

Великие информатики

- **«История рассказывает о том, что было, поясняет смысл уже содеянного и, что самое главное, помогает предвидеть достижения завтрашнего дня».**

А. Частиков

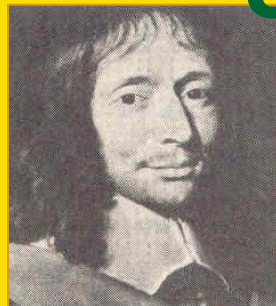
- **«Лучший пророк для будущего – прошлое».**

Лорд Байрон

Наука создается людьми и для людей



Аристотель
(384-322гг.
до н.э.)



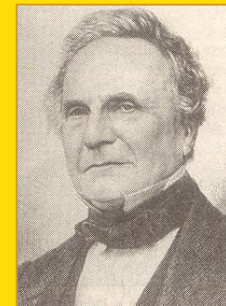
Блез Паскаль
(1623-1662гг.)



Вильгельм
Шиккард
(1592-1636гг.)



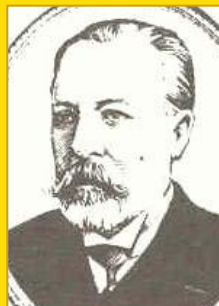
Готфрид Вильгельм
Лейбниц
(1646-1716гг.)



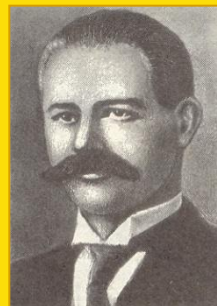
Чарльз
Баббидж
(1792-1871гг.)



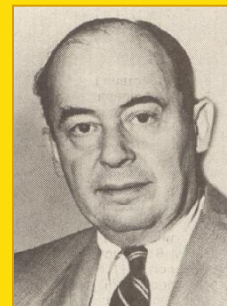
Ада Августа
Лавлейс
(1815-1852гг.)



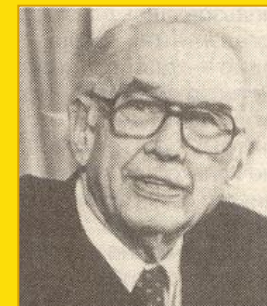
Вильгодт Однер
(1845-1905гг.)



Герман Холперит
(1860-1929гг.)



Джон фон
Нейман
(1903-1957гг.)

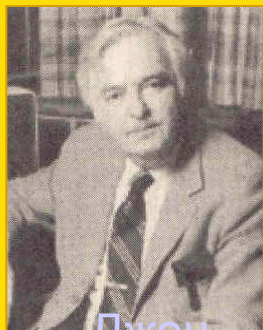


Джон
Атанасов

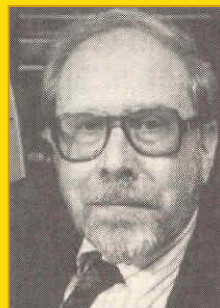
Создатели языков программирования



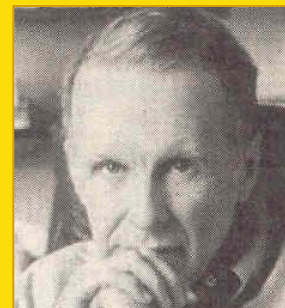
Томас Курц



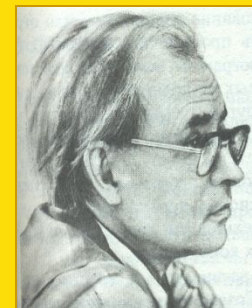
Джон
Кемени



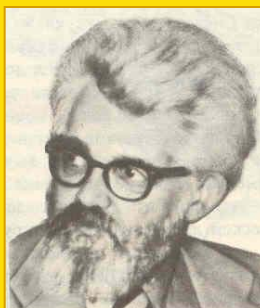
Никлаус Вирт



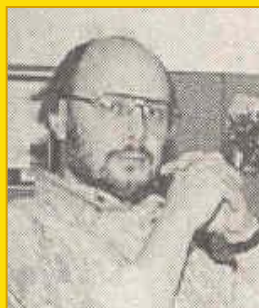
Джон Бекус



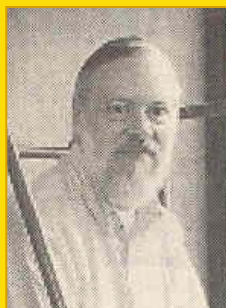
А. П. Ершов



Джон Маккарти



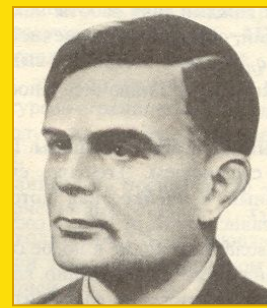
Бьерн
Страуструп



Деннис
Ритчи



Грейс Мюррей
Хоппер



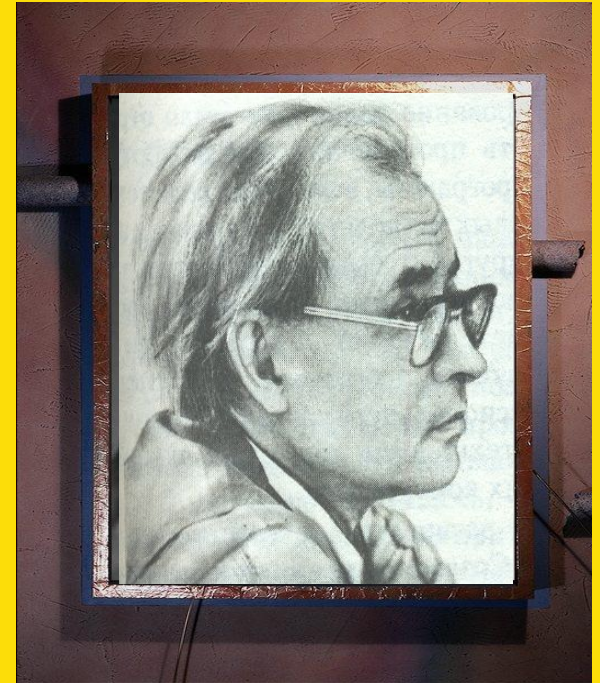
Алан Матисон
Тьюринг
(1912-1954гг.)

А. П. Ершов

Программист должен обладать способностью первоклассного математика к абстракции и логическому мышлению в сочетании с эдисоновским талантом сооружать все, что угодно, из нуля и единицы. Он должен сочетать аккуратность бухгалтера с пронцательностью разведчика, фантазию автора детективных романов с трезвой практичностью экономиста. А кроме того, программист должен иметь вкус к коллективной работе, понимать интересы пользователя и многое другое.

А.П. Ершов

"Андрей Петрович Ершов — известнейший ученый в области информатики, как и многие из его поколения, занялся вычислительным делом почти случайно".



Home



Андрей Петрович Ершов (1931 — 1988) поступил на работу в Институт точной механики и вычислительной техники Академии наук (ИТМ и ВТ) в 1953 году, когда был еще студентом Московского государственного университета им. М.В. Ломоносова. В то время в ИТМ и ВТ формировался один из первых в нашей стране коллективов программистов. Спустя год Ершов заканчивает университет, а в 1957 году его назначают заведующим отделом автоматизации программирования в созданном незадолго до этого Вычислительном центре Академии наук. Вскоре, в связи с образованием Сибирского отделения Академии наук, по просьбе директора Института математики данного отделения академика С.Л. Соболева он берет на себя обязанности организатора и фактического руководителя отдела программирования этого института и затем в 1960 году окончательно переезжает в Сибирь. Позже отдел Ершова вошел в состав созданного в 1964 году Г.И. Марчуком Вычислительного центра Сибирского отделения Академии наук.

Благодаря Ершову новосибирский Академгородок становится одним из ведущих центров в области программирования, а сам он оказывает большое влияние на развитие информатики в стране.



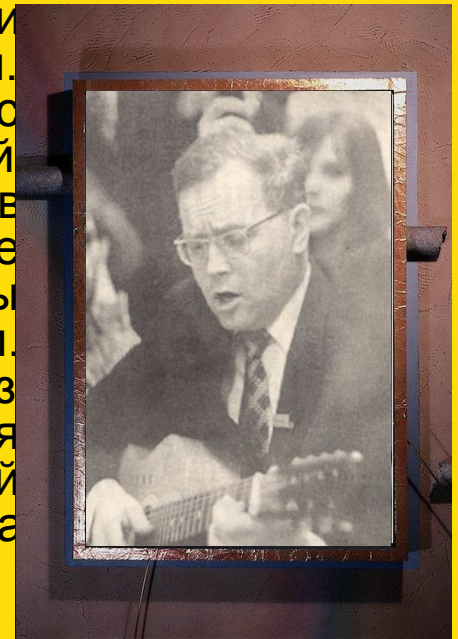
Понимая, как важна научно-организационная деятельность при становлении новых научных направлений, Ершов уделяет ей много внимания не только в масштабе страны. Он работает в редколлегиях международных изданий (Information Processing Letters, Theoretical Computer Science, Acta Informatica и других), а также участвует в мероприятиях, проводимых Международной федерацией по обработке информации (International Federation for Information Processing — IFIP): является членом нескольких ее комитетов и рабочих групп, активным участником ряда конгрессов. [В 1980 году за плодотворную деятельность в IFIP Ершова награждают "Серебряным сердечником" (Silver Core) — высшим знаком отличия, принятым в этой организации.]

Для научной деятельности Андрея Петровича Ершова характерно сочетание теоретических исследований и практических работ. Творческое наследие Ершова обширно, оно включает около 200 работ (в том числе ряд монографий), не считая большого числа предисловий, редакционных статей, отзывов и т.п.

Как и большинство программистов 1950-х годов, он начинал с создания алгоритмов численных методов и стандартных программ. Первая работа Ершова была посвящена предложенному им методу обращения матриц.



В середине 1950-х годов шло становление системного и так называемого теоретического программирования. Развитие системного программирования начиналось с работ в области, названной, тогда автоматизацией программирования и связанной с созданием языков программирования и методов их трансляции. Первые отечественные языки программирования были основаны на операторных схемах, предложенных А.А. Ляпуновым. Под руководством Ершова разрабатывались одни из первых отечественных программирующих программ для машин БЭСМ и "Стрела" (программирующей программой называлась в то время комбинация языка программирования и транслятора).



Позже под его руководством были созданы такие известные системы, как АЛЬФА (первая в мировой практике оптимизирующая система программирования для более сложных языков, чем Фортран), АЛЬФА-6 и БЕТА. Они во многом определили современную методологию оптимизирующей трансляции.

Идейно к работам по автоматизации программирования примыкают и труды Ершова по теоретическому программированию, "чьим объектом изучения являются математические абстракции программ".



Ершов умел оценить текущее состояние науки и практики и наметить направления, которые будут определять развитие программирования. Он одним из первых в стране увидел те новые возможности взаимодействия с ЭВМ, которые способны дать системы разделения времени. В 1966 году Андрей Петрович организует работы, связанные с автоматическими информационными станциями (проект АИСТ). Проект АИСТ предусматривал проведение широкого круга исследований, касающихся архитектуры вычислительных комплексов, их программного обеспечения и моделирования вычислительных систем. В рамках данного проекта подготовлена первая в стране развитая система разделения времени АИСТ-0 (которая была реализована на многомашинном комплексе из отечественных ЭВМ).

Обобщая свой опыт руководства большими программными проектами, Ершов выдвинул несколько тезисов, касающихся организации работы программистов, сформулировал ряд общих принципов программирования как нового и своеобразного вида научной деятельности, затронул тот аспект, который позже будет назван дружелюбностью к пользователю, одним из первых в стране поставил задачу создания технологии программирования.

Благодаря своему умению видеть новые перспективы в программировании, Ершов часто выступал как инициатор новых направлений исследований. Сюда относятся, например, методы использования естественных языков для взаимодействия с ЭВМ, автоматизация редакционно-издательской деятельности.



Большое внимание Ершов уделял подготовке кадров. В Новосибирском университете он организовал и затем беспрерывно руководил коллективом, занимающимся подготовкой студентов и аспирантов в области системного и теоретического программирования.

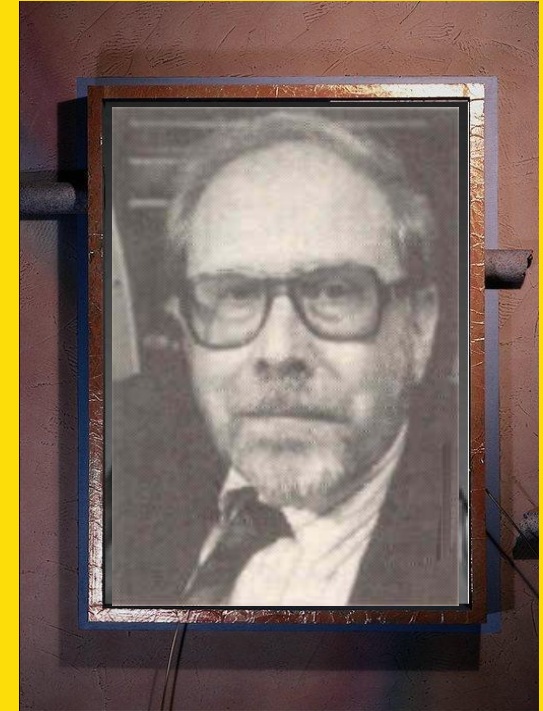
В последний период жизни Андрея Петровича Ершова привлекло обучение программированию и вообще информатике в школе. Он стал одним из создателей и признанным лидером так называемой "школьной информатики", вошел в число ведущих мировых специалистов в этой области. В подготовленной им вместе с Г.А. Звенигородским и Ю.А. Первиным работе "Школьная информатика. Концепции, состояние, перспективы" определены перспективы развития данного направления на годы вперед. Андрей Петрович являлся одним из авторов и редакторов первого школьного курса информатики, а также методического пособия к этому курсу. Незадолго до его кончины появился новый учебник информатики, одним из авторов и редактором которого он был.

