. Компьютерные телекоммуникации: назначение, структура, ресурсы. Локальные и глобальные компьютерные сети. Основные услуги компьютерных сетей: электронная почта, телеконференции, файловые архивы. Гипертекст. Основы технологии World Wide Web. Сеть Интернет: структура, адресация, протоколы передачи. Способы подключения. Браузеры. Информационные ресурсы. Поиск информации.

Электронные таблицы и базы данных

В.М. Нечаев

Продолжение. Начало в № 36/99

Занятие 2. Книжная торговля

Ввод данных

Мелкий торговец компьютерной литературой получает товар со склада по оптовым ценам и продает его в своем киоске по розничным, несколько большим. Производить расчет компаньоны договариваются через месяц, причем поставщик настолько любезен, что согласен принять не распроданный за это время товар, если таковой окажется, обратно на склад. Для подобной формы торгового сотрудничества употребляют термин “взять товар на реализацию”. Она, конечно, очень удобна продавцу, поскольку он, независимо от того, будет продана полученная со склада литература или нет, убытка, по крайней мере прямого, не понесет.

Однако поставщик, заинтересованный в быстром обороте средств, предлагает дополнительно еще и довольно заметную скидку с оптовой цены, если товар будет взят не на реализацию, а с предоплатой, но уже без возможности возврата. Для торговца, с одной стороны, тут есть выгода — товар обходится дешевле. Однако, с другой стороны, это риск: вдруг данная книга совсем не найдет спроса? В конце концов принимается компромиссное решение. Товар в принципе берется на реализацию, но за некоторую его часть требуется предоплата.

Например, торговец заказывает 50 экземпляров какой-то книги, цена которой на складе — 35 рублей за штуку. По какой цене он будет ее продавать — решать ему, и как пойдет торговля — никто не знает, но за 10 штук он платит сразу со скидкой 20%, т.е. по 28 рублей, и это хорошо. Но вот в случае неудачи вернуть можно будет не более 40 книг, даже если вообще не будет продано ни одной книги. И это плохо. Например, если торговец продаст 27 книг, то ему надо отдать деньги за 17 штук (без скидки, по 35 рублей за каждую), а оставшиеся 23 экземпляра просто вернуть.

Для того чтобы продавцу было удобно следить за ходом событий, желательно все эти данные занести в электронную таблицу. Рассмотрим примерную последовательность действий при ее создании.

The screenshot shows the Microsoft Excel 97 interface. The title bar reads "Microsoft Excel - Книга2". The menu bar includes "Файл", "Правка", "Вид", "Вставка", "Формат", "Сервис", "Данные", "Окно", and "?". The toolbar contains various icons for file operations, editing, and formatting. The status bar at the bottom shows "Готово" and "NUM".

The spreadsheet content is as follows:

	A	B	C	D	E	F	G	H	I	J	K	L
1	Текущее состояние дел в книжной торговле.											
2												
3	Название	Автор	Цена опт.	Цена роз.	Кол-во	Оплачено	Продано	Приход	Расход	Баланс	Возврат	
4	Практичес	Долженко	40	50	30	10	8					
5	Excel одн	Вострокну	15	20	50	30	28					
6	Шпаргалк	Столяров	10	15	40	20	35					
7	Разработк	Нортон	75	90	6	6	2					
8	Access 97	О'Брайен	70	85	5	0	2					
9	Excel 97	ЕУэллс	70	85	5	0	1					
10	Access 7.	Гончаров	35	45	15	10	15					
11												
12												
13												
14												
15												
16												
17												
18												
19												
20												
21												
22												
23												
24												

Сначала в ячейку A1 занесем название таблицы, не придавая пока значения тому, как оно выглядит на листе. Потом в клетках третьей строки поместим заголовки колонок. Теперь приступим к вводу исходных данных. Названия книг и фамилии авторов (не уместившиеся в своих клетках, но тем не менее сохраненные там), оптовые и розничные цены, количество заказанных и сразу же оплаченных экземпляров — все это указывается один раз и в дальнейшем изменению не подлежит. Что же касается информации о количестве проданных книг, то она, как раз наоборот, должна меняться неоднократно, отражая, так сказать, динамику событий. Хотя, вообще говоря, она тоже относится к исходным данным в том смысле, что вводится “руками” и от нее зависят приход, расход, баланс и возврат. Ну а в эти четыре колонки надо ввести те формулы, по которым будет производиться подсчет.

Ввод формул

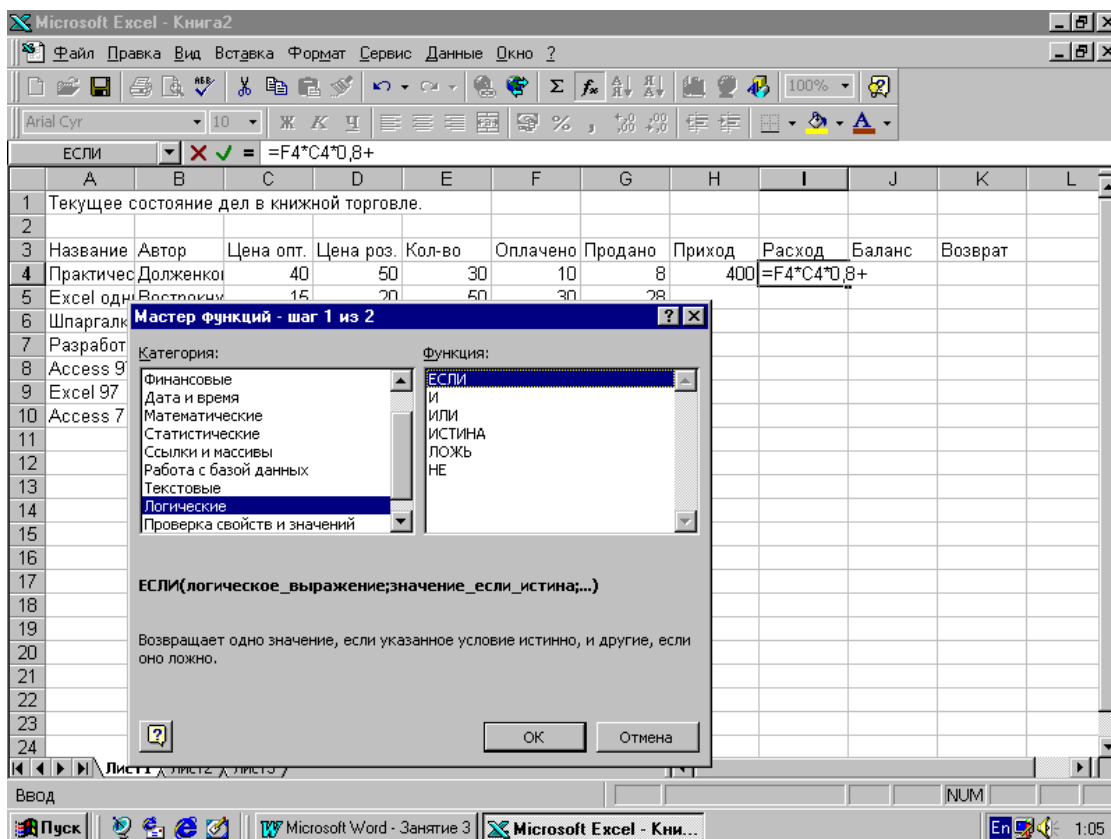
Как рассчитывать приход — ясно: количество проданных книг надо умножить на соответствующую розничную цену. Ввод формулы в ячейку H4 начинаем, разумеется, со знака “равно”. Затем щелкаем по клетке G4, нажимаем клавишу умножения (звездочка) и щелкаем по клетке D4. Все. В заключение нажимаем клавишу **Enter** и мгновенно получаем в качестве результата конкретное число ($8 \cdot 50 = 400$).

А еще ясно, как определяется баланс. От прихода надо отнять расход, и то, что получится, окажется положительным или отрицательным числом. Соответствующую формулу можно ввести в ячейку J4 прямо сейчас, не дожидаясь даже, когда будет заполнена клеточка расхода. Такое нарушение очередности вполне допустимо. Ведь для описания того, как подсчитывать баланс, само значение расхода не играет роли — здесь важно то, какова связь баланса с расходом. Расход в дальнейшем, когда формула для него будет занесена в клетку I4, изменится, соответственно, и баланс будет автоматически пересчитан.

С самим же расходом, вычисляемым в ячейке I4, дело обстоит сложнее. Он состоит из двух слагаемых. Во-первых, это сумма, которая была отдана поставщику за некоторое количество книг сразу в связи с предоплатой. Во-вторых, это деньги, которые придется отдать при окончательном расчете за книги, проданные сверх оплаченной нормы. Если, конечно, сверх такой нормы удастся что-то продать.

Первую часть расхода подсчитать легко, все определяется аналогично приходу (необходимо только учесть, что от оптовой цены надо брать с учетом скидки лишь 80%). Но формула на этом не заканчивается, и клавишу **Enter** нажимать рано. Наоборот, надо продолжить, т.е. ввести “плюс” и указать второе слагаемое.

Но как же его учесть, это второе слагаемое, которое еще неизвестно чему будет равно? Выйти из данного затруднения помогает замечательная встроенная функция ЕСЛИ. Конечно, по своей популярности она (да и любая другая функция) не может сравниться с автосуммированием, и ее значка нет на панели инструментов. Поэтому после введения “плюса” надо щелкнуть по значку, расположенному справа от “кнопки” автосуммирования, который используется для вызова всех остальных встроенных функций.



Точнее, вызывается еще не сама функция, а так называемый мастер функций, обеспечивающий, как и все подобные мастера, управление процессом ввода.

Прежде всего надо указать, к какому разделу относится нужная функция, а попав в этот раздел, выбрать ее по имени. Функция ЕСЛИ является одной из шести логических функций. Выделяем слово ЕСЛИ, щелкаем по кнопке ОК и переходим ко второму этапу работы с мастером — вводу аргументов для выбранной функции.

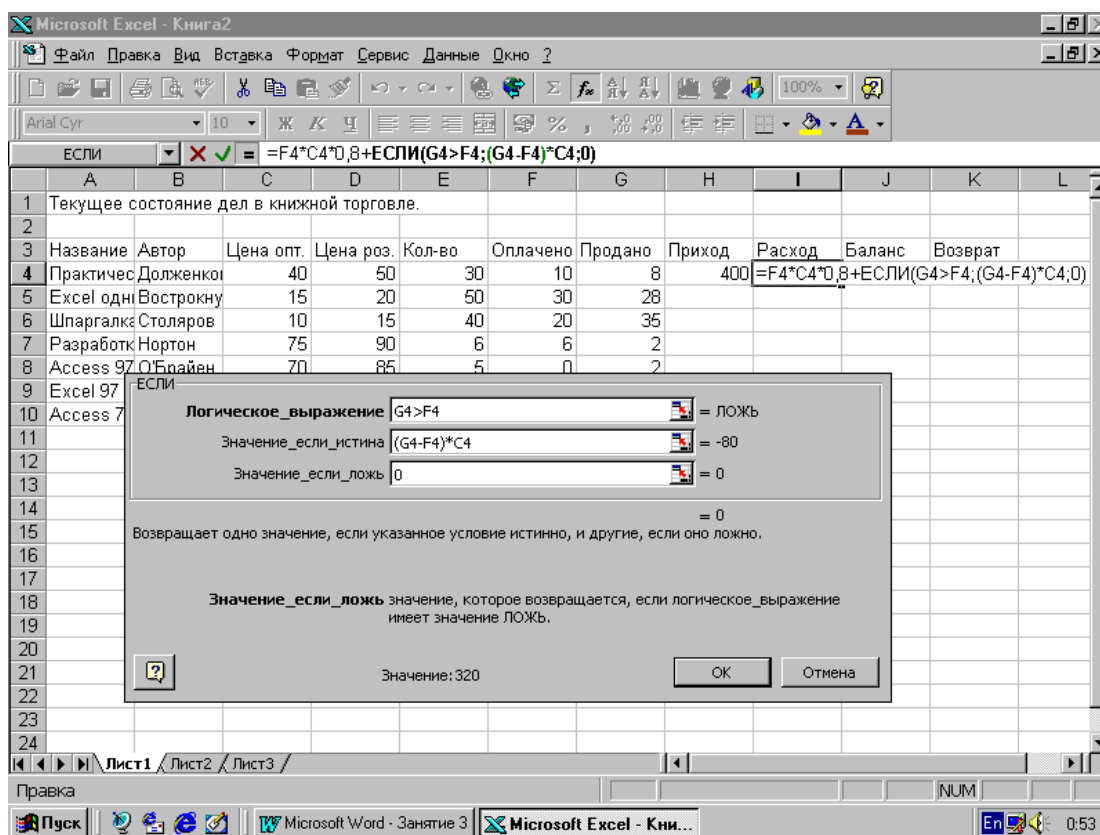
Функция ЕСЛИ

Сколько аргументов у данной логической функции? Уже первая карточка мастера показывала, что три. Они заключены в круглые скобки после имени функции и отделяются друг от друга точкой с запятой.

Логическое выражение — это переменная логического типа, т.е. переменная, способная принимать лишь значение ИСТИНА или значение ЛОЖЬ. В нашем случае дальнейший план действий зависит от ответа на вопрос: “На сегодняшний день книг продано больше, чем было оплачено при заказе?”. Содержимое клетки G4 больше, чем содержимое клетки F4, или меньше? От этого зависит, как будет определяться расход.

Если книг продано больше, чем оплачено, т.е. если логическое выражение $G4 > F4$ истинно, то придется отдавать деньги за эту разницу. Умноженная на оптовую цену, она и составляет то значение, которое будет выдавать функция, если указанное условие выполнено.

Ну а если книг продано меньше, чем оплачено, то тогда какой же может быть дополнительный расход — и так уже одни убытки! Здесь функция должна выдавать значение ноль (для второго слагаемого).



Вторая карточка мастера функций как раз и предназначена для удобного ввода всех перечисленных аргументов. Между прочим, если сама карточка загоразивает нужные клетки таблицы, ее перемещают по экрану, захватив мышью в любой точке серого фона. Или щелкают по красному значку справа от поля ввода.

Полями ввода называются белые полосочки (тут их три, а может быть больше или меньше, для разных функций разное количество). Установив в них курсор, щелкаем мышью по нужным ячейкам, нажимаем на нужные клавиши со знаками — больше, меньше, плюс, минус, умножить, разделить, скобки, цифры. Кстати, удобно пользоваться вспомогательным блоком клавиатуры, расположенным с правого ее края. Надо только следить, чтобы при этом светился индикатор NumLock.

Переходить от одного поля к другому можно, используя мышью или клавишу **Tab**. Если при наборе допущена ошибка, то исправить ее можно, как обычно, с помощью клавиш **BackSpace** и **Del**. Или даже **Esc**, повторяя затем всю операцию заново.

Наконец, если все аргументы введены правильно, следует нажимать клавишу **Enter** или щелкнуть по кнопке ОК. Тогда в ячейке появляется число, рассчитанное по указанной формуле с использованием текущих исходных данных из тех клеток, адреса которых в ней фигурируют. Например, сейчас у нас должно получиться число 320, поскольку $10 \cdot 40 \cdot 0,8 + 0 = 320$. Оно равно сумме предоплаты, так как книг продано еще не очень много.

