

Руслан Богатырев

Летопись языков. Паскаль

Источник: Мир ПК, #04/2001

Паскаль... Кто из программистов не знает о таком языке? Его триумф пришелся на 80-е годы — время революции в индустрии персональных компьютеров и период расцвета структурного программирования. Он стал своего рода точкой отсчета начала новой эры языков. Сегодня будущее Паскаля представляется неопределенным, но, несмотря на все изломы судьбы, он продолжает служить верой и правдой своим почитателям.

Рождение Паскаля

Формально юбилей Паскаля пришелся на ноябрь 2000 г., когда исполнилось 30 лет с момента первой официальной публикации описания языка. Но то был недоступный широкой аудитории технический отчет Швейцарского федерального технологического института ETH (Eidgenoessische Technische Hochschule). В самом начале 1971 г. отчет был перепечатан в первом номере журнала *Acta Informatica*. Так что рождение нового языка можно отсчитывать и с этого момента.

Его автор, швейцарский профессор Никлаус Вирт, во многом стал известен именно благодаря появлению Паскаля. Но и последующие его проекты убедительно доказали миру, что ключ к тайнам компьютеров — в гармонии математики, инженерии и программирования. И если грамотно подойти к делу, то можно реализовать языки, операционные системы и даже создать великолепные компьютеры, превосходящие промышленные образцы, силами обычных студентов!

Паскаль нередко противопоставляют другому языку — Си. Но вот что сказал в присутствии Вирта по этому поводу Деннис Ритчи, автор Си (1993): «Я утверждаю, что Паскаль очень близок языку Си. Одни, быть может, этому удивятся, другие — нет... Даже интересно, насколько они близки друг другу. Эти языки больше расходятся в деталях, но в основе своей одинаковы. Если вы взглянете на используемые типы данных, а также на операции над типами, то обнаружите очень большую степень совпадения... И это несмотря на то, что намерения Вирта при создании Паскаля весьма отличались от наших в языке Си. Он создавал язык для обучения, а потому преследовал дидактические цели. И, как я заметил это по Паскалю и по его более поздним языкам, Вирт был во власти своего стремления ограничить выразительные средства как можно сильнее...»

Да, Паскаль, в отличие от Си, не создавался как язык системного программирования. Во имя простоты и эффективности на том уровне понимания программирования Вирт сознательно пошел на заведомое ограничение возможностей языка, прежде всего в отношении общения с внешним миром (ввод-вывод и системно-зависимые средства). И все же думать, что Паскаль — язык исключительно для преподавания, было бы неверно. Послушаем на этот счет мнение самого Вирта (1984): «Утверждалось, что Паскаль был разработан в качестве языка для обучения. Хотя это утверждение справедливо, но его использование при обучении не являлось единственной целью. На самом деле я не верю в успешность применения во время обучения таких инструментов и методик, которые нельзя использовать при решении каких-то практических задач. По сегодняшним меркам Паскаль обладал явными недостатками при программировании больших систем, но 15 лет назад он представлял собой разумный компромисс между тем, что было желательно, и тем, что было эффективно».

Противостояние Си и Паскаля

Языки Паскаль и Си во многом определили пути развития программирования в конце XX столетия. Их противостояние напомнило романтическую эпоху 60-х годов, когда сторонники Фортрана и Алгола-60 также оказались по разные стороны баррикад.

При всем уважении к таким масштабным языкам, как Кобол, ПЛ/1 и Алгол-68, господствовавшим в 70-е годы, столь явно выраженную борьбу людей и идей нам удалось наблюдать лишь в 80-е (Паскаль и Си) и в самом конце 90-х годов — C++ и Java.

Можно назвать как минимум три десятка языков, которые сыграли заметную роль в развитии программирования, но все же именно эти три пары — Алгол-60 и Фортран, Паскаль и Си, Java и C++ — стали самыми яркими, самыми заметными на компьютерном небосклоне.

Это, конечно, чисто субъективные ощущения, но в определенные моменты эволюции можно было наблюдать едва ли не безоговорочное господство того или иного языка, что тут же отражалось на требованиях промышленности, на составлении учебных планов в вузах и университетах. В 70-е годы на пике популярности среди универсальных языков был Фортран, в 80-е — Паскаль, в 90-е годы — C++. К счастью, языки программирования, в отличие от небесных светил не умирают, а продолжают свою жизнь, растворяясь спустя годы и десятилетия в новых себе подобных звездах.