Ершов готовил телевизионный курс по этой дисциплине, руководил созданием школьных систем программирования и школьного программного обеспечения вообще, вел большую организационную работу. Главным делом последних лет его жизни являлось создание школьной энциклопедии по информатике.



Никлаус Вирт

В середине 1960-х годов Международная федерация по обработке информации — IFIP (International Federation for Information Processing) предложила нескольким специалистам в области информатики принять участие в создании нового языка программирования — преемника Алгола-60. Среди них был швейцарский ученый Никлаус Вирт, который преподавал в то время в Станфордском университете. Вирт участвовал в создании проекта языка, названного Алгол-W (по его фамилии — Wir(h)). И хотя IFIP данный проект отклонила (был принят проект языка Алгол-68), Вирт, возвратившись в Швейцарию, продолжил работу. Он хотел получить инструмент для обучения студентов программированию: в этом смысле Вирта не устраивал не только новый Алгол, но и все ныне используемые основные языки программирования, свойства и конструкции которых зачастую нельзя объяснить логически и убедительно и которые нередко просто оскорбляют здравый смысл.



Вскоре Никлаус Вирт и его сотрудники из Федерального технического университета в Цюрихе подготовили первую версию языка, получившего название Паскаль [в честь знаменитого француза — математика, физика, философа и писателя Блеза Паскаля (1623 — 1662), построившего одну из первых механических счетных машин], затем — первый вариант компилятора, а в 1971 году вышло описание языка. В 1974 году появилась новая версия Паскаля, и спустя еще какое-то время было опубликовано руководство для пользователей, которое, по словам Вирта, вполне можно рассматривать как "стандартное определение" языка Паскаль.

Еще подростком Вирт увлекался конструированием радиоуправляемых моделей самолетов, причем позже он получил в Калифорнийском университете степень бакалавра в области электротехники. К разработке языка Вирт подошел так, как подошел бы инженер к конструированию машины. А "искусство инженера,— говорил он, — состоит в том, чтобы делать сложные задачи простыми".



Язык Паскаль требует от программиста описания всех переменных в отдельном разделе, помещаемом в начале программы. Наличие такого раздела ограничивает свободу программиста. Но одновременно Паскаль требует строгого стиля программирования, который способствует уменьшению количества ошибок. Как и всякий преемник, Паскаль имеет многие черты своего предшественника; например, Паскаль, как и Алгол-60, предусматривает блочную структуру программ (программы состоят из блоков, которые начинаются и оканчиваются словами BEGIN и END соответственно). Структура программы на Паскале позволяет даже специалисту, не являющемуся ее автором, обнаруживать и исправлять допущенные ошибки и вносить изменения. Это делает Паскаль подходящим языком для создания больших программ. В то же время часто говорят, что Паскаль особенно удобен для изучения теории и практики программирования, но не для реальных применений (и ничего удивительного здесь нет: основная цель, которую преследовал Вирт, создавая Паскаль, — разработать язык всего лишь для обучения программированию).

Успех Паскаля превзошел все ожидания. Возможно, главной причиной популярности этого языка было то, что он способствовал развитию зарождающегося тогда движения за так называемое структурное программирование, которое потом очень быстро обрело силу.



Паскаль стал прародителем нескольких языков программирования. Примерно через 10 лет после него Вирт создал язык Модула-2 (Modula 2, от MODUlar LAnguage — модульный язык), который основан на многих концепциях, используемых в Паскале, причем наиболее примечательным тут стало введение средств построения больших и сложных программ, которые можно было теперь создавать в виде отдельных самостоятельных модулей, а затем собирать их вместе.

"Можно считать, что история языка Модула-2 начинается в 1980 году, когда Никлаус Вирт, известный большинству специалистов как создатель языка Паскаль, опубликовал описание нового языка программирования, названного им Модула. В отличие от Паскаля, который был по замыслу языком для обучения программированию, Модула с самого начала представляла собой язык для профессиональных системных программистов, продолжая лучшие традиции своего предшественника и обогащая их новыми идеями, соответствующими таким современным требованиям к языкам программирования, как структурность, модульность и способность к расширению.



Как и множество других языков программирования, Модула подверглась эволюции, во время которой ее первоначальное название было переделано, но в имя Модула-2. Базовым описанием языка считается описание, представленное в третьем издании книги Вирта "Программирование на языке Модула-2", то есть его развитием является язык Модула-3, разработанный совместно фирмами DEC и Olivetti, а Вирт создал еще язык Oberon.)

На Паскаль опирались и все четыре языка, отобранные военным ведомством США в конце 1970-х годов после того, как был объявлен конкурс, в результате которого появился универсальный язык высокого уровня, получивший в дальнейшем название "Ада" (в честь первого программиста — Августы Ады Лавлейс). Следует также отметить, что среди расширений и диалектов Паскаля особую популярность получило семейство систем программирования фирмы Borland для персональных компьютеров, получившее название "Турбо Паскаль".

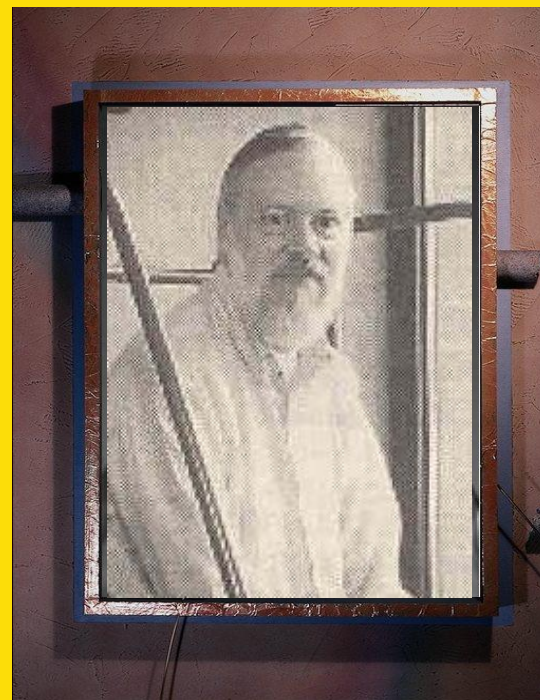


Деннис Ритчи

Язык С — это инструмент, острый как бритва с его помощью можно создать и элегантную программу, и кровавое месиво.

Брайан Керниган

Одним из самых популярных языков программирования является язык С, который был создан как инструмент написания операционной системы UNIX в научно-исследовательской фирме Bell Telephone Laboratories американской корпорации AT&T (American Telephone and Telegraph).



В 1969 году Кен Томпсон, считавшийся среди коллег лучшим программистом в мире, начал разрабатывать эту систему для мини-компьютера PDP-7 фирмы DEC. А в 1972 году Деннис Ритчи, специалист по системному программированию, который получил степень бакалавра по прикладной математике в Гарвардском университете, активно помогавший разрабатывать UNIX, представил язык C, где сочетались лучшие свойства ассемблера и языков высокого уровня. От ассемблера были взяты гибкие и эффективные средства работы с памятью, а от языков высокого уровня — широкий набор управляющих конструкций, возможность использования сложных структур данных, средств ввода и вывода. Ритчи считал, что новый язык пригодится для работы. И он не ошибся. Для повышения скорости функционирования операционных систем при их создании обычно применялся язык низкого уровня ассемблер, а язык C настолько хорошо себя зарекомендовал, что на нем была написана почти вся операционная система UNIX.

Вообще это один из самых универсальных языков программирования. Его можно применять также для создания драйверов внешних устройств, трансляторов языков программирования, текстовых процессоров, программ для решения математических, экономических и многих других прикладных задач.



Название языка соответствует третьей букве английского алфавита. Один из языков, появившийся в 1960-х годах, назывался APL (A Programming Language). Тем самым, как бы была занята первая буква алфавита — А (Эй). Во время работы над операционной системой UNIX Томпсоном был создан язык программирования В (Би), "для внутреннего пользования", который оказал сильное влияние на следующий язык, разработанный Ритчи для тех же целей. Подчеркивая эту преемственность, языку дали название С (Си). Первое описание языка было приведено в книге. "Керниган подготовил почти весь пояснительный материал, я же отвечал за приложение, куда было включено справочное руководство по языку, и за главу о взаимодействии с UNIX", — сообщает Ритчи. Долгое время данное описание считалось стандартом, однако некоторые его места допускали неоднозначное толкование, что породило множество трактовок языка С. Для исправления ситуации при Американском национальном институте стандартов (American National Standards Institute — ANSI) был образован комитет по стандартизации языка, и в 1983 году был утвержден стандарт, получивший название ANSI C. "Прямыми потомками" языка С являются языки: С++ (Б.Страуструп, 1984 г.); Concurrent C (Н.Джехани, 1986 г.); Objective C (Б.Кокс, 1986 г.); С* (Thinking Machines, 1987 г.); С+@, ранее Calico (Bell Labs, 1991). Самым известным здесь стал, безусловно, С++.



После реорганизации AT&T, уже в рамках Lucent Labs, Ритчи возглавлял группу, разрабатывавшую операционную систему Plan 9 (1995 г.). Она явилась основой следующего проекта его группы — операционной системы Inferno, о создании которой было объявлено в апреле 1996 года. Для этой системы Ритчи подготовил язык Limbo. Сейчас Ритчи, имеющий степень бакалавра по физике и магистра по прикладной математике, возглавляет отдел исследований в области системного программного обеспечения в фирме Bell Laboratories. В 1983 году Деннису Ритчи и Кену Томпсону за разработку и реализацию языка программирования C и операционной системы UNIX была вручена премия Тьюринга. В 1988 году Ритчи был избран в Американскую национальную инженерную академию (National Academy of Engineering).

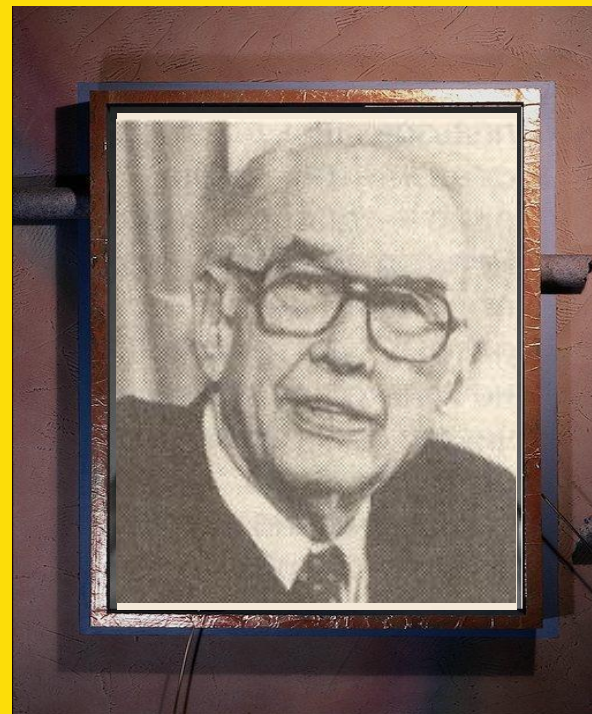


Почему же язык С покорила сердца миллионов программистов? "Главные достоинства С состоят в том, что он предоставляет программисту возможность полного управления реализацией и что программы, написанные на С, работают весьма быстро", — считает Брайан Керниган. А вот ответ самого Денниса Ритчи: "Язык С — это невероятный, оглушительный и огромный успех. Хотя повороты судьбы ему помогали, он, безусловно, удовлетворил потребности в таком языке реализации систем, который был достаточно эффективным, чтобы заменить ассемблер, и в то же время достаточно абстрактным и гибким, чтобы описывать алгоритмы и взаимодействия в широком спектре программных сред". При этом "существует масса прекрасных языков (прекраснее, чем С), которые не привились, — отмечает Ритчи, — но кто-то все же выигрывает..."



Джон Атанасов

В 1945 году был собран цифровой компьютер ЭНИАК (ENIAC, от Electronic Numerical Integrator and Computer — электронный цифровой интегратор и компьютер), который... считался первым действующим электронным компьютером", однако позже право называть ENIAC первым цифровым электронным компьютером было оспорено. "Дело сводилось к тому, что в конце 1930-х годов в более простой машине американского профессора физики и математики Джона Атанасова были использованы те же принципы конструирования электронных переключателей на вакуумных лампах. В 1973 году... суд постановил, что патентные права на основные идеи цифровых электронных машин принадлежат Джону Атанасову".



Поиски путей к автоматизации вычислений Атанасов начал в 1933 году, когда руководил аспирантами, занимавшимися теорией упругости, квантовой физикой и физикой кристаллов. Ему было тогда 30 лет. В большинстве задач, с которыми сталкивались "подопечные" Атанасова, фигурировали дифференциальные уравнения в частных производных. Для их решения приходилось использовать приближенные методы, они же, в свою очередь, требовали решения больших систем алгебраических уравнений. А потому ученый стал предпринимать попытки применения технических средств для ускорения вычислений.