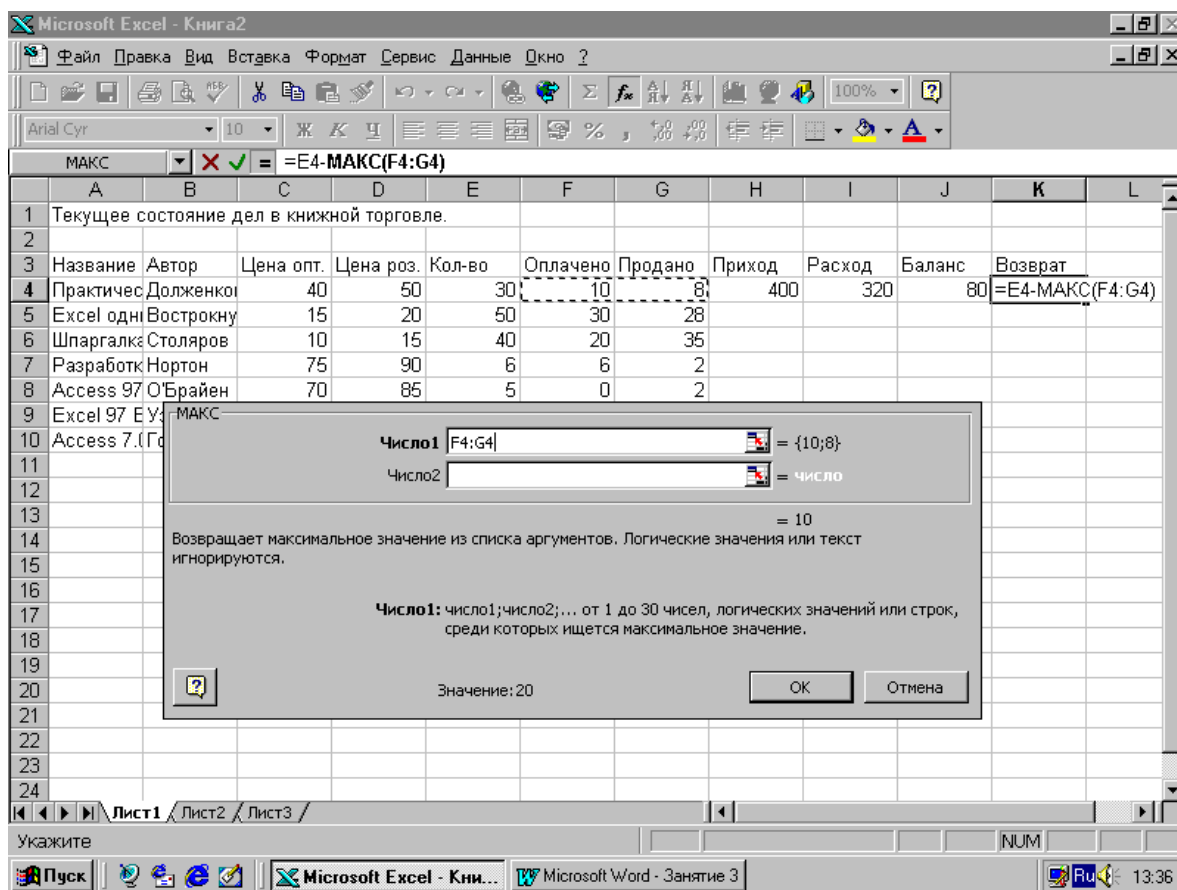
Но, положим, через неделю ситуация изменится и будет продано 13 книг. Согласно формуле расход увеличится: $10 \cdot 40 \cdot 0,8 + (13 - 10) \cdot 40 = 440$.

Функция МАКС

Итак, на текущий день приход составляет 400 рублей, а расход — 320. По мере продолжения торговли расход, конечно, будет увеличиваться, но ведь и приход тоже, иначе не было бы смысла в этой деятельности. Причем приход будет расти чуть быстрее, обеспечивая с некоторого момента “положительность” баланса. В нашем случае для первой книги из списка этот момент позади — приход уже больше, чем расход. То есть торговля себя уже оправдала, однако нераспроданных книг все еще много, и к концу срока реализации возврат, по-видимому, будет иметь место. По какой же формуле его определять?

Опять-таки можно воспользоваться функцией ЕСЛИ. Причем с тем же логическим выражением в качестве первого аргумента, но с другим Значением_если_истина и другим Значением_если_ложь. Каковы эти новые значения? Если продано меньше, чем было оплачено, то хоть бы и хотелось вернуть все, да нельзя по условиям договора. Оплаченное уже не возвращают. Только неоплаченное, т.е. разницу между общим полученным количеством и сразу на месте оплаченным. Если же продано все-таки больше, чем оплачено, то можно вернуть все, что осталось, т.е. разницу между полученным и проданным.

Есть, однако, и другие встроенные функции, и вот еще одна, с помощью которой удастся прийти к тому же результату, причем даже более простым способом. Называется она МАКС и принадлежит к категории статистических. В качестве аргумента здесь обычно указывают некоторый диапазон адресов, какую-то область таблицы, в клетках которой надо найти самое большое число. Таких аргументов может быть много, хоть десятков, если области поиска не образуют одного целого прямоугольника, а расположены в разных местах таблицы. Кроме того, аргументом может служить и просто то или иное конкретное число.



В нашем случае формула для подсчета возвращаемых книг будет такой: из полученного их количества (E4) надо вычесть либо количество оплаченных (F4), либо количество проданных книг (G4), в зависимости от того, какое из них больше. Иначе говоря, вычесть надо будет то число, которое окажется максимальным в ячейках с адресами из диапазона (F4:G4). Область поиска, если даже она состоит всего из двух клеточек, указывается обычным способом — выделением (используется мышь с нажатой левой кнопкой).

Если не нравится видеть мышью, можно сделать по-другому. В первое поле ввода занести один только адрес F4 (щелкнув по нему), а во второе — один только адрес G4. И так, и так в результате будет 20: от числа 30 будет отнято число 10 (потому что оно больше, чем 8). Полученное число (количество книг, подлежащих возврату) появится в клетке K4.

Копирование формул

В принципе для каждого названия в списке книг можно было бы повторить действия, связанные с вводом формул. Но ведь должен иметься способ, позволяющий не составлять эти практически одинаковые формулы десять раз подряд. Надо иметь возможность копировать введенную один раз формулу в другую ячейку или даже в целый диапазон ячеек.

И в самом деле, такие способы есть, точнее, способ один, но в разных реализациях. Однако в любом случае начинать следует с выделения той ячейки, в которую формула уже занесена. А потом — на выбор: одно, другое или третье, через буфер обмена или просто “перетаскиванием” (методом *drag-and-drop*).

В нашем случае удобнее всего сделать так. Наведя рамочку-курсор на ячейку H4 и тем самым выделив ее, поводить мышью (не нажимая на ее клавиши), внимательно наблюдая за видом указателя. Когда указатель находится внутри рамки или, наоборот, вне ее, он представляется нам как широкий значок креста. Когда же такой крест пересекает рамку, все равно где — слева, справа, снизу, сверху, — он становится не крестом, а стрелкой. Ну и третий облик — узенького “плюсового” значка — указатель принимает, когда проходит над черным утолщением рамки в ее правом нижнем углу. Вот как раз третий вариант нам и надо уловить.

Нажав в этот момент левую клавишу мыши и не отпуская ее, протянем раздвижную пунктирную рамочку, которая при этом образуется, вниз до клетки H10 включительно. А потом отпустим клавишу. Все попавшие внутрь рамки ячейки останутся выделенными. Выделение легко снять, щелкнув по любой другой клетке. Но главное в том, что каждая ячейка (из выделенных) теперь содержит в себе формулу, аналогичную той, с которой все начиналось! То есть исходная формула скопировалась, куда ей было указано.

Причем сделано это было не бездумно, один в один, а с учетом измененного местоположения целевой ячейки. Насколько сдвинулась вниз при копировании формула, настолько же изменились и все адресные ссылки, содержащиеся в ней. Скажем, формула в ячейке H5 будет ссылаться не как начальная на G4 и D4 (что было бы при буквальном копировании), а уже на G5 и D5. Такие ссылки, меняющиеся в соответствии со смещением формулы, называются относительными.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Текущее состояние дел в книжной торговле.											
2												
3	Название	Автор	Цена опт.	Цена роз.	Кол-во	Оплачено	Продано	Приход	Расход	Баланс	Возврат	
4	Практичес	Долженко	40	50	30	10	8	400	320	80	20	
5	Excel одн	Вострокну	15	20	50	30	28	560	360	200	20	
6	Шпаргалк	Столяров	10	15	40	20	35	525	310	215	5	
7	Разработк	Нортон	75	90	6	6	2	180	360	-180	0	
8	Access 97	О'Брайен	70	85	5	0	2	170	140	30	3	
9	Excel 97	Е Уэллс	70	85	5	0	1	85	70	15	4	
10	Access 7.1	Гончаров	35	45	15	10	15	675	455	220	0	
11												
12				Всего:		151	76	91	2595	2015	580	52
13												
14												
15												
16												
17												
18												
19												
20												
21												
22												
23												
24												

Точно так же надо поступить и с остальными колонками таблицы, в которых вычисляются расход, баланс и возврат. Причем не обязательно брать каждую колонку по очереди. Гораздо лучше выделить не одну только клетку I4, или одну клетку J4, или одну клетку K4, а все три вместе и за одну “протяжку” вниз скопировать все, что нужно.

То же самое касается и копирования формулы автосуммы, предварительно введенной в ячейку E12, которое производится, естественно, не в клетки столбца E, а в клетки строки 12. Теперь уже в адресах ссылок будут меняться (по сравнению с начальной формулой) не цифры, а буквы.

Ширина столбца

Таблица уже получилась вполне пригодной для повседневной работы. Не очень, правда, хорошо, что названия книг сильно усечены и в них трудно разобраться. Но продавец наверняка знает эти названия наизусть — для него важнее всего следить за текущим состоянием торговли, выраженным в числах. А с ними все в порядке, самое главное, что расчет выполняется правильно, да и просматриваются числа достаточно хорошо. Возможно, только формат числовых ячеек в отдельных колонках стоит поменять так, чтобы сразу было видно, где рубли, а где количество штук.

Однако не все, что может использоваться в рабочем порядке, годится для финансового отчета. Хотя бы уже потому, что ведь когда-то документ нужно будет представить на бумаге, а хорошо ли он выглядит при распечатке? Во-первых, таблица просто-напросто не поместится на одном листе! Слишком она широкая. И поэтому данные, касающиеся баланса и возврата, придется размещать на второй странице. В этом легко убедиться, осуществив предварительный просмотр. Во-вторых, названия книг в отчете надо все-таки указывать полностью. В-третьих, для того чтобы документ хорошо выглядел, требуется хоть какая-то разметка, т.е. нужны разделительные линии, по крайней мере между столбцами (а желательно и между строками). Перечисление недостатков можно было бы продолжить, однако на первый раз достаточно.

Итак, необходимо уместить документ на одном бумажном листе. Первое, что приходит в голову, — это использовать более мелкий шрифт. Чтобы выделить сразу все ячейки таблицы, щелкнем по клеточке в левом верхнем углу, на пересечении ряда букв А, В, С ... и цифр 1, 2, 3 ..., и когда все рабочее поле станет черным, выберем на панели инструментов форматирования экранную кнопку “Размер шрифта”. Сейчас там указано 10, а мы установим 8. Шрифт в самом деле стал мельче, однако само по себе это не приблизило нас к цели, поскольку ширина столбцов не изменилась. Значит, дело не только в шрифте, нужно еще столбцы сужать.

Да, сужать столбцы надо, но только не все. Колонку с названиями книг, наоборот, придется расширять. Подведем курсор к границе между столбцами А и В — именно там, вверху, где указаны сами эти буквы. Курсор при этом меняется: принимает форму двунаправленной черной стрелки. Нажимаем левую кнопку мыши и двигаем границу вправо до тех пор, пока самое длинное название не появится на экране целиком (одновременно высвечивается число, соответствующее подбираемой ширине).

	A	B	C	D	E	F	G	H	I	J	
1	Текущее состояние дел в книжной торговле.										
2											
3	Название	Автор	Цена опт.	Цена роз.	Кол-во	Оплачено	Продано	Приход	Расход	Баланс	
4	Практическая работа с MS Excel 97	Долженков	40р.	50р.	30	10	8	400р.	320р.	80	
5	Excel одним взглядом	Вострокнути	15р.	20р.	50	30	28	560р.	360р.	200	
6	Шпаргалка по Excel 7.0	Столяров	10р.	15р.	40	20	35	525р.	310р.	215	
7	Разработка приложений в Access 97	Нортон	75р.	90р.	6	6	2	180р.	360р.	-180	
8	Access 97 Библиотека ресурсов	О'Брайен	70р.	85р.	5	0	2	170р.	140р.	30	
9	Excel 97 Библиотека ресурсов	Уэллс	70р.	85р.	5	0	1	85р.	70р.	15	
10	Access 7.0 в примерах	Гончаров	35р.	45р.	15	10	15	675р.	455р.	220	
11											
12				Всего:		151	76	91	2 595р.	2 015р.	580
13											
14											
15											
16											
17											
18											
19											
20											
21											
22											
23											
24											
25											
26											
27											

Пока получилось еще хуже, в том смысле, что теперь не только столбцы баланса и возврата, но и колонки прихода и расхода не помещаются на одной странице с другими данными. Об этом свидетельствует пунктирная разделительная линия, которая появилась после первого же предварительного просмотра и в дальнейшем постоянно будет присутствовать на экране.

Что же делать? Ведь еще и фамилии авторов не всегда уместаются!

Перенос по словам

Выход есть, и он очевиден: раз названия занимают очень много места, то почему бы не использовать для названия две строки вместо одной? Есть такой дополнительный режим, именуемый “переносить по словам”, который можно заказать, когда выбираешь способ выравнивания текста в ячейке. Этот режим позволит уместить все, что нужно, в более узкий столбец. Мы зададим для него ширину 18,00 вместо имеющихся 27,00.

Это ничего, что названия снова стали урезанными. Выделим сейчас для простоты весь столбец А (щелчком по самой букве вверху) и, когда он станет черным, в меню Формат обратимся к пункту Ячейки. Появится знакомое по первому занятию диалоговое окно с шестью закладками и Выравниванием в том числе. Напротив упомянутого режима поставим галочку, а заодно отцентрируем текст по вертикали.

