

Содержание

Введение	6
Важное замечание	6
Второе введение.....	7
Часть первая. Я – программист	11
«Кто вы?»	11
Чей нос лучше?	12
Наука и жизнь	16
Где мой дом?	19
Работать в корзину	22
Часть вторая. Юношеское увлечение: Лисп	27
Лисп на «Эльбрусе»: первая предыстория.....	27
Лисп на «Эльбрусе»: вторая предыстория.....	31
Лисп для MS-DOS.....	35
Лисп для .NET: Scheme.....	38
Зачем?	41
Часть третья. Первая любовь: Ада	43
Введение	43
ASIS: интерфейс будущего для языка из прошлого	45
ASIS-for-GNAT: начало	48
Двум богам нельзя служить	53
.NET и Ада	55
AdaCore, GNAT и среды программирования	56
И один в поле воин	65
Чем плох «чёрный ящик».....	68
Bring Ada to .NET!.....	73
Один в поле не воин.....	79
Заключение	80
Часть четвертая. Редкая профессия	82
Введение	82
Летает или не летает?	84

Что нам стоит дом построить?	85
По рукам!	88
Глаза бояться, а руки делают.....	89
Первые радости.....	91
Что такое идентификатор?	93
Компилятор как таковой: таблицы и деревья.....	95
Лебедь, рак и щука, или Гадкий утенок.....	99
Стиль программирования: на вкус и цвет товарища нет	105
Программирование «наизнанку»	108
Настоящая работа	111
Быстро сказка сказывается, да не скоро дело делается	113
Кризис.....	116
Как отремонтировать подгнивший дом	118
Последнее «прости»	120
Любимое дитя.....	122
Confidential.....	124
Заключение. Полетит?	126
Постскриптум.....	129

Часть пятая. Lessons Learned 130

Введение.....	130
Организация лексического разбора: отдельно или вместе?	131
Синтаксический и семантический анализ: вместе или отдельно?	134
Составные имена: лексика или синтаксис?	140
Таблицы и деревья: две структуры или одна?.....	142
Ошибки и их диагностика: несколько соображений на будущее....	146
Заключение	150

Часть шестая. Третий компилятор..... 152

Что дальше?	152
Второй компилятор.....	152
Инструменты и парсеры.....	156
Третий компилятор	159
Вдогонку: что такое СП	161

Часть седьмая. Русские «плюсы»..... 166

Проект Wednesday	166
На следующий день.....	167

Дьявол и детали.....	172
Поехали!	176
Русские «плюсы»	177
Ох уж эта локализация... ..	189
Перекрыть Енисей или перекрыть функцию.....	191
Операторы сложения и инструкции присваивания	195
Начало: планы, персоны и обязанности.....	201
Что говорит компилятор	205
Как организовать работу: инструменты и технологии	207
«Команда».....	211
Кризис.....	215
Исход и «вторая жизнь»	218
Заключение	221
Редкая профессия?	221
Сделать наоборот	225
Предисловие, или Как вывернуть компилятор наизнанку	226

Введение

Я очень люблю рассказывать. Не то чтобы уж так много захватывающе интересного есть у меня, чтобы поделиться с другими. Просто со временем у любого человека образуется некоторый запас, груз, багаж – называйте как хотите – знаний, опыта и просто жизненных ситуаций, которые могут быть любопытны не только ему самому.

К сожалению, в реальной жизни крайне редко удаётся (почти никогда не) рассказать что хочется и именно так, как хочется – спокойно, обстоятельно, не боясь, что тебя прервут, и будучи уверенным, что будут слушать с интересом, дослушают до конца и правильно поймут. Винить в этом некого: во-первых, чтобы удерживать внимание слушателей на протяжении всего рассказа, необходимы врождённые способности, похвастаться которыми могут немногие; во-вторых, просто крайне редко обстоятельства складываются так, что становится возможным спокойный рассказ перед заинтересованной и доброжелательной аудиторией.

Так или иначе, я придумал простой способ реализовать свою склонность к «рассказыванию»: все, что я хотел бы рассказать, я просто возьму и запишу, а там уж как сложится... Интересным это окажется для кого-нибудь – можно и опубликовать; нет – ну и пусть себе лежат эти тексты у меня в ноутбуке. Переживать не буду: ведь я уже и так сделал, что хотел, – рассказал ☺.