Создатели языков старались делать их в меньшей степени похожими на среду общения между человеком и компьютером и в большей — на упорядоченный набор знаков и символов. Первым держать курс на традиционную и устоявшуюся математическую символику предложил Х. Рутисхаузер (1952), ставший родоначальником идеи языков программирования и одним из авторов языка Алгол-60. Широкое распространение и применение его идеи получили лишь в 1957 г., после того, как корпорация IBM опубликовала описание языка Фортран и реализовала для него компилятор, транслировавший программы в машинный код. По сути, с этого момента и началась эпоха языков программирования.

Если взглянуть на историю языков с точки зрения географии их создания, то несложно сделать вывод, что две ныне самые известные ветви генеалогического древа — Паскаль и Си, давшие жизнь многим своим потомкам, являются яркими выразителями соответственно европейской и американской школ. При этом в Старом Свете архитекторы языков активно используют наработки американских исследовательских центров и лабораторий, а представители Нового Света стараются обогащаться идеями европейских специалистов (Швейцария, Норвегия, Англия, Дания). Европейская школа чаще ратует за чистоту и компактность своих языков программирования, тогда как заокеанская не в силах избежать сумбурного включения новомодных решений. Ну почти как в жизни: чистота и консерватизм французского языка против безудержных заимствований английского в его американском диалекте.

ETH Pascal и P-код

Первый компилятор Паскаля (ETH Pascal) был написан в 1970 г. Первое официальное описание Паскаля с изложением синтаксиса и семантики было опубликовано Виртом в конце 1970 г. Новая версия языка вышла в свет в 1972 г. Тогда же Вирт и его английский коллега Чарльз Энтони Хоар (Charles Anthony Richard Hoare) выпустили аксиоматическое описание Паскаля.

«Наш первый компилятор Паскаля был реализован на семействе компьютеров CDC-6000 и написан на самом Паскале. Никакого PL6000 не потребовалось, и я рассматривал это как существенный шаг вперед», — вспоминает Вирт. Несуществующий язык PL6000 Вирт упомянул в связи с тем, что несколькими годами раньше для эффективной реализации компилятора языка Algol-W на компьютере IBM/360 ему понадобилось разработать низкоуровневый язык PL360. Теперь же в этом необходимости не было.

Почему был выбран именно компьютер фирмы Control Data Corporation? Ответ прост: к тому моменту в ETH активно использовались именно эти машины, а выбор языков на них был более чем скромным: ассемблер или Фортран. Конкурировать с Фортраном в эффективности на этой платформе было непросто, ведь набор команд CDC-6000 создавался с прицелом именно на реализацию Фортрана.

Написание компилятора в 1969 г. Вирт поручил одному своему студенту (Э. Мармье). В тот момент Мармье владел лишь Фортраном и писал компилятор на этом языке с последующей трансляцией его в Паскаль. Затем компилятор Паскаля должен был подвергнуться процессу раскрутки (переписан на самом Паскале). Как отмечает Вирт, выбор Фортрана был серьезной ошибкой. Он не мог адекватно выражать сложные структуры данных компилятора, что все больше запутывало программу.

Вторая попытка создать компилятор началась с того, что он сразу формулировался на самом Паскале (в соответствии с описанием 1970 г.). Синтаксический анализ нового однопроходного компилятора осуществлялся методом рекурсивного спуска. Теперь в команду разработчиков вошли У. Амман, Э. Мармье и Р. Шилд. После того как компилятор был написан на еще не существующем языке, Шилд был отправлен к себе домой на две недели, где все это время он вручную транслировал программу во вспомогательный низкоуровневый язык, доступный на CDC-6000. Итак, в середине 1970 г. компилятор ETH Pascal был готов.

Он был интересен не только тем, что стал одной из первых реализаций языков высокого уровня на самом себе, примерно на два года опередив компилятор Си. В ходе работ над ним в 1973 г. была придумана абстрактная Pascal-машина (P-машина), исполняющая специальный P-код. Чтобы решить проблему переноса компилятора Паскаля на разные платформы, Вирт решил воспользоваться испытанными временем методами интерпретации. Из наиболее известных решений, предшествовавших P-коду, можно назвать реализацию языка Snobol-4 (Р. Грисуолд, 1967), где в качестве кода абстрактной машины использовался язык SIL (System Implementation Language).

