

Владимир Лось

## К вопросу о выборе языка программирования

1999, В.Лось

Письмо в форум написано под впечатлением бурных обсуждений темы "Роль языка программирования в жизни человека" на Базарной площади (<http://www.delphikingdom.com>).

Это не вопрос вкуса, симпатий или нелюбви. Здесь не может быть компромиссов, ибо, если мы вступаем в сделку с совестью или отходим от истины, то получается нечто вроде творений Microsoft. Опыт активного программирования у меня — тринадцать лет. Диапазон проектов — от автоматизации аэродинамической трубы до разработки информационной экспертной системы анализа экономической деятельности министерства. Языки реализации — от ассемблера PDP-11 до C++. География — от Новосибирска до Киева. Кроме того, опыт преподавательской деятельности в вузе (микропроцессоры, ОО-технологии, основы программирования).

Теперь — тишина на фланге поклонников Си! Посидите пять минут спокойно, помолчите, не размахивайте руками, отдохните. Дайте слово паскалисту (если кому-то согреет душу — пасквильанту).

Да! Вы (поклонники Си/C++) сейчас на коне! Ни в одной фирме нельзя быть нанятым на работу без экзамена на знание C++. Посмотрите объявления на работу в Интернете. Полистайте околокомпьютерную журнальную макулатуру (это вообще отдельная песня) — так все и пестрит Си/C++ и Java. Скорее вызывает удивление появление статей, в которых авторы пытаются рассказать о том, что существует и нечто еще в мире языков и операционных систем. А ведь есть!

Для начала небольшое лирическое отступление.

Из опыта преподавания последних лет в вузе, я сделал интересное наблюдение. У студентов (даже первокурсников!) стал формироваться некий «американский» стиль мышления и манеры выполнения заданий. То есть главным при обучении становится процесс набора стандартных решений под типовые ситуации. Призывы к осмыслению принципов, на которых получено то или иное решение, пониманию важности развития решения задачи до обобщенного подхода, встречают все больше не то, что нежелание, а просто непонимание со стороны студентов. Поясню на собственном примере. На нашем потоке (специальность АСУ) училось примерно 55 человек. Где-то треть в той или иной степени были заинтересованы в овладении специальностью. Все 19 человек, по моим сведениям, несмотря на «веселые» времена работают по специальности. Об остальных говорить не будем — то был известный балласт, пришедший в вуз кто за «корочкой», кто за мужем, кто чтобы продлить детство. А что сейчас? Первый курс этого года: 63 души, 3 группы. На шестьдесят три человека только четверо с каким-то пониманием, куда они попали и что им нужно делать! Из остальной пассивной массы где-то половина способна проглотить начитанный материал, с «толкача» уяснить его на лабораторных работах и практике и продержаться до ближайшей сессии. Оставшийся контингент, видимо расстанется с институтом после февраля...

Зачем я все это рассказываю? Затем, что таково сегодняшнее положение вещей. А ведь у меня перед глазами студенты всех пяти с половиной курсов нашей специальности. Можно проследить изменение миропонимания, подходов к обучению и восприятию материала, отношения к специальности вообще и к программированию, в частности. Изменение не только в зависимости от года обучения, но и, как я говорил, от года поступления. И вот тут у меня возникает двоякое чувство. С одной стороны мне страшно за них, жалко своих времени и сил, которые я на них трачу в стенах вуза (тем более, что жизнь за этими стенами вносит свои коррективы в виде базарной экономики, и вечные ценности как-то стали выходить из моды). С другой — возникает чувство дьявольского спокойствия: с такого рода контингентом мне нечего бояться остаться без работы!

Можно опять переспросить: ну и о чем этот плач Ярославны? А о том, что активно и критически мыслящий студент и специалист стал выходить из моды. Мышление, настроенное на накопление суммы решений, просто не в состоянии трезво и критично оценить то, что ему предлагается в качестве средства решения встающих перед ним задач. Получилось в одном месте — получится и в другом! Что об этом пишут в толстых журналах? Ах, там еще и иностранный дядька-менеджер в крахмальной рубашке и при красивом галстуке объясняет, что успех его фирмой достигнут за счет применения этого средства? А что местная околокомпьютерная богема со старшего курса по этому поводу вчера на ВЦ трепала? А посмотрим объявления на сайтах в Интернете в требованиях к кандидатам на работу? И там это средство среди обязательных, которые необходимо знать? Заучу-ка я его раз и навсегда и у меня больше никогда проблем не будет! Это мысли пассивной части, т.н. «среднего» студента. А что те, кто поактивней? У них мысли более оформлены на освоение продукта. Эти и книжки по нему достают (даже перечитывают — и не один раз). К преподавателю пристают с вопросами. Но только тематика вопросов опять-таки странновата. Вопросы все больше не по поводу подходов к решению частей задач, а о способах выражения тех или иных понятий предметных областей решаемых задач средствами языка. Вы можете задать вопрос: а что же тут странного? Люди учатся, у них возникают проблемы при освоении нового материала, они консультируются у преподавателя, а этот преподаватель еще и возмущается. Да, согласен, если бы это были студенты первого-второго года обучения. А если к вам приходит на консультацию не самый глупый лоб с четвертого-пятого курса и начинает задавать вопросы, касающиеся чисто синтаксических конструкций языка, на котором он программирует (по крайней мере подразумевается) уже 4 года?