	A	B	C	D	E	F	G	H	I	J	K
1	Текущее состояние дел в книжной торговле.										
2											
3	Название	Автор	Цена опт.	Цена роз.	Кол-во	Оплачено	Продано	Приход	Расход	Баланс	Возврат
4	Практическая работа с MS Excel 97	Долженков	40р.	50р.	30	10	8	400р.	320р.	80р.	
5	Excel одним взглядом	Вострокнути	15р.	20р.	50	30	28	560р.	360р.	200р.	
6	Шпаргалка по Excel 7.0	Столяров	10р.	15р.	40	20	35	525р.	310р.	215р.	
7	Разработка приложений в Access 97	Нортон	75р.	90р.	6	6	2	180р.	360р.	-180р.	
8	Access 97 Библиотека ресурсов	О'Брайен	70р.	85р.	5	0	2	170р.	140р.	30р.	
9	Excel 97 Библиотека ресурсов	Уэллс	70р.	85р.	5	0	1	85р.	70р.	15р.	
10	Access 7.0 в примерах	Гончаров	35р.	45р.	15	10	15	675р.	455р.	220р.	
11											
12				Всего:	151	76	91	2 595р.	2 015р.	580р.	
13											
14											
15											
16											
17											
18											
19											
20											
21											
22											

В первом столбце — почти то, что нужно. Хотелось бы только, чтобы все строки с названиями книг имели одинаковую высоту, независимо от длины названия, от того, одну строчку оно занимает или две.

Высоту строки меняем так, как только что меняли ширину столбца. Например, для строки с заголовками колонок (имеющей номер три) подводим курсор к границе между ней и следующей строкой (эта граница проходит между цифрами 3 и 4, указанными слева). “Зацепив” границу, “протягиваем” ее вниз, до значения, допустим, 24. Точно так же для итоговой строки устанавливаем высоту поменьше, скажем, 15. Для каждой из строк — с четвертой по одиннадцатую — сразу, за один прием (конечно, сначала выделив их, “пройдясь” по номерам), задаем высоту 27 — для этого можно двигать любую границу, попавшую в выделение.

Теперь займемся другими столбцами. Для второго, с фамилиями авторов, произведем автоподбор ширины. Это быстрый и удобный способ установить подходящую ширину столбца. Подводим курсор к границе, проходящей между буквами В и С, и не просто нажимаем левую клавишу мыши, а делаем двойной щелчок. Вот и весь “автоподбор”.

Остальные же столбцы, с С по К включительно, выделим, а затем установим для всех сразу ширину 6 (вместо заданного изначально размера 8,43) — методом проб можно определить, что при таком значении все колонки уже смогут вписаться в пунктирную границу.

Заголовки колонок, правда, срежутся, не будут полностью видны. Но уже известно, как с такими нежелательными явлениями бороться. Надо снова заказать перенос по словам, только теперь для всех ячеек третьей строки. А заодно, чтобы было красиво, произвести центрирование и по вертикали, и по горизонтали.

Однако как же сделать “перенос по словам”, если слово всего одно? А очень просто. Его надо разбить, используя дефис. Например, вместо “Оплачено”, ввести “Опла-чено”. Вот две строки и получатся.

Разлиновка таблицы

Теперь пора заняться проведением разделительных линий в нашей таблице. Самый простой, но и самый непритязательный путь здесь — использование одной из кнопок панели инструментов форматирования. Само собой разумеется, что сначала надо выделить объект действия — прямоугольную область (A3:K12), т.е. практически всю таблицу, за исключением первых двух строк, отведенных для ее названия.

Кнопка, о которой идет речь, называется Границы и имеет с правого своего бока маленький черный треугольничек-стрелку, позволяющий раскрыть список предоставляемых возможностей. Список не так уж велик: всего 12 видов границ предлагается. Какой-то из этих видов мы должны выбрать; остановимся, скажем, на втором слева внизу. Щелкнув по нему, мы получим то, что заказали, — банальную сетку. Стоит только снять оставшееся после операции выделение, и она прекрасно будет видна на экране.

Хотелось бы, однако, иметь что-нибудь более выразительное. Сделаем так, чтобы третья строка с заголовками колонок была отделена от всех последующих не обычной тонкой, а, допустим, жирной линией. Или, скажем, двойной. И того и другого одинаково легко добиться. Действия почти те же самые: выделяем не всю таблицу, а только указанные заголовки (A3:K3), потом обращаемся к списку границ и выбираем нужную границу — первую слева посередине. По виду это двойная черта снизу, под выделенными ячейками. Так получается нагляднее.

Пожалуй, надо, чтобы и нижняя строка таблицы, с итогами, тоже как-то отличалась от других. Можно использовать тот же самый вид границы с двойной чертой внизу, но только выделить тогда нужно будет предыдущую строчку. Не (A12:K12), как, казалось бы, должно быть, а (A11:K11), поскольку граница-то будет нижней.

Всю область (A3:K3), на которую наложена сетка, можно красиво обвести по периметру одной замкнутой жирной линией. Соответствующая толстая внешняя окантовка представлена в общем списке границ (в правом нижнем углу).

Еще большего “визуального” эффекта удастся достигнуть, если слегка закрасить верхнюю и нижнюю строки таблицы, средства для этого тоже имеются. Так мы и поступим. Причем для такой одной общей операции выделим сразу две области: сначала (A3:K3), а потом, удерживая клавишу **Ctrl**, дабы не сбросить первое выделение, (A12:K12). Справа от кнопки Границы расположена другая, которая называется Цвет заливки, она-то нам и нужна. Какой именно цвет выбирать из списка — конечно, дело вкуса, но лучше отдать предпочтение неярким тонам. Скромная 25%-ная градация серого будет здесь наиболее уместна. Ну и, наконец, для ячейки A1 следует отменить перенос по словам и взять шрифт размером 14.

Текущее состояние дел в книжной торговле.										
Название	Автор	Цена опт.	Цена роз.	Количество	Оплачено	Продано	Приход	Расход	Баланс	Возврат
Практическая работа с MS Excel 97	Долженков	40р.	50р.	30	10	8	400р.	320р.	80р.	20
Excel одним взглядом	Вострокнутов	15р.	20р.	50	30	28	560р.	360р.	200р.	20
Шпаргалка по Excel 7.0	Столяров	10р.	15р.	40	20	35	525р.	310р.	215р.	5
Разработка приложений в Access 97	Нортон	75р.	90р.	6	6	2	180р.	360р.	-180р.	0
Access 97 Библиотека ресурсов	О'Брайен	70р.	85р.	5	0	2	170р.	140р.	30р.	3
Excel 97 Библиотека ресурсов	Уэллс	70р.	85р.	5	0	1	85р.	70р.	15р.	4
Access 7.0 в примерах	Гончаров	35р.	45р.	15	10	15	675р.	455р.	220р.	0
Всего:				151	76	91	2 595р.	2 015р.	580р.	52

Все, конец второго занятия! У документа приличный вид, и он достоин того, чтобы его сохранили на диске в папке “Упражнения” в виде файла с именем “Книжная торговля”.

Вопросы для проверки

1. Как записать формулу для подсчета возвращаемого количества книг, используя не функцию МАКС, а функцию ЕСЛИ?
2. Каков будет суммарный баланс, если удастся распродать все книги до единой?
3. Каков будет суммарный баланс в другом крайнем случае, когда торговля идет очень плохо?
4. Каким должно быть минимальное количество проданных книг, чтобы баланс для каждого названия стал положительным?
5. Каков был бы суммарный баланс на настоящий момент, если бы по всем названиям была совершена полная предоплата?
6. Наоборот, если бы предоплаты не было вовсе, то каков был бы суммарный баланс на настоящий момент?
7. Если дополнительно произвести расчет прибыли, принесенной одним экземпляром книги, то какое название в этом смысле было бы наиболее успешным?
8. Можно ли использовать автоподбор ширины для столбца, содержащего названия книг?
9. Сколько всего имеется типов разграничительных линий между ячейками таблицы?
10. Какой может быть заливка клеток, если не считать разные цвета и оттенки серого?

Задания для самостоятельной работы

1. На небольшом предприятии сотрудникам выдается месячная зарплата. Кассир, имея на руках бухгалтерскую ведомость, собирается ехать в банк за деньгами. При этом он имеет возможность не просто подать заявку на общую сумму, а еще и указать, какими именно купюрами ему хотелось бы данную сумму получить. Делается это для того, чтобы потом было удобнее рассчитываться с каждым работником. Предварительно все исходные данные нужно представить в виде таблицы. Как только вводится сумма начисленной сотруднику зарплаты, она тут же указывается против его фамилии в виде соответствующего набора денежных знаков. Одновременно подводится и общий итог по всему коллективу для формирования банковского заказа.

Указания к исполнению

При определении набора следовать принципу наименьшего числа купюр.

Ради упрощения задачи расчет вести только на бумажные деньги с точностью до пяти рублей.

Для расчетов использовать встроенные функции ЦЕЛОЕ и ОСТАТ из категории “математических”.

2. В школе проходит шахматный турнир, в котором участвуют 12 лучших игроков из разных классов. Соревнование проводится по круговой системе: каждый играет с каждым один раз. Результаты заносятся в обычную турнирную таблицу — с диагональю заштрихованных клеточек. В остальные клетки вносятся единицы за победы, нули за поражения и половинки за ничью. Набранные на текущий день очки суммируются, и результаты выводятся в итоговый столбец. Лидеры турнира (если их, как это часто бывает, несколько) должны быть в таблице выделены.

Указания к исполнению

При вводе результатов сыгранных партий исключить во избежание случайных ошибок избыточность данных, иными словами, нули, единицы и половинки следует вводить только в верхнюю часть таблицы, над диагональю. В нижнюю часть, под диагональ, соответствующие показатели должны заноситься автоматически.

Учесть при оформлении таблицы, что она вывешивается на всеобщее обозрение и потому должна быть легко читаемой.

Под таблицей указать возрастной диапазон участников.

Продолжение следует

Задачи по системам счисления

Л.Л. Акуленко–Босова

Предлагаем вашему вниманию подборку задач по теме “Системы счисления”, ориентированную на учеников 5—7-х классов

1. Какой числовой эквивалент у цифры 6 в числах:

- а) 6789; в) 16;
б) 3650; г) 69?

Ответ:

- а) 6 тысяч; в) 6 единиц;
б) 6 сотен; г) 6 десятков.

2. Вы знакомы с римскими цифрами. Первые три из них — I, V, X. Их легко изобразить, используя палочки или спички; так же легко выложить спичками минус, плюс и знак равенства. Ниже написано несколько неверно решенных примеров. Как можно получить верные решения, если разрешается переложить с одного места на другое только одну спичку (палочку)?

$$\begin{array}{ll} VII - V = XI & IX - V = VI \\ VI - IX = III & VII - III = X \end{array}$$

Решение

Будьте внимательны — в каждом примере перенесена только одна спичка (палочка). Существуют и другие варианты решений.

$$\begin{array}{ll} VI + V = XI & XI - V = VI \\ VI = IX - III & VIII + II = X \end{array}$$

3. Сравните числа VVV и 555.

Решение

$$\begin{array}{l} VVV < 555, \text{ так как} \\ VVV = (5 + 5 + 5) = 15 \end{array}$$

4. Запишите в десятичной системе счисления:

- а) MCMXCIX;
б) CMLXXXVIII;
в) MCXLVII.

Решение

$$\begin{array}{ll} \text{а)} & (1000) & \text{M} \\ & (1000 - 100) & \text{CM} \\ + & (100 - 10) & \text{XC} \\ & (10 - 1) & \text{IX} \\ \hline & 1999; & \\ \text{б)} & 988; & \\ \text{в)} & 1147. & \end{array}$$

5. Запишите число, месяц и год своего рождения с помощью римских цифр.

Пример

$$\begin{array}{ll} 1 \text{ августа} & 1989 \text{ года} \\ I & VIII \quad \text{MCMLXXXIX} \end{array}$$

6. Трехзначное десятичное число оканчивается цифрой 3. Если эту цифру переместить на два разряда влево, т.е. так, что с нее будет начинаться запись нового числа, то это новое число будет на единицу больше утроенного исходного числа. Найдите исходное число.

Решение

$$\text{Исходное число } \overline{ab3} = a \cdot 100 + b \cdot 10 + 3.$$

$$\text{Новое число } \overline{3ab} = 3 \cdot 100 + a \cdot 10 + b.$$

По условию

$$\begin{array}{l} 3 \cdot 100 + a \cdot 10 + b = 3 \cdot (a \cdot 100 + b \cdot 10 + 3) + 1; \\ 3 \cdot 100 + a \cdot 10 + b = 3 \cdot a \cdot 100 + 3 \cdot b \cdot 10 + 10; \\ 3 \cdot 100 + a \cdot 10 + b = 3 \cdot a \cdot 100 + (3 \cdot b + 1) \cdot 10 + 0. \end{array}$$

Рассмотрим последнее равенство. Сравнивая коэффициенты при 100 (3 и $3 \cdot a$) и при 10 (b и 0), делаем вывод, что $a = 1$ и $b = 0$.

Таким образом, исходное число — 103.

7. Шестизначное число оканчивается цифрой 4. Если эту цифру переставить из конца числа в начало, то есть приписать ее перед первой, не изменяя порядка остальных пяти, то получится число, которое в четыре раза больше первоначального. Найдите это число.

Решение

Из условия задачи следует, что

$$\overline{4abcde} = 4 \cdot \overline{abcde4}. \quad (1)$$

Представим каждое число в виде разложения по степеням 10.

$$\overline{4abcde} = 4 \cdot 100000 + a \cdot 10000 + b \cdot 1000 + c \cdot 100 + d \cdot 10 + e; \quad (2)$$

$$\begin{array}{l} 4 \cdot \overline{abcde4} = 4 \cdot (a \cdot 100000 + b \cdot 10000 + \\ + c \cdot 1000 + d \cdot 100 + e \cdot 10 + 4) = \\ = 4 \cdot (a \cdot 100000 + b \cdot 10000 + c \cdot 1000 + \\ + d \cdot 100) + 4 \cdot e \cdot 10 + 16 = \\ = 4 \cdot (a \cdot 100000 + b \cdot 10000 + c \cdot 1000 + \\ + d \cdot 100) + (4 \cdot e + 1) \cdot 10 + 6. \end{array} \quad (3)$$

Так как рассматриваемые числа равны, то число единиц в них совпадает и $e = 6$. Подставим значение e в выражение (3):

$$\begin{aligned} & 4 \cdot (a \cdot 100000 + b \cdot 10000 + c \cdot 1000 + \\ & + d \cdot 100 + 4 \cdot 6 + 1) \cdot 10 + 6 = \\ & = 4 \cdot (a \cdot 100000 + b \cdot 10000 + c \cdot 1000 + \\ & + d \cdot 100) + 25 \cdot 10 + 6 = \quad (3') \\ & = 4 \cdot (a \cdot 100000 + b \cdot 10000 + \\ & + c \cdot 1000) + (d + 2) \cdot 100 + 5 \cdot 10 + 6 \end{aligned}$$

Сравнивая (2) и (3'), заключаем, что $d = 5$.

Проводя аналогичные рассуждения, получаем: $c = 2$, $b = 0$, $a = 1$.

Окончательный результат — число 102 564.

8. Некогда был пруд, в центре которого вырос один лист водяной лилии. Каждый день число листьев удваивалось, и на десятый день вся поверхность пруда уже была заполнена листьями лилий.

Сколько дней понадобилось, чтобы заполнить листьями половину пруда? Сосчитайте, сколько листьев выросло к концу десятого дня.

Решение

Для наглядности составим таблицу:

День	1	2	3	4	5	6	7	8	9	10
Листья	2	4	8	16	32	64	128	256	512	1024

Ответ. 9 дней, 1024 листа.