Как известно, введение виртуальной (абстрактной) машины Java преподносилось ее разработчиками из Sun Labs едва ли как не фундаментальное открытие в практике языков программирования. Один из учеников Вирта, Михаэль Франц, заметил по этому поводу следующее: «Переносимость Java основана на наличии виртуальной машины, позволяющей легко имитировать большое число архитектур. Идея виртуальной машины была очень популярна уже более двадцати лет назад, хотя впоследствии о ней забыли. Тогда речь шла о Pascal-P — созданной в ETH реализации Паскаля, которая сыграла решающую роль в распространении этого языка. Интересно, что виртуальные машины для Паскаля и Java весьма схожи по архитектуре: в обеих используются однобайтовые инструкции без адресов (операнды помещаются в стек)».

Идеи P-кода нашли применение не только в платформах Java и NET, не только в других языках и машинах баз данных, но и в реализации аппаратных средств. Например, для непосредственного исполнения P-кода в Western Digital в 1979 г. был разработан специальный набор WD9000 P-Engine. В Стэнфордском университете в 1980 г. был создан экспериментальный процессор POMP. Появившаяся в 1978 г. коммерческая реализация Паскаля — UCSD Pascal стала еще более известной, и многие забыли, где же впервые возникли P-код и P-машина. Вот что говорит об этом Вирт: «После того как стало известно о существовании Паскаля, несколько человек попросили нас помочь в его реализации на различных машинах, подчеркивая, что они намерены использовать его для обучения и что быстрое действие для них не имеет первостепенного значения. После этого мы решили создать версию компилятора, которая генерировала бы код для машины нашей собственной конструкции. Позднее этот код стал известен как P-код... Pascal-P оказался исключительно удачным языком для распространения среди большого числа пользователей. И если бы у нас хватило мудрости предвидеть масштабы такого развития событий, то мы приложили бы больше усилий и тщательности при разработке и документировании P-кода».

Для удобства использования в ETH создали P-инструментарий, в который вошли компилятор в P-код и его интерпретатор, причем все это в виде исходных текстов. Одним из получивших такой набор стал Университет Калифорнии в Сан-Диего.

Третья попытка реализации в ETH компилятора Паскаля, ставшего впоследствии известным под названием P2, была предпринята учеником Вирта Урсом Амманом; в 1974 г. она завершилась успехом. Благодаря высокому качеству P2 получил широкое распространение в университетах и компаниях.

UCSD Pascal и начало коммерческого распространения

В 1975 г. профессор Кеннет Боулес, работавший в Университете Калифорнии в Сан-Диего (University of California at San Diego — UCSD), получил из Цюриха P-инструментарий, который вместе с компилятором P2 и был положен в основу UCSD Pascal. В Институте изучения информации калифорнийского университета Боулес вместе со Стефеном Франклином и Альфредом Борком занялись созданием системы программирования и операционной системы на базе Паскаля для микрокомпьютерных архитектур. В UCSD Pascal были внесены изменения как на уровне языка (в плане расширения и использования низкоуровневых вставок кода, в том числе и P-кода), так и на уровне P-машины. Она была переделана. Для эффективности использования Паскаля на разных компьютерах была включена дополнительная возможность генерирования машинного кода для нужной архитектуры сразу после получения P-кода. Весьма значительным усовершенствованием языка в UCSD Pascal стало введение unit-блоков, необходимых для поддержки отдельной компиляции. Впоследствии они были унаследованы в языке Turbo Pascal.

Интересно, что взята эта идея была из спецификаций языка Ада, которые к тому моменту подходили к своему завершению. В то же время сам Вирт из языка Mesa позаимствовал куда более совершенную конструкцию, ввел ее в Модуль-2 и назвал «модулем» (module). Собственно, она и дала название новому языку Вирта Modula-2 (MODULar LAnguage).

За три года команда Боулеса разработала законченную систему, в которую вошли текстовый редактор, файловая система, а также отладчик. По сравнению с тем режимом работы, к которому уже

привыкли пользователи больших машин (ввод программ и данных с перфолент и перфокарт, а также упрощенный терминальный ввод-вывод), это был огромный шаг вперед. Появился прообраз того, что сегодня называют IDE-средой (интегрированной средой разработки). UCSD Pascal стала прародительницей системы Turbo Pascal, добившейся небывалой популярности.