Сначала Атанасов пробовал использовать так называемый дифференциальный анализатор (специализированный аналоговый компьютер для решения дифференциальных уравнений), однако вскоре убедился, что это устройство не дает достаточной точности. Затем он обратился к табулятору Германа Холлерита, но обнаружил, что при решении больших систем уравнений скорость вычислений на машине, построенной на базе электромеханических элементов, явно недостаточна. И тогда Атанасов задумал сконструировать компьютер, основанный на новых принципах, взяв при этом в качестве элементной базы электронные лампы.



Ему не удалось заинтересовать своим проектом фирмы, производящие счетные машины, поскольку в 1930-е годы мало кто верил в возможность создания надежного вычислительного устройства на базе электронных ламп. Финансировать, работу согласились экспериментальная агрономическая станция штата Айова и частная нью-йоркская фирма, выполнявшая научные исследования.

Осенью 1939 года Джон Атанасов и его ассистент Клиффорд Берри приступили к постройке машины — специализированной ЭВМ, предназначенной для решения системы алгебраических уравнений с 30 неизвестными. Было решено назвать ее ABC (Atanasoff-Berry Computer).

Исходные данные, представленные в десятичной системе счисления, должны были вводиться в машину с помощью стандартных перфокарт. Затем в самой машине осуществлялось преобразование десятичного кода в двоичный, который далее в ней и использовался.

Основными арифметическими операциями являлись сложение и вычитание, а умножение и деление выполнялись уже с их помощью. Имелась также операция сдвига числа влево и вправо на один разряд (что при применении двоичной системы равносильно умножению и делению на два). Для реализации этих операций употреблялись ламповые сумматоры.

В машине было два запоминающих устройства, каждое из которых представляло собой вращающийся барабан, содержащий множество бумажных конденсаторов. Конденсаторы располагались внутри барабана, а на его внешней поверхности находились медные контакты, соприкасавшиеся при вращении барабана со специальными щетками. Во избежание ошибок "содержимое" каждого конденсатора (определяющее значение соответствующего двоичного разряда) периодически обновлялось (регенеративная память).

К весне 1942 года работу над машиной удалось в основном завершить; не было готово только периферийное оборудование. Однако в это время США уже находились в состоянии войны с нацистской Германией; и проблемы военного времени отодвинули работу над первой ЭВМ на задний план. Вскоре машину демонтировали.



Получить патент на машину Атанасову не удалось, но это, по всей видимости, тогда не очень его беспокоило. К работе над ЭВМ он больше не возвращался. В 1963 году Джон Атанасов вышел на пенсию. В том же году умер Клиффорд Берри.

Об их компьютере широкой публике стало известно лишь в связи с вышеупомянутым судебным процессом, закончившимся признанием приоритета Атанасова.

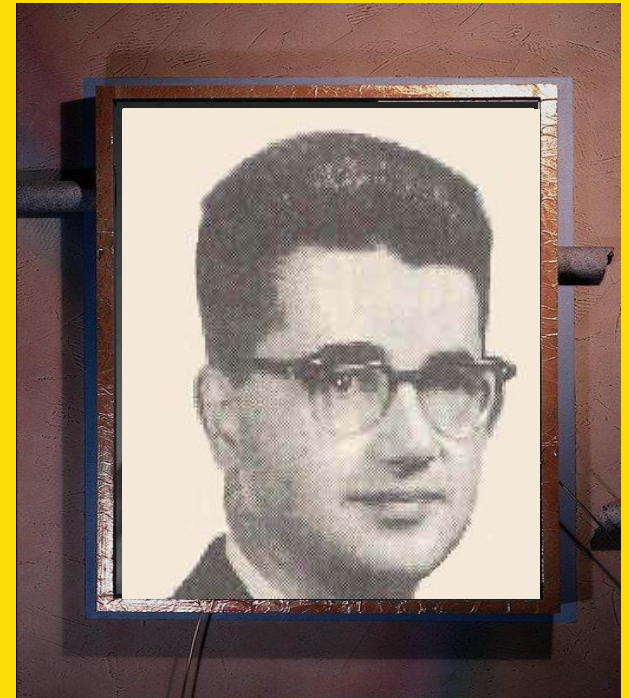
В Болгарии раньше, чем в США оценили вклад Атанасова в науку. В 1970 году он был удостоен одной из высших наград Болгарии — ордена Кирилла и Мефодия.

Ну а ЭНИАК по-прежнему считается первым успешно функционировавшим электронным цифровым компьютером (который к тому же наглядно продемонстрировал преимущества ЭВМ перед механическими и электромеханическими машинами).



Джон Маккарти

В 1950-х годах специалисты по искусственному интеллекту начали искать язык, пригодный для манипулирования понятиями, выраженными словами и фразами на естественном языке. (Искусственный интеллект — раздел информатики, ставящий своей целью разработку методов решения задач, для которых отсутствуют формальные, алгоритмы, понимание естественного языка, обучение, доказательство теорем, распознавание образов.) Первый результат тут был получен в виде семейства языков под названием ИПЛ (IPL, от Information Processing Languages — языки обработки информации), созданного одним из пионеров в области искусственного интеллекта Алленом Ньюэллом и его сотрудниками. Центральным для ИПЛ являлось понятие списка.



Представляя данные в виде списка слов и символов, программист мог связать понятия в памяти компьютера приблизительно таким же образом, как, по мнению специалистов по искусственному интеллекту, они связываются в памяти человека. Понятием списка заинтересовался и Джон Маккарти, разносторонний математик, один из ведущих исследователей в области искусственного интеллекта (причем сам термин искусственный интеллект был предложен именно им в 1956 году). Маккарти родился в 1927 году в семье профсоюзного деятеля. Среди его многочисленных работ наиболее важные относятся к концу 1950-х годов, когда он работал в Массачусетском технологическом институте (МТИ). Маккарти был генератором самых фантастических идей, а также "человеком поразительной внешности с шокирующими манерами, например, привычкой поворачиваться и уходить, оборвав разговор на полуслове".

В 1957 — 1958 годах по инициативе Маккарти при МТИ сформировалась лаборатория искусственного интеллекта, и он вместе с несколькими студентами стал создавать язык высокого уровня, в котором понятие списка сочеталось с системой обозначений, заимствованной из специального раздела математики — так называемого лямбда-исчисления (λ -исчисления). Введенное в 1931 году математиком Алонзо Черчем, лямбда-исчисление оперирует всего тремя типами элементов: символами, представляющими переменные и константы; скобками для группировки символов; обозначениями функций с использованием греческой буквы лямбда.



Появившийся вскоре язык получил название Лисп (LISP, от LISt Processing — обработка списков). Спустя какое-то время вышла статья Маккарти "Рекурсивные функции и символьные вычисления" с математическим обоснованием применения Лиспа. "Лисп являлся для нас не просто языком, который используют для определенных целей, — говорил один из студентов, помогавших Маккарти, — им можно было любоваться, как прелестной вещью".

Программы и данные Лиспа существуют в форме символьных выражений, хранимых в виде "спусковых" структур. Употребляются два вида объектов: атомы и списки. Атомы — это символы, применяемые для идентификации объектов, которые могут быть числовыми или символьными. Список — это последовательность (возможно, пустая) элементов, заключенных в круглые скобки, каждый из которых является либо атомом, либо списком. Для манипулирования списками употребляются так называемые списочные примитивы (примитивные функции), из которых получаются все другие функции обработки списков. Первая реализация Лиспа была осуществлена на ламповом компьютере 1BM 704. [У нас первый транслятор (интерпретатор) Лиспа был создан для БЭСМ-6 в конце 1960-х годов.]



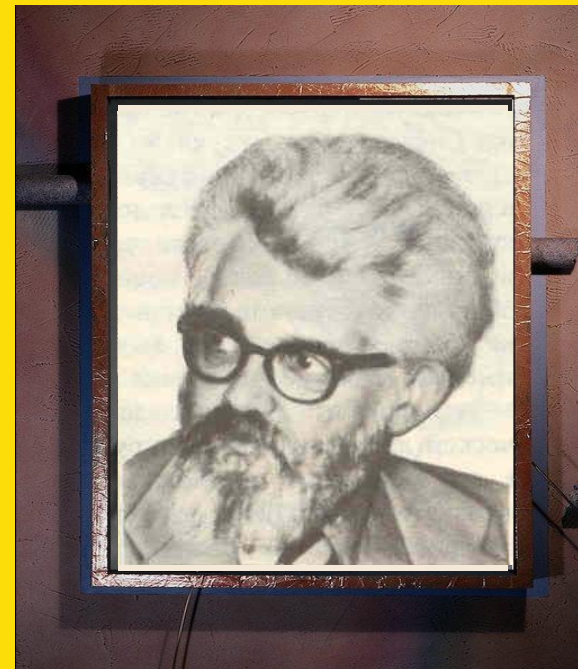
Даже небольшой фрагмент программы на Лиспе способен содержать десятки пар скобок, куда заключены списки. Часто пары скобок вкладываются друг в друга, образуя список внутри списка, который, в свою очередь, также находится внутри списка (иногда бывает 8 — 10 уровней вложенности). Характерная особенность языка Лисп состоит в том, что и программы, и данные, обрабатываемые с их помощью, имеют одинаковую структуру.

В 1960-е годы предпринимались попытки аппаратной реализации Листа, однако реальная возможность решения этой задачи появилась позже, когда начало интенсивно развиваться производство интегральных схем. А с 1982 года Лисп-компьютеры (с реализацией различных версий языка) стали выпускаться серийно целым рядом фирм.

Скорость и легкость, с которой можно писать, выполнять и изменять программы на Лиспе, способствовали появлению многочисленных поклонников этого языка и вне лабораторий искусственного интеллекта. Подобно большинству популярных языков, Лисп имеет достаточно большое "потомство". Так, в Великобритании употреблялся появившийся в 1960-х годах язык ПОП-2 (POP 2), названный в честь одного из его создателей Робина Дж. Поплстона из Эдинбургского университета.



Лиспу уже более сорока лет, и, несмотря на то, что этот язык имеет определенные недостатки (например, связанные с уже отмечавшейся необходимостью использования большого количества скобок), он весьма широко применяется на всех этапах создания систем искусственного интеллекта и вообще решения сложных логических задач. "LISP... в течение длительного времени рассматривался как кандидат на стандарт языка в области искусственного интеллекта. Позднее в эти притязания вмешался Prolog".

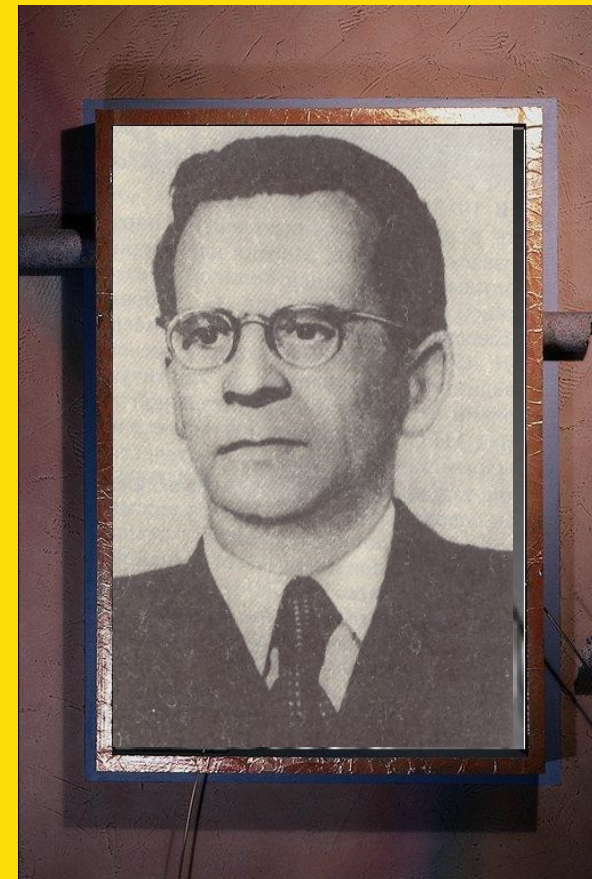


В 1959 году Маккарти предложил концепцию разделения времени (режим работы компьютера, позволяющий нескольким программистам, работая за разными терминалами, одновременно использовать его память и процессор). Тем самым специалисты избавлялись от необходимости ждать несколько часов или даже дней, пока их задания будут обработаны машиной. (По другим данным, идею разделения времени впервые высказал англичанин К.Стрейчи.)



Сергей Алексеевич Лебедев

Характерной чертой научной деятельности Сергея Алексеевича Лебедева (1902 — 1974), проявившейся с самого ее начала, являлось органическое сочетание большой глубины теоретической проработки с конкретной практической направленностью. При этом научный поиск и конструкторскую деятельность он всегда сочетал с работой, связанной с подготовкой кадров. Лебедев преподавал в вузах, являлся в свое время заведующим кафедрой релейной защиты и автоматизации электрических систем Московского энергетического института, был одним из инициаторов создания Московского физико-технического института, где основал кафедру вычислительной техники.



Активным сторонником работ, связанных с автоматизацией научных исследований и математических расчетов, Лебедев стал в 1936 — 1937 годах: тогда в возглавляемом им отделе автоматики Всесоюзного электротехнического института в Москве началось создание анализатора для решения дифференциальных уравнений, и имеются свидетельства, что уже в то время ученый задумывался над принципами построения вычислительных машин, использующих двоичную систему счисления.