9. Этот случай вполне мог иметь место во времена “золотой лихорадки”. На одном из приисков старатели были возмущены действиями Джо Макдональда — хозяина салуна, принимавшего от них в уплату золотой песок. Очень уж необычными были гири, с помощью которых тот взвешивал золото: 1, 2, 4, 8, 16, 32 и 64 грамма. Джо утверждал, что с помощью такого набора гирь он может взвесить с точностью до грамма любую порцию золотого песка, вес которой не превышает 100 граммов.

Прав ли Джо Макдональд? Какой наибольший вес могут “взять” такие гири? Как с помощью названных гирь набрать вес:

- а) 25 г; в) 72 г;
б) 48 г; г) 105 г?

Решение

- а) $25 = 16 + 8 + 1$;
б) $48 = 32 + 16$;
в) $72 = 64 + 8$;
г) $105 = 64 + 32 + 8 + 1$.

Наибольший вес получится, если задействовать все гири:

$$1 + 2 + 4 + 8 + 16 + 32 + 64 = 127.$$

10. Найдите такой набор из пяти гирь, чтобы, располагая их на одной чаше весов, можно было бы взвесить с точностью до 1 кг любой груз до 31 кг включительно.

Ответ. 1, 2, 4, 8 и 16 кг.

11. Каким наименьшим числом гирь можно взвесить груз от 1 до 63 кг с точностью до 1 кг, помещая гири только на одну чашку весов?

Ответ. 1, 2, 4, 8, 16 и 32 кг.

12. Можно ли с помощью трех гирь (1, 3 и 9 кг) взвесить с точностью до 1 кг любой груз до 13 кг включительно, если гири можно располагать на обеих чашах весов, в том числе и на чаше с грузом?

Решение

Чаша с грузом		Чаша с гирями
Груз	Гири	
1	—	1
2	1	3
3	—	3
4	—	1, 3
5	1, 3	9
6	3	9
7	3	1, 9
8	1	9
9	—	9
10	—	1, 9
11	1	3, 9
12	—	3, 9
13	—	1, 3, 9

13. Один кладовщик оказался в большом затруднении: заказанный комплект гирь для простых чашечных весов не прибыл в срок, а на соседнем складе лишних гирь тоже не было. Тогда он решил подобрать несколько кусков железа разной массы и временно пользоваться ими как гирями. Ему удалось выбрать такие четыре “гири”, с помощью которых можно было бы взвешивать с точностью до 100 г товар от 100 г до 4 кг. Подумайте, какой массы были эти “гири”.

Ответ. “Гири” имели массу 100, 300, 900 и 2700 г.

14. Заполните следующую таблицу:

Система счисления	Основание	Базис
Десятичная	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Восьмеричная	8	
	7	
		0, 1, 2, 3, 4, 5
Пятеричная		
Троичная		
		0,1

15. Заполните следующую таблицу, пользуясь для записи чисел десятичной системой счисления:

Система счисления	Основание	Разряды (степени)				
		10 000	1000	100	10	1
Десятичная	10					
Восьмеричная	8					
Шестеричная	6					
Пятеричная	5					
Троичная	3					
Двоичная	2					

16. “Чудесная таблица”. Запишем все числа от 1 до 15 в двоичной системе.

- | | |
|-------------------|--------------------|
| $1_{10} = 0001_2$ | $10_{10} = 1010_2$ |
| $2_{10} = 0010_2$ | $11_{10} = 1011_2$ |
| $3_{10} = 0011_2$ | $12_{10} = 1100_2$ |
| $4_{10} = 0100_2$ | $13_{10} = 1101_2$ |
| $5_{10} = 0101_2$ | $14_{10} = 1110_2$ |
| $6_{10} = 0110_2$ | $15_{10} = 1111_2$ |
| $7_{10} = 0111_2$ | |
| $8_{10} = 1000_2$ | |
| $9_{10} = 1001_2$ | |

Выпишем теперь десятичные числа в четыре строки, придерживаясь следующего правила:

- в строку I записываем все числа, в двоичной записи которых есть единицы первого разряда (сюда попадут все нечетные числа);
- в строку II записываем все числа, у которых есть единицы второго разряда;
- в строку III — числа, имеющие единицы третьего разряда, и в строку IV — числа, имеющие единицы четвертого разряда. Таблица будет иметь вид:

I	1	3	5	7	9	11	13	15
II	2	3	6	7	10	11	14	15
III	4	5	6	7	12	13	14	15
IV	8	9	10	11	12	13	14	15

Теперь можно кому-нибудь предложить задумать любое число от 1 до 15 и указать все строки таблицы, в которых оно записано. Пусть, к примеру, указали, что задуманное число находится в строках I и III. Значит, задуманное число содержит в двоичной записи единицы первого и третьего разрядов, а единиц второго и четвертого разрядов в нем нет. Следовательно, задумано число $101_2 = 5_{10}$.

Запишите все числа от 1 до 31 в двоичной системе и заполните соответствующую таблицу из пяти строк.

17. Переведите следующие числа из двоичной системы счисления в десятичную:

- | | | |
|---------|---------|---------|
| 1) 1000 | 2) 0001 | 3) 0110 |
| 4) 0011 | 5) 0101 | 6) 0111 |
| 7) 0100 | 8) 1001 | 9) 0010 |

Полученные числа впишите в соответствующие клетки квадрата.

1	2	3
4	5	6
7	8	9

Сложите числа в каждом столбце, каждой строке или в любой из диагоналей — и убедитесь, что данный квадрат магический.

Решение

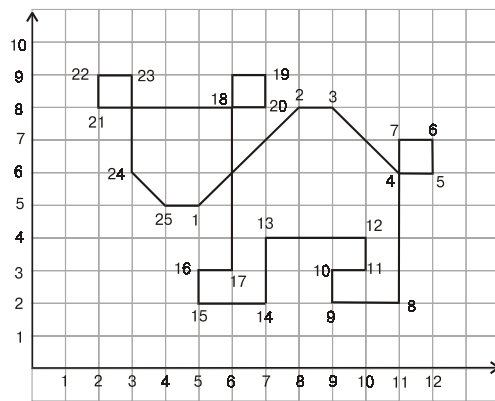
8	1	6
3	5	7
4	9	2

Сумма чисел в каждом столбце, каждой строке или в любой из диагоналей равна 15.

18. Отметьте и последовательно соедините на координатной плоскости точки, координаты которых записаны в двоичной системе счисления.

- 1(101,101), 2(1000,1000), 3(1001,1000), 4(1011,110), 5(1100,110), 6(1100,111), 7(1011,111), 8(1011,10), 9(1001,10), 10(1001,11), 11(1010,11), 12(1010,100), 13(111,100), 14(111,10), 15(101,10), 16(101,11), 17(110,11), 18(110,1001), 19(111,1001), 20(111,1000), 21(10,1000), 22(10,1001), 23(11,1001), 24(11,110), 25(100,101).

Ответ



19. Воспользуйтесь таблицей из номера 15, чтобы записать следующие числа:

- а) в восьмеричной системе счисления:
7, 9, 24, 35, 57, 64;
- б) в пятеричной системе счисления:
9, 13, 21, 36, 50, 57;
- в) в троичной системе счисления:
3, 6, 12, 25, 27, 29;
- г) в двоичной системе счисления:
2, 5, 7, 11, 15, 25.

Ответ

- а) 7, 11, 30, 43, 71, 100;
 б) 14, 23, 41, 121, 200, 212;
 в) 10, 20, 110, 221, 1000, 1002;
 г) 10, 101, 111, 1011, 1111, 11001.

20. Для записи десятичных чисел в других системах счисления с основанием меньше 10 надо данное число нацело разделить на основание новой системы, частное опять разделить на основание новой системы — и так до тех пор, пока не получим частное, меньшее основания новой системы. Запись числа в новой системе составляется так: слева направо записывается последнее частное, за ним последовательно все остатки от деления “с конца”.

Пример

$$\begin{array}{r|l}
 2000 & 8 \\
 \hline
 0 & 250 \\
 & \hline
 & 2 & 31 & 8 \\
 & & \hline
 & & 7 & 3
 \end{array}$$

Воспользуйтесь этим правилом для перевода числа 2000 в следующие системы счисления:

- а) в восьмеричную;
 б) в пятеричную;
 в) в двоичную.

Ответ

- а) 3720; б) 31000; в) 11111010000.

21. Задача-игра “Угадывание задуманного числа по отрезкам”. Один из учеников (ведущий) задумывает некоторое трехзначное число, мысленно делит задуманное число пополам, если число нечетное, то от него отбрасывается единица. Полученную половину опять делит пополам — и так до тех пор, пока не получится 0. При каждом делении ведущий чертит на доске новый отрезок ломаной, направленный вертикально, если делится нечетное число, и горизонтально, если делится четное число. Как на основании полученной фигуры безошибочно определить задуманное число?

Ответ

Ломаная “расшифровывается” так. Движемся по отрезкам от первого до последнего. Вместо каждого отрезка **справа налево** записываются цифры. Вертикальный отрезок “заменяется” единицей, горизонтальный — нулем. Это двоичная запись задуманного числа.

22. Какое минимальное основание имеет система счисления, если в ней записаны числа 123, 222, 111, 241? Определите десятичный эквивалент данных чисел в найденной системе счисления.

Решение

Минимальное основание системы счисления — 5. Чтобы найти десятичный эквивалент чисел, записанных в пятеричной системе счисления, представим каждое число в виде суммы соответствующих разрядных слагаемых:

$$\begin{aligned}
 123_5 &= 1 \cdot 25 + 2 \cdot 5 + 3 \cdot 1 = 38_{10} \\
 222_5 &= 62_{10}; \quad 111_5 = 31_{10}; \quad 241_5 = 71_{10}.
 \end{aligned}$$

23. Запишите наибольшее двузначное число и определите его десятичный эквивалент для следующих систем счисления:

- а) восьмеричной системы счисления;
 б) пятеричной системы счисления;
 в) троичной системы счисления;
 г) двоичной системы счисления.

Решение

- а) $77_8 = 63_{10}$; в) $22_3 = 8_{10}$;
 б) $44_5 = 24_{10}$; г) $11_2 = 3_{10}$.

24. Запишите наименьшее трехзначное число и определите его десятичный эквивалент для следующих систем счисления:

- а) восьмеричной системы счисления;
 б) пятеричной системы счисления;
 в) троичной системы счисления;
 г) двоичной системы счисления.

Решение

- а) $100_8 = 64_{10}$; в) $100_3 = 9_{10}$;
 б) $100_5 = 25_{10}$; г) $100_2 = 4_{10}$.

25. Найдите десятичные эквиваленты следующих чисел:

$$\begin{array}{lll}
 3221_8; & 4321_6; & 1221_3; \\
 1221_3; & 1011_2. &
 \end{array}$$

Решение

Перевод в десятичную систему можно проводить так:
 $((3 \cdot 8 + 2) \cdot 8 + 2) \cdot 8 + 1 = 1681$;
 $((4 \cdot 6 + 3) \cdot 6 + 2) \cdot 6 + 1 = 985$;
 $((1 \cdot 5 + 2) \cdot 5 + 2) \cdot 5 + 1 = 186$;
 $((1 \cdot 3 + 2) \cdot 3 + 2) \cdot 3 + 1 = 52$;
 $((1 \cdot 2 + 0) \cdot 2 + 1) \cdot 2 + 1 = 11$.

26. Упорядочите следующие числа по убыванию:

$$\begin{array}{lll}
 143_6; & 50_9; & 1222_3; \\
 1011_4; & 110011_2; & 123_8.
 \end{array}$$

Решение

Переведем все числа в десятичную систему:

$$\begin{array}{lll}
 143_6 = 63_{10}; & 50_9 = 45_{10}; & 1222_3 = 53_{10}; \\
 1011_4 = 69_{10}; & 110011_2 = 51_{10}; & 123_8 = 83_{10}.
 \end{array}$$

Ответ

$$123_8, 1011_4, 143_6, 1222_3, 110011_2, 50_9.$$

27. В классе $111100_2\%$ девочек и 1100_2 мальчиков. Сколько учеников в классе?

Решение

Переведем числа в условии задачи в десятичную систему счисления.

В классе 60% девочек и 12 мальчиков. Сколько учеников в классе?

Ответ

В классе 30 учеников.

28. У меня 100 братьев. Младшему 1000 лет, а старшему 1111 лет. Старший учится в 1001-м классе. Может ли такое быть?

Ответ

Может, если считать, что все данные приведены в двоичной системе.

29. В двоичной системе счисления таблица сложения имеет вид:

$$0 + 1 = 1,$$

$$1 + 1 = 10.$$

Составьте таблицы сложения в следующих системах счисления:

- а) в восьмеричной системе счисления;
- б) в пятеричной системе счисления;
- в) в троичной системе счисления.

Решение

Оформим таблицы сложения аналогично той, что приводится на обложках тетрадей в клетку:

а)

	1	2	3	4	5	6	7
1	2	3	4	5	6	7	10
2	3	4	5	6	7	10	11
3	4	5	6	7	10	11	12
4	5	6	7	10	11	12	13
5	6	7	10	11	12	13	14
6	7	10	11	12	13	14	15
7	10	11	12	13	14	15	16

б)

	1	2	3	4
1	2	3	4	10
2	3	4	10	11
3	4	10	11	12
4	10	11	12	13

в)

	1	2
1	2	10
2	10	11

30. Выполните операцию сложения над двоичными числами:

- а) $1011 + 100$; г) $1001 + 11$;
- б) $10010 + 101$; д) $11101 + 101$;
- в) $1011 + 1100$; е) $1101 + 1011$.

Для того чтобы убедиться в правильности полученных результатов, найдите десятичные эквиваленты слагаемых и суммы. Сложение удобно выполнять в столбик.

Ответ

- а) 1111; г) 1100;
- б) 10111; д) 100010;
- в) 10111; е) 11000.

31. Найдите суммы следующих чисел в троичной системе:

- а) $101+121$; б) $2012+1211$.

Ответ

- а) 222; б) 11000.

32. Найдите суммы следующих чисел в пятеричной системе:

- а) $221 + 104$; б) $432 + 114$.

Ответ

- а) 330; б) 1101.

33. Найдите суммы следующих чисел в восьмеричной системе:

- а) $66+43$; б) $515+324$.

Ответ

- а) 131; б) 1041.

34. В классе 1000_q учеников, из них 120_q девочек и 110_q мальчиков. В какой системе счисления велся счет учеников?

Решение

$$\begin{array}{r} 120 \\ + 110 \\ \hline 1000 \end{array}$$

Проанализировав результат, получим $q = 3$, так как только в троичной системе счисления $2 + 1 = 10$.

35. В саду 88_q фруктовых деревьев, из них 32_q яблонь, 22_q груш, 16_q слив и 17_q вишен. В какой системе счисления посчитаны деревья?

$$8 \cdot q + 8 = 3 \cdot q + 2 + 2 \cdot q + 2 + 1 \cdot q + 6 + 1 \cdot q + 7;$$

$$8 \cdot q - 3 \cdot q - 2 \cdot q - 1 \cdot q - 1 \cdot q = 2 + 2 + 6 + 7 - 8.$$

Ответ

Деревья посчитаны в девятеричной системе счисления.