Что касается UCSD Pascal, то эта реализация для Паскаля означала серьезный прорыв: благодаря Боулесу сфера применения языка значительно расширилась, что, в свою очередь, дало заметный импульс развитию микрокомпьютерной революции.

Разработкой UCSD Pascal, сумевшей стать законченным коммерческим продуктом, заинтересовались многие компании. Причем их внимание привлекла и перспективная P-машина, на которую UCSD продал немало лицензий. В качестве одного из примеров можно привести процессор ITS, созданный в Nirpon Electric (1980) и предназначенный для исполнения P-кода UCSD Pascal. Но и сейчас, в эпоху Java, интерес к UCSD-реализации не остыл. Так, в частности, английская компания Cabot International продает новые версии P-машины по лицензии UCSD. Это предлагается в качестве альтернативы для Java в области ТВ-приставок и бытовой электроники (интерактивное цифровое ТВ).

Turbo Pascal

Огромную роль в массовом распространении Паскаля сыграла компания Borland International. На основе идей UCSD Pascal она сумела создать знаменитую Turbo-среду разработки. Это был значительный шаг вперед в облегчении процесса программирования. Удобство визуальных средств в сочетании с тесной интеграцией инструментария стали для сотен тысяч программистов большим подспорьем.

Правда, язык в исполнении Borland теперь уже отдаленно напоминает то, что когда-то носило имя Паскаль. Безжалостные законы рынка раздавили оригинал и родили на свет Turbo-Borland-Object Pascal. В результате не язык стал определять реализацию, а наоборот. Как только все было повернуто с ног на голову, как только деликатные вопросы изменения языка оказались уделом узкого круга лиц внутри одной компании, стало ясно, что рассчитывать на поддержку промышленностью не переносимого на другие платформы языка просто бессмысленно. И это в то время, когда давным-давно существовали новые языки Вирта — Модула-2 и Оберон-2.

Что касается Модулы-2, добившейся утверждения своего ISO-стандарта, то компилятор для этого языка (Turbo Modula-2) не только был создан в компании Borland (для CP/M), что весьма тщательно скрывается, но и поступил в продажу (в Северной Америке и Европе). Однако руководство компании (Филипп Кан) отказалось породить собственного конкурента крайне успешному Turbo Pascal. Возмущенный вице-президент Borland Нильс Йенсен (один из основателей компании) вместе со своей командой разработчиков в 1987 г. ушел из Borland, выкупил права на Turbo Modula-2 и создал компанию JPI (Jensen & Partners International). В Англии ею под маркой TopSpeed была выпущена одна из лучших линеек компиляторов для процессоров семейства x86: Assembler, Modula-2, Pascal, C/C++, Ada.

А что же Turbo Pascal? Язык видоизменялся едва ли не с каждой версией среды разработки! В версии 3.0 появилась встроенная графика, в версии 4.0 — модули, в версии 5.5 — средства объектно-ориентированного программирования. Начиная с версии 7.0 Turbo Pascal был переименован в Borland Pascal, а с появлением Delphi — в Object Pascal.

В отношении языка Turbo Pascal Никлаус Вирт высказывается довольно дипломатично (1993): «Фактический стандарт для Паскаля был определен компанией Borland просто потому, что ими был создан компилятор, который распространялся широко и дешево. Borland расширяла Паскаль на протяжении ряда лет, и некоторые из этих нововведений были не столь хорошо интегрированы в язык, как мне бы того хотелось. Я был менее компромиссным в отношении собственных расширений и потому дал новому проекту иное имя (речь идет о языке Модула-2. — Прим. ред.) Например, то, что Borland назвала UNIT, мы называли MODULE. Однако UNIT — это вставки в виде исходного текста, тогда как MODULE является отдельной единицей компиляции, которая допускает полный контроль типов и компоновку на этапе загрузки». На вопрос о том, какую бы оценку Вирт поставил Модуле-2, если исходить из того, что Фортран заслужил «2», а Паскаль — «5», он ответил: «6». (В школах Швейцарии это наивысший балл.)