В конце 1940-х — начале 1950-х годов уже в Киеве, в лаборатории моделирования и вычислительной техники Института электротехники Академии наук Украины, под руководством Лебедева создается первая отечественная электронная цифровая вычислительная машина МЭСМ малая электронная счетная машина (являющаяся одной из первых в мире и первой в Европе ЭВМ с хранимой в памяти программой). Эта работа, по сути дела, и определила всю его дальнейшую научную и организационную деятельность.

В 1948 году в Москве организуется Институт точной механики и вычислительной техники (ИТМ и ВТ) Академии наук, а через два года, когда основные работы, связанные с созданием МЭСМ, подходили к концу, Лебедев принимает приглашение перейти в этот институт и стать главным конструктором БЭСМ — быстродействующей электронной счетной машины. (Вскоре он был назначен директором указанного института).

В качестве макета новой машины Лебедев рекомендует принять МЭСМ. Такой подход дал возможность применить весь накопленный опыт и в весьма короткие сроки построить ЭВМ БЭСМ-1 со средним быстродействием около 10 000 оп./с. Осенью 1952 года новая машина была введена в эксплуатацию, а в следующем году принята Государственной комиссией. Тогда БЭСМ-1 являлась самой быстродействующей ЭВМ в Европе и не уступала лучшим компьютерам США [1]. Вскоре машина была немного модернизирована и в 1956 году стала серийно выпускаться под названием БЭСМ-2. На БЭСМ-2 выполнялись расчеты при запуске искусственных спутников Земли и первых космических кораблей с человеком на борту.

В 1958 году появилась еще более мощная универсальная ЭВМ — М-20, которая примерно через год была запущена в производство. Эта машина, в которой использовалось 4500 электронных ламп и 35 000 полупроводниковых диодов, была самой производительной отечественной ЭВМ первого поколения. Она стала последней созданной под руководством академика Лебедева машиной этого поколения и одновременно послужила исходной моделью ряда машин второго поколения: БЭСМ-3М, БЭСМ-4, М-220, М-222. Совершенствование шло в основном по пути модернизации внешних устройств, увеличения емкости оперативной памяти на магнитных сердечниках и внешних запоминающих устройств.

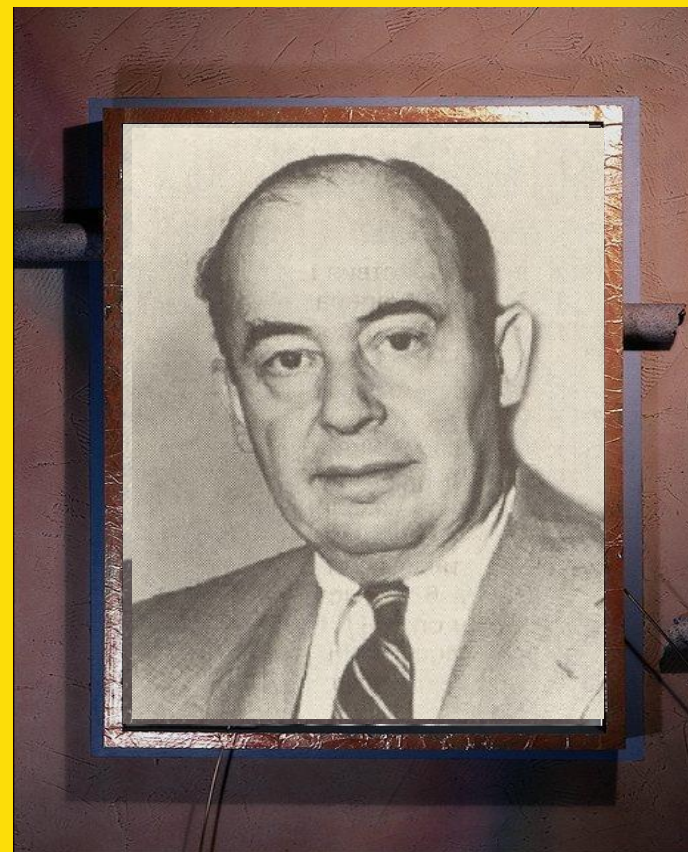
По структурной организации первые три из указанных ЭВМ мало чем отличались от М-20, а БЭСМ-4 называли даже ее "полупроводниковым вариантом", который "отличается от машины М-20 только большей емкостью оперативной и внешней памяти и более широким набором устройств ввода и вывода". Быстродействие БЭСМ-4 было несколько ниже быстродействия машины М-20, а системы команд у них являлись совместимыми — в том смысле, что любая программа ЭВМ М-20 могла быть "правильно выполнена на машине БЭСМ-4".

Наконец, в 1967 году начала серийно выпускаться созданная под руководством С.А. Лебедева и В.А. Мельникова оригинальная по архитектуре ЭВМ БЭСМ-6 с быстродействием около 1 млн оп./с. В ней использовалось 60 тыс. транзисторов и 200 тыс. полупроводниковых диодов. "Это полупроводниковая машина нового типа с хорошо развитыми периферийными устройствами, которые могут работать одновременно с центральным вычислительным устройством машины. На машине могут одновременно находиться в решении несколько задач, между которыми и распределяются отдельные участки памяти и периферийные устройства".

Тогда БЭСМ-6 стояла в ряду самых производительных ЭВМ в мире. Она являлась "первой большой отечественной машиной, которую начали поставлять пользователям вместе с развитым программным обеспечением".

Джон фон Нейман

Американский математик и физик Джон фон Нейман (1903—1957) был родом из Будапешта, второго по величине и значению после Вены культурного центра бывшей Австро-Венгерской империи. Своими необычайными способностями этот человек стал выделяться очень рано: в шесть лет он разговаривал на древнегреческом языке, а в восемь освоил основы высшей математики.



Работал он в Германии, но в начале 1930-х годов, предвидя ухудшение политической обстановки в стране, которое не могло не сказаться отрицательно на научной жизни, принял решение обосноваться в США. Джон фон Нейман внес существенный вклад в создание и развитие целого ряда областей математики и физики, оказал значительное влияние на развитие компьютерной техники. Он выполнил фундаментальные исследования, связанные с математической логикой, теорией групп, алгеброй операторов, квантовой механикой, статистической физикой, развил теорию игр и теорию автоматов. Фон Нейман является одним из создателей метода Монте-Карло (называемого еще "методом статистических испытаний") — численного метода решения математических задач, основанного на моделировании случайных величин. (Название *Монте-Карло* происходит от города Монте-Карло в княжестве Монако, знаменитого своими игорными домами. Дело в том, что одним из простейших устройств для получения случайных величин является рулетка.)

Преимущество метода Монте-Карло состоит в том, что он способен "сработать" там, где отказывают другие методы.



В 1945 году был опубликован доклад фон Неймана, в котором он наметил основные принципы построения и компоненты современного компьютера. Идеи, отраженные в докладе, развивались, и примерно через год появилась статья "Предварительное рассмотрение логической конструкции электронного вычислительного устройства". Здесь важно, что авторы, отвлекшись от электронных ламп и электрических схем, сумели обрисовать, так сказать, формальную организацию компьютера.

"Универсальная вычислительная машина должна содержать несколько основных устройств: арифметики, памяти, управления и связи с оператором. Мы хотим, чтобы после начала вычислений работа машины не зависела от оператора".

"Необходимо, чтобы машина могла запоминать некоторым образом не только цифровую информацию, требуемую для данного вычисления, но также и команды, управляющие программой, с помощью которой должны производиться эти вычисления".



"Если приказы машине представить с помощью числового кода и если машина сможет каким-то образом отличать число от приказа, то память можно использовать для хранения как чисел, так и приказов".

"Помимо памяти для приказов, должно существовать еще устройство, способное автоматически выполнять приказы, хранящиеся в памяти. Будем называть это устройство управляющим".

"Поскольку наша машина является вычислительной, в ней должен быть арифметический орган... Устройство, способное складывать, вычитать, умножать и делить".

"Наконец, должно существовать устройство ввода и вывода, с помощью которого осуществляется связь между оператором и машиной".

Отмечалось также, что машина должна работать с двоичными числами, быть электронной, а не механической и выполнять операции последовательно, одну за другой.



Таким образом, фактически "по фон Нейману" главное место среди функций, выполняемых компьютером, занимают арифметические и логические операции. Для них предусмотрено арифметико-логическое устройство. Управление его работой и вообще всей машины — осуществляется с помощью устройства управления. (Как правило, в компьютерах устройство управления и арифметико-логическое устройство объединены в единый блок — центральный процессор.) Роль хранилища информации выполняет оперативная память. Здесь хранится информация как для арифметико-логического устройства (данные), так и для устройства управления (команды).

После выхода этих работ компьютер был признан объектом, представляющим научный интерес, причем вскоре компьютеры, построенные в соответствии с приведенными положениями, стали называть "машинами фон Неймана".

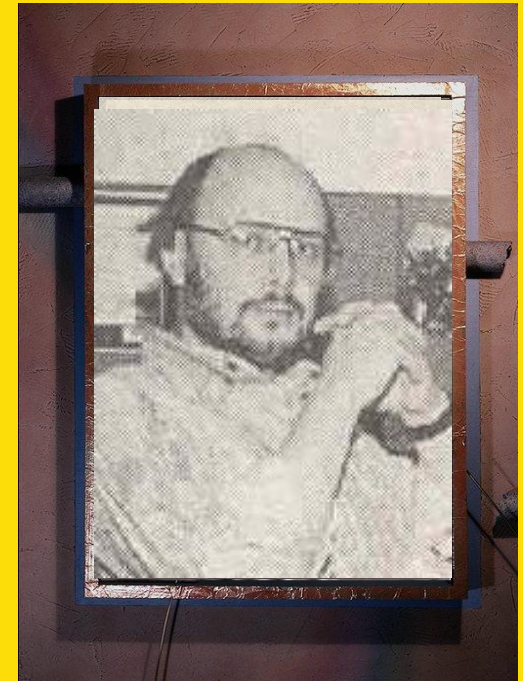
Архитектурные принципы организации ЭВМ, указанные Джоном фон Нейманом, долгое время оставались почти неизменными, и лишь в конце 1970-х годов в архитектуре супер-ЭВМ и матричных процессоров появились отклонения от этих принципов.



Бьерн Страуструп

Появление благодаря работам Бьерна Страуструпа языка Си++, ставшего фактически главным языком программной индустрии, а также повальное увлечение объектно-ориентированным проектированием и программированием заставили уйти в тень его именитого предшественника — язык Си.

"Сообщество сторонников Си++ страдало от чрезмерной популярности этого языка: он постоянно становился объектом нападок, ведь в современном мире, живущем по законам коммерции, честная борьба — большая редкость".



Бьерн Страуструп в начале 1980-х годов в научно-исследовательской фирме Bell Telephone Laboratories американской корпорации AT&T (*American Telephone and Telegraph*) Бьерном Страуструпом в результате дополнения и расширения языка Си был создан язык, получивший название "Си с классами". В 1983 году это название было заменено на Си++. Здесь как бы сделано указание на то, что Си++ — язык следующего поколения по отношению к Си как известно, операция инкремента ++ увеличивает значение переменной на единицу. (При создании Си++ Страуструп искал базовый язык для включения в него концепции классов, взятой из языка Симула. Как альтернатива Си, по словам Страуструпа, рассматривались языки Модула-2, Ада, Смолток, Mesa и CLU. Но предпочтение было отдано Си)

Язык Си++ является языком объектно-ориентированного программирования.

Идея использования программных объектов развивалась разными исследователями в течение целого ряда лет. Одним из первых языков такого типа является Симула-67. Утвердило же свой статус объектно-ориентированное программирование в созданном Аланом Кеем языке Смолток (Smalltalk), который появился в 1972 году. "Объект есть логическая единица, которая содержит данные и правила

(методы) обработки этих данных. В языке Си++ в качестве таких правил обработки выступают функции, т.е. объект здесь объединяет в себе данные и функции, обрабатывающие эти данные".



Объектно-ориентированный язык создает так называемое программное окружение в виде множества независимых объектов, каждый из которых отличается своими свойствами и способами взаимодействия с другими объектами. Каждый объект тут в каком-то смысле подобен компьютеру: он способен получать сообщения от других объектов и, в свою очередь, посылать сообщения, запоминать информацию и обрабатывать ее определенным образом. Описывая структуру обмена сообщениями между объектами, программист задает совокупность операций, которые и образуют программу. Можно использовать объекты для решения задач, не понимая механизма работы объектов, т.е. рассматривать их как своего рода "черный ящик". Объектно-ориентированный язык скрывает многие детали функционирования системы внутри самих объектов. Но при необходимости программист может изменять детали внутреннего устройства объектов, формируя тем самым другие объекты, нужные для решения новых задач.

Каждый объект в объектно-ориентированном окружении обладает своими собственными характеристиками, однако объекты, имеющие одинаковые свойства и ведущие себя сходным образом, объединяются в группы, или классы.