36. Было 53_q яблока. После того как каждое из них разрезали пополам, стало 136_q половинок. В системе счисления с каким основанием вели счет?

Решение

$$\begin{array}{r} 53 \\ + 53 \\ \hline 136 \end{array}$$

$5 + 5 = 13$; в семеричной системе счисления.

37. Один мальчик так написал о себе: “У меня 24 пальца, на каждой руке по 5, а на ногах 12”. Как это могло быть?

Решение

Так как $5 + 5 = 12$, то речь идет о восьмеричной системе счисления. Так что мальчик наш — абсолютно нормальный ребенок, изучивший восьмеричную систему счисления.

38. В бумагах одного математика найдена была его биография. Она начиналась следующими удивительными словами: “Я окончил курс университета 44 лет от роду. Спустя год, 100-летним молодым человеком, я женился на 34-летней девушке. Незначительная разница в возрасте — всего 11 лет — способствовала тому, что мы жили общими интересами и мечтами.

Спустя немного лет у меня была уже и маленькая семья из 10 детей. Жалованья я получал в месяц всего 200 рублей, из которых $1/10$ приходилось отдавать сестре, так что мы с детьми жили на 130 руб. в месяц”; и т.д.

Чем объяснить странные противоречия в числах?

Решение

Недесятичная система счисления — вот единственная причина кажущейся противоречивости приведенных чисел. Основание этой системы определяется фразой: “Спустя год (после 44 лет), 100-летним молодым человеком...”. Если от прибавления одной единицы число 44 преобразуется в 100, то, значит, цифра 4 — наибольшая в этой системе (как 9 — в десятичной), а

следовательно, основанием системы является 5. Можно высказать предположение, что все числа в автобиографии записаны в пятеричной системе счисления, и путем несложных преобразований восстановить их истинный смысл.

39. Выпишите целые десятичные числа, принадлежащие числовому промежутку $[101_2; 1100_2]$.

Ответ

5, 6, 7, 8, 9, 10, 11, 12.

40. Запишите в двоичной системе целые числа, принадлежащие числовому промежутку $[101101_2; 110000_2]$.

Ответ

101101, 101110, 101111, 110000.

41. Какое число предшествует каждому из данных:

а) 10001_2 ; б) 1000_2 ?

Ответ для каждого числа запишите в двоичной и десятичной системах счисления.

Ответ

а) $10000_2 = 16_{10}$; б) $111_2 = 7_{10}$.

42. Выполните операцию вычитания над двоичными числами:

а) $1100 - 10$; в) $1100 - 111$;
б) $1000 - 11$; г) $11011 - 1110$.

Ответ

а) 1010; в) 101;
б) 101; г) 1101.

43. Восстановите неизвестные цифры, обозначенные знаком вопроса, в следующих примерах на сложение и вычитание, определив вначале, в какой системе изображены числа.

$$\begin{array}{r} \text{а) } \quad 2?21 \quad \quad \quad \text{г) } \quad 4?5 \\ + 123? \quad \quad \quad - 136 \\ \hline \quad ?203 \quad \quad \quad \quad ?56 \end{array}$$

$$\begin{array}{r} \text{б) } \quad 5?55 \quad \quad \quad \text{д) } \quad 1536 \\ + ?327 \quad \quad \quad - ?42 \\ \hline \quad ?16?4 \quad \quad \quad \quad 67? \end{array}$$

$$\begin{array}{r} \text{в) } \quad 21?02 \\ + ?1212 \\ \hline \quad ?2?021 \end{array}$$

Ответ

а)

$$\begin{array}{r} 2421 \\ + 1232 \\ \hline 4203 \end{array}$$

пятеричная с/с

б)

$$\begin{array}{r} 5255 \\ + 4327 \\ \hline 11604 \end{array}$$

восьмеричная с/с

в)

$$\begin{array}{r} 21102 \\ + 21212 \\ \hline 120021 \end{array}$$

троичная с/с

г)

$$\begin{array}{r} 425 \\ - 136 \\ \hline 256 \end{array}$$

семеричная с/с

д)

$$\begin{array}{r} 1536 \\ - 642 \\ \hline 674 \end{array}$$

восьмеричная с/с

44. Дайте “серьезные” ответы на “несерьезные” вопросы.

Вопросы

Когда $2 \cdot 2 = 100$?

Когда $2 \cdot 2 = 11$?

Когда 10 — число нечетное?

Когда $2 \cdot 3 = 11$?

Когда $3 \cdot 3 = 13$?

Когда $21 + 24 = 100$?

Когда $22 + 44 = 110$?

Когда $3 + 4 = 7$, а $3 \cdot 4 = 13$?

Когда $6 \cdot 6 = 44$?

Когда $4 \cdot 4 = 20$?

Ответы

В 2-ичной с/с.

В 3-ичной с/с.

В любой с/с с нечетным основанием.

В 5-ричной с/с.

В 6-ричной с/с.

В 5-ричной с/с.

В 6-ричной с/с.

В 9-ричной с/с.

В 8-ричной с/с.

В 8-ричной с/с.

45. Выполните операцию умножения над двоичными числами:

а) $101 \cdot 10$;

б) $11 \cdot 11$;

в) $110 \cdot 11$;

г) $101 \cdot 111$.

Ответ

а) 1010;

б) 1001;

в) 10010;

г) 100011.

46. Выполните операцию деления над двоичными числами:

а) $1100 : 100$;

б) $100100 : 1100$;

в) $10010110 : 101$;

г) $100000011 : 111$.

Ответ

а) 11;

б) 11;

в) 1111;

г) 100101.

47. Вычислите значения двоичных выражений:

а) $(11001—1111) : 10$; в) $110001 : 111 — 100$;

б) $1100100 : 100—1111$; г) $11 \cdot 11 + 1011$.

Ответ

а) 101;

в) 11;

б) 1010;

г) 10100.

48. Расставьте знаки арифметических операций так, чтобы были верны следующие равенства в двоичной системе:

а) $1100 ? 11 ? 100 = 100000$;

б) $1100 ? 10 ? 10 = 100$;

в) $1100 ? 10 ? 10 = 110000$;

г) $1100 ? 10 ? 10 = 1011$;

д) $1100 ? 11 ? 100 = 0$.

Решение

Перепишав условие в десятичной системе, получим:

а) $12 \cdot 3 — 4 = 32$;

г) $12 — 2 : 2 = 11$;

б) $12 : 2 — 2 = 4$;

д) $12 — 3 \cdot 4 = 0$.

в) $12 \cdot 2 \cdot 2 = 48$;

49. Фокусник высыпает на стол монеты на сумму 3 рубля и предлагает задачу: разложить деньги по 9 кошелькам так, чтобы можно было уплатить любую сумму до 3 рублей, не открывая кошельков. Как можно разложить монеты? (Напоминаем, что существовали монеты достоинством в 1, 2, 3, 5, 10, 15, 20 копеек.)

Ответ

Монеты можно разложить так:

1, 2, 4, 8, 16, 32, 64, 128 и 45 копеек.

Литература

1. Босова А.А. Системы счисления. М.: Информатика, № 7/97.

2. Мазаник А.А. Реши сам. Мн.: Нар. асвета, 1980.

3. Программы общеобразовательных учреждений. Начальные классы (1—3) по системе Л.В. Занкова. М.: Просвещение, 1996.

4. Программы общеобразовательных учреждений. Начальные классы (1—5) по системе Д.Б. Эльконина — В.В. Давыдова. М.: Просвещение, 1996.

5. Семакин И., Залогова Л., Русаков С., Шестакова Л. Информатика. Учебник по базовому курсу. М.: ООО “Издательство Лаборатория Базовых Знаний”, 1998.

6. Фалина И., Андреева Е. Системы счисления и компьютерная арифметика. М.: ООО “Издательство Лаборатория Базовых Знаний”, 1999.

Транслятор?.. Это очень просто!

— А.А. Дуванов

Напрасно я объяснял своим друзьям, что телевизионная техника не только не проста, но, согласно выражению Незнайкина, дьявольски сложна, что она затрагивает различные области физики, что положение еще усложняется из-за отсутствия международного стандарта.

Ничего не помогло. Я должен был покориться и написать: "Телевидение?.. Это очень просто!"

Е.Айсберг

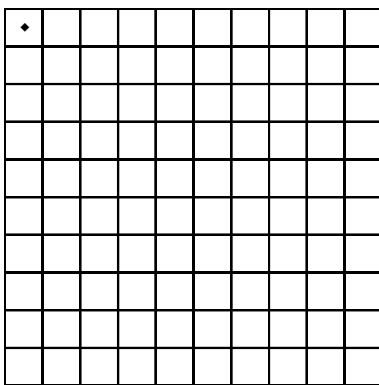
Эти заметки для тех, кто твердо уверен, что кулинария и мясорубка — это не одно и то же, но тем не менее готов ради хорошей кухни покрутить ручку, пренебрегая всякими электрическими излишествами. Ведь в “ручной” котлете сохраняется больше витаминов!

Но перейдем от красивых аллегорий к реальной жизни. Речь пойдет о Роботландском университете, где школьники пишут трансляторы в среде Кукарачи и Корратора.

Те, кто знаком с этими чудаками (исполнителями), выразительно засмеялись. Ну что ж, чтобы веселья прибавилось, скажу пару слов о Кукараче. Для тех, кто не знает. Пусть посмеются тоже.

ИСПОЛНИТЕЛЬ КУКАРАЧА

Средой исполнителя является клетчатое поле, обычный размер которого — 10×10 . Но поле может быть продолжено вправо и вниз, если это диктуется условием задачи.



На рисунке символом ♦ изображен исполнитель. Кукарача может переползать из клетки в клетку вверх, вниз, влево, вправо, но не по диагонали. Выход исполнителя за пределы поля запрещен (ситуация “Не могу!”). Кроме самого исполнителя, в каждой клетке может находиться кубик с нанесенным на его грань символом. Кукарача, перемещаясь по полю, может толкать один или несколько кубиков перед собой и даже “сбрасывать” кубики за пределы поля.

Некоторые кубики на поле могут иметь “скрытые” надписи: считается, что такие кубики расположены в клетке “надписью вниз”. Толкая скрытый кубик, Кукарача перемещает его в следующую по ходу движения клетку и переворачивает.

Толкая кубик, исполнитель в состоянии прочитать символ на его грани независимо от того, была ли надпись обычной или скрытой (перевернутой).

Система команд Кукарачи (СКИ)

- ВПРАВО — сместиться на одну клетку вправо на поле;
- ВЛЕВО — сместиться на одну клетку влево на поле;
- ВВЕРХ — сместиться на одну клетку вверх на поле;
- ВНИЗ — сместиться на одну клетку вниз на поле;
- СТОЯТЬ — “пустая” команда, исполнитель не выполняет никаких действий.

Как видите, Кукарача — прямой наследник Муравья Звенигородского. И таких насекомых от информатики в России развелось немало. Предполагается, что программирование робота, визуальное представление на экране, гораздо нагляднее, чем возня с абстрактными вычислениями на Фортране, Бейсике или Паскале.

И это, конечно, правильно, хотя обсуждать этот вопрос мне неинтересно.

Но теперь самое время дружно посмеяться еще раз всем вместе: о какой трансляции можно говорить в этой примитивной клетчатой среде? Что здесь транслировать-то и во что?

Вы заинтригованы?

Ну, тогда, читатель, за мной! К вершинам тараканьей трансляции.

Для начала помянем добрым словом стариков Бэкуса и Наура.

ЯЗЫК БЭКУСА—НАУРА

Язык, при помощи которого описывают другие языки, называется метаязыком. Понятно, что русский язык является метаязыком. Ведь на этом языке мы описываем, например, язык исполнителя Кукарачи.

В программировании очень часто применяют метаязык Бэкуса—Наура для описания формальных конструкций. Впервые этот язык был использован для описания языка программирования Алгол-60.

В языке Бэкуса—Наура используются следующие символы (метасимволы):

- <> — скобки, в которые заключается определяемое понятие;
- := — составной символ, который читается “по определению есть”;
- | — символ, обозначающий альтернативу, читается “или”.

Примеры определений

Пример 1

<знак сложения> := +

Знак сложения по определению есть символ “+”.

Пример 2

<цифра> := 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

Цифра по определению есть 0, или 1, или 2, или 3, или 4, или 5, или 6, или 7, или 8, или 9.

Наиболее интересным является использование языка Бэкуса—Наура для создания рекурсивных определений.

Пример 3

<слово> := <буква> | <слово><буква> (1)

<буква> := a | m (2)

Прочитать это определение можно так:

“Слово” по определению есть: или “буква”, или объект, про который мы знаем, что он “слово”, вслед за которым записана “буква”.

“Буква” — это либо символ “a”, либо символ “m”.

Как видите, первая строка в определении содержит рекурсию!

Проверим, соответствуют ли определению “слово” следующие объекты: “a”, “ma”, “ava”.

1. Объект “a” — слово, потому что: “a” — буква (правило 2), а буква является словом (правило 1).
2. Объект “ma” — слово, потому что: “m” — это слово (так как “m” — это буква по правилу 2, а значит, слово по правилу 1). Слово, за которым следует буква, является словом (правило 1), значит, “ma” — слово.
3. Объект “ava” — не слово, потому что за словом “a” следует не буква в смысле правила 2.

Как проверить соответствие объекта определению?

Нужно следовать правилам, заданным в этом определении. Если правила рекурсивные, значит, и проверка будет содержать рекурсию. Проверим, например, что объект “мама” является словом.

Объект “мама” — слово, если словом является “мам” (правила 1 и 2).

Объект “мам” — слово, если словом является “ма” (правила 1 и 2).

Объект “ма” — слово, если словом является “m” (правила 1 и 2).

Объект “m” — слово (правило 2).

Заметим, что по определению 1 словом является любая (не пустая!) последовательность из букв “a” и “m”. Это легко доказать, используя метод математической индукции.

Напоминание. Пусть $F(n)$ — некоторое утверждение о целом числе n .

Доказательство методом математической индукции проводится в три шага.

1. База индукции.

Утверждение $F(n)$ проверяется при некотором начальном $n = n_0$ (например, при $n_0 = 1$).

2. Индуктивное предположение.

Выдвигается индуктивное предположение. Оно заключается в том, что утверждение $F(n-1)$ считается справедливым ($n > n_0$).

3. Индуктивный переход.

Доказывается справедливость $F(n)$ с учетом индуктивного предположения, то есть предположения о справедливости $F(n-1)$. Доказательство основывается на попытке представить зависимость $F(n)$ через $F(n-1)$.

Это доказательство наряду с проверкой F для одного конкретного значения n_0 позволяет утверждать, что $F(n)$ справедливо всегда. Почему?

Мы доказали, что утверждение $F(n)$ справедливо, если справедливо $F(n-1)$. Этого достаточно, чтобы утверждать, что $F(n)$ справедливо для любого $n > n_0$, если $F(n_0)$ проверено непосредственно.

...

