

ГЛАВА 2

История развития языка программирования Паскаль и среды разработки Borland Delphi

Сегодня среда Delphi использует одноименный язык программирования. Однако его корни восходят к языку программирования Паскаль. Таким образом, история системы Delphi началась в конце 60-х годов XX века.

2.1. История языка Паскаль

В 1971 году *Никлаус Вирт* (Niclaus Wirt), профессор швейцарского технологического института ETH (www.ethz.ch) города Цюрих, опубликовал описание нового языка программирования *Паскаль*. Название было выбрано в честь французского монаха и математика Блеза Паскаля, жившего в XVII веке и сконструировавшего механический калькулятор. Язык Паскаль был придуман во многом как учебный язык¹, предназначенный для обучения программированию. Он опирался на идеи уже существовавшего языка Алгол 60.

В середине 1960 годов Международная федерация по обработке информации IFIP (www.ifip.org) открыла научную программу по созданию нового языка программирования. Он должен был унаследовать лучшие черты популярного языка Алгол 60. Впоследствии проект был расширен поддержкой языка Алгол 68. Никлаус Вирт работал в те годы в Стэнфордском университете. Он занимался созданием компилятора для одного из диалектов Алгола 68 и предложил свой вариант нового языка, названный Algol W. И хотя эксперты IFIP в конечном счете остановились на Алголе 68, Вирт по возвращении в Швейцарию продолжил работу над своим детищем. В ходе создания Паскаля профессор применял инженерно-конструкторские навыки, приобретенные при создании моделей радиоуправ-

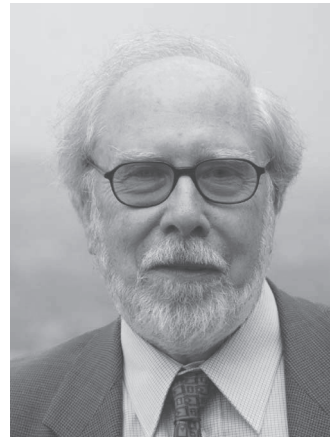


Рис. 2.1. Никлаус Вирт

¹ Профессор Вирт неоднократно отмечал, что инструмент может быть успешным в обучении, только если он позволяет эффективно решать сложные практические задачи. Паскаль удачно сочетает академическую простоту и лаконичность с мощными выразительными средствами построения программ.

ляемых самолетов. Он также заинтересовался набиравшим в то время популярность *структурным подходом к программированию*.

Структурная методология разработки программного обеспечения была изобретена голландским ученым Эдсгером Дейкстрой (Edsger Dijkstra). Он впервые сосредоточил внимание на эффективных способах внутренней организации программы. Дейкстра, как и Вирт, участвовал в проектах IFIP и работах по проектированию разных версий Алгола.

В 1969 году в Риме прошла научная конференция НАТО по программированию. Вирт на этой конференции рассказывал о Паскале, а Дейкстра выступил со знаменитым докладом «Структурное программирование». Он показал, что при программной реализации любого алгоритма можно обойтись без оператора перехода. Достаточно придерживаться последовательного выполнения частей программы. Сама программа создается методом «сверху-вниз». Сначала в реализуемом алгоритме выделяются крупные модули. Их функционирование детализируется до более и более мелких элементов, но все они выполняются последовательно.

Структурное программирование — технология создания программ в виде иерархически связанных модулей, представляющих собой линейную последовательность операторов цикла, присваивания и условных операторов, а также вызовов вложенных модулей.

Концепция структурного программирования позволила полностью отказаться от применения оператора перехода GoTo, обусловила возможность автономного написания и отладки модулей и стала необходимой предпосылкой для перехода к концепции объектно-ориентированного программирования.

Вирт, изучив идеи Дейкстры, взял из Алгола лучшие элементы и дополнил их современными на тот момент кибернетическими веяниями. Он, в частности, выдвинул идею свободного определения пользовательских типов. Особое внимание Вирт уделил поддержке в Паскале упомянутой концепции структурного программирования. Он стремился упростить задачи выделения частей крупной системы в отдельные модули. Эти модули, по задумке профессора, должны быть максимально независимыми друг от друга.

Детище Вирта оказалось на редкость удачным. Паскаль стал провозвестником множества современных языков программирования. Его модификации широко распространены у программистов всего мира и продолжают активно применяться в проектах самой высокой сложности.

2.2. Императивное программирование

Классический Паскаль считается императивным языком программирования.

