

Pascal — Lurkmore

Эта статья о языке программирования; об учёном, в честь которого он назван см. [Блез Паскаль](#).



A long time ago, in a galaxy far, far away...

События и явления, описанные в этой статье, были давно, и помнит о них разве что пара-другая олдфагов. Но Анонимус не забывает!

«Всех, кого ты не спроси,

Программируют на Си. Нынче встретишь ты едва ли Тех, кто пишет на Паскале

»

— Фольклор

```
<Ashuga> По дороге на работу решил выбросить мусор.Подхожу к контейнеру.
<Ashuga> (р-н Б.Дорогомилловской, Москва)
<Ashuga> В соседнем контейнере роятся два бомжа. И вот один вытягивает из мусора книгу.
<Ashuga> Название: "Программирование на Турбо-Паскале". И выдает фразу:
<Ashuga> - Глянь, Петрович, неужели кто-то еще на "Трубе" пишет?
```

 763

Pascal (*Паскаль, ПаскакалЪ, ПоссалЪ, Пасквиль, Поцкаль, Пасцал, Пашка, В-Пасть-Кал, Турба* (устар.), Турбопаскаль (устар.), Турбомоскаль(УкронимЪ.), Труба (устар.), Делфя) — **мертвый, но еще живой**, язык программирования, созданный в тех годах, когда ваши родители играли кирпичами и кидались ссаными валенками, известным **быдлокодером** Никлаусом СкВиртом, чтобы учить говнокодить **школьников** и студентом, и изначально являлся упрощённым **Алголом**, освоив который нетрудно перейти на настоящий Алгол. Но, позже **ВНЕЗАПНО** обнаружилось, что сей учебный язык может быть пригоден для написания **реальных программ** и даже **игор** (от чего Алгол, утратив свою необходимость постепенно загнулся). Когда-то имел весьма солидную популярность, но сейчас ее полностью потерял, сохранившись в основном в виде Delphi для учебных и академических целей. Язык невозбранно привлекает возможностью писать почти как на обычном английском языке, а не ломать голову и пальцы о `++i + ++i`, эзотерику истинности выражений `(1/3 == 0)` и `(-1 > (unsigned int) 1)` и прочих извращенных приёмов, принятых в C-подобных языках.

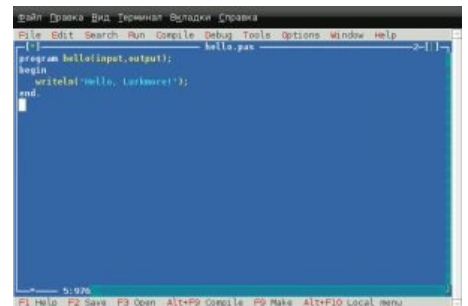
Borland

Что такое было Borland?

Компания «Borland» была основана в начале 80-х годов прошлого века **математиком и саксофонистом** Филиппом Каном. «Borland» стала популярной на рынке программно обеспечения именно благодаря своим Turbo-версиям компиляторов языков программирования, выпускаемых для **широких масс населения** в те годы. А первым компилятором, который они сделали, был «Turbo Pascal». Вернее не сделали, они купили некоего Андреаса Хейлсберга, который до этого понаписал свою реализацию под названием **Blue Label Pascal**, чуть допилили, упаковали, PROFIT!

Этот компилятор с языка «Паскаль» в ранних версиях представлял собой диалект на базе UCSD Pascal, до того использовавшегося в основном на яблоках. Позднее, в Turbo Pascal были **нагло украдены** невозбранно позаимствованы из Mac Pascal такие расширения, которые позволяли его признать языком объектно-ориентированного и даже системного программирования, наряду с параллельно развивающимся языком **C** и **C++** (например, передача нетипизированных указателей в функцию, свободное приведение типов, модификатор «absolute», директива «interrupt», ассемблерные вставки и пр). Да-да, анон, это невероятно, но факт, Object Pascal был создан именно в недрах Apple и занимал там примерно ту же нишу, которую сейчас занимает Objective C. Однако, несмотря на популярность, Turbo Pascal не суждено было стать языком системного программирования. На PC. На Маках, как известно, он вплоть до 90-х годов был основным языком системного и прикладного программирования (сумневающимся стоит посмотреть исходники Photoshop 1.0), однако впоследствии был почти полностью выпилен и заменен C++ (примерно в седьмой редакции Mac OS).

Одним из самых удачных решений «Borland» было объединение в одной программе (помимо самой оболочки) редактора кода, компилятора, отладчика и доступа к интерпретатору командной строки ОС (то есть они изобрели всё то, что теперь называется IDE, а **Emacs** для задротов). Кодер мог запускать свою программу непосредственно из среды разработки — это было первое пришествие RAD. Была также возможность



Думаете это Borland? **Автхуй!** Это Free Pascal

компилировать программы в ОЗУ и исполнять (и, главное, отлаживать) их оттуда, а не с диска (в те годы многие пользователи загружались с дискет (НГМД), и подобная отладка через «память» заметно увеличивала скорость работы, компиляция и запуск занимали секунды, а не минуты). Правда, придумано это было не впервые, ибо отладчиком кода в памяти невообразимо пользовались ещё на TX-0 и PDP-1, когда для них были только Ассемблер, Фортран и Лисп, и не было никаких дискет. Одни перфоленты, да. Но главное не в этом, там отладчик не был отдельным приложением, как тот же FLIT для упомянутого TX-0, поэтому тут у Borland [win](#).

Начиная с версии 5.5 у среды радикально сменился диалект. Сохраняя изрядное влияние UCSD-Pascal и все введённые в ранних версиях низкоуровневые плюшки, в основе своей она окончательно перешла на разработанный в [Apple Object Pascal](#), чем и послужила его широчайшей популяризации. Тот факт, что под названием Pascal в настоящее время понимается именно Object Pascal, и именно на нём основывается общепринятый стандарт языка — заслуга в первую очередь Borland и TurboPascal, ибо за давностью лет все уже давно забыли, кто же изначально этот Object Pascal создал.

В версии 6.0 среда переписана на Turbo Vision (впрочем, лишь невозбранно спизженном с Mac-ов тех времен). Но те, кому доставляла лишь алгоритмическая часть их задач (то есть [школота](#)), а не новомодные понты Turbo Vision (смысл которых они понять не могли в силу слишком юного возраста), как только могли, старались продолжать пользоваться версией 5.0 и даже 3.0. Даже не смотря на труЪ-защищенный режим, появившийся в BP 7.0, и возможность программировать под Форточку 3.11 там же. Версия 7.0 получила также IDE под Win3.x и, кстати сказать, была первой, где отошли от программирования приложений под Виндус посредством функции Winmain и предложив объектный подход в создании приложений.

Параллельно «Borland» пыталась выпускать подобные программы-компиляторы для C, Prolog и BASIC. Первым склеил лапы [Пролог](#), потом Филю замучила совесть — выпилил Бейсик (просуществовал до трубо-паскаля 4.0, присутствовал бинарный, в формат EXE, компилятор). Turbo C — представлял собой, на то время, вполне вменяемый скоростной компилятор (в отличие от мелкомягких поделок), но получив в нагрузку два плюса и Трубо Вижен (TV) — потихоньку скурвился.

Скурвился, впрочем, лишь по мнению тех, у кого от неприятия гениального, по тем временам, Turbo Vision наступило окончательное окукливание ФГМ: им казалось, что мелкомягкие стали единственными евангелистами Цэ, во вселенной ДОСа и и даже винDOSa. На самом деле Borland C++ невозбранно имел развитие и в Windows 3.11 (ололол, да-да, тот самый легендарный OWL, просто все, кто его знал, [уже уехали в Лондон](#)), и потом заново выкуклился в весьма популярной в отдельных кругах серии Borland C++ Builder (есть мнение, что лучше бы и не выкукливался), в результате чего на плюсах тоже можно стало кодить мышкой, как в Delphi. Количество быдлокодеров сильно возросло, отправив плюсы в быдлокодерские языки. Правда до тру оптимизации руки у борманов уже так и не дошли, а потом команда и вовсе ушла пилить бюджет, выделенный Бени Гейком под свой мерзкий .NET, гибрид идей визуалбейсика, Delphi и Java в одном лице. Алсо, Borland Builder до сих пор не поддерживает стандарт C++98 (который, в общем-то, полностью почти никто и не поддерживает).

Но то было уже потом. А пока в середине 90-х годов фирма поменяла политику, хитрожопо сменив ориентацию на выпуск фирменной RAD для объектного Pascal-диалекта, спёртого ими у яблока, под ОС Windows — Object Pascal. Тем самым она стала раскручивать продукт под названием «Delphi», и, надо сказать, настолько преуспела, что Бени Гейц впоследствии выкупил команду гениальных разработчиков вместе с их домиками на колесах, чуть более, чем целиком и насовсем. Стоит, впрочем, заметить, что переход на Object Pascal случился ещё в ДОС-эру и ничего, кроме охуевания школоты от Turbo Vision, не вызвал. Скатывание среды в сраное говно началось именно в Дельфях с появлением там возможности «кодить мышкой».

Почему Pascal стал популярен?

В 80-е и 90-е годы прошлого века, в связи с ~~развитием технологий~~ появлением большого числа ленивых долбоебов, DOS-программисты желают послать нахуй ассемблер. В то же самое время, фирмы «Microsoft», «Intel» и «IBM» (i.e. [ZOG](#)) проектируют концепт под названием «PC» (писюк), позднее «PS/2» (полуписюк), а позднее и вовсе Wintel, и даже выпускают все это (вызывая приступы входа в стену у юниксоидов и Аллы Пугачевой). Дешевые настольные компьютеры, очевидно, производят промышленную революцию. Теперь пользователем компьютера может стать практически любой человек, а не только нерд в очках. Выпускаемые персоналки имеют спрос среди колхозников, и такой же спрос существует на быдлопрограммистов, готовых говнокодировать под эту платформу. Основное требование к ним — говнокодить быстро и много. Правда, [КО](#) намекает, что дешёвые доступные настольные 8-битные компьютеры появились гораздо раньше писюков, еще в 75 году, и под них клепали интерпретаторы Basic, на чём поднялся дядюшка Билли из Мелкомягкого.

Вот тут-то «Borland» и подсуетился. Но компилятор «TP» был-таки выпущен для этой набирающей обороты платформы. И «TP» по возможностям написания кода для процессоров x86 не уступал языку «C». Но ограничение на максимальный размер массива в 64к и безумная реализация функции Delay() вызывала повышение ЧСВ у Сионистов. Устранено было только в [ТП 7.01](#).