Поначалу (в первый год преподавательской активности) я с готовностью шел навстречу. В этом была польза и для меня: шлифовались навыки объяснения, заполнялись пробелы, все раскладывалось по полочкам в голове. Но потом я провел анализ зависимости задержек сдачи проектов и количества посещений консультаций от выбранного языка реализации. Догадайтесь о результатах сами. Кто-то опять поднимет тезис примитивности Паскаля. Ребята! Простота и примитивность, согласитесь, разные вещи. Людям (студентам, программистам) надо решать задачи, а не бороться с синтаксическими извращениями, опциями среды и командной строки или ворошить 800-страничные руководства в поисках ответа, почему же что-то идет не так. Я, как системотехник, работаю в терминах предметной области, мне не нужно знать, что при таком-то ключе компилятор с линковщиком разместят элементы данных по адресам кратным четырем, а иначе спячат все вместе, и следить при этом за моей подпрограммой копирования областей памяти. Я могу и не знать о всех частных условиях вызовов конструкторов копирования. Мне более понятно понятие модуля как конкретной единицы компиляции и группирования близких по смысловому значению переменных и функций в файл исходника, чем «размазанное» по нескольким файлам понятие пространства имени. Что легче: прочитать 20-ти страничное руководство с описанием общих принципов, поняв которые, можно обращаться к этому руководству раз в год или постоянно таскать с собой двухтомник с описанием кучи исключений из правил и рассмотрением частных случаев?

А как насчет перехода на другую систему программирования (без смены языка)? Помнится создатели С++ говорили о мобильности? А о совместимости средств разработки забыли? А что по поводу файлов помощи в среде? Я имею в виду их размеры, — прямое следствие «успешных» реализаций сред разработки. Сюда еще приплюсуйте описания API-интерфейсов тех ОС, на которых эти среды установлены! А ведь есть еще и масса всяких инструментов третьих фирм, направленных на «улучшение» этих сред и «облегчение» труда разработчика. Никто не подсчитывал время, затраченное на их изучение? А различного рода библиотеки, причем с идеологией, отличной от первоначальной идеологии среды разработки и операционных систем? После этого не приходится удивляться недоверчивым взглядам собеседников, когда рассказываешь им об инсталляторах операционки, помещающихся на одной дискете или средах разработки в единицы мегабайт (кстати две трети из этого объема — help-системы).

Говоришь с людьми о новых технологиях и слышишь, что кто-то создал ActiveX-элемент размером всего 56 килобайт, и начинаешь ловить многозначительные переглядывания, когда рассказываешь, что элемент с такими же функциональными возможностями может быть размером в полтора килобайта. А при словах, что операционка может быть размером в 200 Кбайт (и при этом воплощать все передовые тенденции развития ПО), явственно чувствуешь желание собеседников вызвать скорую или просто набить вам физиономию — как же! ведь тут не использовался С++.

Так что же? Я могу еще долго мыслью по древу расплываться, приводить всяческие аргументы. Но все это перекрывает один и тот же контраргумент. Выражается он в вопросах вроде: кто на этом работает или что на этом написано? Не то, чтобы это удар ниже пояса, но ежиться заставляет. Крыть нечем. Все, что сделано Виртом по проекту Oberon и Component Pascal (в

среде BlackBox) знакомо ограниченной части академической среды и десятку энтузиастов. Какие-то фирменные вещи мне не известны. Публикаций (по крайней мере на русском языке) очень мало, да и то, по-моему, только в Интернете. О выдающихся качествах как самого языка Component Pascal, так и фреймвока в среде BlackBox знают очень мало. Я, как преподаватель, просто не решаюсь делать курс по этой системе: мне жаль студентов и своих усилий. Применения (в обозримом будущем) они этой системе не найдут и канут их знания как вода в песок. Причина та же: спрос на тот или иной язык (среду разработки) формируется рынком — а, значит, все теми же троечниками менеджерами-недоучками, обчитавшимися рекламных врезок а-ля «новое средство от NuMega поможет вам избежать ошибок при работе с памятью», которые смыслят в программировании, как ежик в пылесосах. Вот и приходится возвращаться к монстрам типа VisualStudio на шести(!) CD-дисках.