Императивный язык — это язык программирования, инструкции которого предназначены для последовательного, пошагового изменения состояния вычислительного устройства.

Программа на Паскале конструируется из сравнительно небольшого набора операторов: условного, присваивания, цикла. Они выполняются последовательно и работают с данными явно заданных типов. Паскаль считается *языком со строгой типизацией* — все определяемые в программе переменные принадлежат конкрет-

ным типам данных. Это позволяет избежать ошибок неверной обработки данных разных типов еще на этапе компиляции. Группы команд можно выделять в *подпрограммы* (процедуры и функции). В связи с этим Паскаль также называют *процедурным языком*.

Паскаль удачно вписался в историю программирования, так как он последовательно реализует структурный подход. В императивном программировании основным синтаксическим элементом является оператор. Императивным оператором считается как простейший оператор (условный, присваивания, цикла), так и составной, объединяющий последовательность простейших операторов. Понятие составного оператора стало основой структурной методологии и проектирования программ по принципу «сверху-вниз».

У императивного подхода имеется существенный недостаток. Он заключается в том, что сложность разработки крупных проектов на императивных языках растет быстрее, чем число операторов в программе. Понимание и модификация программы из сотен тысяч и миллионов операторов сильно затрудняется, а нередко становится невозможной.

Императивный подход не является единственно возможным. Другие подходы порождают особые категории языков программирования.

- *Объектные языки программирования* ориентированы на манипуляции с объектами. К ним относятся, например, C++, Java, C#, современный вариант языка Delphi. Программирование в объектной модели происходит не столько кодированием мелких шагов, сколько организацией взаимодействия объектов.
- *Функциональные языки программирования* опираются на теорию лямбда-исчислений. Они описывают алгоритм в терминах функциональных преобразований. Это, например, LISP и РЕФАЛ.
- *Логические языки программирования* позволяют описать набор фактов и правил их обработки. Поиск результата выполняется автоматически системой логического вывода.

2.3. Прикладные реализации Паскаля

Первый компилятор Паскаля был написан программистами EТН под руководством Вирта в 1970 году для суперкомпьютера CDC 6000. Примерено в то же время в американском университете штата Иллинойс появился компилятор Паскаля для ЭВМ PDP-11. Он был разработан при активном участии канадского математика Дональда Джиллиеса (Donald Gillies).

Запущенный в 1957 году первый советский спутник вызвал панику в правительстве США — зарубежные политики озаботились научным отставанием от СССР, и вскоре физика и математика в американских школах были введены в качестве обязательных предметов. Джиллиес за двое суток рассчитал на компьютере ILLIAC I траекторию движения спутника и опубликовал результаты в журнале Natural. Его статья частично успокоила американскую общественность, взволнованную технологическим отставанием.

В швейцарской версии компилятора Паскаля была экспериментально реализована предложенная Виртом идея *виртуальной машины*. Программа на Паскале

преобразовывалась не в машинный код, а в промежуточный, так называемый Р-код. Такой подход теоретически позволял переносить скомпилированный код на разные аппаратные архитектуры. Для них достаточно было создать виртуальную машину, исполняющую Р-программы. Сегодня эта идея воплощена в платформенно-независимом языке Java.

В 1970-е годы появилась еще одна законченная реализация Паскаля и виртуальной Р-машины. Она известна как интерпретирующая система UCSD Pascal² и была создана в калифорнийском университете Сан-Диего в 1975 году под руководством профессора Кеннета Боулеса (Kenneth Bowles). Он еще в те годы предсказывал, что в будущем появится множество компьютерных платформ и возникнет немало проблем с переносимостью программного обеспечения. Система UCSD Pascal в силу своей простоты оказалась популярной не только у студентов, но и у профессиональных программистов. Значительная часть операционной системы Apple II и III для компьютеров Макинтош была написана на системе UCSD Pascal.

Полное формальное описание Паскаля было разработано и опубликовано Виртом в 1974 году совместно с Кетлин Йенсен (Kathleen Jensen). В 1983 году появился стандарт Паскаля *ISO/IEC 7185*, а в 1990 — расширенный стандарт *ISO/IEC 10206*. Единые требования к компиляторам Паскаля пошли на пользу данному языку. Достаточно было разработать программу в соответствии со стандартом, и после этого ее можно было переносить между разными платформами и поддерживающими стандарт компиляторами, не внося модификации в исходные тексты.