В результате, в то время:

- Вакантные места программистов для писюков

про Delay()

Тот самый «Runtime error 200» при запуске программы на машинах старше 386-х, заставлявший срать кирпичами пользователей. Но джедаям было [эквипенисуально](#) — они знали, какие 5 байт EXE-файла надо было слегка поправить. Ошибка заключалась в секции инициализации стандартного модуля CRT во время попытки вычисления миллисекундных пауз в тиках

захватывает мафия бывших студентов Вирта (ну им так казалось, на самом деле там было все схвачено вплоть до конца XX-го века ушлыми клипперистами, фокспрошниками и дибейзниками, да да). Но паскалисты (студенты) пишут программы (лабораторные) быстро, решительно и безошибочно (ибо среда прерывает компиляцию при любой попытке написать явно школьный код).

- Программисты на «С» делают в это время фатальные ошибки, постоянно забывая, что в этом их языке символ «*» обозначает, в зависимости от контекста, четыре совершенно разные операции (вообще, ошибкам при программировании на «С» был посвящён целый раздел в Википедии (выпилен)), из-за чего сессия сдаётся не вовремя, девки не дают, и все такое разное, ну вы понимаете. Правда, это всё дескать более от ФГМ недокодеров, ибо там чётко написано, что программировать нужно уметь (но мы то знаем...).
- А тем временем GW/Quick/Visual BASIC массово начинают преподавать в школах как язык дрессировки обезьян программированию. И, что характерно и удивительно, обезьяны крайне успешно дрессируются, при всех мыслимых нарушениях азов программирования обычные пользователи, вооружённые VB6, Access и макросами Excel (не говоря уже про FoxPro) до сих пор порой показывают чудеса в написании полезных для бизнеса программ учета товара на складе и не только (но это уже из другой оперы). А **программисты** на Паскале, не имея тогда толком возможности ходить в базы данных, лишь тупо паялились на свой Turbo Vision; хотя и существовали вполне себе сетевые btrieve и Paradox Engine, масса DBF engines в виде сторонних библиотек (некоторые даже с поддержками индексов и крутых наворотов), но всякие эти ваши гриды и биндинги визуальных контролов еще не были придуманы, и далеко не каждый мог придумать подобное сам. Более-менее поправил эту ситуацию Delphi — там уже появилась полностью нативная поддержка BDE (DBase/Paradox/InterBase/FireBird/Oracle/DB2), впоследствии ADO (Oracle/MSSQL/Access/...) и т. д.
- Кроме того, нельзя не отметить влияние на популярность Паскаля и сугубо местной специфики: видимо, благодаря позиционированию Паскаля как **обучающего** языка, он стал **дикой популярной** на всякого рода IT факультетах вузов **этой страны**. (Да, да, дорогой анонимус, за пределами МКАД тоже есть жизнь, и там даже иногда встречаются всякие университеты/академии/институты.) Что характерно, многие из этих вузов до сих пор не вылезли из каменного века, и их студики до сих пор пишут свои программистские шедевры на Turbo Pascal 7.0, кроя матом несовместимость от слова «совсем» TP7 с Windows 7 и избеживаясь с режимами совместимости и всякими **DOSBox-ами**. (спойлер: Программист-тян рапортует: можно просто поставить WSL, FreePascal и любой консольный текстовый редактор. Пару тысяч нервных клеток это точно спасёт. Не благодари, анон) Вдвойне не повезло тем, кто может себе позволить **более 4 гигабайт оперативной памяти** и поставил себе 64-разрядную версию винды. Кстати говоря, классическая Delphi 7 также имеет некоторые проблемы совместимости с **последними версиями Windows**, но хотя бы запускается под ними.

процессора. С одной стороны в D0Se этого sleep'a не было, с другой — производители железа (советского в том числе) особенно не заморачивались аппаратной реализацией таймера (хотя вектор 15h для часов и таймера и был заточен (вот только появился он, лишь начиная с биосов для PC-AT, а программы нужно было запускать и на XT)). Хотели сделать универсально — получился фейл.

Появления Borland Delphi

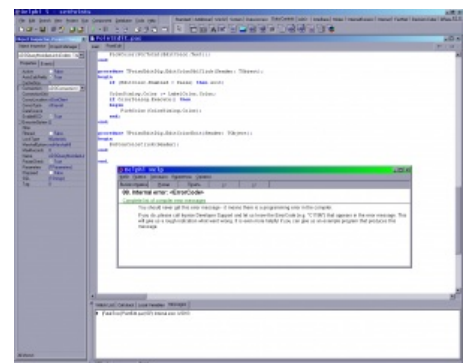
Но с выходом Windows, «Borland» решил таки поддержать своих фанатов. На любое новое Технологическое введение Microsoft (VBX, ODBC, ADO, ActiveX, COM, .NET, **тысячи их**), среда разработки Delphi и тогда, и сейчас отвечает полной (иногда бездумной) поддержкой всего этого.



Логотип
Borland Delphi

Фейлы Delphi

- «Тупорылый» оптимизатор — правда, в винде и с наплывом этих ваших пентиумов и кордваду кодеры оптимизаторами уже **не сильно заморачивались** — ибо редкий пользователь заметит разницу реакции программы на клик мышки между 0.1 сек и 0.01 сек, вот честно.
- Якобы «тупорылый» компоновщик — минимальное оконное приложение, сделанное на основе стандартных компонентов, весило ненормально много для привыкших к минимализму системщиков (около 300 кб). Причина была в том, что Delphi не может исключить из класса часть методов, даже если они нигде не используются, то есть пихает в приложение код класса и всех его предков полностью, а в VCL классы очень тяжелые и с длинной генеалогией. Алсо, вправив Delphi мозги (ну или рисуя окна непосредственно через функции WinAPI), можно заполучить более **компактные экзешники**. Особенно в этом деле помогает библиотека **KOL**, с помощью которой можно писать оконные программы весом в несколько десятков килобайт. То есть дело вовсе не в компоновщике, а в библиотеках. Так или иначе — в наше время все это выглядит особенно ржачно, когда **внезапно** замечаешь, что системный mspaint.exe (наследник pbrush.exe) в Windows 7 весит уже порядка 6376960 байт, а системная библиотека .NET (или Java), будучи скомпилированной и сжатой — весит не менее 20 мегабайт.
- Настоящий фейл был связан с написанием софтинок с кнопочками. Дело в том, что аж до 2008 года (когда вышла Delphi 2009) родная библиотека VCL весьма хреново поддерживала Юникод, что делало



You should never get this **error** message. If you see it, you'll **shit bricks**

локализацию приложений под все возможные языки (в смысле чтоб одновременно все сразу в одной запущенной программе) несколько затруднительной (хотя если язык в момент времени предполагался только один, который системный, то было глубоко пофиг, ибо даже китайский там весьма и весьма неплохо работал). В принципе, еще в 2001 году (год выхода Delphi 6) эта проблема решалась чуть более, чем полностью сторонними наборами стандартных компонент с поддержкой Юникода, а начиная с версии Delphi 2009 — решена раз и навсегда сама по себе, «из коробки», но многие кодеры и тогда, и даже сейчас — смутно это знают и понимают. Такие дела

- Kylix был неплохим начинанием, однако выпилен [по слухам] в угоду [микрософту](#) в обмен на .NET-ништяки. Но не смотря на это, скомпилированные в Kylix 3 (или CrossKylix) бинарники вполне нейллюзорно могут работать под CentOS 5.5, OpenSuSE 11 и даже Ubuntu 10.04 LTS — просто редкий школьник способен обнаружить данный факт.
- После релиза Delphi XE в 2011 году, по официальным заявлениям Embarcadero на одной из конференций, снова добавится поддержка Linux и прочие плюшки на зло этим вашим VS. Однако пока это лишь обещания, да будет превентивно записано в фэйлы.

Что же есть Delphi?

[Педивикия](#) о Delphi:

Delphi (по-русски обычно произносят [бóрланд дэ́лфи] или [бóрланд дэлфи]) — это [интегрированная среда разработки ПО](#) фирмы [Borland](#). Delphi является средой [RAD](#) (от [англ.](#) rapid application development — быстрая разработка приложений).

В 1986 году фирма Apple разработала объектное расширение языка Паскаль, получив в результате Object Pascal. В 1989 году аналогичные объектные средства были добавлены фирмой Borland в систему Turbo Pascal (языковые различия между объектным Turbo Pascal 5.5 и Object Pascal от Apple крайне незначительны). Последующие версии Turbo Pascal, который стал затем называться Borland Pascal, дополняли объектное расширение новыми средствами, этот процесс продолжился и в ходе развития языка системы Delphi. Название языка менялось: от Pascal, через Object Pascal до Delphi, в литературе встречалось название Delphi Pascal, в последних версиях Delphi язык именуется так же, как и сама система (любопытный момент: в Delphi 7 пункт системного меню, запускающий справку по языку, как и в более ранних версиях, называется «Object Pascal Reference», а сам запускаемый им модуль помощи имеет заголовок «Delphi Language Manual», в тексте язык также везде называется «Delphi»). Автор же языка — расовый швейцар Мыкола Вирт — заявляет, что язык, используемый в Delphi — это Delphi (всегда ваш, К. О.), а не [изобретённый](#) им Паскаль.

Матчасть

Система типов этого языка включает в себя ссылочные типы, выраженные словом class, и типы-значения, именуемые «записями» (record). Отличие таких типов состоит в том, что, объявляя экземпляр ссылочного типа, вы получаете ссылку на объект в куче, для инициализации которой необходимо явно вызвать конструктор, имеющий вид статического метода с названием Create(), в случае со записями получаем экземпляр, созданный на стеке, либо внедрённого в экземпляр класса, если речь идёт о полях значащего типа. Чуть менее, чем полное незнание предмета порождает среди школоты мифы о «двух моделях ООП», поддерживаемых Delphi, и срачи на тему object vs class. Анон, запомни: Delphi — статически типизированный язык, ни о каких различных моделях ООП касательно этого языка речи быть не может. На самом деле, если копать глубже, "вторая модель ООП" в Delphi таки есть. Это старый-добрый object, доставшийся в наследство от Borland Pascal, который сам по себе не является ссылочным типом (но на него можно сделать ссылку, как и на всё остальное, если надо), а как и все остальные паскалевские типы, получает на себя память сразу при объявлении переменной. При переходе к Delphi ещё 1.0 от этого подхода отказались - всё равно для полноценного ООП нужно создавать ссылку, а лишнее нагромождение знаков ^ выглядит некрасиво. Но для совместимости с наработками, особенно TurboVision (и немного - с Object Windows Library) это всё оставили, и держат аж вплоть до Delphi 10, и ещё могут держать. Тем не менее, использовать эти object иначе как в древних наработках, которые, возможно, продолжают работать в консольном окне без серьёзных правок, не рекомендуется. А если в одном модуле будут встречаться и object, и class - проблем не оберёшься (чего стоит TObject, который есть в обеих моделях, и, очевидно, и близко не совместим).