$F(5)$ справедливо, если справедливо $F(4)$;

$F(4)$ справедливо, если справедливо $F(3)$;

$F(3)$ справедливо, если справедливо $F(2)$;

$F(2)$ справедливо, если справедливо $F(1)$;

$F(1)$ — база индукции, справедливость проверена непосредственно.

Мы как будто заставляем работать некоторого исполнителя для проверки справедливости утверждения F для каждого конкретного n по следующей рекурсивной программе:

ЭТО Индукция (n)

ЕСЛИ n = 1

ТО ОТВЕТ ("F — справедливо для любого n")

ИНАЧЕ Индукция (n-1)

КОНЕЦ

Например, процедура Индукция (3) рекурсивно выполнит процедуру Индукция (2). Ведь из доказательства индуктивного перехода следует, что $F(3)$ справедливо, если справедливо $F(2)$. Процедура Индукция (2) аналогично выполнит процедуру Индукция (1). В последней процедуре рекурсия заканчивается: ведь $F(1)$ справедливо — это проверенная база индукции.

Приведенная программа, конечно, лишена всякого практического смысла. Она просто демонстрирует математическую индукцию как рекурсивный процесс, который, пользуясь доказанным индуктивным переходом, сводит проверку $F(n)$ к проверке $F(1)$.

Вернемся к нашей задаче. Докажем, что любая последовательность $F(n)$, состоящая из n символов, каждый из которых принимает значение “а” либо “м”, является словом в смысле заданного определения.

- База индукции. $F(1)$ состоит из одного символа “а” или “м” и в силу правила (2) является буквой, а значит, в силу правила (1) — словом.
- Индуктивное предположение. Считаем, что $F(n-1)$ — слово.
- Индуктивный переход. Докажем, что $F(n)$ — тоже слово.

Представим $F(n)$ для $n > 1$ в виде двух частей: последний символ и все остальное. Тогда $F(n)$ можно записать как

$$F(n) = F(n-1)s \quad (*)$$

Здесь s — это последний символ в $F(n)$, он может принимать значение “а” или “м”, то есть является буквой в силу (2).

Итак, мы представили $F(n)$ как слово $F(n-1)$, за которым следует буква. Но тогда в силу (1) — $F(n)$ — тоже слово.

Ниже следует список задач с ответами. Эти задачи можно использовать на занятиях со школьниками для практической работы с определениями, заданными в нотации Бэкуса—Наура.

ЗАДАЧА 1

Определение 1

$\langle \text{имя} \rangle ::= \langle \text{буква} \rangle | \langle \text{имя} \rangle \langle \text{буква} \rangle | \langle \text{имя} \rangle \langle \text{цифра} \rangle$
 $\langle \text{буква} \rangle ::= x | y$
 $\langle \text{цифра} \rangle ::= 1 | 2$

Проверить, являются ли приведенные ниже объекты именами в смысле определения 1.

- | | |
|---------------|--------------|
| 1) x ; | 6) $y321$; |
| 2) $xу$; | 7) 1212 ; |
| 3) $x1$; | 8) $x21$; |
| 4) $2у$; | 9) $x21$; |
| 5) $xxxxxx$; | 10) $x1y1$. |

Ответ: именами не являются объекты 4, 6, 7, 9.

Докажите следующее утверждение.

Именем является любая последовательность из строчных букв и цифр, которая начинается с буквы.

ЗАДАЧА 2

Определение 2

$\langle \text{имя} \rangle ::= \langle \text{буква} \rangle | \langle \text{имя} \rangle \langle \text{буква} \rangle | \langle \text{имя} \rangle \langle \text{цифра} \rangle$
 $\langle \text{буква} \rangle ::= \langle \text{пусто} \rangle | x | y$
 $\langle \text{цифра} \rangle ::= 1 | 2$
 $\langle \text{пусто} \rangle ::=$

Проверить, являются ли приведенные ниже объекты именами в смысле определения 2.

- | | |
|-----------|----------------|
| 1) x ; | 5) $xxxxxxx$; |
| 2) $xу$; | 6) $z321$; |
| 3) $x1$; | 7) 1212 ; |
| 4) $2у$; | 8) $x1y1$. |

Ответ: именем не является объект 6.

Верны ли следующие утверждения?

- Любая последовательность из символов “ x ” и “ 1 ” является именем в смысле определения 2.
- Совпадает ли множество таких “имен” со множеством имен, задаваемых определением 2?

ЗАДАЧА 3

Определение 3

$\langle \text{выражение} \rangle ::= \langle \text{число} \rangle | (\langle \text{выражение} \rangle | \langle \text{число} \rangle + \langle \text{выражение} \rangle)$
 $\langle \text{число} \rangle ::= 20 | 30$

Проверить, являются ли приведенные ниже объекты выражениями в смысле определения 3.

- | | |
|----------------|---------------------------------|
| 1) 2 ; | 6) $(20 + 30)$; |
| 2) 20 ; | 7) $20 + (30 + 30)$; |
| 3) 2030 ; | 8) $20 +$; |
| 4) $20 + 30$; | 9) $+$; |
| 5) (20) ; | 10) $(20 + (30 + (20 + 20)))$. |

Ответ: выражениями не являются объекты 1, 3, 8, 9.

ЗАДАЧА 4

Запишите следующее определение при помощи языка Бэкуса—Наура.

Пример — это несколько цифр, соединенных между собой знаком “—”.

Ответ:

$\langle \text{пример} \rangle ::= \langle \text{цифра} \rangle - \langle \text{цифра} \rangle | \langle \text{пример} \rangle - \langle \text{цифра} \rangle$
 $\langle \text{цифра} \rangle ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9$

ЗАДАЧА 5

Запишите при помощи языка Бэкуса—Наура определение записи натурального числа в десятичной системе счисления.

Ответ:

$\langle \text{натуральное число} \rangle ::= \langle \text{ненулевая цифра} \rangle | \langle \text{натуральное число} \rangle \langle \text{цифра} \rangle$
 $\langle \text{ненулевая цифра} \rangle ::= 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9$
 $\langle \text{цифра} \rangle ::= 0 | \langle \text{ненулевая цифра} \rangle$

ЗАДАЧА 6

Запишите при помощи языка Бэкуса—Наура определение записи целого числа.

Ответ:

$\langle \text{целое} \rangle ::= \langle \text{целое без знака} \rangle | + \langle \text{целое без знака} \rangle | - \langle \text{целое без знака} \rangle$
 $\langle \text{целое без знака} \rangle ::= \langle \text{цифра} \rangle | \langle \text{целое без знака} \rangle \langle \text{цифра} \rangle$
 $\langle \text{цифра} \rangle ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9$

ЗАДАЧА 7

Определение 4

$\langle \text{выражение} \rangle ::= \langle \text{терм} \rangle | \langle \text{выражение} \rangle + \langle \text{терм} \rangle$
 $\langle \text{терм} \rangle ::= \langle \text{первичный} \rangle | \langle \text{терм} \rangle * \langle \text{первичный} \rangle$
 $\langle \text{первичный} \rangle ::= x | y | z$

Проверить, являются ли приведенные ниже объекты выражениями в смысле определения 4.

- 1) z ;
- 2) $x * y + z$;
- 3) $x + y + z$;
- 4) $(x + y) * z$;
- 5) $x * x$;
- 6) $y + y * y + y$;
- 7) xu ;
- 8) $+ x$;
- 9) $z +$.

Ответ: выражениями не являются объекты 4, 7, 8, 9.

ЛЕКСИЧЕСКИЙ АНАЛИЗ ВЫРАЖЕНИЙ

Открою обман: я не умею писать трансляторы в среде Кукарачи! На самом деле это обман только наполовину:

- я умею писать в среде Кукарачи лексические анализаторы, а это все же составная часть транслятора;
- я на самом деле умею писать трансляторы в среде исполнителя Корректор!

Как известно, исполнитель не понимает языка программирования. Когда мы записываем в программе ПОВТОРИ 5 ВПРАВО, то знаем, что эта запись исполнителю недоступна. Ведь такой команды нет в его СКИ. Тем не менее, когда программа работает, Кукарача перемещается на пять клеток вправо.

Дело в том, что запись ПОВТОРИ 5 ВПРАВО — это команда языка программирования, а язык программирования “понимает” не исполнитель, а компьютер, и то не сам, а с помощью транслятора.

Транслятор — это специальная программа, которая переводит текст с языка программирования в последовательность команд из СКИ робота. Эту последовательность команд компьютер передает исполнителю. И Кукарача делает пять своих прыжков.

Одна из важных задач трансляции — сборка из последовательных символов отдельных элементов текста программы. Эти элементы называются лексемами.

Лексический анализ выражений — это выделение в тексте лексем, кирпичиков, из которых строятся конструкции языка. До лексического анализа текст рассматривается как однородная цепочка символов, после — как цепочка лексем.

Например, текст

ПОВТОРИ 5 ВПРАВО

содержит 16 символов (среди них два символа пробела). После лексического анализа текст преобразуется в цепочку из трех лексем:

- ПОВТОРИ — ключевое слово, начало цикла;
- 5 — число;
- ВПРАВО — ключевое слово, команда из СКИ.

ДИАГРАММА ПЕРЕХОДОВ

Метод решения задач лексического анализа никак не связан именно с Кукарачей и Корректором, а имеет общепрограммистское значение. Он основан на составлении схем (диаграмм) переходов с последующим простым (и буквальным) программированием составленной схемы на языке реализации.

Содержательная часть работы — именно составление схем. Когда схема нарисована, остается просто переписать ее на языке программирования. Само программирование (кодирование) при этом выполняется почти механически.

Суть метода рассмотрим на следующем примере.

Определение 5

$\langle \text{число} \rangle ::= \langle \text{целое} \rangle | \langle \text{дробь} \rangle$
 $\langle \text{целое} \rangle ::= \langle \text{цифра} \rangle | \langle \text{целое} \rangle \langle \text{цифра} \rangle$
 $\langle \text{дробь} \rangle ::= \langle \text{целое} \rangle / \langle \text{целое} \rangle$
 $\langle \text{цифра} \rangle ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9$

Диаграмма переходов для анализа записи по определению 5 изображена на рис. 1.

Схема показывает состояния алгоритма (они обозначены словами в круглых скобках) и переходы алгоритма из одного состояния в другое (они изображены стрелками) в зависимости от значения очередного символа входной записи (символы обозначены на стрелках-переходах).

Рассмотрим схему в динамике работы алгоритма.

В начальный момент алгоритм находится в состоянии (вход).

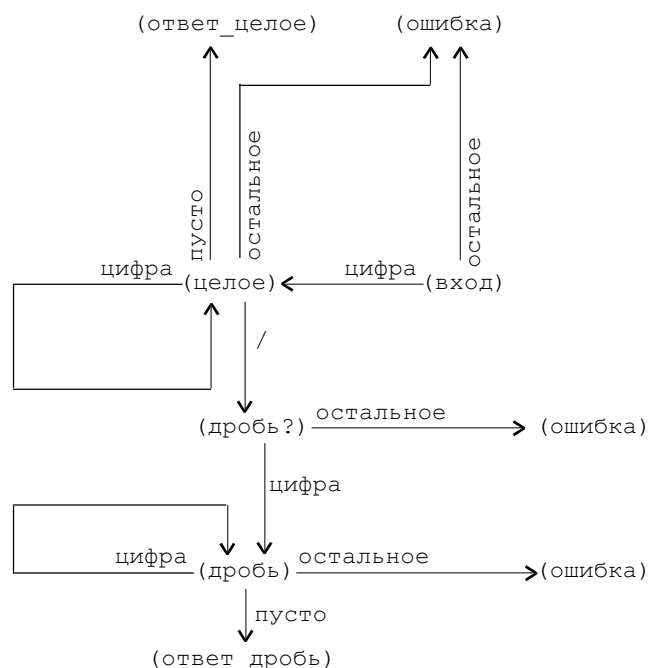


Рис. 1

Любой поступивший символ в этом состоянии, кроме символа “цифра”, приводит к переходу в состояние (ошибка).

Если все же в состоянии (вход) поступает символ “цифра”, алгоритм переходит в состояние (целое).

Какой символ допустим в этом состоянии? Из определения следует, что таких допустимых символов три:

“цифра” — алгоритм остается в состоянии (целое);

“/” — алгоритм переходит в состояние (дробь?);

“ПУСТО” — алгоритм переходит в состояние (ответ_целое).

Любой другой символ приводит к переходу в состояние (ошибка). Состояние (дробь?) вводится дополнительно к состоянию (дробь) для того, чтобы расценить запись, завершающуюся косой чертой, как ошибочную.

Состояние (дробь) обрабатывает очередной символ следующим образом:

“цифра” — алгоритм остается в состоянии (дробь);

“ПУСТО” — алгоритм переходит в состояние (ответ_дробь);

любой другой символ — алгоритм переходит в состояние (ошибка).

Когда диаграмма построена, остается простая и фактически рутинная работа: нужно закодировать диаграмму на языке программирования. Рекомендуемая методика предполагает, что каждое состояние схемы отображается одноименной процедурой, а переходы между состояниями — вызовами соответствующих процедур-состояний. Такой способ записи буквально повторяет схему и практически не требует дополнительных комментариев.

Программа для Кукарачи

Пусть запись расположена во второй строке, а исполнитель — перед ней:

	1	2	3	4	5	6	7	8	9	10
1										
2		•	з	а	п	и	с	ь		
3										

Задание

Проверить запись на соответствие определению (*). Если запись — “целое”, установить исполнителя за записью в первой строке:

	1	2	3	4	5	6	7	8	9	10
1			з	а	п	и	с	ь	•	
2										
3										

Если запись — “дробь”, в конце записи во второй строке:

	1	2	3	4	5	6	7	8	9	10
1			з	а	п	и	с	ь		
2									•	
3										

Если в записи есть ошибки, установить исполнителя под первой из них:

	1	2	3	4	5	6	7	8	9	10
1			з	а	п	и	с	ь		
2					•					
3										

ЭТО вход
шаг
ЕСЛИ ЦИФРА ТО целое i
ИНАЧЕ ошибка
КОНЕЦ

Состояние (вход)
Смотрим следующий символ

ЭТО целое
шаг
ЕСЛИ ПУСТО
ТО ответ_целое
ИНАЧЕ ЕСЛИ ЦИФРА
ТО целое
ИНАЧЕ ЕСЛИ /
ТО дробь?
ИНАЧЕ ошибка
КОНЕЦ

Состояние (целое)
Смотрим следующий символ

ЭТО дробь?
шаг
ЕСЛИ ЦИФРА ТО дробь
ИНАЧЕ ошибка
КОНЕЦ

Состояние (дробь?)
Смотрим следующий символ

ЭТО дробь
шаг
ЕСЛИ ПУСТО
ТО ответ_дробь i
ИНАЧЕ ЕСЛИ ЦИФРА
ТО дробь
ИНАЧЕ ошибка
КОНЕЦ

Состояние (дробь)
Смотрим следующий символ

ЭТО ошибка
шаг
ЕСЛИ НЕ ПУСТО
ТО ошибка
ВЛЕВО
КОНЕЦ

Перемещение
остатка записи
в первую строку
Возврат к месту
ошибки

ЭТО ответ_целое
ВВЕРХ
КОНЕЦ

ЭТО ответ_дробь
СТОЯТЬ
КОНЕЦ

ЭТО шаг
ВНИЗ ВПРАВО ВВЕРХ
КОНЕЦ

"Потрогать"
следующий символ

Не правда ли, программа получилась пусть и не очень короткая, но простая, предельно понятная и даже скучная? Она полностью соответствует построенной диаграмме.