Вирт разработал еще несколько языков, развивавших идеи Паскаля: *Модула* и *Модула-2*. В них были усилены идеи модульной разработки, позволившие обойти ряд недостатков классической императивной концепции. Одним из последних языков Вирта стал *Оберон*, в котором появились средства объектно-ориентированного программирования³. Особое внимание при создании языков программирования Вирт уделял простоте. Он не без основания полагал, что чем выразительнее, компактнее и проще средства языка, тем строже процесс разработки и тем проще отслеживать правильность программ.

Описание языка программирования Модула было опубликовано в 1975 году. Модула первоначально создавалась в ходе экспериментов с параллельным (многозадачным) программированием для систем с несколькими процессорами. Основанная на ней Модула-2 также была разработана в институте ЕТН. Оригинальный проект первоначально назывался «Лилит». В Модуле-2 появились развитые механизмы ввода-вывода и прямого доступа к памяти, средства программирования низкого уровня и сопровождения синхронно выполняющихся задач (процессов). Впервые компилирующая система для Модулы-2 заработала в 1979 году на ЭВМ PDP-11.

² В системе UCSD Pascal была впервые воплощена фундаментальная концепция модулей (Units), принятая и в современной версии языка Delphi.

³ Общедоступные компиляторы для языков Модула-2 и Оберон и сопутствующую информацию можно найти, например, на сайтах www.oberon.ethz.ch, www.excelsior-usa.com/xds.html и blackbox.thundersign.su.

Модуля-2 проектировалась с ориентацией на крупные проекты и большие коллективы разработчиков. Однако корпорация Borland вложила основные средства именно в Паскаль. Поэтому Модуля и Оберон, как и многие другие, не менее замечательные языки программирования, так и не смогли завоевать массовую популярность.

Язык Оберон был реализован Виртом в 1987 году на 32-разрядном однопользовательском компьютере Ceres на базе процессора RISC NS32000. Автор сделал Оберон очень простым. Из него была исключена любая избыточность, а синтаксическая структура языка минимизирована. Описание этой структуры уместается на одной страничке.

Некоторые сильные черты Модуля и Оберона воплощены в версии Паскаля под названием язык программирования Delphi. Первоначально этот язык представлял собой вариант Паскаля, близкий к классическому. Затем он был расширен множеством самых разных нововведений, в том числе и объектных. Современная версия языка Delphi, в отличие от Оберона, не избежала избыточности. Однако она стройна, хорошо продумана и, главное, прекрасно реализована.

2.4. История системы Delphi

В 1980 году программист Андерс Хейлсберг (Anders Hejlsberg) разработал компилятор Паскаля Blue Label Pascal для персонального компьютера Nascom-2 с процессором Z80. Этот компилятор был лицензирован корпорацией Borland. На его основе в 1983 году под руководством Хейлсберга была выпущена первая интегрированная среда разработки Turbo Pascal 1.0 для IBM PC стоимостью в 50 долларов. Она включала редактор, простой отладчик и среду исполнения, вызываемые из удобно спроектированной текстовой оболочки.

Хейлсберг устроился на работу в Borland⁴. Он занимался вопросами проектирования архитектуры всех версий среды Turbo Pascal и первых трех версий системы Delphi. Ему, в частности, принадлежит идея расширить первые версии Turbo Pascal объектными концепциями.

Работа над интегрированной графической средой программирования Delphi была начата корпорацией Borland в середине 1993 года. Новая среда коренным образом отличалась от всех предыдущих систем разработки. В ней была впервые воплощена полноценная концепция визуального создания программ. В частности, появился Проектировщик форм и библиотека визуальных компонентов. На рынке этот продукт позиционировался как «убийца Visual Basic».

Название Delphi было выбрано на основе существовавшей в Древней Греции поговорки: «Хочешь поговорить с оракулом — иди в Дельфы». В этом названии обыгрывается слово Oracle (оракул) — название известной компании, производящей серверы баз данных. Первоначально среда Delphi должна была называться AppBuilder (Построитель приложений). Однако перед ее выходом компания Novell выпустила собственную систему визуальной разработки Visual AppBuilder.

⁴ В 1996 году Хейлсберг перешел на работу в корпорацию Microsoft. Впоследствии он разработал язык C#, а ныне отвечает за развитие архитектуры .NET.

Поэтому менеджерам корпорации Borland пришлось экстренно менять название продукта.