Free Pascal

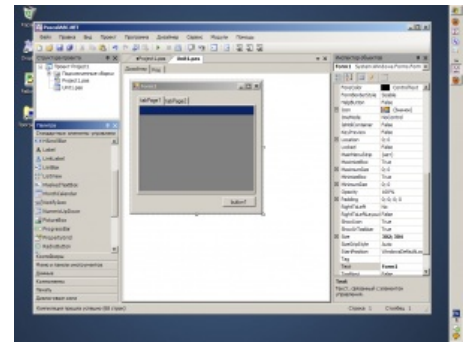
Существует также вполне себе нормальный компилятор Pascal. Поддерживает все возможные диалекты, но с видимым уклоном в сторону подражания Borland. В нем даже есть свой клон Turbo Vision. В процессе развития в него запилили порты библиотек и возможность компиляции кода написанного для Delphi, gpc, hp/ux, МАКовских и ИСОшных стандартов паскаля и вообще всего, что Pascal хотя бы отдаленно напоминает. Также, в отличие от своего старшего собрата, [прибитого гвоздями к x86](#), реально является кроссплатформенным и посему [винраден чуть более, чем полностью](#). А ещё есть IDE [Lazarus](#), внешне похожее на Delphi, в которое даже можно импортировать некоторые сторонние компоненты для той самой Delphi (ну и встроенные VCL клоны тоже многие уже переписаны, при том реально работают и под Linux и под Mac OS X, ну и под винду — ясен пончик). Правда, крайне тупой анон жалуется на размер получаемых им экзешников — дескать гигантский даже в сравнении с Delphi, не зная про нужные ключики^[1].

Применение на практике сего продукта требует готовности к борьбе с сиюминутным недоделками типа несоответствия прототипов библиотечным функциям или полному отсутствию оных в стандартных модулях. Однако при должной осторожности, если не наступишь на мину, можно получить годный бинарник, который будет работать под всеми линуксами вне зависимости от версий этих ваших libc.

PascalABC

Также была разработана действительно **труЪ** (только для обучения) версия Паскалю — PascalABC (устарела морально и физически, больше не развивается) и **PascalABC.Net**. Хотя бы встроенный задачник стоит того, чтобы взглянуть на эту систему (но он, сука, **платный** и **требует регистрации**).

PascalABC появился на свет усилиями измученных Турбо Паскалем преподавателей одного смешного университета, потому как в Windows эти ваши пенсионерские Crt, InitGraph и прочее **говно мамонта** дико бесили. Система программирования получилась довольно приятная, использовалась для обучения школьников и первокурсников, и поначалу успешно маскировалась под старших старших товарищей TP/FP/Delphi. А потом разработчиков накрыло – с ростом проекта поддерживать и развивать его становилось всё сложнее, PascalABC похерили и перепилили всё под .NET, чем обеспечили стогание кучи стульев у обладателей «зверь-сборок» Windows XP и прочего раритета (ибо требуется Windows с вмняемой версией установленной платформы .NET). Но теперь в PascalABC.Net стало возможно писать код практически на **C#** (то есть пользоваться почти всеми возможностями платформы .NET).



PascalABC.NET с формочками

И тут Остапа понесло! Всё ещё немного оглядываясь на совместимость со старичком Turbo Pascal (ибо **ЕГЭ** никто не отменял, а там до сих пор насилуют труп TP 7.0), в PascalABC.NET начали тащить из других языков программирования всё, что под руку попадалось – объявление переменных в коде, автовывод типов, кортежи, лямбды, запросы, срезы, и т.д. Сверху приправили это всё отличными модулями для работы с графикой – и растровой (классика), и графическими объектами, и 3D туда же. Как вам 3D-анимация в пару строк кода? Да, и про .NET не забываем – почти всё, что можно написать на **C#**, с косметическими изменениями будет работать и в PascalABC.NET.

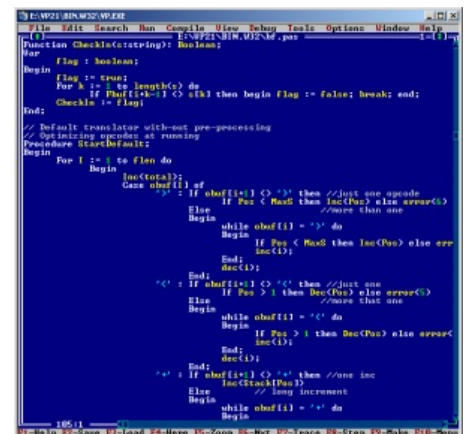
Были, конечно, и неудачи – например, веб-среда WDE для написания программ прямо в браузере потихоньку загнулась, не хватало сил на её поддержку. Или, к примеру, редактор форм тоже особо не развивается – разрабы вмняемые люди, и понимают, что школьников и без клепания формочек есть чему учить, да и студентов насилуют Паскалем всего один семестр, после чего спокойно пересаживают на **C++**.

Получилась лютая солянка из стилей и возможностей различных языков программирования, школьные учителя в прострации, и не успевают за постоянными изменениями в языке, предпочитая забыть на это дело и учить старому доброму Turbo Pascal 7.0, используя PascalABC.NET просто как среду для запуска программ. Это создаёт PascalABC.Net репутацию устаревшего и «детского», что в корне неверно! Если найдёте хорошую современную литературу по языку, то окажется, что это прекрасный и современный язык программирования, идеально подходящий для обучения.

Отдельно следует упомянуть проблемы с **ЕГЭ**, ведь PascalABC.NET по лаконичности и эффективности не уступает этому вашему Python'у, позволяя решать многие задачи в пару строк и поиметь при этом проблемы от какого-нибудь проверяющего работы старого пердуна с комментарием «Так писать на Паскале нельзя!», **sad but true**. Ситуация потихоньку меняется, но всё ещё остаётся довольно печальной, поэтому даже знающие современные возможности языка преподаватели предпочитают натаскивать школьников к **ЕГЭ** по информатике с использованием только подмножества языка, совместимого со старым TP.

Virtual Pascal

В довершение, нельзя не отметить еще один вполне годный компилятор — Virtual Pascal. Представляет из себя 32-битный компилятор с заявкой на кроссплатформенность. Имеет встроенную IDE с подробной помощью и отладчик. Может создавать исполнимый код для операционных систем OS/2, Win32 и ограниченной поддержкой линукс через костыли, навроде EMX. Впрочем для линупса никто на нем не писал, а вот для олдфажной OS/2 вполне кодили, пока в Free Pascal глюк на глюке сидел и глюком погонял. Раньше был платный, ныне бесплатен и доступен для свободного скачивания. Увы, проект заброшен с 2005 года. Поговаривают, что он написан чуть менее чем полностью на ассемблере, из-за чего поддержка кода адово затруднена. **Исходники закрыты**. С другой стороны, можно сказать, что Virtual Pascal является законченной программой. Реализована полная совместимость с языками Turbo Pascal (v7.0), Borland Delphi (v2.0) и Free Pascal (v1.0.x).

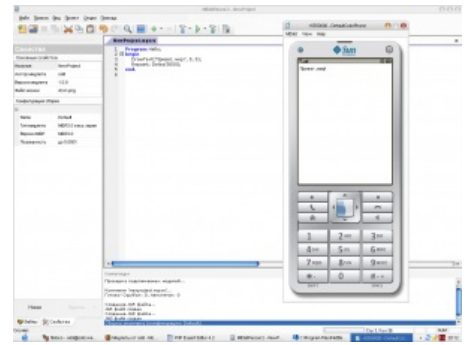


Virtual Pascal v2.1 win32

Почему именно Virtual Pascal? Размер создаваемого кода — вот главная фишка компилятора. Так, например, консольная программа, выводящая Hello World, скомпилированная Virtual Pascal v 2.1 занимает 11,776 байт. Та же программа, скомпилированная Free Pascal v2.4.0 — 31,548 байт. Delphi v7.0 — 15,350 . С оптимизацией тоже дела обстоят неплохо, иногда получается быстрее Free Pascal. Знакомый всем с детства IDE также доставляет (хотя она же давно есть и во Free Pascal).

MIDlet Pascal

Это ещё один занятный компилятор Pascal. Позволяет создавать программы для мобильных под **Java ME**, причём компилирует сразу с Паскаля в **жаба-байткод**. Тормозной (впрочем, небольшие программы компилирует довольно быстро, а вот по-настоящему сложные проекты компилирует минут по десять, чем эпично сливает NetBeans) и не слишком удобный инструмент для создания приложений и игр для мобилы. Кроме того встроенная функциональность обрезана по самые яйца: никакого ООП, никаких ссылок, следственно вообще никаких динамических переменных (только статика, только хардкор), никаких параметров-переменных в функциях (если хочешь передать наружу более чем одно значение, то тебе помогут только глобальные переменные). Распространяется и работает бесплатно. Имеет бешеную популярность среди школоты, которая только недавно учила Паскаль в школе на информатике, а тут вдруг выяснилось, что на данном языке можно писать и что-то более востребованное для рынка. Увы, из-за отсутствия прямых рук и верхней головы неспособны овладеть даже таким элементарным языком и занимаются только тем, что просят помощи у более продвинутых товарищей на [одном широко известном в узких кругах форуме](#). Кроме того, MP имеет возможность подключения библиотек на Java, что более чем актуально, ибо сам по себе язык скуден.