В тени марки Borland оказалось имя автора Turbo Pascal датчанина Андерса Хейльсберга. В 1983 г. Borland выкупила лицензию на компилятор Хейльсберга и приняла автора на работу. В ноябре того же года на рынке появился Turbo Pascal 1.0 для CP/M и компьютеров 8086. История сохранила даже данные о компактности той реализации: размер исполняемого .COM-файла компилятора составлял 33

Кбайт, а все файлы занимали 130 Кбайт. В январе 1989 г. Microsoft выпустила Quick Pascal, позиционировавшийся как конкурент Borland Pascal. Но он продержался недолго. Microsoft с Borland поделили рынок, разведя ветви Си и Паскаля еще дальше друг от друга.

Что касается Хейльсберга, то он 13 лет проработал в компании и стал руководителем проекта Delphi. В октябре 1996 г. Андерса Хейльсберга за 3 млн. долл. приобрела корпорация Microsoft, и теперь он ведущий архитектор языков .NET, а также автор языка C#. (При разработке C#, по словам Хейльсберга, рассматривались C++, Java, Модуль-2, Smalltalk.)

Его прежняя компания, вернув себе имя Borland после нескольких лет работы под вывеской Inprise, предпринимает теперь активные шаги по укреплению позиций своего Паскаля в бурно развивающейся ОС Linux. Обеспечивая простоту миграции программ, созданных для Windows на Visual Basic и Object Pascal, в среду Linux под единым инструментарием Kylix, Borland Software намерена нанести серьезный удар по, казалось, незыблемым позициям Microsoft.

Особенности Паскаля и его преемники

Язык Паскаль явился не просто ревизией Algol-W. Он создавался Виртом под воздействием идей Чарльза Энтони Хоара, опубликованных впоследствии в работе «Заметки по структуризации данных» (Hoare C.A.R. Notes on Data Structuring Academic Press, 1972). Вклад английского ученого в разработку языка был столь значителен, что его смело можно назвать крестным отцом Паскаля.

От языка Паскаль принято отсчитывать эпоху структурного программирования. А все началось с того, что известный голландский специалист Эдсгер Дейкстра опубликовал статью «Структурное программирование» (Dijkstra E.W. Structured Programming // NATO Science Committee, 1969). В ней он предложил ограничить логику управления программы всего тремя формами: следованием (sequence), ветвлением (selection) и циклом (iteration). Из этого вытекало, что в языках Алгол и ПЛ/1 оператор безусловного перехода (goto) был уже попросту не нужен. Вирт, правда, не рискнул изъять его из Паскаля. Но главное было в другом: структурное программирование задавало нисходящий принцип разработки (пошаговая декомпозиция), предусматривало структурирование логики и данных, за счет простоты и математической основы повышало надежность ПО. Все это органично вписывалось в возможности лаконичного Паскаля.

«Почти все в программном обеспечении может быть реализовано, продано и даже использовано, если проявить достаточную настойчивость... Но существует одно качество, которое нельзя купить таким образом, — это надежность. Цена надежности — это погоня за крайней простотой. Это цена, которую очень богатому труднее всего заплатить» — так Энтони Хоар определил для себя главный критерий, ставший доминантой и для Вирта.

Взвешенный подход, простота и лаконичность — вот залог надежности. Вирт отмечает: «Поддержание языка максимально простым и регулярным всегда было приоритетом в моей работе: описание Паскаля занимало около 50 страниц, Модуль-2 — около 40, а Оберона — и вовсе 16. И я рассматриваю эту тенденцию как прогрессивную. Истинная ценность языков программирования зависит от качества и практичности их абстракций».

Характеризуя замысел своего языка, Вирт пишет: «Главной инновацией Паскаля было введение вариативности структур и типов данных подобно тому, как Алгол ввел вариативность управляющих структур. Алгол предлагал только три базовых типа данных: целые и вещественные числа, значения истинности, массивы; Паскаль ввел дополнительные базовые типы и дал возможность определять новые базовые типы (перечисление, диапазоны), а также новые виды структурирования: запись, множество, файл (последовательность), часть которых была представлена в Коболе. Наиболее важной стала, конечно, рекурсивность структурных описаний и вытекающая из нее возможность осуществлять комбинирование и вложение структур».