Итак, совершенно очевидно, что в этой задаче (и всех других аналогичных) самое трудное — построить правильную диаграмму. Вопрос кодирования диаграммы на языке программирования является второстепенным и несущественным. Программирование здесь не содержит никаких идей и могло бы выполняться автоматически при наличии специального транслятора, который получал бы на входе построенную диаграмму, а на выходе получал бы готовую программу. Эту программу даже не нужно было бы тестировать.

Конечно, правильность программы означает всего лишь, что она полностью соответствует диаграмме, является другой формой ее записи.

Но если диаграмма построена неверно, то неверной будет и эквивалентная ей программа.

Итак, успех построения лексического анализатора выражений по заданному формальному определению, подобному определению 5, полностью зависит от успеха построения правильной диаграммы переходов.

Отступление в "большую" информатику

Задачи на лексический анализ выражений занимают в "большой" информатике важное место. Описанный способ решения применяется достаточно часто. При этом диаграмма переходов записывается обычно в виде двумерного массива (таблицы).

Для нашего определения 5 этот массив выглядит так:

Предыдущее состояние	Класс нового символа			
	Цифра 0	/ 1	ПУСТО 2	Все остальное 3
(вход) 0	1	4	4	4
(целое) 1	1	2	0	4
(дробь?) 2	3	4	4	4
(дробь) 3	3	4	0	4
(ошибка) 4	0	0	0	0

Этот массив, например, на Си можно записать так:

```
int diagramma [5][4]=
{
  1, 4, 4, 4,
  1, 2, 0, 4,
  3, 4, 4, 4,
  3, 4, 0, 4,
  0, 0, 0, 0,
};
```

Элемент `diagramma[i][j]=m` означает, что на диаграмме стрелка, выходящая из состояния i при обнаружении символа с номером j , приводит в состояние m или приводит к завершению работы, если $m = 0$. В последнем случае программа должна выдать в качестве результата число i (или соответствующий ему текст — название состояния).

Когда массив переходов задан, анализ выражения выполняется при помощи следующего кода:

```
int i = 0;
int j = diagramma [i] [TypeNextSymbol()];
do
{
  /*АНАЛИЗАТОР*/
  i = j;
  j = diagramma [i] [TypeNextSymbol()];
} while (j);
```

Здесь функция `TypeNextSymbol()` читает следующий символ входной записи и возвращает класс этого символа, то есть в соответствии с таблицей переходов выдает:

- 0 — если символ "цифра";
- 1 — если символ "/";
- 2 — если символ "ПУСТО";
- 3 — в остальных случаях.

Как видите, анализатор совершенно не зависит от вида определения 5. При смене определения на другое (какой угодно сложности) лексический анализатор останется прежним и будет содержать те же 7 строк программного кода. Будет меняться массив `diagramma`, а лексический анализатор останется неизменным.

В "большой" информатике проблема та же, что и в малой: как правильно составить диаграмму переходов по заданному определению?

Надо уметь понимать и "чувствовать" рекурсивность определения. А для этого нужен хороший практикум. Такие вещи не приходят по наитию, нужно достаточно долго с этим работать. Не только смотреть, как это делают другие, а на самом деле практиковаться самому, решать задачи.

* * *

Вы не заметили? А ведь в приведенной программе для Кукарачи есть одно тонкое место — процедура ошибка:

ЭТО ошибка
шаг
ЕСЛИ НЕ ПУСТО
ТО ошибка
ВЛЕВО
КОНЕЦ

Перемещение остатка
записи в первую строку
Возврат к месту ошибки

Кажется, что автор статьи сам ошибся: как может эта простая процедура установить исполнителя точно под обнаруженной ошибкой? НЕУЖЕЛИ ОНА ЭТО ДЕЛАЕТ?

Кто понял, в чем дело, — молчите! Пусть остальные немного помучаются! В следующий раз я обязательно займусь "разоблачением" этого рекурсивного фокуса.

А теперь несколько задач из конкурса Роботландского сетевого университета прошлого года.

ЗАДАЧА 1 (5 БАЛЛОВ)

Введем такое определение:

Определение 6

$\langle \text{число} \rangle ::= \langle \text{цифра} \rangle | \langle \text{число} \rangle \langle \text{цифра} \rangle$ (1)

$\langle \text{цифра} \rangle ::= 0 | 1$ (2)

Задание. Напишите программу для Кукарачи, которая проверяет, является ли запись на его поле числом в смысле определения 6.

В начальный момент исполнитель расположен перед кубиками с записью во второй строке:

◆ запись

Если проверяемая запись — число, установить исполнителя в конец записи:

запись◆

Если запись — не число, поставить исполнителя в десятую строку поля.

ЗАДАЧА 2 (7 БАЛЛОВ)

Решить задачу 1, но если запись не является числом, установить исполнитель под первым неверным символом в записи. Например:

1021

◆

Если запись — число, установить исполнитель за последним ее символом. Например:

1011◆

ЗАДАЧА 3 (8 БАЛЛОВ)

Введем следующее определение:

Определение 7

$\langle \text{число} \rangle ::= \langle \text{целое} \rangle | \langle \text{дробное} \rangle$

$\langle \text{целое} \rangle ::= \langle \text{цифра} \rangle | \langle \text{целое} \rangle \langle \text{цифра} \rangle$

$\langle \text{дробное} \rangle ::= \langle \text{целое} \rangle \langle \text{целое} \rangle \langle \text{целое} \rangle$

$\langle \text{цифра} \rangle ::= 0 | 1$

Задание. Напишите программу для Кукарачи, которая проверяет, является ли запись на его поле числом в смысле определения 7.

В начальный момент исполнитель расположен перед кубиками с записью во второй строке:

◆ запись

Если проверяемая запись — число, установить исполнитель в конец записи:

запись◆

Если запись — не число, поставить исполнителя под первым неверным символом:

запись

◆

Примеры результатов выполнения программы:

1) 1010◆

3) .00◆

2) 121.34

4) .10.0

◆

◆

ЗАДАЧА 4 (9 БАЛЛОВ)

Введем следующее определение:

Определение 8

$\langle \text{число} \rangle ::= \langle \text{целое} \rangle | \langle \text{дробь} \rangle$

$\langle \text{целое} \rangle ::= \langle \text{цифра} \rangle | \langle \text{целое} \rangle \langle \text{цифра} \rangle$

$\langle \text{дробь} \rangle ::= \langle \text{простая} \rangle | \langle \text{десятичная} \rangle$

$\langle \text{простая} \rangle ::= \langle \text{целое} \rangle / \langle \text{целое} \rangle$

$\langle \text{десятичная} \rangle ::= \langle \text{целое} \rangle \langle \text{целое} \rangle \langle \text{целое} \rangle$

$\langle \text{целое} \rangle \langle \text{целое} \rangle$

$\langle \text{цифра} \rangle ::= 0 | 1$

Задание. Напишите программу для Кукарачи, которая проверяет, является ли запись на его поле числом в смысле определения 8.

В начальный момент исполнитель расположен перед кубиками с записью во второй строке:

◆ запись

Если проверяемая запись — число, установить исполнитель в конец записи:

запись◆

Если запись — не число, поставить исполнитель под первым неверным символом:

запись

◆

ЗАДАЧА 5 (9 БАЛЛОВ)

Введем следующее определение выражения:

Определение 9

$\langle \text{выражение} \rangle ::= \langle \text{число} \rangle \langle \text{число} \rangle |$

$\langle \text{число} \rangle \langle \text{выражение} \rangle$

$\langle \text{число} \rangle ::= \langle \text{цифра} \rangle | \langle \text{число} \rangle \langle \text{цифра} \rangle$

$\langle \text{цифра} \rangle ::= 0 | 1$

Задание. Напишите программу для Кукарачи, которая проверяет, является ли запись на его поле выражением в смысле определения 9.

В начальный момент исполнитель расположен перед кубиками с записью во второй строке:

◆ запись

Если проверяемая запись — выражение, установить исполнитель в конец записи:

запись◆

Если запись — не выражение, поставить исполнитель под первым неверным символом:

запись

◆

Продолжение следует

Что такое алгоритм

А.Л. Брудно

От редакции

Эта заметка известного российского специалиста, доктора физико-математических наук, профессора, организатора московских олимпиад по программированию, автора многочисленных книг и учебных пособий Александра Львовича Брудно поступила в редакцию больше года назад. Александр Львович неоднократно напоминал о ней, интересовался, когда она будет опубликована, а мы все медлили: нам казалось, что текст заметки слишком конспективен, слишком много требуется читателю “додумать” самостоятельно, чтобы в полной мере осознать, что хочет сказать автор. И далеко не всегда мнение А.Л. Брудно совпадает с мнением авторов школьных учебников информатики, ряд различий (вообще говоря, существенных) виден невооруженным глазом. Но в конце концов... мы решили, что необходимо опубликовать статью одного из основателей российской информатики. А что касается наших сомнений... Мы уверены, что в любом случае эта заметка будет интересна и полезна для наших читателей.

Алгоритм является фундаментальным понятием математики. Представление о нем необходимо для эффективного применения вычислительной техники к решению практических задач. Мы сначала дадим определение алгоритма, потом рассмотрим ситуации, приведшие к такому определению, и в конце познакомимся с историей возникновения и установления понятия алгоритма.

1. Начнем с неформального определения. *Алгоритм* — это точное и безотказное предписание действий, которые должны быть выполнены.

Нужно понять смысл слов, входящих в это определение, и то, что никакими иными свойствами алгоритм может и не обладать.

Как понимать то, что предписание должно быть точным? Оно должно быть настолько точным, чтобы его мог выполнить компьютер с неограниченной памятью. Здесь иногда вместо “выполнить” надо сказать “промоделировать”. Возможно здесь и еще одно расширение понятия “точный”, к которому мы вернемся в п. 5.

Как понимать то, что предписание должно быть безотказным? Это значит, что в любых условиях, на которые объявлен алгоритм, он должен выдавать свои предписания. Особую трудность обычно представляют те условия, которые могут возникнуть в результате выполнения самого алгоритма.

2. Иногда говорят не *алгоритм*, а *алгоритм решения некоторой задачи (или группы задач)*. Понятия эти близкие, но разные, ибо алгоритм может и не решать никаких задач. Они могут в нем вовсе не упоминаться.

Алгоритм решения некоторой задачи — это алгоритм, приводящий к решению этой задачи за конечное число действий.

Алгоритм решения группы задач — это алгоритм, приводящий к решению каждой задачи (из этой группы) за конечное число действий.

3. Алгоритм может быть хорошим или плохим (по чьей-то оценке), он может быть ясным или запутанным, коротким или длинным. Он может быть пригоден для решения одной или многих задач, может приводить к решению быстро или медленно. Может быть пригоден для выполнения на простом автомате или требовать компьютера с большой памятью. Можно назвать и иные качества алгоритма, но нужно понимать, что никакими свойствами, кроме указанных в п. 1, алгоритм может и не обладать.
4. Необходимо также понимать, что мы можем уметь решать задачу и не знать алгоритм ее решения, т.е. не знать (достаточно точно), как мы ее решаем. С такими случаями мы познакомимся на примерах.
5. Облегчим применение понятия *алгоритм*. По определению, приведенному в п. 1, получается, что алгоритм — это программа для компьютера. Строго говоря, так оно и есть. Но проверять это всякий раз было бы долго. Поэтому мы будем считать алгоритмом и такие указания, про которые нам будет ясно, как сделать их понятными компьютеру, т.е. как довести их до программы. Это, конечно, не совсем точно, так что в сомнительных случаях следует возвращаться к определению из п. 1.

6. Перейдем к примерам алгоритмов.

6.1. Все мы знаем алгоритм сложения десятичных чисел. Надо заранее выучить таблицу сложения пар однозначных чисел, а затем складывать цифры слагаемых поразрядно, справа налево, учитывая единицы переноса. Вы можете сами аккуратно выписать этот алгоритм. Легко его и запрограммировать. Стоит заметить, что алгоритм решает за конечное число действий каждую задачу, а не все задачи, на которые объявлен.

6.2. Напомним задачу о волке, козе и капусте. По условию волк, коза, капуста, перевозчик и лодка находятся на одном берегу и их надо перевезти на другой. В лодке, кроме перевозчика, может находиться только одно животное или только капуста. Оставлять (без присмотра перевозчика) волка с козой или козу с капустой нельзя. Решите эту задачу. Заметьте, что для самого решения вам не понадобилось ни лодки, ни животных. Вы “промоделировали” решение. И эту модель можно запрограммировать.

6.3. Алгоритм Евклида. Для двух заданных натуральных чисел m , n требуется найти их наибольший общий делитель $d(m, n)$. Заметим предварительно, что если $m = n$, то $d(m, n) = m$. Если же $m \neq n$ и, для определенности, $m < n$, то всякий общий делитель пары чисел m , n будет и общим делителем пары m , $n - m$. И наоборот. Следовательно, если $m < n$, то $d(m, n) = d(m, n - m)$.

Это подсказывает следующий алгоритм отыскания $d(m, n)$:

1) если $m = n$, то ответ равен m и вычисление закончено;

2) иначе надо заменить большее число разностью между ними и меньшим и вернуться к п. 1.

Ясно, что этот алгоритм приводит к решению за конечное число шагов (докажите!). Алгоритм реализуется программой:

```
10 input m, n
20 goto 40
30 if m < n then n = n - m
   else m = m - n
40 if m <> n goto 30
50 print m
60 end
```

Решение можно ускорить. Действительно, если $m < n$, то число m будет вычитаться из n до тех пор, пока разность станет меньше или равна m . Но эту (последнюю)

разность можно получить сразу, как остаток ($n \bmod m$) от деления n на m . Нужно только учесть, что остаток может обратиться в 0. Программе можно придать вид

```
10 input m, n
20 if m < n goto 40
30 swap m, n
40 n = n mod m
50 if n > 0 goto 30
60 print m
70 end
```

Разберите ее работу. Поупражняйтесь с нею на компьютере.

7. Мы успешно программируем задачи, в которых у человека нет интуиции. Но когда мы переходим к решению задач, где приходится конкурировать с механизмом, созданным самой природой, наши программы значительно уступают решениям людей и животных. Выясняется, что мы не знаем, как мы думаем, и не умеем это запрограммировать — не знаем алгоритмов природы.

7.1. Завязать шнурки на ботинках бантиком трудно. Многие дети в 6 лет умеют это делать. Но написание алгоритма может затруднить и взрослого.

7.2. Разбор кучи. В начале автоматизированной линии обработки поковок коленчатых валов стоит человек. Он вынимает заготовки из короба и вставляет в конвейер. Автоматизировать эту работу не удастся. Надо узнавать деталь в разных ракурсах, частично заложенную другими деталями. Решать, с какой начать и по какой траектории ее вынимать.