Программирование — не искусство. Это наука. Проектирование программ — отчасти искусство, согласен. Искусство в программировании возникает там, где пытаемся преодолеть трудности, вызванные недостаточной проработкой решения задачи на стадии проектирования. Тогда возникает законный вопрос: а какая разница в том, какой язык мы выбираем для реализации проекта? Первая мысль — никакой! Вторая — если четко проработан проект системы, ясно определена предметная область и ее сущности (с их поведением, наследованием, внутренней структурой и т.п.), произведена разбивка всего этого на логически связанные блоки (модули), то зачем нам какие-то сложные навороты в языке, он должен быть как можно проще (повторюсь: проще — не примитивнее!), для того чтобы просто выразить в терминах математических абстракций решения стадии проектирования.

Кстати, простота языка позволяет легко применять его не только программистам, но и на более ранних этапах, — аналитикам, алгоритмистам, дизайнерам, — что повысит степень их взаимопонимания и координации (при перепроектировании). (Вы попробуйте какого-нибудь консультанта по предметной области — если он не бывший программист-практик — попросить почитать третье издание Страуструпа! Отличное упражнение на заведение себе врагов.) Кроме всего прочего, для простого языка проще разрабатывать средства поддержки (вроде CASE-средств). Именно для этого и создавались Oberon и Component Pascal: с одной стороны — способность быть языком описания проектируемых систем, с другой — удобным средством реализации принятых решений.

Язык Component Pascal очень прост. Из моего опыта: хватает одной лекции для аудитории, знакомой с Delphi или средами программирования Си/C++. Правда, я говорю *только* о языке. Изучение классов, формирующих BCF (система классов в BlackBox Component Builder, фреймвок), занимает более значительное время (что естественно для такого класса ПО — вы спросите любого поклонника Си, сколько времени ему понадобилось, чтобы понять идеологию, если таковая существует, и научиться свободно применять MFC). Что интересно, типичной реакцией студентов после лекции был вопрос: зачем понадобилось это объяснять? *Это и так понятно!* Вам ни о чем это не говорит? Вирт вычленил самую суть, убрал все наносное, оставил только то что нужно, основы. На этой основе создал минимальный набор классов (типов) для повседневной работы, а дальше — счастливого пути — берите ваши задачи и просто решайте их в этой среде. Решайте задачи, а не постигайте потаенные и неописанные нюансы функционирования компилятора, среды разработки или библиотеки классов. Если вам что-то не нравится — перекраивайте среду, режьте, добавляйте новые модули, переделывайте Help — все в вашей власти. Есть ряд запрещенных территорий — но это именно то, что позволяет вам быть таким свободным, — лучше (поначалу) в них не лезть.

Гибкость среды поражает. Первая реакция на размеры модулей (реакция поклонников Си, конечно): «Ну в Visual Basic тоже модули миниатюрные!». И сразу удивленно-недоверчивый взгляд, когда говоришь, что это чистый 32-разрядный код. А вы когда-нибудь видели фокус с добавлением в формочку (в режиме конструирования) дополнительного управляющего элемента так, чтобы он тут же появился в ее рабочем аналоге (рабочая форма здесь же на экране)? И это *без перекомпиляции* проекта. Кстати, вы можете, «не отходя от кассы» (не перекомпилировав код!), назначить нововведенному элементу опции того, с какой переменной какого модуля ему общаться. Без перекомпиляции! И это не интерпретатор!

На людей, пообщавшихся несколько лет с продуктами Microsoft, это производит впечатление! А вы никогда не пытались в какой-нибудь среде откомпилировать надпись на форме? Или заголовок кнопки? А отформатировать Help, атрибуты его текста? Построить свои документы гипертекста? Вставлять управляющие элементы в текст? И это без всяких дополнительных пакетов! Это все просто заложено в среде, во фреймвоке, в языке. А знаете сколько занимает единственный запускаемый EXE-файл среды? Вы стоите? Сядьте! 78 Кбайт! Все остальное — от

надстройки над файловой системой до компилятора — компоненты, причем реально работающие! Кстати, не только на PC, но и на Macintosh. Здесь все можно вставить во все! По сути дела нет привычного разбиения на этапы компиляции, линковки, запуска. Вы просто постоянно развиваете среду вместе с теми задачами, которые решаете.