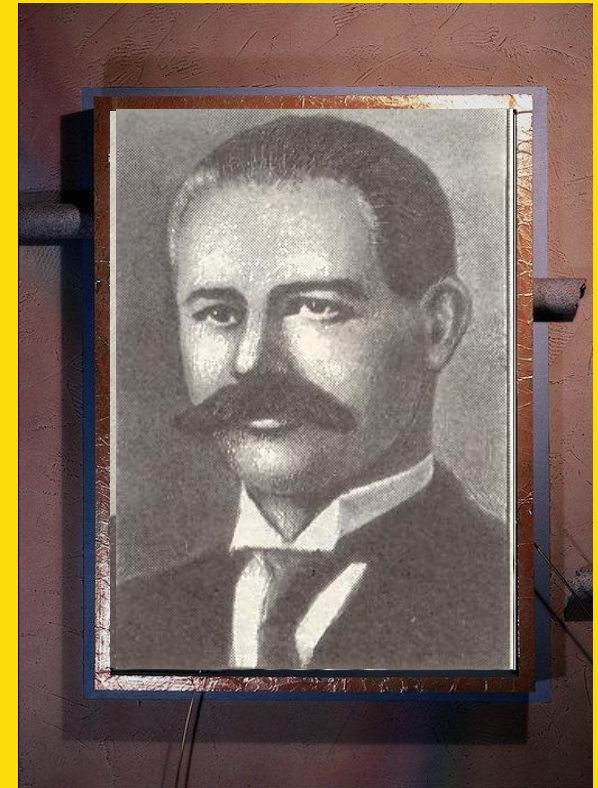
"Понятие класса является центральным понятием языка Си++. Вообще класс — это определенный программистом тип, к которому относятся любые структуры данных, определенные посредством ключевых слов *class*, *struct* и *union*. При определении класса расширяются возможности языка Си++, так как в дальнейшем этот класс можно использовать как любой другой тип Си++. Как только класс определен, можно создавать объекты этого класса, каждый из которых имеет члены, определенные в классе".

Пример класса — графическая точка на экране. Данными-членами тут являются ее координаты, а также цвет. Функции-члены могут обеспечивать инициализацию точки, определение ее цвета, задание нового цвета. Вообще говоря, "в принципе возможен объектно-ориентированный подход к проектированию даже в традиционных языках программирования, например, таких, как Си; однако использование объектно-ориентированных языков типа Си++ позволяет существенно облегчить работу...". Сейчас Страуструп возглавляет отдел программных исследований в лаборатории AT&T Labs (корпорация AT&T была в свое время реорганизована). Область его научных интересов охватывает технологии распределенных систем и операционных систем, сферу имитационного моделирования, проектирования и программирования.



Герман Холлерит

Основателем счетно-перфорационной техники, прямой предшественницы релейных машин, является американец немецкого происхождения Герман Холлерит (1860 — 1929), который в 1890 году, занимаясь обработкой данных переписи населения, впервые построил ручной перфоратор для нанесения цифровой информации на перфокарты. Он также ввел механическую сортировку для раскладки перфокарт в зависимости от места пробивок и построил суммирующую машину, названную табулятором. Она "прощупывала" отверстия на перфокартах, воспринимая их как соответствующие числа, и подсчитывала эти числа.



В 1900 году стала применяться автоматическая загрузка перфокарт, а еще через тринадцать лет появились первые печатающие и накапливающие табуляторы. У нас в стране в 1929 году был создан первый в мире итоговый перфоратор, а в 1934 — 1935 годах были разработаны принципы использования фотоэлемента в перфорационных устройствах.

Обычно использовались целые комплексы счетно-перфорационных или, как их еще называли, счетно-аналитических машин, причем простейшим из применявшихся у нас подобных комплексов являлся так называемый комплект счетно-аналитических машин.

Отверстия на перфокартах пробивались посредством перфоратора. Достаточно опытному оператору удавалось обработать с его помощью приблизительно 230 карт в час.

Поскольку оператор мог допустить ошибки, после пробивки данных перфокарты следовало проверять. Для проверки употреблялась специальная машина — контрольник. При работе на контрольнике оператор производил те же операции, что и на перфораторе. Если отверстия на карте были пробиты правильно, то она беспрепятственно проходила через воспринимающее (считывающее) устройство контрольника. Если же при пробивке была допущена ошибка, то карта останавливалась, причем колонка с ошибкой оказывалась над воспринимающим устройством. В этом случае карта вынималась, а необходимые данные пробивались повторно на новой карте. Оператор мог проверить примерно 250 карт в час.



Иногда перед вычислениями требовалось произвести сортировку данных, пробитых на перфокартах, по тем или иным признакам. Для этого применялась специальная машина — сортировка. Колода перфокарт вкладывалась в магазин подачи машины, после чего следовало с помощью рукоятки установить, по какой колонке сортировать карты, а затем — нажать пусковую кнопку. Карты начинали подаваться под воспринимающее устройство машины и складываться в ее карманы. Таких карманов было тринадцать (двенадцать соответствовали возможным вариантам пробивок в колонке, тринадцатый предназначался для карт, не имеющих там пробивок). За час машина была способна обработать примерно 24 000 картоколонок.

Далее карты поступали на счетно-записывающую машину — табулятор, который автоматически считывал исходные данные, пробитые на перфокартах. Табулятор выполнял расчеты и печатал результаты в соответствии с настройкой, осуществлявшейся при помощи соединений на коммутационной доске и выключателей. Табулятор Т-5 работал со скоростью приблизительно от 6000 до 9000 карт в час — в зависимости от того, какая информация выводилась на печать.

Часто результаты, полученные с помощью табулятора, требовалось вновь использовать при работе на нем же. В таких случаях представлялось целесообразным не только печатать результаты, но и выводить их на перфокарты. Это осуществлялось посредством итогового перфоратора, работающего совместно с табулятором.



Перенос данных с одних перфокарт на другие (в том числе с добавлением новых пробивок, исключением ненужных данных и т.п.) осуществлялся посредством перфоратора-репродуктора, работающего со скоростью примерно 6000 карт в час.

На табуляторе было трудно выполнять умножение. И здесь положение могла исправить еще одна машина — умножающий перфоратор, способный также выполнять сложение и вычитание. Результаты, полученные с помощью умножающего перфоратора, выводились на перфокарты, а скорость вычислений зависела тут от характера работы. (При перемножении шестизначных чисел и вывода на перфокарты результата с двенадцатью знаками можно было произвести около 1200 действий в час.)

Счетно-аналитические машины являлись электромеханическими устройствами и потому работали достаточно медленно, причем комплект не содержал устройств, выполняющих деление.

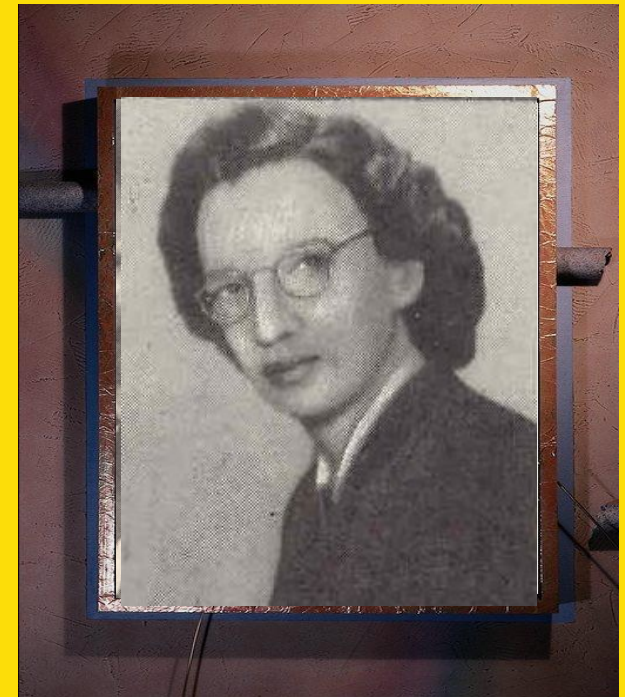
Приспособление таких машин для решения сложных математических задач было сопряжено с большими трудностями.

Грейс Хоппер

Если есть кто-то в компьютерном мире вообще и в истории языков программирования в частности, кто не нуждается в представлении, - это, конечно, капитан Грейс Хоппер...

Джин Сэммит

Слова, вынесенные в эпиграф, принадлежат Джин Сэммит, одной из известных ученых в области программирования, автору многих книг по истории программирования. Для зарубежных специалистов по информатике эти слова не удивительны, однако нашим читателям имя Грейс Хоппер малоизвестно. Более того, в некоторых переводах зарубежных книг по программированию ее фамилия (если она упоминалась) преподносилась читателю в мужском роде.



Грейс Мюррей Хоппер родилась в Нью-Йорке 9 декабря 1906 г. в семье страхового агента Уолтера Флетчера. С детства она увлекалась не куклами, а различными механическими устройствами — «машинками», как она их называла. В 1928 г. она закончила с отличием колледж Вассара в Пугкигси и продолжила свое образование в Йельском университете. Здесь она в 1930 г. получила степень магистра и в 1934г.— доктора философии по математике . Затем, вернувшись в Вассар, она преподавала математику. В 1941 г. Г. Хоппер прослушала несколько курсов в университете Нью-Йорка.

В декабре 1943 г., в самый разгар второй мировой войны, она приняла присягу и в мае 1944 г. поступила на службу в ВМФ США. Получив в июне 1944 г. звание младшего лейтенанта, Г. Хоппер была включена в отдел ВМФ, который занимался разработкой программ для компьютера ASCC, детища Говарда Айкена.

Она вспоминает, что стала «третьим в мире программистом первого в мире большого цифрового компьютера». И далее продолжает: «В те дни мы не назывались программистами. Это слово еще не дошло до нас из Англии, мы были кодировщиками». Однако самый маленький морской офицер и ее коллеги — мичманы заложили в те годы фундамент здания с названием «программирование». Г Хоппер доказала, чего она стоит как программист. Впоследствии она писала: «Я имела то преимущество, что изучала как технику, так и математику и знала, как работает машина от начала и до конца. Конечно же я была счастлива. Заканчивая в 1928 г. колледж, я и не подозревала, что в 1944 г. появится компьютер». Вместе с Говардом Айкеном она опубликовала первое описание компьютера «Марк-1»



При работе на компьютере «Марк-1» Г. Хоппер и ее группой впервые были введены некоторые приемы, которые стали в дальнейшем широко использоваться в программистской практике. В частности, первыми инструментами, которые сэкономили труд программистов, были подпрограммы. Так вот, в августе 1944 г. для «Марк-1» была написана первая подпрограмма для вычисления $\sin x$. После нее последовали другие. «Мы стали писать подпрограммы», вспоминала Г. Хоппер, хотя термин «подпрограмма» был введен позже (Морис Уилкс в Англии. — А. Ч.).

И еще одно фундаментальное понятие техники программирования впервые ввели Г. Хоппер и ее группа — «отладка». Причем случилось это при курьезных обстоятельствах. Жарким летним днем 1945 г. неожиданно произошла остановка компьютера «Марк-1». Обнаружилась неисправность одного реле, контакты которого были заблокированы мотыльком, неизвестно каким образом туда попавшим. Вспоминает Г. Хоппер: «Когда к нам зашел офицер, чтобы узнать, чем мы занимаемся, мы ответили, что очищаем компьютер от насекомых (debuging). Термин «debuging» (отладка) с тех пор прижился и стал использоваться для обозначения поиска неисправностей в компьютере, особенно в программном обеспечении». «Я столкнулась с проблемой жизни в двух мирах,— вспоминала Г. Хоппер, — вероятно, это было одной из причин, побудивших меня по возможности избавиться от восьмеричных чисел».



В 1949 г. Г. Хоппер перешла на работу старшим математиком во вновь образовавшуюся фирму Маучли — Эккперта (создателей электронного компьютера ENIAC), где приняла участие в разработке программных средств коммерческого компьютера UNIVAC-1. Фирма располагалась в помещении старой фабрики на севере Филадельфии. «Когда машина UNIVAC-1 не работала, мы, бывало, говорили, что не мешало бы ее выбросить за забор, на ту сторону, где была свалка, а самим махнуть на другую сторону — на кладбище», — вспоминала те годы Г. Хоппер.

Для облегчения процесса составления программ на машинном языке вместо записи команд в двоичной системе счисления (длинные серии единиц и нулей) в то время использовалась восьмеричная система счисления, запись в которой, как известно, в три раза короче двоичной, а взаимный переход от одной к другой формален. «Все в фирме были убеждены, что единственный способ писать эффективные программы — это пользоваться восьмеричной системой». Г. Хоппер тоже освоила эту систему: научилась складывать, вычитать, умножать и делить в ней. Однако погружение в мир восьмеричной арифметики чуть было не стоило ей неприятностей в жизни, когда она подводила баланс своего банковского счета. (Выручил брат-банкир, напомнив ей, что банки все-таки работают с десятичной системой.)



Надо заметить, что не одну Г. Хоппер занимала проблема облегчения труда программистов, работающих в машинных кодах. В 1949 г. до ее прихода в фирму Джон Маучли создал систему под названием «Short Code», которая являлась примитивным языком программирования высокого уровня. В ней программист записывал решаемую задачу в виде математических формул, а затем, используя таблицу перевода символ за символом, преобразовывал эти формулы в двухлитерные коды. В дальнейшем специальная программа компьютера превращала эти коды в двоичный машинный код. Система, разработанная Дж. Маучли, была по существу одним из первых примитивных интерпретаторов, т.е. в последующие годы одним из двух типов транслятора. Эта система оказала большое влияние на Г. Хоппер. «Short Code» был первым шагом к чему-то такому, что давало программисту возможность писать программы на языке, отличном от машинного», — писала Г.Хоппер. Второй шаг предстояло сделать ей.