MIDlet Pascal 2.02 Rus

Достоинства языка Паскаль

- Использование вполне понятных английских слов `begin`, `end`, `or`, `and`, `not` вместо фигурных скобок и закорючек.
- Строгая типизация — защита от досадной возможности вступить в компост в парадном, [выстрелить себе в ногу](#) при этом, и от прочих видов мазохизма, так любимых писателями на C/C++. Ну и от дураков алсо (хотя от них помогает обычно плохо, точнее ровно до той поры пока они не узнают про касты).
- Элементы массива можно нумеровать начиная хоть с нуля, хоть с единицы, хоть даже с тысячи (а не только с нуля).
- Возможность прямой работы с памятью практически полностью приближает Pascal к C/C++.
- Внятные языковые конструкции, то есть «*» — это «умножить», а не «умножить, указатель, разыменование указателя, указатель на функцию, возвращающую указатель, или вообще поди пойми чего там в C++ перегружено, etc». Также в Паскале отсутствуют знаменитые инкременты `++i` `++i` (хотя есть аналог в виде процедуры `inc()`), ввиду чего программмер больше времени посвящает собственно решению задачи, а не [выносу мозга](#) себе и [остальным](#) вида «почему 13, а не 14» или «почему 14, а не 13» (нужное подчеркнуть) — в этом языке напрочь отсутствуют операторы, способные изменять значения переменных внутри выражений (хотя в новой Delphi компилятор поддерживает `record helpers for intrinsic types`, при желании можно реализовать нужную функцию, и писать конструкции вроде `i.PlusPlus + i.PlusPlus`, при некоторых ограничениях), что более чем разумно, а случаев неопределённого поведения нет вовсе.
- Благодаря полной поддержке ООП в паскале (а точнее в некоторых его потомках, да и не полная), можно создать мутанта Процедурного Программирования и Объектно-Ориентированного Программирования. Некоторые языки нервно курят в стороне.
- Таки язык программирования промышленных контроллеров (ПЛК) — язык ST (aka SCL) основан на Паскале, и учившему Паскаль освоить его зело легко.
- Лёгкое и быстрое изучение. Delphi вообще даёт возможность приступить к работе без годов задрочерства.
- Лёгкое и понятное построение GUI, MFC на этом фоне смотрится вы знаете как.

Достоинства формочек Delphi

- Быстрая компиляция по сравнению с любой IDE с C++ — проект на Delphi в миллион строк компилируется и собирается за секунды, тогда как C++ ники обычно тоскливо втыкают в процесс сборки минутами
- Намного более читабельный и даже более лаконичный код, не смотря на чуть более громоздкие ключевые слова языка (достаточно посмотреть на любой код на C++, написанный с использованием MFC, Qt, GTK+, с аналогичным на VCL, чтобы понять истинность данного утверждения), хотя в принципе, лаконичность исходников наблюдается только потому, что большая часть функциональности уже вшита в родные или купленные сторонние компоненты (однако если даже посмотреть на один и тот же код, просто написанный на C++ и Pascal, к примеру, реализацию уасс, то миф о какой-то особой лаконичности C++ также теряет практический смысл). А стоит попробовать изобрести велосипед, чем обычно занимаются C++ ники (у которых обычно нет компонент, и вообще библиотеки недоразвиты, да и использовать чужой код не позволяет ЧСВ), и размер папки с проектом также начинает невообразно расти. Но вот читабельностью кода Pascal, да, изначально принципиально превосходит Си, ибо на Си наводит «чёрную магию» в одну строку, которая непонятно почему работает», гораздо проще (по этой причине даже случился эпичный фейл с переписыванием 7zip под Delphi, когда быдлокодер-портировщик ниасилил код на C++, и реализовал ее по-своему, неоптимизированно, из-за чего в бенчмарках код 7zip, скомпилированный на Delphi — стал проигрывать аналогичному на C++ где-то в два-три раза, правда лень уже искать места с обсуждением этого факта, выловленного

профилировщиком).

- В Delphi есть уже готовые и весьма разнообразные средства работы с БД (всякие там ADO, FIB+, ODBC и прочие AnyDAC, за которые нужно выложить приличную сумму), до которых C++нудым еще [фапать](#) и [фапать](#) (и которые типовому C++ку в голову вообще не придет использовать в повседневной работе — он сразу же начнет свои писать, особо [оптимизированные](#) велосипедные, затратив сразу своего времени на сумму, раз в двадцать превышающую стоимость какого-нибудь AnyDAC, а потом потратив еще по столько же раз десять, но уже в последующие лет пять исправлений постоянных глюков своего чудного велосипеда, вместо того, чтобы делом заниматься). В результате, пока красноглазика на C++ сидят со своим говнокодом, дельфиодеры обычно сидят уже с гешефтом, правда небольшим, но уже!
- Благодаря действительно неплохому (за некоторыми исключениями) редактору форм в Delphi, ровно как и в Borland C++ Builder, можно легко (Use help, Luke!) и быстро наклепать как-то работающий прототип, под пристальным взглядом начальства, без всякого соблюдения минимально разумных правил ООП — просто из собственной лени (однако, открывать такой проект через полгода-год второй раз не захочется). Однако, это и сыграло с продуктом злую шутку (и привлекло [over 9000](#) безруких [щкодников](#)) — вместо изучения [основ](#) программирования, [принципов](#) ООП, быдло обычно быстро учится программировать мышкой — клепать элементы управления, втыкать им нужные свойства и писать «код» в обработчики Button1Click, но не более, справедливо аргументируя «А зачем, если и так все работает?»).

Сперва добейся

Тем не менее, на Delphi написано множество кошерных и православных программ, наиболее известные:

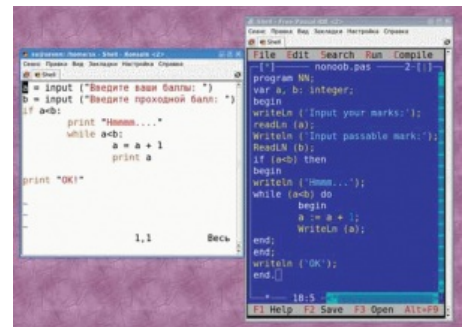
- AdAware
- Age of Wonders
- AIMP2 (и AIMP3 же!)
- Altium Designer
- [ArtMoney](#)
- Auslogics Disk Defrag (и все их продукты)
- AvERP - немецкая ERP система, очередной убийца SAP, типовые клиенты - нищевродские машиностроительные заводы Германии. Огромная текучка кадров.
- BS Player
- D-Fend Reloaded (графическая оболочка для эмулятора [DOSBox](#))
- [Doom 2D: Forever](#)
- Doom: The Roguelike
- Download Master
- EssentialPIM
- FastStone Capture
- FastStone Image Viewer
- [FL Studio \(ex-Fruity Loops\)](#)
- ICE Book Reader
- IconLover
- Inno Setup
- Macromedia Captivate
- Macromedia HomeSite
- MultiEdit
- Partition Magic — C++ Builder
- Goldwave — C++ Builder
- IDA Pro — C++ Builder
- Offline Explorer
- PE Explorer
- PhotoFiltre
- [Photoshop 1.0](#) — часть асм, часть паскаль.
- PHPEdit
- [PHP DevelStudio](#) — среда разработки программ без знаний программирования, которая начиная с середины [2010-x](#) начала использоваться [школотой](#) для создания [вирусов](#), [троянов](#) и прочих [винлоков](#). Ныне переписана на [Java](#) и называется DevelNext.
- PL/SQL Developer
- PortableApps.com Platform
- Resource Hacker
- SASPlanet
- [Scanner](#)
- Старые версии [Skype](#) для Windows (какая-то часть. Чтобы убедиться, достаточно заглянуть в ехе-файл: присутствуют дельфовые TApplication, TWinControl, TForm и прочие имена VCL). Ныне переписан на HTML5/JavaScript и является веб-версией, обёрнутой в кастрированный [Chrome](#).
- Space Empires 4,5 (игра 2000)
- Space Rangers 2 (игра 2004)
- Spybot — Search & Destroy
- The Bat!
- The KMPlayer (не тот [KMPlayer](#), который для [KDE](#), а [другой](#), сделанный в [Южной Корее](#) и, по некоторым данным, нарушающий [GPL](#))
- TOAD
- Total Commander (а то!)

- TuneUp Utilities
- UltraISO
- UVScreenCamera (программа для записи обучающих видеороликов с экрана компьютера)
- Webroot Spy Sweeper
- WinRAR до версии 3.80 (2008 года выпуска) включительно компилировался с помощью Borland C++ Builder. Начиная с версии 3.90 (2009 года выпуска) для компиляции WinRAR используется Microsoft Visual Studio.
- Карточная игра в дурака (англ. *Durbetsel*)
- Поднебесье
- Тысячи их!
- Ну и да, ненавистный всем QIP. Который, вообще-то, был вполне себе кошерной программкой, прежде чем его автор связался с РБК и скатил проект в странное говно.
- Алсо, первый вариант этого вашего кошерного GCC был напейсан самим Столлманом на посцале.

Во избежание срачей следует заметить: вышеприведенный список нужен не для демонстрации превосходства над другими языками, а чтобы показать, что сам по себе Паскаль (и даже его реализация — Delphi) вполне кошерен и годен к использованию.

Недостатки языка Pascal

- Сравнительно громоздкие конструкции языка. Использование длинных английских слов `begin` и `end` вместо фигурных скобок. В IDE они обычно выделяются шрифтом, и ни у кого проблем не вызывают, кроме тех, кто привык к С-подобным языкам — их мозг упорно пытается воспринять их как названия функций/операторов/объектов. Их дольше набирать (5 и 3 байта вместо одного). Они не создают подсознательного впечатления графических элементов. Что особенно не доставляет, когда `begin/end` вкладываются друг в друга 3 и более раз — от обилия `begin`'ов появляется лёгкое головокружение и для любителя C++ код становится неудобочитаемым. (Use code formatting, Luke!) То же самое касается логических и двоичных операторов. Ну а краткость и красота ООП же заставляет просто рыдать от счастья.
- Статическая типизация — защита от явно ненормальных программистов. Хотя, использование `move(..)` и явное приведение типов позволяет присвоить что угодно чему угодно (впрочем, во многих языках точно так же, кроме некоторых особо извращенных с динамической типизацией).
- Слишком низкий порог вхождения, как ни странно — даже поболее низкий, чем у C# и Java, что порождает любовь школьников, студентов, преподав и просто случайных прохожих к этому языку. Однако после небольшого анализа рынка труда оные обычно дружно идут заново осваивать 1С, C# или Java (на самом деле — учиться программировать хоть как-нибудь и что-нибудь полезное), порождая кучи ненависти к Delphi лично и громогласно потрясая «мнением», что Delphi дескать мертв и буржуи на нем, дескать, вообще ничего не пишут, и вакансий и в РФ нет в частности [1].
- Локальные переменные, как правило, нельзя определять в том месте, где они начинают использоваться (кроме PascalABC.NET) — это, по мнению профессионалов C++/C#/Java — приводит к засорению пространства видимости, проблемам с оптимизацией (привет цикл `for`!) и нервному тик у кодера, который должен лезть через 2 страницы (впрочем, не обязательно ему это делать самостоятельно — современные средства разработки, в том числе даже бесплатный Lazarus, умеют определять локальную переменную и даже предполагать её тип из операторов присваивания по одной команде) в блок `var` (да, представьте, какой ужас, переменные в Pascal/Delphi нельзя определять где попало, а только в строго отведенных для того местах, что противоречит принципу «локальные переменные — это не такой дюже важный объект, чтобы ему уделять столько внимания»). При этом, конечно, типичный студент, ненавидящий эти `var` в Delphi как-то и не догадывается, что писать метод, который не вмещается в экран (то есть больше 10-15 строк) — это признак явного непрофессионализма и непрочитанной книги Refactoring. Однако человек, таки прочитавший Фаулера, замечает, что Extract Method применяется уже ПОСЛЕ того как метод разросся на 2-3 экрана. А вот человек, таки не только прочитавший, но и осмысливший Фаулера и Макконела, замечает, что у нормальных программистов ничего не разрастается на 2-3 экрана, они сразу делают функциональную декомпозицию, если видят, что метод не вписывается в 75% экрана. Однако другой человек, который реально применял рефакторинг в реальных фирмах скажет, что рефакторинг без юнит тестов проводить НЕЛЬЗЯ. А так как в дельфи полноценной поддержки тестов с моками и стабами нет (The Art Of Unit Testing с примерами на ПАСКАЛЕ? Нет, не слышали.), то ни один ВМЕНЯЕМЫЙ прогер не будет делать рефакторинг БЕЗ тестов. Never touch the running system. Oga.
- Case-insensitive. То есть, `begin` = `Begin` = `BEGIN` = `BeGiN`. Вот это раздражает, потому что хорошему программисту далеко не пох: хоть он всё равно пишет идентификаторы в правильном регистре, но чужой код, без соблюдения стиля — может довести до рвотных инстинктов (см. исходники Free Pascal). В то же время служит важной защитой от быдлокода со введёнными похожими переменными вида `BiasX`, `BIASX` и `biasX`, разницу между назначением которых забывает сам же быдлокодер через полчаса.
- Лишний `then` после условий `if`. Алсо, перед `else` никогда не ставится точка с запятой. Никогда, сука!!!