И еще. Не знаю как у других, а у меня больше всего ошибок в программах связано с отслеживанием выделения/освобождения памяти (знаю, что я не самый аккуратный и внимательный человек на этой планете). Так вот, забудьте об этих проблемах! Решайте задачи, а о памяти (ее подчистке за вами) позаботится сборщик мусора. Кроме того, что вы можете просто постоянно работать в среде, вам иногда понадобится сделать автономную программу или динамическую библиотеку (в духе Windows). И здесь у вас есть интересная возможность. Вы можете и тут получать варианты EXE и DLL с возможностью сборки мусора или без нее. А такой гибкой системы настройки меню я вообще нигде не видел.

Кто-то упомянет о COM+, но последний появляется только сейчас, а виртовскому детищу уже около пятнадцати лет! Так что же? Вся эта братия во главе с Microsoft пыхтит, раздувает щеки, произносит фразы, но в конце концов приходит к принципам, заложенным командой Вирта (только в других формах). Так не лучше ли изначально не водить народ сорок лет по пустыни (чего кстати никогда не было...), а прямо заняться внедрением умных вещей с умной организацией. Или по-прежнему нам будут тыкать в лицо аргументами совместимости, соблюдения стандартов и массового переучивания народа? Но что касается первого: а не дороже ли обходится поддержание старого маразма в рабочем состоянии (и попыток его развития), чем построения новых аккуратных систем? А по поводу второго: те стандарты тоже когда-то были приняты... По третьему же замечу, что когда-то ходила шутка, что даже внутри IBM никто полностью не знает PL/1. Это я к тому, что не дороже ли обходится обучение народа кучеобразному нагромождению средств в языке, все нюансы которого знают только гуру, чем показ средства, воплощающего на практике основные подходы и принципы построения программных систем?

И еще одно замечание. Не за горами очередная революция в области полупроводниковой техники. И гораздо раньше, чем вы ожидаете! Вы не боитесь, что она приведет к изменению не только архитектуры компьютеров, но и к необходимости пересмотра самих принципов построения программ (в общеархитектурном смысле)? Ну, например, полетит к чертовой бабушке само понятие наследования. Я серьезно! (Кто хочет поподробнее обсудить эту тему — милости просим! Жду вопросов.) Да, так вот, что тогда? Вы что, еще раз расширите C++ добавлением новых синтаксических уродств? А нет риска, что они придут в противоречие с имеющимися? И вы опять нагрузите имеющиеся в языке слова еще несколькими значениями? Или добавьте новые? Да, еще не забудьте о новых библиотеках (и стандартах)! А еще надо обеспечить совместимость со старьем! Сколько страниц будут занимать руководства по ++C++ (или C\*\*, или C!) через 10 лет?

Вы спросите: ну хорошо, а что предлагается взамен? А я ничего не предлагаю! Я за виртовскую линию языков спокоен. Может быть, она и останется не столь популярной, но кардинальных изменений не претерпит. И не потому, что костна или неповоротлива. Просто в основе ее лежат строгие математические основания (а не сиюминутные модные поветрия). Ну а математика, ребята, это вещь, претендующая на носительницу истины. Единственная проблема — это то, что ее (математики) изучение и понимание требует прилежания и великого напряжения. Ни к первому, ни к другому народ, в массе своей, не привык или не желает на это тратить время — отсюда и нераспространенность идей профессора Вирта, воплощенных в его языках и системах.

Что-то понесло меня. Вообще-то я человек неразговорчивый. И со статьями — только по крайней необходимости. А тут, видимо, накопело. Да и надоело уже видеть, как воинствует серость, прикрывая собственные огрехи в программировании сложностью используемых средств. Был у меня знакомый по кафедре — полтора года сортировал матрицу десять на десять. Работал на C++. С шаблонами классов. Полтора года. С высшим образованием. Дал я втихаря это задание студенту-третьекурснику (не блиставшему особыми успехами в программировании). Через две недели пришлось ставить зачет.

Тут один однокашник прирулил. Говорит, что их фирма начала на буржуев работать. В фирме большинство на Delphi работает. А буржуи рогом уперлись: у нас, говорят, все профессионалы на "плюсах" пишут — он у нас стандарт. Посидел народ наш, покумекал, и решил основные алгоритмы и компоненты на Паскале ваять, а те исходники, к которым буржуи будут иметь доступ — на "плюсах". Работали в Delphi 5.0 и C++Builder 4.0. Теперь все довольны. Буржуи — соблюдением стандарта и сделанной качественно и раньше срока работой, наши — зарплатой и

уверенностью в надежности, вразумительности, проработанности и краткости основных частей системы.

В общем, кому-то может показаться, что ушел от темы, но я лишь хотел показать, что вопрос выбора языка — это не вопрос вкусов и симпатий. За этим стоит система ценностей и приоритетов, которыми живет человек, давящий на клавиши. Либо он плывет по течению, проглатывая то, что ему навязывают («все так поступают»), либо, проведя критичный анализ избирает средство простое и адекватное решаемой задаче.