Немного отвлечемся и напомним читателю о двух типах трансляторов с позиций сегодняшнего дня.

Первый тип, упоминаемый выше,— интерпретатор в каждый момент времени транслирует строку за строкой программу, написанную на языке высокого уровня, в программу на машинном языке и одновременно ее выполняет.



Второй тип — компилятор транслирует весь текст программы, написанной на языке высокого уровня, в машинный код в ходе одного непрерывного процесса. При этом создается полная программа в машинных кодах, которую затем можно выполнить без участия компилятора. Работа компилятора состоит из трех этапов: сначала текст программы создается при помощи редактора текстов или другой программы текстовой обработки, затем текст компилируется, наконец скомпилированная программа выполняется. Разница между компилятором и интерпретатором аналогична разнице между переводчиком литературного произведения и переводчиком устной речи.

Итак, в 1951 г. Г. Хоппер создала первый компилятор и ею же был введен сам этот термин. Компилятор Г. Хоппер осуществлял функцию объединения и в ходе трансляции производил организацию подпрограмм, выделение памяти компьютера, преобразование команд высокого уровня (в то время псевдокодов) в машинные команды. «Подпрограммы находятся в библиотеке (компьютера), а когда вы подбираете материал из библиотеки — это называется компиляцией» — так она объясняла происхождение введенного ею термина. Конечно, между компилятором, созданным Г. Хоппер, и сегодняшними сходство небольшое, но начало было положено. В 1952 г. на одной из первых конференций Ассоциации компьютерной техники в Питтсбурге Г. Хоппер сделала доклад «The Education of a Computer», посвященный компилирующим программам.



В этот период фирму Маучли — Эккерта поглотила компания Remington Rand (как это часто бывает, талантливые изобретатели оказались неумелыми коммерсантами), но Маучли, Эккерт и Хоппер остались работать в новой компании.

Свой первый компилятор Г. Хоппер назвала А-О, затем по мере его расширения и улучшения в последующие годы появились его версии А-1, А-2, А-3 (две последние системы базировались на трехбуквенных псевдокодах для указания математических операций). В 1954 г. группа под руководством Г. Хоппер разработала систему АТ-3, включающую язык программирования и компилятор, которая в дальнейшем получила название МАТЬ-MATIC и продавалась компанией Remington Rand вместе с компьютером UNIVAC-I.

После удачного завершения работ по созданию МАТЬ-MATIC Г. Хоппер и ее группа принялись за разработку нового языка и компилятора, который позволил бы пользователям программировать на языке, близком к обычному английскому. Они выбрали около 30 английских слов типа: add (сложить), compare (сравнить), transfer (переслать) и т. п., а затем разработали компилятор, который транслировал программы, написанные на этом языке, в машинный код.



Для адекватного распознавания компилятором выбранных слов Г. Хоппер придумала способ, который сохранился в операторах будущих языков: каждое слово содержит неповторимую комбинацию из первой и третьей букв, поэтому компилятор при генерации машинного кода может игнорировать все остальные буквы в слове.

Создание «необычного» языка и компилятора, который получил название В-О, было завершено в 1956 г. Необходимость появления подобной системы Г. Хоппер объясняла следующим образом: «Существует много различных людей, которым нужно решать разные задачи. Некоторые из них связаны с обработкой символов, другие — с обработкой слов, и им нужны языки другого типа, а не наши попытки превратить их всех в математиков». Действительно, в дальнейшем язык с успехом использовался людьми, работающими в бизнесе, в конторах фирм и в правительственных учреждениях. В 1958 г. система В-О поступила на рынок с новым названием FLOW-MATIC и компания, в которой трудилась Г. Хоппер, поменяла свое название на Sperry Rand Corp. В отличие от ФОРТРАНа — языка для научных приложений, FLOW-MATIC был первым языком для задач обработки коммерческих данных. Он был проще ФОРТРАНа, и опыт, приобретенный за годы работы с ним, привел к созданию универсального языка КОБОЛ. Справедливости ради необходимо отметить, что и другие языки, но в меньшей степени послужили основой КОБОЛа: язык AIMACO, созданный в ведомстве ВВС США в 1958 г. (в разработке этого языка принимала участие и группа Г. Хоппер), язык COMTRAN, созданный фирмой IBM в 1958 г., язык FACT, разработанный фирмой Honeywell.



Создание универсального, машинно независимого языка высокого уровня для деловых применений и бизнеса стало очередной задачей, за разрешение которой в начале 1959 г. взялась энергичная Грейс Хоппер. Она обратилась за поддержкой в Министерство обороны США, которое вскоре организовало конференцию по языкам программирования. Эта конференция, получившая в дальнейшем название КОДАСИЛ, организовала три группы по разработке нового универсального языка и Исполнительный комитет, осуществлявший контроль за разработкой. К осени 1959 г. создание языка было завершено и он получил имя КОБОЛ (COBOL— Common Business Oriented Language). Одним из основных консультантов при создании языка была Грейс Мюррей Хоппер.

Курьезный случай (и в то же время неожиданная реклама КОБОЛа) произошел с Г. Хоппер во время ее посещения компьютерного центра в Японии. Гостеприимные японцы никак не могли понять ее просьбу, пока она не обратилась к операторам КОБОЛа. «Move (переместить) — сказала она, показывая на себя, — goto (перейти к) отель «Осака». После чего японцы сразу отвезли ее в гостиницу.

В 1967 г. Хоппер вернулась на действительную службу в ВМФ и к 1980 г. стала капитаном Управления вычислительной техники в Вашингтоне. В 1983 г. она получила звание капитана первого ранга, а в ноябре 1985 г. — звание контр - адмирала. В 1982 — 1986 гг. она была старейшим действующим офицером американской армии. После отставки из армии в 1986 г. она еще некоторое время работала консультантом фирмы DEC

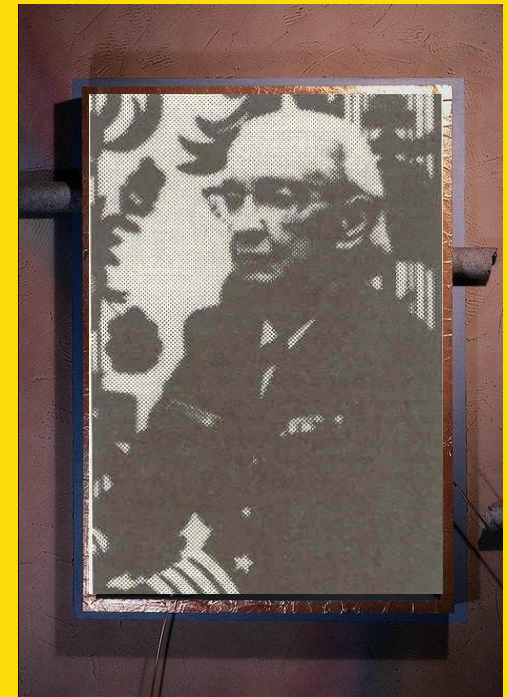


В начале января 1992 г. из Америки пришло сообщение (Newsbytes News Network, January, 3, 1992): «Недалеко от Вашингтона в возрасте 85 лет после продолжительной болезни скончалась Грейс

Мюррей Хоппер, одна из изобретателей языка КОБОЛ».

Хоппер начала работать с компьютерами с конца 40-х гг. Она участвовала в работах над «Марк-1», «Марк-II», «Марк- 111» и UNIVAC-1. Ее часто называли Мамой КОБОЛа и Бабушкой КОБОЛа...»

Заслуги Г. Хоппер в программировании оценены по достоинству. Более 40 университетов и колледжей присвоили ей различные звания. Г. Хоппер была обладательницей первой награды по информатике «Человек года», присуждаемой Ассоциацией управления и обработки данных. На церемонии отставки в 1986 г. в Бостоне на борту авианосца морской министр Джон Ф. Леман младший представил адмирала Грейс Хоппер к правительственной награде.



Автор одной из первых книг по проектированию компиляторов (1971) профессор Корнеллского университета Дэвид Грис, оценивая значимость работ Г. Хоппер, пишет, что в ее наследии имеются «наметки многих инструментов и способов в отношении компиляторов, которые в настоящее время считаются общепризнанными: библиотека подпрограмм, дополненная спецификациями, трансляция формул в их элементарные компоненты; первичные функции компилятора; подпрограммный интерфейс и относительная адресация; компоновщик загрузчик и оптимизация текста программы». И далее продолжает: «Она одной из первых поняла, что именно программные средства, а не аппаратные потребуют большего числа затрат. Она предвидела тот день, когда для математиков уже не будет необходимым знание системы команд компьютера... Хоппер предвосхитила символьную обработку и хорошо понимала, что произойдет даже в отношении искусственного интеллекта (хотя в то время этого термина еще не существовало. — А. Ч.), когда говорила: «В настоящее время задача состоит в том, чтобы в наиболее возможной степени заменить человеческий мозг на электронный цифровой компьютер».

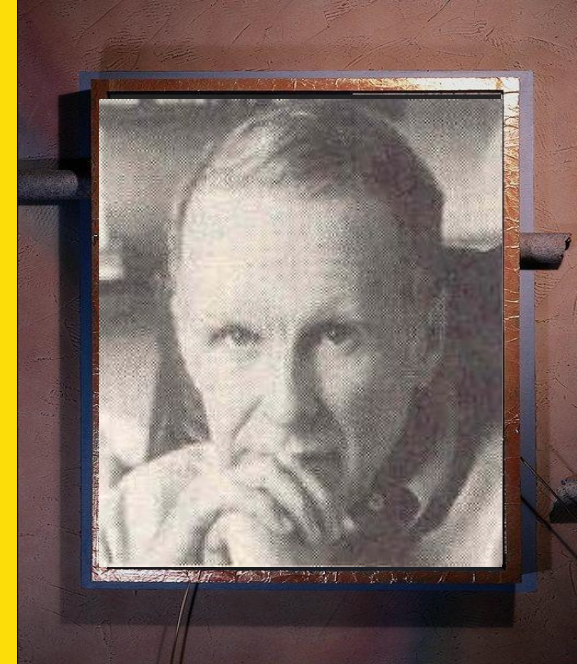


Джон Бекус

Создатели Фортрана, работавшие в фирме IBM, не подозревали, что этот язык получит такое признание. Когда в 1954 году они приступили к работе, информатика развивалась стихийно, и трудно было что-то планировать. Руководителем группы разработчиков был Джон Бэкус.

После службы в армии во время Второй мировой войны Бэкус попал в Нью-Йорк, где стал учиться в радиотехнической школе. "Тогда пределом моих стремлений,— признавался он позднее, — было сделать высококачественное звуковоспроизводящее устройство".

Однако один из преподавателей убедил Бэкуса продолжить образование, и он поступил в Колумбийский университет. Так началась одна из самых плодотворных карьер в истории компьютерной техники.



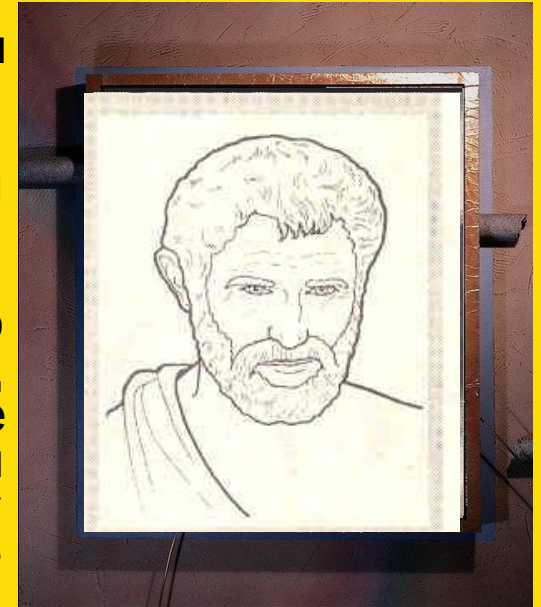
В 1950 году Джон Бэкус, уже имея степень магистра математики, пришел в фирму IBM. Ему было тогда 25 лет. Вскоре он возглавил группу, разрабатывавшую интерпретатор "Быстрый кодировщик" для компьютера IBM 701, а позже принял участие в создании усовершенствованного варианта этой машины, модели IBM 704. В 1953 году Бэкус предложил создать для компьютера IBM 704 язык, позволяющий записывать команды почти в обычной алгебраической форме, и компилятор для него.

Данное предложение быстро нашло поддержку, так как в фирме искали пути, обеспечивающие увеличение сбыта своих компьютеров, и потому стремились сделать их более привлекательными, "дружественными" пользователю. Как правило, инженеры, преподаватели, научные работники не имели ни времени, ни терпения изучать машинные коды или даже язык ассемблера, тем более что это требовало довольно глубокого знания того, как устроен и действует компьютер. Однако все они могли бы использовать язык высокого уровня, если бы его было достаточно легко освоить. Джон Бэкус активно помогал развитию Алгола и попутно разработал специальную систему определений для языков программирования. Например, определяя элемент "цифра", он указывал: "<цифра>:=: 0.|1|2|3|4|5|6|7|8|9|" Вскоре такой способ записи назвали нормальной формой Бэкуса, или БНФ. Потом БНФ долго употреблялась при описании языков высокого уровня.