А что же говорили противники языка? Из всех критических работ по Паскалю, пожалуй, наибольшую известность получила статья Брайана Кернигана «Почему Паскаль не является моим любимым языком программирования». В 1981 г. она появилась на свет в виде препринта AT&T Bell Laboratories. Поскольку ряд авторитетных журналов отказались ее публиковать, она стала расходиться «нелегальными» путями. В широкой печати ей довелось выйти лишь в 1984 г. в сборнике «Comparing and Assessing Programming Languages» (Prentice-Hall, 1984).

Как известно, Керниган вместе с Ритчи готовил подробное описание языка Си, а потому его мнение особенно интересно. Началось все с того, что Керниган решил адаптировать исходные тексты своей

книги «Software Tools» с Си для Паскаля. К работе над примерами из книги, как пишет Керниган, он приступил весной 1980 г. и завершил ее лишь в январе 1981 г.

Среди достоинств языка Керниган отметил следующие: механизм рекурсии, тип «перечисление», тип «запись», булевы переменные. Из серьезных недостатков он выделил отсутствие поддержки массивов с открытыми границами, неудобство работы со строками, отсутствие статических переменных (по отношению к процедурам и функциям), настоятельную потребность в отдельной компиляции, ограниченные средства ввода-вывода.

Керниган пишет: «Паскаль может быть превосходным языком для обучения новичков тому, как писать программы... Он определенно оказал воздействие на проектирование новых языков, из которых Ада, пожалуй, является наиболее важным. Но в своем стандартном виде (как нынешнем, так и предлагаемом) Паскаль не подходит для написания реальных программ».

Многие конкретные претензии сделаны по существу. Однако, по всей видимости, Керниган немного лукавил, говоря о недостатках Паскаля и умалчивая о работах Вирта, направленных на их устранение. А ведь к весне 1980 г. Вирт и его коллеги не только уже завершили работы по компьютеру Лилит и языку Модула-2, но и опубликовали их результаты. К тому же в 1977 г. в известном журнале *Software — Practice & Experience* вышла статья Хоара, Уэлша и Снинрингера с анализом проблем Паскаля.

Просчеты при создании Паскаля были устранены в последующих языках швейцарской школы (Вирта и его коллег). Все они четко следовали основным тенденциям развития технологии программирования. Программирование структурное (Паскаль), модульное (Модула-2), объектно-ориентированное (Оберон-2), компонентное (Component Pascal) — все это значительные шаги в индустрии ПО.

«Наша конечная цель, — пишет Вирт, — расширяемое программирование (extensible programming). Под этим я понимаю возможность конструирования таких иерархий модулей, когда каждый модуль добавляет новую функциональность в систему. Расширяемое программирование подразумевает, что добавление модуля возможно без необходимости вносить какие-либо изменения в существующие модули — не должно быть необходимости даже их перекомпилировать. Новые модули не только добавляют новые процедуры, но, что более важно, добавляют также новые (расширенные) типы данных. Мы продемонстрировали практичность и экономичность этого подхода при проектировании Oberon System».

Паскаль жив?!

Говоря об отстраненном и предвзятом отношении людей к «чужим» языкам, Вирт отмечает: «Многие относятся к стилям и языкам программирования, как к религиозным конфессиям: если вы принадлежите к одной из них, то не можете принадлежать к другой. Но это ложная аналогия, и она сознательно поддерживается по причинам коммерческого порядка».

В самом деле, противостояние двух лагерей — Си и Паскаля — все эти годы намеренно поощрялось. Между ними методично возводили высокую стену, а затем принялись уничтожать поодиночке. В лице UNIX язык Си нашел себе надежную защиту от чересчур агрессивных модернистов-реформаторов. У Паскаля же такого «панциря» не оказалось. Он продолжает свой путь во многом благодаря преданности своих почитателей, которая подчас просто поражает.

* * *

В заключение предоставим слово оппоненту Никлауса Вирта — Деннису Ритчи: «Паскаль — очень элегантный язык. Он по-прежнему жив. Он породил немало своих последователей и оказал глубокое воздействие на проектирование языков».