Система Delphi 1 появилась на рынке в 1995 году. Ее отличали мощные средства проектирования и программирования работы экранных форм. В Delphi 1 входили средства поддержки популярных СУБД, в том числе клиент-серверных. Сам язык на первых порах получил официальное название Object Pascal (объектный Паскаль).

С тех пор версии Delphi выходят почти каждый год. Версия Delphi 2 (1996 год) обзавелась быстрым 32-разрядным компилятором. Она была настроена на программные интерфейсы появившейся тогда же операционной системы Windows 95. Delphi 2 также пополнилась улучшенными наборами компонентов для работы с базами данных и для построения визуальных табличных элементов, связанных с СУБД. Ее девизом стала фраза «легкость Visual Basic и мощь Си++».

В версии Delphi 3 (1997 год) появились компоненты, позволившие использовать веб-технологии на корпоративном уровне. Возникла возможность отладки библиотек DLL, создания шаблонов и пакетов компонентов. Были добавлены компоненты для построения отчетов и анализа данных, средства оптимизации работы в редакторе.

Возможность создания распределенных приложений стала отличительной чертой версии Delphi 4 (1998 год). В нее были включены средства работы с компонентной архитектурой CORBA, высокопроизводительные драйверы СУБД, технологии стыковки с модулями Java, интерфейсы доступа к настольным и серверным приложениям Microsoft и поддержка интерфейсов Windows 98. В языке появились динамические массивы и средства перегрузки методов.

Стабильная, производительная и многофункциональная версия Delphi 5 (1999 год) стала продуктом, который завоевал массовое признание программистов во всем мире. В нее был добавлен большой набор компонентов для создания приложений для Интернета и поддержки технологии XML. Система Delphi 5 позволила настраивать и хранить несколько вариантов рабочих столов среды, в ней появились средства локализации. Доступ к СУБД из программ на Delphi мог организовываться по новой технологии Microsoft ADO. Название языка (объектный Паскаль) окончательно сменилось на сегодняшний вариант *язык Delphi*, совпадающий с названием среды.

Версия Delphi 6 (2000 год) отличалась развитой поддержкой веб-сервисов. Был реализован набор «тонких» компонентов dbExpress для быстрого доступа к СУБД. Наряду с Delphi 6 корпорация Borland выпустила новый пакет Kylix, ориентированный на создание программ на языке Delphi для платформы Linux. В результате линейка продуктов Delphi стала кросс-платформной. Один и тот же исходный код приложения на языке Delphi мог компилироваться и в среде Delphi, и в среде Kylix для разных операционных систем.



Рис. 2.2. Андерс Хейлсберг

Версия Delphi 7 (2001 год) стала последней и наиболее развитой версией среды, реализованной для платформы Win32. В ней, в частности, появился компилятор для .NET, пока в консольном варианте. Добавилось множество компонентов и Мастеров, автоматизирующих создание сетевых приложений. Была продолжена поддержка кросс-платформной технологии создания программ на базе Delphi и Kylix.

Версия Delphi 8 (2003 год) считается первой версией Delphi для платформы .NET. Переход был кардинальным: из Delphi 8 был полностью исключен компилятор для платформы Win32. В то же время допускалась создание приложений как с помощью стандартного набора компонентов .NET, так и с помощью новой версии библиотеки Borland VCL.NET.

Версия Delphi 2005 (2004 год) вновь включила средства разработки для Win32 — в дополнение к поддержке .NET. В нее также вошел компилятор языка C# — стандартного языка разработки для платформы .NET.

Версия Delphi 2006 и все последующие версии среды развиваются не только наращиванием функциональных возможностей, вводом новых наборов компонентов или включением новых информационных технологий. Практически все современные технологии уже вошли в Delphi в той или иной форме — ведь жизненный цикл технологии программирования составляет не менее трех лет, а версии Delphi выпускаются каждый год. Среда Delphi теперь активно расширяется внешними продуктами, охватывающими все большее число этапов создания программного продукта. Автоматизируется работа по анализу требований, модельному проектированию, организации дистанционного доступа к версиям кода, автоматическому тестированию и развертыванию. Таким образом, система Delphi становится законченным комплексом, предлагающим все необходимые инструменты для организации жизненного цикла программного обеспечения.

2.5. Вопросы

1. Кто автор языка Паскаль?
2. Что такое императивный язык программирования?
3. В чем заключается структурный подход к разработке программ?
4. Что такое виртуальная машина?
5. Какие языки развивают идеи Паскаля?
6. Кто разработал систему Delphi?
7. Как развивается среда Delphi в настоящее время?