Аристотель

Математическая логика тесно связана с логикой и обязана ей своим возникновением. Основы логики, науки о законах и формах человеческого мышления, были заложены величайшим древнегреческим философом Аристотелем (384 — 322 гг. до н. э.), который в своих трактатах обстоятельно исследовал терминологию логики, подробно разобрал теорию умозаключений и доказательств, описал ряд логических операций, сформулировал основные законы мышления, в том числе законы противоречия и исключения третьего. Вклад Аристотеля в логику весьма велик, недаром другое ее название — аристотелева логика.



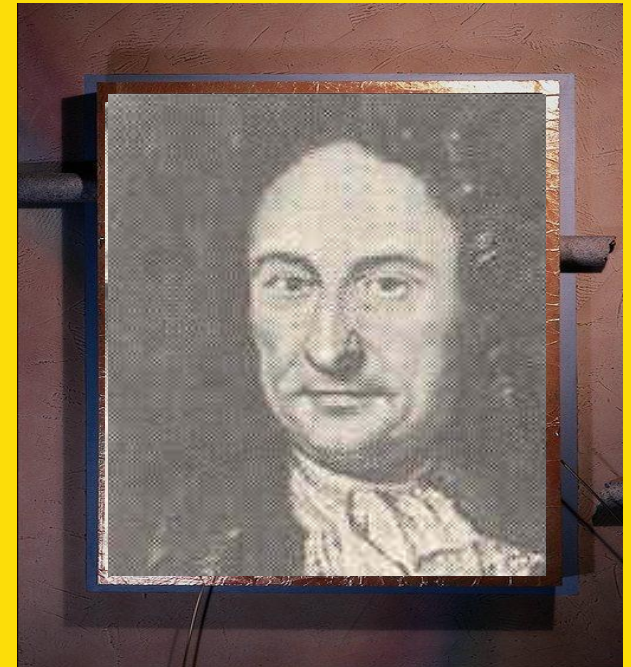
Еще сам Аристотель заметил, что между созданной им наукой и математикой (тогда она именовалась арифметикой) много общего. Он пытался соединить две эти науки, а именно свести размышление, или, вернее, умозаключение, к вычислению на основании исходных положений. В одном из своих трактатов Аристотель вплотную приблизился к одному из разделов математической логики — теории доказательств.

В дальнейшем многие философы и математики развивали отдельные положения логики и иногда даже намечали контуры современного исчисления высказываний.

Г.В. Лейбниц

Ближе всех к созданию математической логики подошел уже во второй половине XVII века выдающийся немецкий ученый Готфрид Вильгельм Лейбниц (1646— 1716), указавший пути для перевода логики "из словесного царства, полного не- определенностей, в царство математики, где отношения между объектами или высказываниями определяются совершенно точно". Лейбниц надеялся даже, что в будущем философы, вместо того чтобы бесплодно спорить, станут брать бумагу и вычислять, кто из них прав. При этом в своих работах Лейбниц затрагивал и двоичную систему счисления.

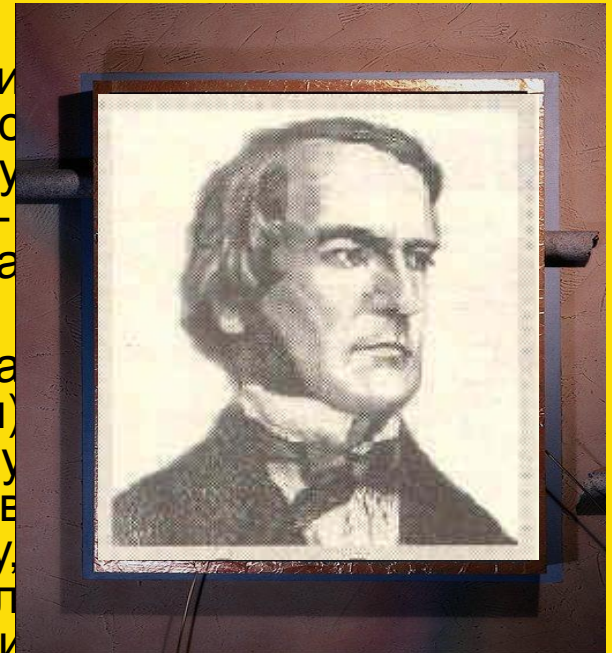
Следует отметить, что идея использования двух символов для кодирования информации очень стара. Австралийские аборигены считали двойками, некоторые племена охотников-сборщиков новой Гвинеи и южной Америки тоже пользовались двоичной системой счета. В некоторых африканских племенах передают сообщения с помощью барабанов в виде комбинаций звонких и глухих ударов. Знакомый всем пример двухсимвольного кодирования— азбука Морзе, где буквы алфавита представлены определенными сочетаниями точек и тире.



Джордж Буль

После Лейбница исследования в этой области вели многие выдающиеся ученые, однако настоящий успех пришел здесь к английскому математику-самоучке Джорджу Булю (1815 — 1864), целеустремленность которого не знала границ.

Материальное положение родителей Джорджа (отец которого был сапожным мастером) позволило ему окончить лишь начальную школу для бедняков. Спустя какое-то время Буль, сменив несколько профессий, открыл маленькую школу, где сам преподавал. Он много времени уделял самообразованию и вскоре увлекся идеями символической логики.



В 1847 году Буль опубликовал статью "Математический анализ логики, или Опыт исчисления дедуктивных умозаключений", а в 1854 году появился главный его труд "Исследование законов мышления, на которых основаны математические теории логики и вероятностей".



Буль изобрел своеобразную алгебру— систему обозначений и правил, применимую ко всевозможным объектам, от чисел и букв до предложений. Пользуясь этой системой, он мог закодировать высказывания (утверждения, истинность или ложность которых требовалось доказать) с помощью символов своего языка, а затем манипулировать ими, подобно тому, как в математике манипулируют числами. Основными операциями булевой алгебры являются конъюнкция (И), дизъюнкция (ИЛИ) и отрицание (НЕ).

Через некоторое время стало понятно, что система Буля хорошо подходит для описания электрических переключательных схем. Ток в цепи может либо протекать, либо отсутствовать, подобно тому, как утверждение может быть либо истинным, либо ложным. А еще несколько десятилетий спустя, уже в XX столетии, ученые объединили созданный Джорджем Булем математический аппарат с двоичной системой счисления, заложив тем самым основы для разработки цифрового электронного компьютера.

Отдельные положения работ Буля в той или иной мере затрагивались и до, и после него другими математиками и логиками. Однако сегодня в данной области именно труды Джорджа Буля причисляются к математической классике, а сам он по праву считается основателем математической логики и тем более важнейших ее разделов — алгебры логики (булевой алгебры) и алгебры высказываний.

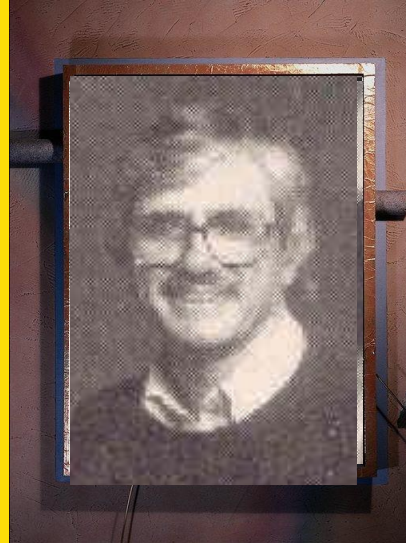


Изобретателями первых логарифмических линеек являются англичане — математик и педагог Уильям Отред и учитель математики Ричард Деламейн. Сын священника, Уильям Отред учился сначала в Итоне, а затем в Кембриджском королевском колледже, специализируясь в области математики. В 1595 году Отред получил первую ученую степень и вошел в совет колледжа. Ему было тогда чуть больше 20 лет. Позже Отред стал совмещать занятия математикой с изучением богословия и в 1603 году стал священником. Вскоре он получил приход в Олбьюри, близ Лондона, где и прожил большую часть жизни. Однако настоящим призванием этого человека являлось преподавание математики. "Все его мысли были сосредоточены на математике, — писал современник Отреда, — и он все время размышлял или чертил линии и фигуры на земле... Его дом был полон юных джентльменов, которые приезжали отовсюду, чтобы поучиться у него". Плату за обучение Отред не брал, хотя не был богатым. В 1631 году он написал для своих учеников учебник арифметики и алгебры "Ключ математики", пользовавшийся большой популярностью не только в XVII, но и в XVIII столетии. Летом 1630 года у Отреда гостил его ученик и друг, лондонский учитель математики Уильям Форстер. Коллеги разговаривали о математике и, как бы сказали сегодня, о методике ее преподавания. В одной из бесед Отред критически отозвался о шкале Гюнтера, отметив, что манипулирование двумя циркулями отнимает много времени и дает низкую точность.

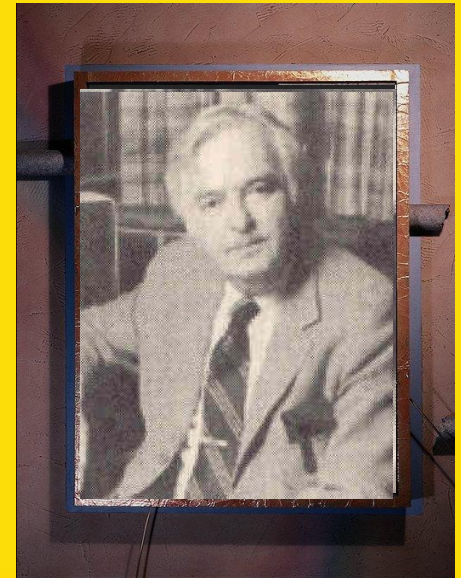


Томас Курц и Джон Кемени

История создания языка Бейсик (BASIC: *Beginner's All purpose Symbolic Instruction Code*) связана с поисками решения проблемы обучения основам программирования всех студентов (независимо от их специализации), поставленной сотрудниками математического факультета Дартмундского колледжа (США) Джоном Кемени и Томасом Курцем. Тогда данное предложение уже само по себе являлось необычным, однако Кемени и Курц намеревались пойти еще дальше.



Томас Курц



Джон Кемени



Бейсик прост, но на этом языке можно писать сложные программы. "Язык программирования Бейсик во многом является антиподом современных языков программирования, например, языка Паскаль. В отличие от последнего, в Бейсик включались не столько "правильные", сколько "удобные" конструкции и средства.

В частности, в нем широко используются разного рода умолчания, что считается совершенно недопустимым в современных языках. В результате процесс написания простой программы на Бейсике для человека, не склонного к педантичности, представляется гораздо более естественным" – Б. Уолш – английский программист, автор книги «Программирование на Бейсике».

В отличие от преподавателей из других учебных заведений, которых устраивала методика обучения студентов приемам работы с компьютером путем чтения лекций, они хотели, чтобы студенты осваивали машину, создавая работающие программы. И здесь прежде всего следовало избавиться от пакетного режима, при котором колоды перфокарт с текстами программ и исходными данными передавались операторам вычислительного центра, а потом приходилось много часов (или даже дней) ждать результатов.



Кемени и Курц решили использовать систему разделения времени, впервые предложенную в Массачусеттском технологическом институте. Система предполагала подключение к одному главному процессору нескольких терминалов. Специальные программы позволяли процессору обращаться по очереди к каждому терминалу, выполняя при этом небольшую часть каждого задания за определенное время, измеряемое обычно миллисекундами.

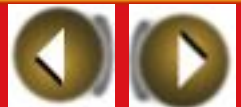
Процессор обрабатывал информацию очень быстро, и потому у каждого программиста создавалось впечатление, что машина обслуживает только его одного. Следующая задача заключалась в создании языка программирования, который легко бы осваивали студенты. Предполагалось, что в языке должны употребляться простые английские слова, такие, как SAVE (сохранить), PRINT (печатать), RUN (выполнить), причем его операторы должны быть сходны с операторами языка Фортран, но отличаться от них большей простотой. Некоторые предложения Кемени и Курца являлись совсем новыми. Примером тут может служить оператор INPUT. Поскольку ожидалось, что студенты будут сами вводить тексты программ (а не передавать колоды перфокарт операторам), то и в языке предусматривалась возможность работы в диалоговом режиме.



Первая программа на Бейсике с использованием системы разделения времени заработала в мае 1964 года.

Для нового языка было выбрано удачное название — универсальная система символического кодирования для начинающих. Действительно, с его помощью студент-социолог мог написать программу обработки данных статистического обследования, студент-филолог, изучающий французский язык, — создать программу спряжения правильных глаголов, а будущий инженер-строитель — подготовить программу для расчета характеристик прочности моста в зависимости от используемых материалов. В начале 1970-х годов появились первые микрокомпьютеры.

Первые версии интерпретатора языка Бейсик для микрокомпьютера "Альтаир-8800" были созданы первокурсником Биллом Гейтсом и его приятелем Полом Алленом и проданы изготовителю этой машины, фирме MITS (*Micro Instrumentation and Telemetry Systems* — микроприборы и телеметрические системы). Вскоре Гейтс и Аллен основали фирму Microsoft, которая внесла существенный вклад в развитие языка и способствовала его реализации на микрокомпьютерах, в том числе на персональных компьютерах.