Из биографии Н. Вирта

Профессор Никлаус Вирт (Niklaus K. Wirth), автор языка Паскаль, закончил Швейцарский федеральный технологический институт ETH (Eidgenoessische Technische Hochschule) в родном Цюрихе (1958). В Лавальском университете в Квебеке (Канада) он получил степень магистра (1960). В 1963 г. в Университете Калифорнии в Беркли (США) Вирт под руководством профессора Гарри Хаски реализовал расширение Алгола-60 (язык Euler) и защитил диссертацию. В 1963—1967 гг. Вирт преподавал в Стэнфордском университете (США). В это же время он был приглашен в международную экспертную группу IFIP Working Group 2.1, занимавшуюся проектированием языка Алгол-68.

В 1967 г. Вирт вернулся на родину и стал доцентом Университета Цюриха. В 1968 г. он перешел в ETH, где занялся разработкой языка Паскаль. В 1970 г. был завершен первый компилятор Паскаля. В период 1978—1981 гг. Вирт возглавлял проект, в результате которого был разработан язык Модула-2, ориентированный на него 16-разрядный персональный компьютер Лилит (Lilith) и ОС Medos. Все ПО, включая системное, было полностью реализовано на Модуле-2. В 1984 г. Никлаус Вирт за большой вклад в развитие языков программирования и за создание персонального компьютера Лилит был удостоен премии Алана Тьюринга — самой престижной и почетной в компьютерном мире, которая по своему значению стоит в одном ряду с Нобелевской премией.

В период 1986—1989 гг. Вирт вел проект по созданию нового языка Oberon, расширяемой объектно-ориентированной ОС Oberon и 32-разрядной рабочей станции Ceres. Многие идеи того проекта были положены сотрудниками Sun Labs в основу языка и технологии Java.

С 1990 г. Вирт руководил Институтом компьютерных систем при ETH. В 1999 г. он ушел на заслуженный отдых и стал почетным профессором родного ETH.

Стандарты Паскаля

В 1977 г. во время конференции по Паскалю в Саутгемптоне Э. Эддман попросил оказать помощь в формировании комитета по стандартизации при Британском институте стандартов BSI (British Standards Institute). В 1978 г. для обсуждения расширения Паскаля на конференции в Сан-Диего, проходившей под руководством Кеннета Боулеса, встретились представители промышленных кругов. С этого и началась стандартизация языка.

В 1982 г. Международная организация по стандартизации (ISO) выпустила стандарт ISO 7185:1983. В это же время вышел американский стандарт ANSI/IEEE 770X3.97:1983. Принято выделять три принципиально разных стандарта: Unextended Pascal (исходный), Extended Pascal (расширенный) и Object-Oriented Extensions to Pascal (объектно-ориентированный). В 1989 г. стандарт ISO 7185:1983 был подкорректирован, после чего надобность в самостоятельном ANSI-стандарте отпала и он был заменен ссылкой на ISO 7185:1990. Одним из главных отличий Extended Pascal стало введение модульности, необходимой для раздельной компиляции. Extended Pascal закреплен в международном стандарте ISO/IEC 10206 (1991) и американском ANSI/IEEE 770X3.160:1989. В 1993 г. объектно-ориентированные расширения по отношению к Extended Pascal были сформулированы в техническом отчете ANSI/X3-TR-13:1994.

Что касается самой распространенной реализации языка — Turbo Pascal, а также Object Pascal (Delphi), то ни одному из этих стандартов они не соответствуют. В свою очередь, они неявно сами задают соглашения, которым пытаются следовать некоторые реализации.

Табл. Наиболее важные реализации Паскаля

Система	Год	Руководитель	География	Организация
ETH Pascal	1970	Никлаус Вирт	Европа	ETH
UCSD Pascal	1978	Кеннет Боулес	Америка	UCSD
Turbo Pascal	1983	Андерс Хейльсберг	Америка	Borland Int.