Free Pascal против Python.
Почувствуй разницу, школьник!



Типичная программа на дельфи

После end ставится всегда (не всегда, можно не ставить точку с запятой перед end). После последнего end **ВНЕЗАПНО** ставится точка. Что как бы **символизирует** автора.

- Delphi провоцирует **быдлокодеров** на внесение логики во всякие непотребные места: в код обработки событий (**w:Магическая кнопка**), в обработчики события Button1Click, что есть нарушение этой модели Model-Controller-View.
- Как ни странно, основной: Pascal/Delphi лютой **НЕНАВИСТЬЮ** ненавидели в университетах этого вашего **Пиндостана**. Как следствие — Паскаль оказался дико нелюбим всеми этими вашими программистами из Калифорнии, которые, как ни странно, в основном и занимали аутсорсеров в этих ваших **странах**, что привело к тому, что привело. Как и яблофагия, Паскаль, в какой-то степени, пал жертвой моды, так как количество ещё более уёбищных языков, цветущих и пахнущих (достаточно вспомнить ООП-уродство от Страуструппа, которое плодило совершенно уродский код до того, как вокруг него возвели ограничения).
- Unicode появился только в Delphi 2009, благодаря чему переносимость кода на другие версии такая ахуенная.
- Толстые прожекты заточенные под когда-то кошерный Delphi 7 могут быть перенесены на Delphi 2007 только с помощью напильника и бубна. На Delphi 2009 и более поздние версии только с помощью **утюга и паяльника** или не могут быть перенесены **вовсе**.
- Литература. Из-за любви быдла к этому языку, книги пишутся той же самой аудиторией (Флёнов и т. п. профессора Аналы-хакеры). Самое большее что из синтаксиса там описывается — что такое begin, end, if, for. Остальное — VCL, как тыкать формочку.
- Отсутствие GC. Вообще, недостаток двуликий — с обратной стороны это очень нужно. Но, как язык для системного кодирования Delphi не слишком то предназначен, а отсутствие сборки мусора вызывает бурление мозгов у быдла, которое не удаляет объекты (подобие GC + куча сомнительной хуйты запилили в XE4 для ARM компилятора). Капитан очевидность добавляет, что GC на самом деле в Delphi есть изначально (правда сделан был только для строк). GC есть и в случае этих ваших Interface, _AddRef, Interface, _Release, кому ну очень сильно надо (подсчет ссылок). Да компонентная модель Parent-Owner также автоматически занимается вопросами удаления мусора. Говоря проще — элементы GC в Pascal/Delphi есть (там где оно реально нужно и полезно, а не тупо принудительно везде), но считается его что нет вообще, ибо нефиг. И вообще, в новых дельфах легко кодится свой GC, **с подсчетом ссылок и дженериками**.
- IDE. Начнём с того, что выбор нормальной IDE небогат, тут или бажная RAD Studio стоящая немалых вечнозеленых, либо бажный Lazarus, но зато бесплатный. После Delphi 7-й версии до сей поры не было выпущено ни одной быстро работающей и устанавливаемой с первого раза, без костылей, IDE. Деталью нормальности она начала обрастать только с XE версии.

А вообще, полный разбор недостатков Паскаля был «выполнен» Брайаном Керниганом в статье «Почему Паскаль не является моим любимым языком программирования». Оценить градус неадекватности автора можно [тут](#), или в каком-никаком, но переводе [тут](#).

Это, правда, слабоактуальный даже на момент выпуска статьи бред из конца 70-х, но все равно кагбе намекает. Вирт по выходу сего сборника пытался вяло отгавкиваться, что он-де сторонник идеи о том, что для каждой задачи необходим свой инструмент, и что Паскаль разрабатывался только как учебный продукт и ни для чего более использоваться **не должен**, но никого не убедил. А потом и вовсе запилит Модулу-2, оказавшуюся таким **хтоническим пиздецом**, что он аж сам испугался, и, узрев свет энтерпрайза, ушёл писать сборщик мусора для **жабки**. При этом особенно доставляет то, что эта смешная и неадекватная статья вышла в начале 1980-х, когда уже существовали более развитые диалекты Паскаля, избавленные от большинства указанных в статье пороков, но которые, похоже, Кернигану были принципиально неизвестны. Некоторые недостатки Паскаля были также исправлены в ISO-стандарте 1982 года, в частности, в языке появились открытые массивы, давшие возможность использовать одни и те же процедуры для обработки одномерных массивов различных размеров. Никлаус Вирт понимал недостатки созданного им языка, но, следуя традициям академической среды и собственным принципам, согласно которым «неподходящий инструмент надо не исправлять, а заменять», не стал его самостоятельно развивать дальше, а разработал новые языки семейства: Модула-2 и Oberon, а на Паскаль положил шланг. Вот только Oberon и Модула так и остались популярны в узких кругах (то, что на Модуле написаны модули управления Буран, а также современными спутниками — не считается). При этом типовой Delphi-кун искренне верит, что Вирт после создания Паскаля впал в маразм и перестал понимать разницу между практическим инструментом и академическим упражнением (не верящим — достаточно посмотреть на blackbox, как реализацию Oberon).

Глубинные причины ненависти к Delphi/Pascal

В Интернетях уже стойко закрепилась ситуация, когда постоянно, буквально уже на третьем или четвертом посте в форумах (да и вообще где бы то ни было) какой-нибудь долбоеб нет-нет да и обязательно вставит свои пять копеек про то, что:

- Delphi мертв
- на Delphi никто ничего не пишет ни в США, ни в Европе
- на Delphi нет приличных вакансий в РФ, а то, что есть — платят гроши и заставляют допиливать старые задачи, сидя на коробках из-под мониторов перед ЭЛТ-мониторами в госучреждениях
- на Delphi никто не открывает новых проектов
- Delphi давно продан непонятно кому, и уже не развивается (ну... версии 2006, 2007, 2009, 2010, XE, XE2, XE3, XE4, XE5, XE6, XE7, XE8 уже не торты, само собой)
- вся команда из Borland ушла в Microsoft
- на Delphi можно только программировать мышкой, и ничего сложнее двух кнопок и поля ввода написать

нельзя

и тысячи, тысячи подобной ерунды.

На Delphi и CBuilder написана целая российская SCADA-система и система программирования логических контроллеров, это вам не палец обоссать, и не ОБНОм воспользоваться, импортозамещение во все поля. Что, как несложно догадаться, не может не вызывать недоумения, ибо вроде как продукт уже мертв, а почему-то и регулярно его новые версии выходят, и работа твоя никуда не девается, и новых проектов на следующий год сейлзы навалили уже пару штук, а нормальных сотрудников с опытом работы даже в Москве просто так не найти, и все нормальные хотят ЗП строго от 80-90 штук на руки, и это еще без бонусов (что, к слову, соответствует зарплате .NET/Java джуна без опыта), и почему-то до сих пор ничего в родной фирме (и во всех трех, в которых удалось поработать за последние лет десять) так и не переписали на .NET (хотя все собирались и даже .NET-чиков наняли, аж два отдела). И т. д.

Так стоит ли разобраться в ситуации? Откуда столько мегатонн ненавистей к, в общем-то, вполне обычному коммерческому продукту для сверхбыстрого написания всяких окошек к базам данных и не только?

А причины, наш дорогой читатель, просты и очевидны, лежат, так сказать, на поверхности: человеческий, сука, фактор.

Вернемся в школьные и студенческие годы нашего типового будущего говнокодера, но пока еще такого юного и полного подающих надежд дарования. В чем суть его радужного детства-студенчества? Правильно! Он старательно изучает в школе Pascal (Turbo/Free), и потому что преподаватель информатики ничего особо другого и не знает, а будучи помещен в ВУЗ-ик — наше дарование натужно изучает много чего, от С++ и Java, но вот почему-то все курсовые получается писать только на Delphi, да и многие однокурсники, хоть и пускают сопливые пузыри про Ruby и .NET, но курсовой по дискретной математике, с рисованием графов, все равно шпарят на Delphi.

Вот и получается, что отучившись в ВУЗе, наше юное дарование кое-как умеет программировать на Delphi, в теории знает основы .NET и/или Java (в рамках семестра), ну и туманно помнит, что в результате целой ночи так и не написало лабораторную на С++, ибо, блин, как же непонятны эти дурацкие указатели там. Ну и PHP еще, слышал, есть такой язык...

Но... это все хорошо. Но диплом защищен, ура-ура, пора бы уже устраиваться и на работу, правда (ну или третий курс уже стукнул, и тоже поработать хочется)? И... вот тут и случается Epic fail! Что имеет и что получает наше дарование?