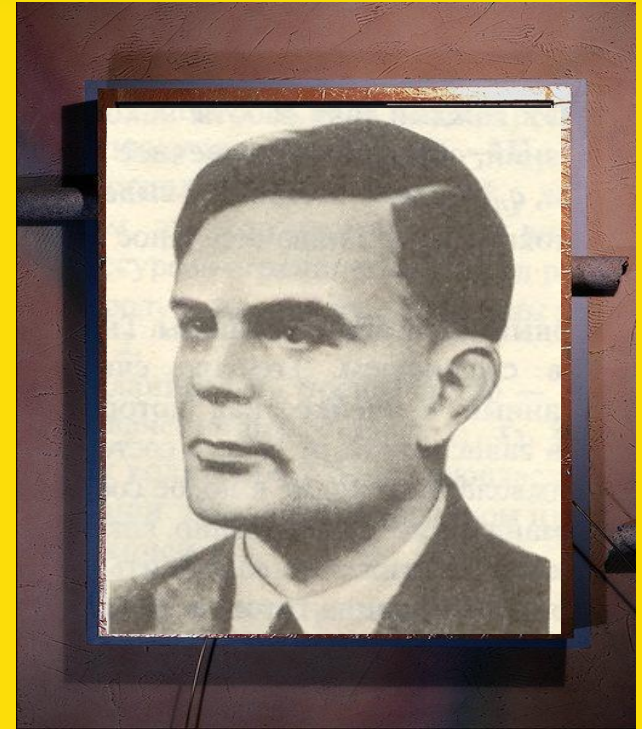
У нас Бейсик впервые был реализован в 1969 — 1970 годах на вычислительной машине М-20 (в пакетном режиме) сотрудниками Нижегородского (тогда Горьковского) университета под руководством Юлия Лазаревича Кеткова. Он же подготовил первую доступную книгу на русском языке, посвященную программированию на Бейсике. В свое время Бейсик имел необычайную популярность в студенческой среде, в том числе и в нашей стране. В 1975 году одному из студенческих строительных отрядов Нижегородского университета присвоили название BASIC. Вскоре после этого слово BASIC можно было встретить на заборах, ограждающих строительные площадки города и области. Вряд ли какой-нибудь другой язык программирования удостоивался подобных почестей.

За время своего существования язык Бейсик прошел большой путь (и имеет огромное количество "диалектов"). "Своей популярностью на микрокомпьютерах BASIC обязан реализациям Tine BASIC и Microsoft BASIC. Среди других известных реализаций: CBASIC (Compiled BASIC), Integer and Applesoft BASIC (для компьютеров Apple 11), BASICA и GW-BASIC (для IBM PC и совместимых с ними компьютеров), TrueBASIC (пересмотренная редакция авторов-разработчиков Кемени и Курца) и Microsoft QuickBASIC". Достаточно широко известен также Turbo BASIC (фирмы Borland). Если в 1960-х годах этот язык мог рассчитывать только на внимание начинающих любителей, то постепенно положение менялось и позже вместо BASIC все чаще стали писать Basic (базовый), как бы настаивая на другой трактовке названия. А еще Бейсик называют "питоном, пожирающим все, что лежит на его пути". Так, в 1980-х годах он закончил "переваривать" язык Паскаль со всеми его управляющими структурами...

Алан Тьюринг

Алан Тьюринг окончил Кембриджский университет в 1935 г. В это время внимание многих математиков было приковано к работам, посвященным тем логическим основам, на которые опирается эта самая строгая из всех известных человечеству наук. Фундаментальные понятия математики подвергались тщательному изучению и обоснованию. Особенно те, которые традиционно использовались без уточнения их смысла, на интуитивном уровне.

Алгоритм — одно из таких понятий. И вот настал момент, когда формализация его стала необходимой. А Тьюрингу удалось дать определение понятия алгоритм .



В качестве уточнения понятия он предложил некоторую гипотетическую конструкцию— машину, получившую вскоре название "машина Тьюринга". Он сделал свое изобретение в 1937 г. В то время Тьюринг стажировался в Принстонском университете в США. В Англию он вернулся уже знаменитым ученым. Долгие годы А.Тьюринг работал в Манчестерском университете. Его конструкция начала свою вторую жизнь после появления ЭВМ, для которых понятие алгоритма — центральное.

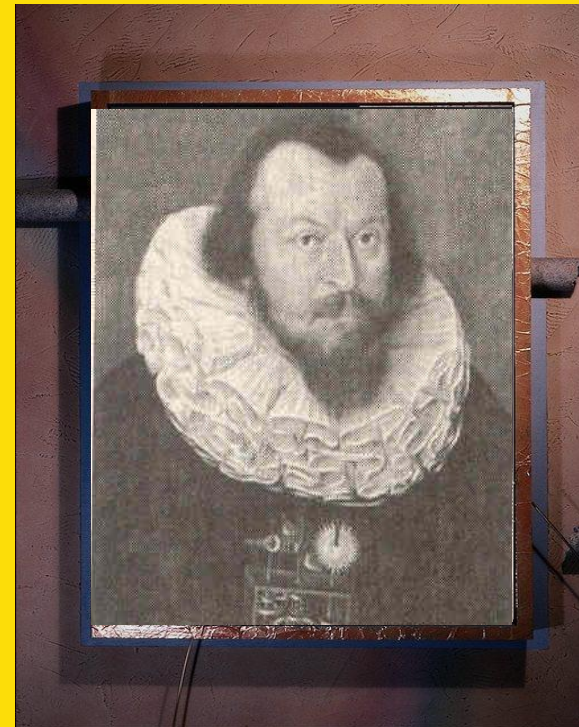
В 1951 г. Тьюринг стал членом Королевского общества Великобритании. В этот период он активно занимался теоретическими проблемами программирования, строил интерпретаторы для новых ЭВМ. Тьюринг ввел научный оборот понятие стека и сделал значительный вклад в технологию программирования. В последние годы жизни А. Тьюринг живо интересовался математической биологией, на которую возлагал большие надежды.



Вильгельм Шиккард

Более 300 лет считалось, что автором первой счетной машины является Блез Паскаль. И вот в 1957 году директор Кеплеровского научного центра Франц Гаммер сделал в Германии на семинаре по истории математики сенсационный доклад, из которого следовало, что проект первой счетной машины появился, как минимум на два десятилетия раньше "паскалева колеса", а сама машина была изготовлена в середине 1623 года.

Работая в городской библиотеке Штутгарта, Гаммер обнаружил фотокопию эскиза неизвестного ранее счетного устройства. Гаммеру удалось установить, что этот эскиз есть не что иное, как отсутствовавшее приложение к опубликованному ранее письму Иоганну Кеплеру профессора университета в Тюбингене Вильгельма Шиккарда (от 25 февраля 1624 года), где Шиккард, ссылаясь на чертеж, описывал изобретенную им счетную машину.



Машина содержала суммирующее и множительное устройства, а также механизм для записи промежуточных результатов. Первый блок-шестиразрядная суммирующая машина — представлял собой соединение зубчатых передач. На каждой оси имелись шестерня с десятью зубцами и вспомогательное однозубое колесо — палец. Палец служил для того, чтобы передавать единицу в следующий разряд. При вычитании шестеренки следовало вращать в обратную сторону. Контроль хода вычислений можно было вести при помощи специальных окошек, где появлялись цифры. Для перемножения использовалось устройство, чью главную часть составляли шесть осей с "навернутыми" на них таблицами умножения. После доклада Гаммера исследователи вспомнили и про другое письмо Шиккарда Кеплеру — от 20 сентября 1623 года, — на которое прежде не обращали должного внимания. В нем Шиккард сообщал, что осуществил механически то, что Кеплер делал алгебраически, а именно — сконструировал машину, автоматически выполняющую сложение, вычитание, умножение и деление. Шиккард писал, что Кеплер был бы приятно удивлен, если бы увидел, как машина сама накапливает и переносит влево десяток или сотню и как она отнимает то, что держит в "ум однозначного ответа. Из уже упоминавшегося письма Шиккарда от 25 февраля 1624 года следует, что один, еще не совсем готовый, экземпляр машины, который находился у механика Пфистера, сгорел во время пожара.



На вопрос, использовал ли Паскаль при построении своей суммирующей машины идеи Шиккарда, ответ должен быть отрицательным. Документы свидетельствуют о том, что никакие сведения о счетном устройстве 1623 года не дошли до научных кругов Парижа и, значит, Паскаль о нем знать не мог.

Машина Шиккарда была известна, по-видимому, лишь узкому кругу лиц и не повлияла на последующее развитие механизации счета, тем не менее, имя Вильгельма Шиккарда находится в одном ряду с именами Блеза Паскаля, Готфрида Вильгельма Лейбница, Чарльза Бэббиджа, Пафнутия Львовича Чебышева, Германа Холлерита и других выдающихся изобретателей счетных устройств XVII — XIX столетий.

Через десять лет после того, как Франц Гаммер сделал свое сенсационное сообщение, в Национальной библиотеке Мадрида были найдены два тома неопубликованных рукописей Леонардо да Винчи. Среди чертежей обнаружили эскиз тринадцатиразрядного суммирующего устройства с десятизубыми колесами. В целях рекламы оно было собрано фирмой IBM и оказалось вполне работоспособным.

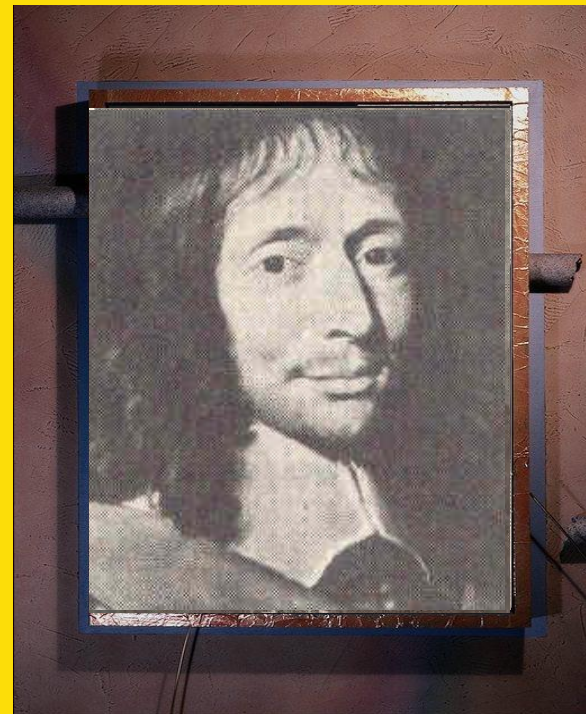
Но являлся ли и данный проект первым?



Блез Паскаль

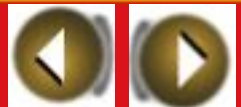
Блез Паскаль (1623 — 1662)— один из самых знаменитых людей в истории человечества. Ему посвящено огромное количество литературы. Каких только сторон жизни и наследия Паскаля не касалось "паскале-ведение"! Конечно, особенно популярен он во Франции.

Паскаль умер, когда ему было 39 лет, но, несмотря на столь короткую жизнь, он вошел в историю как выдающийся математик, физик, философ и писатель. Его именем названы единица давления (*паскаль*) и получивший чрезвычайно широкое распространение язык программирования.



Работы Паскаля охватывают самые разные области. Он является одним из создателей математического анализа, проективной геометрии, теории вероятностей, гидростатики (широко известен *закон Паскаля*, в соответствии с которым изменение давления в покоящейся жидкости передается в остальные ее точки без изменений), создателем механического счетного устройства — "паскалева колеса" — как говорили современники. Паскаль продемонстрировал, что воздух обладает упругостью, и доказал, что он имеет вес, открыл, что показания барометра зависят от влажности и температуры воздуха и потому его можно использовать для предсказания погоды. Философские мысли Паскаля оказывали влияние на многих знаменитых людей и, в частности, на русских писателей — И.С. Тургенева, Ф.М. Достоевского, Л.Н. Толстого.

Некоторые из практических достижений Паскаля удостоились высшего отличия — сегодня мало кто знает имя их автора. Например, сейчас очень немногие скажут, что самая обыкновенная тачка — это изобретение Блеза Паскаля. Ему же принадлежит идея омнибусов — многоместных конных экипажей с фиксированными маршрутами — первого вида регулярного общедоступного городского транспорта.



Уже в шестнадцатилетнем возрасте Паскаль сформулировал теорему о шестиугольнике, вписанном в коническое сечение (*теорема Паскаля*). Через несколько лет Блез Паскаль создал механическое вычислительное устройство — суммирующую машину, которая позволяла складывать числа в десятичной системе счисления. В этой машине цифры задавались путем соответствующих поворотов дисков (колесиков) с цифровыми делениями, а результат операции можно было прочитать в окошках по одному на каждую цифру. Диски были механически связаны, при сложении учитывался перенос единицы в следующий разряд. Диск единиц был связан с диском десятков, диск десятков — с диском сотен и т.д. Если при повороте диск проходил через ноль, то следующий диск поворачивался на, единицу "вперед". Этот поворот, в свою очередь, мог вызвать поворот на единицу следующего диска и т.д. Главный недостаток суммирующей машины Паскаля состоял в неудобстве выполнения с ее помощью всех операций, кроме сложения. Однако изобретенный Паскалем принцип связанных колес стал основой, на которой строилось большинство вычислительных устройств на протяжении следующих трех столетий.



Но, наверное, самой известной математической работой Блеза Паскаля является трактат об "арифметическом треугольнике", образованном биномиальными коэффициентами (*треугольник Паскаля*), который имеет применение в теории вероятностей. Что же касается замечательной кривой 4-го порядка— *улитки Паскаля*, — то она названа так в честь' отца Блеза Паскаля Этьена, совмещавшего государственную службу с занятиями математикой.