Преемники языка Паскаль

- Модула-2 —> Оберон —> Оберон-2 —> Component Pascal
- Concurrent Pascal —> Модула —> Edison
- Euclid
- Mesa —> Ада —> Cedar —> Модула-3
- UCSD Pascal —> Turbo Pascal —> Borland Pascal —> Object Pascal

Популярные языки						
Название	Год	Вид	Автор	География создания	Организация	Стандарт
Фортран (FORTRAN)	1954	A	Джон Бэкус	Америка	IBM	ISO 1539:1997
Лисп (LISP)	1958	F	Джон Маккарти	Америка	MIT	-
Алгол-60 (Algol 60)	1960	A	Питер Наур+	Межд.	IFIP	-
Кобол (COBOL)	1960	A	+	Межд.	CODASYL Committee	ISO 1989:1985
Симула (Simula)	1962	B	Кристен Нигаард+	Европа	-	-
Бейсик (BASIC)	1963	A	Джон Кемени+	Америка	Dartmouth College	ISO 10279:1991
ПЛ/1 (PL/I)	1964	A	Джордж Радин	Америка	IBM	ISO 6160:1979
Алгол-68 (Algol 68)	1968	A	Аад ван Вайнгартен+	Межд.	IFIP	-
Паскаль (Pascal)	1970	C	Никлаус Вирт	Европа	ETH	ISO 7185:1990
Форт (FORTH)	1970	A*	Чарльз Мур	Америка	Mohasco Industries	ISO 15145:1997
Си (C)	1972	C*	Деннис Ритчи	Америка	AT&T Bell Labs	ISO 9899:1999
Smalltalk	1972	B	Алан Кей	Америка	Xerox PARC	-
Пролог (Prolog)	1973	E	Алан Кольмеро+	Европа	Univ. of Aix-Marseille	ISO 13211:1995
Ада (Ada)	1980	H*	Джин Ишбиа+	Америка	CII Honeywell	ISO 8652:1995
C++	1984	H*	Бьорн Страуструп	Америка	AT&T Bell Labs	ISO 14882:1998
Java	1995	H	Джеймс Гослинг	Америка	Sun Labs	-
Экспериментальные и промышленные языки						
АПЛ (APL)	1957	I	Кеннет Айверсон	Америка	Harvard Univ.	ISO 8485:1989
Снобол (Snobol)	1962	I	Ральф Грисуолд	Америка	AT&T Bell Labs	-
Сетл (SETL)	1969	I	Джек Шварц	Америка	IBM	-
Параллельный Паскаль Concurrent Pascal	1974	G	Пер Бринч Хансен	Америка	CIT	-
CLU	1974	D	Барбара Лисков	Америка	MIT	-
Scheme	1975	F	Гай Стил+	Америка	MIT	-
Mesa	1976	D*	Дж. Мичел+	Америка	Xerox PARC	-
Icon	1977	I	Ральф Грисуолд	Америка	AT&T Bell Labs	-
Модула-2 (Modula-2)	1979	D*	Никлаус Вирт	Европа	ETH	ISO 10514:1996

Оккам (Occam)	1982	G*	Дэвид Мэй+	Европа	Inmos	-
Cedar	1983	H*	Батлер Лэмпсон+	Америка	Xerox PARC	-
Common Lisp	1984	F	Гай Стил+	Америка	MIT	-
Objective C	1986	H*	Брэд Кокс	Америка	Productivity Products	-
Эйфель (Eiffel)	1986	D*	Бертран Мейер	Европа	ISE	-
Оберон (Oberon)	1988	D*	Никлаус Вирт	Европа	ETH	-
Модула-3 (Modula-3)	1988	H*	Билл Калсов+	Америка	DEC SRC	-
Оберон-2 (Oberon-2)	1991	D*	Ханспетер Мёссенбек+	Европа	ETH	-
Limbo	1996	D*	Деннис Ритчи	Америка	Bell Labs (Lucent)	-
Component Pascal	1997	D*	Куно Пфистер+	Европа	Oberon microsystems	-
C#	2000	H*	Андерс Хейльсберг+	Америка	Microsoft	-

Примечание. Выделены языки Паскаль-семейства.

УСЛОВНЫЕ ОБОЗНАЧЕНИЯ	
Виды (парадигмы)	Сокращения
A — процедурное программирование	MIT — Massachusetts Institute of Technology
B — объектно-ориентированное программирование	PARC — Palo Alto Research Center
C — структурное программирование	ETH — Swiss Federal Institute of Technology
D — модульное (компонентное) программирование	SRC — Systems Research Center
E — логическое (реляционное) программирование	ISE — Interactive Software Engineering
F — функциональное программирование	ISO — International Standard Organization
G — параллельное программирование	CIT — California Institute of Technology
H — гибрид (смесь парадигм; B+C+D+G)	* Поддержка системного программирования.
I — специализированные языки	+ Несколько авторов.