Во-первых, ЧСВ изначально дает о себе знать. Ибо или красный диплом получен, или самомнение о себе явно зашкаливает до уровня второго если не Билла Гейца, то Торвальдса.

Во-вторых... а вот реалии почему-то говорят строго об обратном. Попытка собеседования вызывает неумолимый, жгучий, и очень даже реальный **Butthurt!** Дяденька в очках на собеседовании (на всех двадцати фирмах куда была попытка устроиться со знанием любимого Delphi) весьма жестко, непонятными вопросами и задачками показывает, что ты — практически полный d'Bill и программировать умеешь чуть менее, чем вообще не умеешь (а реалии таковы, что на имеющиеся вакансии Delphi действительно нужны прокачанные годами не кислые скиллы, ступидентов туда уже давно не берут, ибо это чревато совсем неуправляемыми потоками говнокода вплоть до потери бизнеса). А даже те дяденьки, которые говорят «ну хорошо, мы подумаем» — почему-то не перезванивают! А это больно! Это очень больно! А папа, мама настойчиво намекают, что пора, сынок, и на работу устраиваться, не хотят они за комнату съемную в Москве твою платить, кризис, дескать. И за одежду и питание, пивасик — тоже (а это ваще плохо).

В-третьих, посмотрев по сторонам, наше дарование весьма неиллюзорно убеждается, что более расторопные одноклассники быстренько уже понаустроивались во всякие фирмы, где нужны PHP, Python, .NET и даже Java-кодеры. И вообще без опыта работы! Вообще, просто пришли говорят, порешали задачи на сообразительность, и все, взяли. Во как!

И тут наше дарование принимает стратегически верное решение. Покупает книги вида «PHP5 и MySQL», «Программирование на ASP.NET», «Ruby on Rails», «Основы Tomcat», «Python» Лутца и «Django» Головатого, изучает их все, и... с пятой попытки устраивается хоть куда-то. WIN! PROFIT!

И... ну короче, надоело уже. Все вы уже поняли. Юное дарование получает неплохие денежки (как правило, больше, чем в вышеупомянутых Delphi вакансиях, на которые не берут студентов), начинает прилично питаться, потом начинает даже говнокодить (через год) что-то уже самостоятельно, а не под строгим наблюдением рядом сидящего говнокодера с чуть менее нулевым, чем у тебя самого опытом, даже снимает через года два отдельную квартирку. И... понимает! Что во всем виновата Delphi! Во всем виноваты преподаватели! Которые совали ему мерзкую Delphi, он ее всю изучил, а она не дала ему зарабатывать даже на питание, одежду и съем квартиры...

Впоследствии это тело дорастает даже до тимлида какого-нибудь, в ЕРАМ-е, процесс окукливания пациента в его ненависти к Delphi окончательно цементируется, и уже вполне половозрелый балбес лет 27 от роду решает донести миру ПРАВДУ!

Примеры ПРАВДЫ про Delphi

Собственно ПРАВДА



НЕНАВИСТЬ!

Данный текст содержит зашкаливающее количество НЕНАВИСТИ. Мы настоятельно рекомендуем убрать от мониторов людей, животных со слабой психикой, кормящих женщин и детей.

Народный эпос правды про Delphi сам по себе достоин отдельного изучения и даже, можно сказать, осмысления, потому мы, пожалуй, начнем собирать в этом разделе особенно, ацки доставляющие примеры, которые периодически сюда вносят правками некоторые, особо одаренные и интересные личности (привет Ынтырпрай!), сохраняя стилистику, орфографию и прочую грамматику аффторов

эта ваша ПРАВДА практически без изменения, итак:

- Делфи — это быдлокодерская среда разработки, созданная в недрах быдлофирмы Багланд как запоздалая попытка создать конкурента Visual Basic, который в Цивилизованном Мире уже получил к 95-му году заслуженную популярность благодаря своей простоте и дешевизне.
- то что ты сейчас пытаешься сделать (использовать ассемблер в программах на Delphi) — ошибка. Генетическая. Российская. Всей образовательной системы страны. Таких преподав надо увольнять сразу же. Если выжигать эту дурь из мозгов каленым железом не получается. Правильно им копейки платят. Они это заслужили.
- Создание любого маломальски сложного приложения БД начинается с очень долгого и мучительного выбора компонент грида, отчетов и для доступа к БД — BDE (этот труп продолжают тащить из версии к версии), dbGo, dbExpress, IBX или покупать сторонние. Но для бедных совковых быдлокодеров сделали исключение — сниженные цены. Особо нищие качают у китайцев. Таким образом рядовой набор патриота-нищеврода состоит как минимум из FIBPlus, FastReports, EhLib. Да, а в Access это все уже есть!
- Был такой случай, когда американский программист спросил: а что, в России прогеры так много получают, что у каждого есть делфи энтерпрайз дома. у нас Visual Basic Standard только. И он потом долго смеялся над зарплатой прогера в 100 \$ и ценой диска в 3\$. Причем К. О. как бы намекает, что это высказывание было актуально на начало 2000-х. Анонимус помнит, как декан одного ВУЗа съезвил по поводу зарплат делфи-программистов: от 4 тысяч и ниже. Другой же почтенного возраста, преподающий «современное программирование» в том же ВУЗе, пошутил, что «программист» в этом городе (а может и всем замкадье?) считается средним между «водитой» и «охранником». Как по зарплатам, так и по отношению. Такие дела.
- Вакантные места программистов для писюков захватывает мафия бывших студентов кулинарных техникумов со знанием турбопаскаля
- а вы что наивно полагаете что дельфи не дёргает кучу при копировании дин массивов? он даже при увеличении размера дёргает кучу.
- Free Pascal — поделие немчика с претензией на кроссплатформность и совместимость с дельфями, которое ВНЕЗАПНО начали использовать сами разработчики Дельфы при запиле беты своего компилятора под эти ваши iPhone
- А сейчас политика Багланда — делать деньги на ежегодном выпуске новых версий с минимальными и спорными изменениями, оно и понятно, у Microsoft — есть Windows и Office, у Oracle есть дофига серверного ПО, а у текущего владельца делфи нет ничего, кроме делфи.
- это следствие того, что компилятор вообще «вываливает огромные портянки ошибок», а не выплевывает их по одной.
- старые Delphi программисты — это те, которым в среднем 45-55 лет, толстые, каждый час выходят покурить и имеют очень похожую на них собаку.
- Выпуск под каждую версию отдельного дистрибутива компонентов как бы намекает, что DCU-шки несомместимы между версиями или в случае с поставкой с исходниками делфинисту придется ебаца с заточкой исходников под текущую версию компайлера, если об этом не позаботился производитель с опциями условной компляции)

А библиотеки Java/C# начиная от версии 1.x/2.0 работают без каких либо переделок и требований перекомпиляции

- на Delphi никто ничего не пишет в США и в Европе (ну за исключением Quest, Skype, Allroundautomations, Databacker (интересно, кто же его пишет, и где?) — клепают коробочный софт, который потом никто не покупает — все лежит на прилавках за 1 евро
- В замкадье Delphi вакансий просто НЕТ, все давным давно свалили на 1Ску, а зарплата «выпускника со знанием делфи» < зарплате уборщицы. Анонимус помнит, что в миллионном городе в 2003-году не было НИ ОДНОЙ ВАКАНСИИ по делфи, а был один сплошной 1С. Ну ты понел.

А фирмочка, связавшаяся с делфи продолжает искать годами «суперкрутого специалиста по никому не нужной делфи», пока не придет новый владелец и как жет — мы переходим на .NET. И на самом деле юное дарование на 100% право, делфи — это удел старых пердунов с бородкой, которые сидят на сладком месте с какого нибудь 94-го в фирмочке и обслуживают свою супер-нетленку на каких нибудь делфи 3 + Интербэйс. А айти знаете ли за это время не кисло так шагнул вперед и программисты должны решать совсем другие задачи, где делфи **ВООБЩЕ** нет места.

- Lazarus для Free Pascal на деле оказывается далеко не кроссплатформенным (вы пробовали запускать и компилировать что-то на нем под маки?) и тем более не совместимым с делфи
- Вирт создавал Оберон вообще под действием сильнодействующих веществ, которыми затарился будучи в командировке в Амстердаме. Осознавая бессмыслие что-то изменить, он стал нападать на мейнстрим языки типа Java и C# — что мол все идеи у него сперли буквально из головного мозга. Но поезд ушел и дедушке остается сидеть на пенсии и пить пиво вместе с местными пеннерами.
- В рашке же делфи получил распространение только благодаря дискам с пиратками, купленными на радиорынке за столярник, где вы найдете все версии делфи и еще 1000 крякнутых компонент в придачу. Сейчас варезные версии делфи распространяются через интернет, в том числе и всякие Lite и прочие Edition.
- `for(int i=1;i<10;i+=2);` такова в дельфи не сделать стандартным циклом(Use while, Luke!)
- Когда-то давно пришлось (заставили под страхом смерти делать прогу) писать код на Delphi. Освоившись со средой, начал кодить. Банальнейшая задача по нахождения следующего символа (a->b, e->f и т. д.) на Дельфах обратилась в ад (дельфины: предоставьте, пожалуйста, хороший код, который делает эту задачу, я не осилил (хороший код: **AnsiChar(Ord(chSym)+1))**) (на самом деле **succ(ch)**). Также доставило странное поведение среды: при возникновении ошибки чтения программа молча, без сообщений закрывалось. For у D — самый топорный из ВСЕХ языков. Даже у Basic есть Step, который развеивает грустные мысли. Тут пришлось извращаться, чтобы организовать цикл сложнее «от А до В». Позже доставила таблица импорта у результатов действий компилятора Delphi. Сейчас, возможно, это уже пофиксили — не проверял.
- так детишки как раз и пишут на паскале, а на промышленных языках с C-like синтаксисом пишут серьезные дяди.
- Ну не только, конечно, но я недавно смотрел лекцию Степанова, где он ругал Паскаль за то, что в нём нет арифметики указателей. Это означает, что в Set нельзя положить указатели, так как нельзя вычислить хэш-ключ и прочие полезные штуки нельзя делать, о которых поцкалисты даже не знают.
- Плюсы вообще можно долго изучать и постоянно будет попадаться что-то новое. А паскаль заостенел уже давно.
- операции над целыми числами и над отдельными битами... и их отсутствие^[ЩИТО?] в паскале это эпический фейл последнего...
- Ты вообще хоть знаешь, что такое макроассемблер? Это ассемблер с макросами. Если сделать макрос для FOR, который будет принимать три аргумента — начало цикла, условие продолжения и шаг, то получится почти то же, что и в Си, только в Си пошли дальше и получили язык, почти такой же быстрый, как и ассемблер, но гораздо удобнее его и паскаля, заодно. А в паскале такого намутили, что и сами не могут понять, как это нормально компилировать, потому и имеете вы то, что имеете.
- То что паскаль еще держится говорит о наличии людей которые не могут освоить нормальный язык программирования.
- Всилу обстоятельств и по просьбе знакомых сегодня кодил на дельфи. Столько матов еще ни разу не сгребал в одно предложение, при каждой попытке высказать свое мнение. МЛЯ, язык, где нельзя управлять счетчиками и приходится юзать while там, где он по смыслу не нужен, УЖААААСНАЯ отладка, убогая система «оптимизации», эксепшены типа «выход за границу массива» выбрасываются ЧЕРЕЗ РАЗ и то не всегда, блин, причем не в НОРМАЛЬНОМ виде типа «IndexOutOfRangeException», а как крик оперативки типа «Access violation...at x0000» и т. д.
- помню в школе нам долго объясняли что в паскале = это операция сравнения, а := присвоения... почему-то люди не знавшие ни одного языка этого не понимали...
- with — костыль.
- На Паскаль, а Пасрал (варианты Пасцал или Паскал), Delphi — Дельфин (или Выбляделфи, Быдлодельфа), Делфисты — Дельфинисты, Быдлодельфятники
- нет, просто на дельфи только такие долбо*бы могут программировать :D :D

сорри всем кодерам дельфи, это чисто мое мнение если что :D :D

-Разве Delphi ,мозг так выносит?