7.3. Чтение. Человек легко читает книгу. Может читать и разборчивую рукопись. Рукописей машины пока вообще не читают, а при распознавании сложного машинописного текста, содержащего различные шрифты, формулы и т.п., часто допускают ошибки. Человек делает это как-то проще!

7.4. Если в предыдущих примерах можно что-либо свалить на несовершенство внешних устройств, то при программировании шахмат недостатки нашего понимания того, “как мы думаем”, проявляются в чистом виде. Рассмотрим для примера одну позицию: б.: Крб5, Фб6; ч.: Крс3. Ее без труда выиграет едва научившийся играть. Но даже мастер не сможет быстро написать выигрывающий алгоритм. Да и написанный им алгоритм будет играть хуже, чем едва научившийся человек.

8. О неаккуратных и неудачных определениях и пониманиях алгоритмов.

8.1. Начнем с известного примера “бытового алгоритма” перехода улицы.

“Посмотри налево. Если машин нет — пройди до середины улицы. Если есть — подожди, пока они проедут”. И т.д. Представьте себе, что машина слева есть, но она не едет — у нее меняют колесо. Если вы думаете, что надо ждать, то вы поняли этот алгоритм. Если же вы решили, что улицу переходить можно, считая алгоритм подправленным ввиду непредвиденных (по вашему мнению!) обстоятельств, то вы не усвоили понятия алгоритма. Еще раз подчеркнем: алгоритм всегда должен быть рассчитан на выполнение (не размышляющей) машины.

8.2. Некоторые думают, что алгоритм обязательно должен быть “массовым”, то есть применимым к группе задач, и вводят это требование в само определение алгоритма. Этого не следует делать. Такого требования нет в п. 1, нет его в понятии алгоритма в математической логике (см. п. 9). Если его ввести, то мы не сможем говорить об алгоритме решения одной определенной задачи (см. п. 6.2).

8.3. Иногда спрашивают: чем же отличается алгоритм решения задачи от решения этой задачи? Разница в том, что решение содержит ответ, а алгоритм может давать всего лишь способ для получения ответа. Рассмотренный в п. 6.3 алгоритм Евклида дает лишь способ найти наибольший общий делитель чисел 429 и 455, а решение этой задачи должно содержать ответ: 13.

8.4. Некоторые говорят, что алгоритм “должен быть понятен исполнителю, на которого рассчитан”, и включают это в определение понятия алгоритма (ср. с п. 1). Такое определение алгоритма делает его недостаточно ясным и не универсальным.

8.5. Рассмотрим для примера такой *алгоритм* игры в шахматы: поручить провести игру квалифицированному шахматисту. Нас не устраивает ни такая “алгоритмизация”, ни такой “исполнитель”. Для нас это не алгоритмическое решение задачи.

9. Из истории понятия “алгоритм”.

9.1. Алгоритмы появились с самого зарождения математики. Появились они в качестве правил для вычисления разного рода величин. Например:

формула для вычисления суммы арифметической прогрессии:

$$s = (a + b) n / 2,$$

где a и b — это первый и последний члены; формулы для вычисления площади треугольника:

$$s = ab/2,$$

или

$$s = \sqrt{p(p-a)(p-b)(p-c)},$$

где $p = (a + b + c) / 2$;

алгоритм Евклида, с которым мы познакомились в п. 6.3, и другие.

9.2. Слово *алгоритм* возникло в средние века, когда европейцы познакомились со способами выполнения арифметических действий (сложения, вычитания, умножения столбиком и деления уголком), описанными математиком из Хорезма. Таким образом, слово *алгоритм* оказывается европеизированным произношением слов *аль Хорезм*, и первоначально под алгоритмом понимали способы выполнения арифметических действий, описанные в этом руководстве математика из Хорезма.

9.3. Научное (строгое) определение алгоритма дал А.Черч в 30-е годы. Он придумал логическую систему *рекурсивных функций*, назвал алгоритмами построения в этой системе, доказал важные теоремы и высказал “постулат Черча”: “Все, что сможет доказать математик, можно построить в системе рекурсивных функций”. После Черча и другие математики (А.Тьюринг, Э.Пост, А.Н. Колмогоров, А.А. Марков и др.) давали определения алгоритма. Все эти определения эквивалентны определению Черча.

Заключение. От редакции

В конце статьи содержалось очень краткое описание “машины Тьюринга”. Мы сочли возможным его опустить, учитывая, что имеются более подробные публикации на эту тему (см., например, № 41, 42/97). Кроме того, мы рекомендуем познакомиться со статьями, посвященными обсуждению понятия алгоритма, которые были опубликованы в нашей газете: № 35/97 — А.Г. Гейн. “Алгоритм”; № 33/98 — В.А. Носов. “О понятии алгоритма”.

Конференция учителей информатики в Челябинском государственном педагогическом университете

24 сентября в ЧГПУ состоялась конференция учителей информатики. Благодаря помощи регионального представителя нашей газеты по Челябинской области учителя информатики школы № 121 г. Челябинска А.Л. Королева на ней присутствовал главный редактор "Информатики" С.Л. Островский. Конференция оставила очень приятное впечатление. На ней не только обсуждались вопросы преподавания информатики в соответствии с новой редакцией обязательного минимума содержания образования по информатике (см. № 38/99), но и рассказывалось о практическом опыте использования программных продуктов, созданных учителями информатики школ г. Челябинска совместно с учителями других предметов (истории, физики и др.).

Мы надеемся, что с развитием и расширением сети региональных представителей встречи редакции с учителями станут более частыми и регулярными.



Конференция учителей информатики г. Челябинска.

На фотографии: М.И. Солодкова — зам. начальника управления по делам образования г. Челябинска, А.Л. Королев — учитель информатики школы № 121, региональный представитель газеты "Информатика".

Региональные представители "Информатики"

Летом этого года было объявлено об организации института региональных представителей нашей газеты. Мы получили и продолжаем получать большое количество заявок. Мы очень благодарны всем, кто прислал письма, но, к огромному сожалению, число наших представителей ограничено. Однако все заявки бережно хранятся, и, если в будущем количество региональных представителей будет увеличено, те, кто прислал письма сейчас, обязательно получат персональные приглашения. На данный момент список региональных представителей "Информатики" выглядит следующим образом (здесь указаны только те, от кого мы уже получили необходимую информацию). Данный список будет публиковаться в нашей газете ежемесячно.

Фамилия, имя, отчество	Город	Адрес	E-mail	Телефон
Алексеев Александр Владимирович	Красноярск	66000, Красноярск, Главпочтамт, а/я 5264	—	(3912) 27-96-70, факс. 27-66-86
Богданова Ольга Алексеевна	Воронеж	394051, Воронеж, ул. Ю.Янониса, 6, с/ш № 73	—	(0732) 33-45-75, каб. информатики
Ветохина Надежда Викторовна	Хабаровск	680051, г. Хабаровск, ул. Ворошилова, 5а-40	—	—
Королев Александр Леонидович	Челябинск	454080, Челябинск, ул. Володарского, 7-30	—	(3512) 61-97-93
Матвеева Людмила Краснославна	Калуга	248023, г. Калуга, а/я 1306	school16@kaluga.ru	(08422) 4-50-58
Окулов Станислав Михайлович	Киров	610017, г. Киров, ул. Ленина, 111, ВятГПУ, кафедра информатики	—	—
Смолянинов Алексей Александрович	Кострома	157930, Кострома, Караваево, ул. Штеймана, средняя школа	lpkar@kosnet.ru	(0942) 66-19-44
Филиппов Михаил Алексеевич	Владимир	600028, Владимир, пр. Строителей, 44в, с/ш № 17	—	(09222) 33-78-98, 33-87-66
Чшиева Татьяна Лазарбековна	Владикавказ	362035, Владикавказ, а/я 512	ipc@pno.ssc.ac.ru	(8672) 33-70-43, 54-39-12
Шестаков Александр Петрович	Пермь	614060, ГСП, Пермь, ул. Сибирская, 24, кафедра информатики и ВТ	shestako@pspu.ac.ru	(3422) 835-37-93
Ширинкин Ефим Дмитриевич	Нижний Тагил	622052, Свердловская обл., г. Нижний Тагил, ул. Парковая, 13, с/ш № 55	—	(3435) 23-15-31
Ямкина Елена Владимировна	Ульяновск	432008, Ульяновск, ул. Автозаводская, 4-21	—	—

Еще раз большое спасибо всем, кто прислал заявки!

Внимание!



Издательский дом «Питер», специализирующийся на выпуске компьютерной литературы, представляет



Важно!

Курс информатики с 6-го по 11-й класс:

- ✓ опирается на объектно-информационный подход;
- ✓ подробно освещает современные компьютерно-информационные технологии;
- ✓ является наиболее полным из существующих на сегодняшний день учебных пособий по информатике для средних и старших классов;
- ✓ получил гриф «Рекомендовано Комитетом по образованию С.-Петербурга».

Чем хороши

эти учебники!

- ✓ Сложные вопросы изложены доступным детям языком, живо и образно.
- ✓ Книги отлично иллюстрированы.
- ✓ Содержание учебников соответствует проекту образовательного стандарта по информатике, созданного под руководством А.А. Кузнецова и признанного победителем Всероссийского конкурса Министерства образования Российской Федерации в 1997 г.

Комплект создан по инициативе Центра информационных систем обучения Университета педагогического мастерства Санкт-Петербурга. Методика прошла испытания в ряде школ города на специально созданных экспериментальных площадках и опирается на опыт педагогов-практиков.

комплект учебников по информатике

под редакцией проф. Н.В. Макаровой (Санкт-Петербург)

ПО ВОПРОСАМ ЗАКУПОК ОБРАЩАЙТЕСЬ ПО АДРЕСАМ:

Москва, 1-й Шипковский пер., 3, оф. 207, тел. (095) 235 55 83, факс 234 38 15,
С.-Петербург, ул. Благодатная, 67, тел.: (812) 327 93 37, 294 54 65,
e-mail: sales@piter-press.ru

Вы можете заказать книги наложенным платежом через службу «КНИГА—ПОЧТОЙ». В этом случае книги обойдутся вам дешевле, а почтовые расходы будут оплачиваться при получении. Помните, что почтовые расходы на каждую книгу **уменьшаются** при заказе нескольких экземпляров. Кроме того, при заказе 10 книг цена уменьшается на 5%, 20 книг — на 7%, 30 книг и более — на 10%. Отправьте почтовую карточку с заказом по адресу: Россия, 197198, Санкт-Петербург, а/я 619-ИО; Украина, 310093, Харьков, а/я 9130-ИО; Беларусь, 220012, Минск, а/я 104-ИО. Укажите названия, коды и количество заказываемых книг, ваш индекс и адрес и, если вы ранее уже пользовались услугами службы «Книга—почтой», ваш регистрационный номер.

Фамилия И. О. _____		Ж		_____	
Адрес: _____		Тел. _____		_____	
цена	название книги	код	кол-во		
56 руб.	Информатика. 6-7 класс	1170	1170		
56 руб.	Информатика. 7-8 класс	1169	1169		
56 руб.	Информатика. 9 класс	1168	1168		
56 руб.	Информатика. 10-11 класс	1167	1167		
15 руб.	Информатика. Учебно-методическое пособие	—	—		

ДОЛГОЖИТЕЛЬ

45 лет назад (10 ноября 1954 года) был опубликован первый отчет, связанный с созданием языка Фортран (версия Фортран I)

История языков программирования тесно связана с историей развития компьютерной техники. За прошедшие с 40-х годов десятилетия были созданы тысячи языков программирования высокого уровня. Большинство из них остались достоянием их создателей, десятки нашли применение в узких областях, и только немногие языки, такие, как Фортран, Алгол, ПЛ/1, Бейсик, Паскаль, Си, получили широкое мировое признание.

Один из первых языков программирования — Фортран (FORmula TRANslator — транслятор, или переводчик формул), который предназначался для описания инженерных и научных задач, был разработан группой сотрудников фирмы IBM под руководством Дж. Бэкуса. Многие идеи, заложенные в Фортране, нашли потом развитие в других языках программирования.

К характерным особенностям языка Фортран относятся:

- сравнительная простота (и легкость его изучения);
- удобные и развитые средства для ввода и вывода данных;
- “близость” записи арифметических выражений к обычной математической записи;
- возможность работы с комплексными переменными и переменными двойной точности.

Первая версия языка реализована на ламповой ЭВМ IBM 704. Для подготовки первого компилятора потребовалось 18 человеко-лет [1]. В 1958 году появилась

усовершенствованная версия — Фортран II, а затем другая версия — Фортран III, не получившая, однако, распространения. Создание примерно в это же время языка Алгол 60 не нанесло существенного “ущерба” Фортрану — главным образом из-за того, что разработчики языка Алгол практически не уделили внимания организации ввода и вывода [2]. Очень большую популярность получила версия под названием Фортран IV, появившаяся в 1962 году. Однако стандарт языка еще отсутствовал, и это мешало перенести программы с машин одного типа на машины других типов. Положение исправилось в 1966 году, когда был введен стандарт, известный как Фортран 66.

Позже появились еще несколько версий, на основе которых была создана переработанная версия стандарта языка — Фортран 77, тоже ставшая весьма популярной. В 1984 году были рассмотрены предложения, связанные с выработкой новой стандартной версии Фортран 8x (позднее получившей название Фортран 88).

В ней предусматривались некоторые изменения, касающиеся, в частности, работы с файлами.

Фортран до сих пор употребляется при решении инженерных и научных задач, но за прошедшие годы он значительно расширился и усовершенствовался (появились еще Фортран 90, предусматривающий возможность параллельной обработки данных, и даже Visual FORTRAN), превратившись в язык, удобный для использования во многих сферах. “Фортран будет всегда” — примерно таков смысл высказываний “поклонников” этого языка-долгожителя.

Литература

1. *Малыхина М.П., Частиков А.П.* Языки программирования: Фортран // Вычислительная техника и ее применение. № 8/88.

2. *Белеуки Я.* Фортран 77: пер. с польск. М.: Высшая школа, 1991.

<p>Гл. редактор С.Л. Островский Зам. гл. редактора Е.Б. Докшицкая Редакция: И.Н. Фалина, Н.Л. Беленькая, Н.П. Медведева Дизайн и компьютерная верстка: Н.И. Пронская Корректоры: Е.Л. Володина, С.М. Подберезина</p>	<p>©ИНФОРМАТИКА 1999 выходит четыре раза в месяц При перепечатке ссылка на ИНФОРМАТИКУ обязательна, рукописи не возвращаются</p>	<p>121165, Киевская, 24 тел. 249 4896 Отдел рекламы тел. 249 9870</p>	<p>Учредитель: ООО “Чистые пруды” Регистрационный номер 012868 Отпечатано в типографии ОАО ПО “Пресса-1”. 125865, ГСП, Москва, ул. “Правды”, 24. Тираж 5000 экз. Заказ №</p> <p>Internet: inf@1september.ru Fidonet: 2:5020/69.32 WWW: http://www.1september.ru</p>
<p>ИНДЕКС ПОДПИСКИ для индивидуальных подписчиков 32291 комплекта приложений 32744</p>			
<p>Тел. (095)249 3138, 249 3386. Факс (095)249 3184</p>			