Нет, просто писать на нем можно без наличия мозга.

да вряд-ли, человек который хотя бы один раз работал с поинтерами и с битовыми операциями на С никогда не съест бабушки :D :D

- Отсутствие нормальной современной среды, окромя поделки Борланда. Хотя есть Lazarus для Free Pascal, и еще пяток поделок под него-же, ну и даже в саму Visual Studio можно привинтить Pascal, правда в вариации под .NET (см. Delphi Prism, PascalABC.NET), но все равно нормальных сред нет, ну вы поняли.
- Во Всем Мире малораспространен **пруф** (ну да, какие-то несчастные 1.6-2.5% на ТЮБЕ [2], с сезонными колебаниями — то есть где-то на уровне Javascript, между Lisp и Perl), да и кол-во вакансий уже лет 10-ть стремится к нулю, к примеру крайне мало вакансий для специалистов на dice.com — 15 и monster.com — 9, для сравнения: Cobol — 90 на dice.com и 34 на monster — nuff said. В России (Yandex.Rabota) — 140 вакансий Delphi против 1781 программист 1С, 900 на С# и 70 на VB.NET. Но старым Delphi программистам (это которым в среднем 25-35 лет, а не 12-16) пока работы хватает.
- Также малая распространенность связана с никакой маркетинговой политикой производителя и его нежеланием и неумением работать с совсем начинающими программистами и выпускать бесплатные (не говоря уже о свободных и кроссплатформенных) версии своего продукта и распространять их среди учебных заведений [2]. До сих пор не существует бесплатной версии Delphi (TD 2006 уже умер — лицензии просто так не раздаются). Выпущенная недавно Starter Edition «по-цене мобильного» вызвала **тонны ненависти** со стороны самих дельфинистов и подбросила тонны еды троллям. Особенно **доставляет** то, что предоставлена возможность upgrade с Notepad++, Eclipse, Visual Studio. **Нувывонели**. А у практически всех других производителей массового ПО (даже 1С!) есть бесплатная полнофункциональная учебная (и не только) версия — Eclipse, Visual Basic CCE, VB 6 Working Model Edition, Visual Studio Express и Access 2007. Embarcadero в этом отношении практически полностью отошел от своего принципа времен ДОС «слон должен стоить 99 долларов» (тогда это было действительно дешево). А ведь нулевая стоимость вхождения является КРИТИЧЕСКИ важной для нищоброда. В Америке и Европе просто НЕТ программистов Delphi, так как он стоит ОХУЕННО МНОГО для их начального обучения (ну да, аж примерно \$899, то есть 1/5 месячной средней зарплаты переднека в США — чуть дороже, чем хорошая стиральная машина^[3]), ну а дисков со всеми версиями Delphi и 1000 компонентами в придачу там как бэ нет и про торренты в Европах/США тоже якобы не знают — это же страны Эльфов, а не **Мордор** какой-нибудь. Однако КО намекает, что обычный западный студент вряд ли выложит свои кровные 1000 долларов на систему с сомнительными перспективами. А Java/Eclipse и VS Express можно элементарно бесплатно скачать с интернета. Плюс практически полное отсутствие литературы в местной университетской библиотеке [**ЩИТО?**] и аж 361 книга по Delphi на amazon.com в продаже против 30500 по Java — это как бэ говорит нам, никто не спорит даже). Для сравнения — последняя книга в Германии по турбопаскалю вышла в 93-м году [**ЩИТО?**], а в России продолжают тратить лес на макулатуру и выпускать чуть ли не каждый год очередную книжку по сабжу (и их постоянно кто-то покупает — вот идиоты, да?).
- VCL работает по принципу LEGO — пиши из того что есть, а о том, чего нет, без вазелина и не мечтай. Не отследишь, например, отрисовку компонентов — если на форме 100500 кнопок, и каждой в цикле поменять параметр Visible, то каждая же из них и будет перерисовываться (вместо того чтобы потом руками вызвать Repaint всего родительского окна как в MFC). Алсо, TApplication перегрузить не получится.
- Уже давно обратил внимание, что паскаль пользуется хроническим успехом у массы клинических идиотов.

Только за это нужно сказать дедушке Вирту отдельное спасибо. Анонимный аналитик с Лора

- Непродуманная система разделения описательной части и реализации. Ни в одном другом языке, кроме C/C++, где это якобы исторически оправдано поставкой header- и obj- файлов без исходников (отметим, что понятие «без исходников» для сообщества Delphi изначально является малоадекватным бредом — все, даже купленные компоненты обычно поставляются с исходными кодами, хотя в мире Java, С# и С++ все с точностью до наоборот. Делфинисты же раздельное описание это оправдывают практической ненужностью JavaDoc тегов (хотя у них есть всякие эти ваши DelphiDoc) и ненужностью фолдинга в их любимом редакторе (то есть отдельное описание — это на самом деле уже такой способ документации интерфейса модуля, о чем любители Javadoc обычно даже и не догадываются. Во всех других таки нормальных средах/языках пишут сразу заголовок метода с модификаторами доступа, его тело и окончание — пользуясь при этом костылями ~~современными средствами~~ — doc-тегами (и для документации, и для интеллигенсы!) или фолдингом, в том числе и на уровне конструкций if/while/ (затрачивая в на эти глупые и малоосмысленные кликания мышкой: фолдинг/анфолдинг, и на javadoc «документацию» просто невероятное количество непродуктивного

времени, даже не замечая этого, а лазанье от имплементационной к интерфейсной части — это как бэ гут?).

Типичная прога на Пацкале, написанная когда автором, тепер пишушим эту вашу ПРАВДУ про Pascal



НЕНАВИСТЬ!

Данный текст содержит зашкаливающее количество НЕНАВИСТИ. Мы настоятельно рекомендуем убрать от мониторов людей, животных со слабой психикой, кормящих женщин и детей.

```
if Length(PP.curdir)<36 then
  pt:=' '+PP.curdir+' '
  else pt:=' \...'+Copy(PP.curdir,Length(PP.curdir)-30,31)+' ';
Ptr:=PP.f_ptr;
if PP.vol='' then PP.vol:='No Label';
WriteString(1,cc+(40-length(pt)) div 2,pt,Pal[8]);
WriteString(22,cc+1+(cc*(34-length(PP.vol))) div 40,['+PP.vol+'],' ,Pal[7]);
if PP.vol='No Label' then PP.vol:='';
{if PP.empty then exit;}
inc(ptr,sizeof(filrec)*row);
for i:=row to n+row do begin
  if (i <PP.files) and (not PP.empty) then
    begin fname:=ptr^.filename;
      {if (ptr^.fileattr and 16) = 0 then StrLwr(fname);}
      if (Ptr^.fileattr and 2) <> 0
        then fname[1]:=upcase(fname[1]);
      if ((Ptr^.fileattr and Hidden) <> 0) and (Pos('.',Fname)>0)
        then fname[Pos('.',Fname)+1]:=upcase(fname[Pos('.',Fname)+1]);
      SortFile(fname);
      for a := length(fname) to 11 do fname:=fname+' ';
      fname:=fname+'|';
      if (Ptr^.fileattr and 16) = 0
        then Str(Ptr^.filesize:10,pt)
        else if fname[1]='.'
          then pt:='⓪UP---DIR⓪'
          else pt:='⓪SUB--DIR⓪';
      fname:=fname+pt+'|';
      DateTimeCnv(pt,Ptr^.filedtm);
      fname:=fname+pt; {SetFattr}
      if Ptr^.selected then WriteString(2+i-row,cc,fname,Pal[1]) else
        WriteString(2+i-row,cc,fname,Pal[2]);
      inc(ptr,sizeof(filrec));
    end
    else WriteString(2+i-row,cc,' | | | ',Pal[1]);
  end;
end;
procedure Put_File(PP:Panel;cc,row,x:word;active:boolean);
var
  fname,pt:string;
  {ptr:P_filrec;}
begin
  if PP.empty then exit;
  ptr:=PP.F_ptr;
  inc(ptr,sizeof(filrec)*(row+x));
  fname:=ptr^.filename;
  {if (ptr^.fileattr and 16) = 0 then StrLwr(fname);}
  if (Ptr^.fileattr and 2) <> 0 then fname[1]:=upcase(fname[1]);
  if ((Ptr^.fileattr and Hidden) <> 0) and (Pos('.',Fname)>0) then fname[Pos('.',Fname)+1]:=upcase(fname[Pos('.',Fname)+1]);
  SortFile(fname);
  for a := length(fname) to 11 do fname:=fname+' ';
  fname:=fname+'|';
  if (Ptr^.fileattr and 16) = 0 then Str(Ptr^.filesize:10,pt)
  else if fname[1]='.' then pt:='⓪UP---DIR⓪'
  else pt:='⓪SUB--DIR⓪';
  fname:=fname+pt+'|';
  DateTimeCnv(pt,Ptr^.filedtm);
  fname:=fname+pt;
  if active
    then If Ptr^.selected
      then WriteString(2+x,cc,fname,Pal[3])
      else WriteString(2+x,cc,fname,Pal[4])
    else if Ptr^.selected
      then WriteString(2+x,cc,fname,Pal[1])
      else WriteString(2+x,cc,fname,Pal[2])
```

Ночной кошмар (эпичная копипта)

Летом 1930-го мы снова посетили Провиденс, где у дяди был большой старый дом, кажется, доставшийся ему от родственников по материнской линии. Однажды вечером, зайдя в кабинет дяди по какому-то делу, я обнаружил у него довольно странного посетителя. При ближайшем рассмотрении он оказался Джоном Уизерби, молодым художником, которого я встречал пару раз в доме профессора Каулера. Причина, по которой я поначалу не узнал его, была весьма прозаичной - со времен нашей последней встречи Уизерби, казалось, сбросил целый десяток килограммов. Одежда висела на нем, словно мешок, а под глазами пролегли темные круги.

Уизерби показал мне рисунок, по его уверению, сделанный им самим в сомнамбулическом состоянии. Стоило мне только взглянуть на полотно, как волны космического ужаса захлестнули мой разум, и я с трудом поборол в себе желание закричать. В нижней части рисунка был воспроизведен текст на совершенно незнакомом мне, противоестественном языке. Он был словно отражением обычного языка, но искривленным, недоступным человеческому восприятию. Вместо привычного `int arr[4]` я увидел мерзостную запись `arr: array [0..3] of integer`, а еще ниже был приведен вообще недоступный моему пониманию фрагмент: `pElem^ := 4`.

Но не текст вызвал во мне столь неподдельный ужас. Чуть выше на рисунке было изображено ужасное существо с крыльями, отдаленно напоминающее человека, и стояла подпись: БОРЛАНД.

Уизерби рассказал нам, что накануне ему приснился кошмар, чуть было не лишивший его рассудка. Во сне он стоял перед огромной желтой колонной, по своей форме напомнившей ему цифру 7. Уизерби несколько раз повторил, что архитектура колонны была совершенно неправильной, словно не принадлежащей нашему миру. Вокруг колонны ходили ужасающие существа, походившие одновременно и на человека, и на дельфина. Каждую минуту один из них протягивал руки к небу и произносил фразу, видимо, являющуюся частью какого-то отвратительного ритуала: "С'тудий'а мас, тда'и! Холии'вар йей!" Во сне Уизерби откуда-то знал, что дельфины поклоняются огромному сверхсуществу по имени Борланд, также называемому Иин'праиз. Уизерби клялся нам, что знание о существе пришло к нему извне, как будто бы из самых темных глубин космоса.

Охваченные сверхъестественным ужасом, мы с дядей перевели взгляд на картину молодого Уизерби..

Учебники Фаронова

Отдельной иконой российских паскаляфагов являются учебники Валерия Васильевича Фаронова. Учебники представляли собой, особенно для того времени, лютейший вин — там просто, понятно и структурировано объяснились как основы программирования, так и собственно язык. По сути, многие программисты девяностых (включая автора правки) учились быдлокодить именно по этим учебникам. Вообще, надо понимать, что вменяемых учебников по любому языку, что у нас, что за бугром, считанные единицы — 95% авторов выебываются и намешивают в текст какой-то романтической хуйни, вместо того чтобы планомерно излагать смысл. Фаронов — как раз такой.

На данный момент существуют издания Фаронова для всех версий Паскаля и Делфи. Начинающим хакерам крайней рекомендуется — там вам на пальцах объяснят, что такое условные операторы, циклы и объекты. Этого вполне достаточно, чтобы понять большинство современных объектных языков.

Алсо, [интервью с автором](#) и [его труды](#).

Вывод

Pascal (Free Pascal/Delphi) — среди олдфагов считается расово верным языком программирования, но для современных задач (WEB, Enterprise Mobility), которые обычно уже не разрабатываются только под нативный Win32 безнадежно устарел.



Книжные новинки

Но сам язык Pascal, как и Delphi — вовсе не мертв, и новые версии выходят практически каждый год, радуя олдфаговцев и не только. А для эталона мертвости языка советуем болезным обратить взор на такие вещи, как Visual Basic 6.0, Visual FoxPro, J++. Или на BeOS и OS/2.

Кроме того, в виду наличия Free Pascal и Oxygene (где компиляция в .NET и JVM) современный Паскаль демонстрирует практически уникальную кроссплатформенность, не достигнутую даже C/C++. Хотя типовому говнокодеру подобные моменты не нужны и непонятны. К слову, в последних дельфах можно писать даже для Linux, а под все эти ваши айфоны [дельфины давно говнокодят](#).

Классический же Delphi, в текущем виде, весьма удобен для быстрого написания относительно несложных оконных приложений под Windows для массового тиража (вроде Skype или Total Commander, всего-то несколько десятков человеко-лет в сумме, ну вы поняли). В этой дисциплине он выигрывает у C++ в скорости написания, а у .NET — в простоте развертывания (КО как бы говорит нам, что хоть размер инсталлятора .NET Framework 2.0 составляет всего лишь 23 мегабайта, но для инсталляции всех апдейтов, версий .NET Framework 3.0, 3.5, 4.0, 4.5, 4.6, 4.7, 4.8 и вот уже 5.0 нам предлагается скачать и установить 1 гигабайт (то есть 1000 мегабайт), одна только установка этого всего может занять час на машине клиента, да еще и в три этапа с перезагрузкой), ну и в защищенности кода от прямого реинжиниринга (получить исходный код любого скачанного поделя на .NET или Java из его байт кода посредством просто чудных программ вроде

Spices.Net Decompiler, .NET Reflector, JAD decompiler, Mocha и прочие тысячи их — обычно проще пареной репы для любого школьника).

Правда, любой более-менее серьезный софт на .NET или Java защищают с помощью обфускаторов, и разобраться в такой каше не намного легче, чем в ассемблере, но кого это останавливало?

А вот низкоуровневое системное программирование на Pascal (драйверов каких-то там или ещё чего-либо вроде ядер операционных систем) — вообще полное исключение — там традиционно правит исключительно C/C++ (потому что никто DDK под Delphi даже не пробовал портировать в виду мизерности рынка — хотя слухи были).

Также следует отметить, что до недавних пор Pascal практически никак не использовался для разработки Web приложений, так как совершенно не был для этого приспособлен (как и для любого типа серверных приложений). Ололо, это не проблема Pascal, как языка - [просто нехватает годных библиотек.. UniGUI? Нет, не слышали](#). Гру-Web-технологии (Java, ASP.NET, Django, Ruby on Rail и таки PHP) используют совершенно другие подходы (автоматическая сборка мусора, JIT компиляция, динамическая/нестрогая типизация), которые в итоге позволяют в срок написать поддерживаемое приложение не вызывая баттхерт у программистов, его пишущих. Тогда как типичное Delphi приложение — это незащищенный код (как и в C/C++), и типичный чумазый говнокодер, который попытается обратиться по ссылке к уже несуществующему объекту или выйти за границы массива, способен по-тихому разрушить всю память серверного приложения (тогда как в ~~языках~~ технологических платформах вроде Java/Spring/Grails, C#/ASP.NET, PHP/JavaScript/Node.js, Python/Django Ruby/Rails данная типичная проблема решена на фундаментальном уровне, то есть запрещена изначально). Однако, свято место пусто не бывает и теперь его заняла Smart Mobile Studio.

Стоит заметить, что сабжевый компилятор есть и для микроконтроллеров. Поделие одной известной фшыстской конторы для PIC выдает вполне себе годный и вменяемого размера код, а расово сербские сербы запилили целую [линейку](#) компиляторов, начиная от баянных AVR и заканчивая моднявыми ARM и экзотикой типа dsPIC.

Отсюда непосредственно следуют выводы о популярности языка у работодателей и говнокодеров (на самом деле нет проблемы открыть или найти вакансию, проект, проблема скорее найти вменяемого разработчика, за соизмеримую зарплату, так что делайте выводы).

Тем не менее современная среда разработки — это не только язык программирования, но и огромный обвес из прикладных библиотек. К сожалению на данный момент не нашлось достаточного количества платных/бесплатных энтузиастов, которые смогли бы вывести инфраструктуру и синтаксис Delphi на современный уровень.

В целом отказ от Delphi произошел по двум объективным причинам. Во времена расцвета Delphi, когда программы ходили пешком под стол (то есть прямо в базу, что называлось 2x уровневой архитектурой), он активно использовался в корпоративной среде. Но когда программисты смекнули, что не все компьютеры могут иметь Windows, а браузер таки имеют все, даже смартфоны — получила распространение трехуровневая архитектура и сферу корпоративных приложений прочно заняла Java/J2EE. С другой стороны с появлением .NET при активной поддержке Microsoft Delphi был выброшен и с рынка десктопных приложений Windows. Неосведомленному читателю все еще может показаться, что все-таки язык жив, но это скорее стадия вялотекущей поддержки. Об этом красноречиво говорит тот простой факт, что в статистике GitHub по языкам программирования Delphi/Pascal в списке топ-50 просто [нет](#).

См. также

- [Программист](#)
- [Индус](#) — имеют отношения к Pascal, так как программу можно растянуть на >9к строчек из за пластичности языка

Примечания

- 1 ↑ Например, от ключика -Xg, или постановки галочки в настройках — «Использовать внешний отладочный файл» бинарник сразу раз в 7-8 похудеет
- 2 ↑ [в рашке это напоминает скорее распил](#)
- 3 ↑ Тут фишка в том, что обучение в США платное. На обучение чаще всего берут кредит. Так, как у добпорядочного американца еще должен быть кредит на дом, то там становится уже не до дополнительных трат. Впрочем это все уже не актуально, так как теперь Delphi дают для обучения бесплатно.



Языки программирования

++i + ++i 1C AJAX BrainFuck C Sharp C++ Dummy mode Erlang Forth FUBAR
God is real, unless explicitly declared as integer GOTO Haskell Ifconfig Java JavaScript LISP

My other car Oracle Pascal Perl PHP Prolog Pure C Python RegExp Reverse Engineering
Ruby SAP SICP Tcl TeX Xyzzy Анти-паттерн Ассемблер Быдлокодер
Выстрелить себе в ногу Грязный хак Дискета ЕГГОГ Индусский код Инжалид дежице Капча
КОИ-8 Костыль Лог Метод научного тыка Очередь Помолясь Проблема 2000 Программист
Процент эс Рекурсия Свистелки и перделки Спортивное программирование СУБД
Тестировщик Умение разбираться в чужом коде Фаза Луны Фортран Хакер
Языки программирования

[w:Паскаль \(язык программирования\)](#) [en.w:Pascal \(programming language\)](#)