Важное замечание

Обязательно надо сказать вот что. То, что здесь написано, – *не мемуары*. По-моему, ничего нет скучнее, чем описывать свою жизнь, – тем более когда искренне считаешь её вполне ординарной. Да, сейчас, может быть, и любопытно читать воспоминания людей, живших в XIX веке, проникать в их психологию, узнавать детали быта, бесследно исчезнувшие к настоящему времени. Возможно, что-то из сегодняшнего тоже скоро исчезнет и останется лишь в подобных воспоминаниях. Но считать, что обстоятельства *собственной* жизни настолько значимы и достойны того, чтобы о них узнали другие, – увольте, у меня никогда не было подобных претензий. Да и, по совести говоря, нет в них, в этих обстоятельствах, ничего хоть сколько-нибудь интересного...

Нет, ей-богу, пусть это делает кто-нибудь другой. А я просто расскажу о нескольких софтверных проектах, в которых участвовал. И еще попытаюсь более-менее внятно описать идеи, которые кажутся мне интересными и перспективными. И которые обязательно будут – в свой черед – реализованы.

Второе введение

Мне 55 лет.

Вроде бы по всем «обыденным» критериям активный период уже позади. Многие скажут: пора переходить от программирования к более «спокойной» работе: рулить командами разработчиков, заниматься «стратегическим планированием», писать технические проекты и книги, учительствовать, прозревать будущее и вообще подводить итоги: писать мемуары, например ☺.

К счастью или к сожалению, у меня всё складывается как-то совсем по-другому. Еще очень давно я заметил, что моя личная эволюция заметно отстаёт от «обычного» хода событий у людей сходного рода деятельности, да и вообще от временных вех, интуитивно понимаемых как «нормальные». Отставание иногда незначительное, иногда разительное, но оно наблюдается почти во всём и почти всё время.

Я потерял девственность... не скажу когда, но значительно позднее большинства своих ровесников. И ещё позднее начал толком понимать, как вообще следует обращаться с противоположным полом.

Я женился в 30 лет, а сына заимел в 33. Не бог весть какая задержка, по сравнению с другими, но она всё-таки есть – и хорошо ложится в общую картину. А ведь очень многие мои ровесники и ровесницы – уже давно дедушки/бабушки...

Я впервые побывал за границей в 40 лет – иные к этому времени успели там пожить, второе гражданство получить, а некоторые – даже вернуться домой. (Правда, первой моей заграницей оказалась Швейцария. Наверное, именно поэтому я навсегда полюбил эту страну и по сей день считаю ее лучшей в мире.)

Я стал более-менее прилично зарабатывать годам к 44-м, несмотря на широкие к тому возможности для людей моей специальности. (Знаю программистов, которые даже при советской власти умудрялись получать весьма неплохие доходы, устанавливая свои системы по всему Советскому Союзу.)

Я научился водить только в 45 лет и тогда же купил свою первую машину. Некоторые мои одноклассники и однокурсники умели во-

дить и реально водили – не важно, что машины были родительскими (или угнанными ☺) – лет с восемнадцати...

С последним обстоятельством связано одно очень яркое и важное для меня воспоминание. Как-то летом мы с семьёй отдыхали на море в Словении и однажды поехали на экскурсию в Люблян. Дорога проходила по очень живописным местам, среди высоких холмов (почти гор), поросших густым красивым лесом. На какой-то стоянке автобус остановился, чтобы дать пассажирам отдохнуть от долгой поездки, как это обычно бывает. Прогуливаясь по парковке, я увидел два или три «кемпера» – мини-автобуса, предназначенных для семейных путешествий. Очень удобная вещь: и средство передвижения, и достаточно комфортный дом (почти в любом кемпинге его можно подключить ко всем коммуникациям). Такие караваны, конечно, не покупают, а берут напрокат на три-четыре недели и колесят в них по всей Европе. Получается значительная экономия на гостиницах при сохранении свободы от predetermined туристических маршрутов, характерной для автопутешественников.

Люди вокруг «караванов» занимались своими обычными делами – отдыхали, копались в вещах, готовились продолжать свой путь. Я наблюдал, застыв на месте. Мгновенно, потрясаяще отчетливо и безысходно мне стало ясно, что такого у меня не случится уже *никогда*. Было абсолютно невозможно представить, что за оставшийся у меня период активной жизни я успею научиться водить (и получить нужный опыт управления машиной!) и попасть в обстоятельства, позволяющие таким образом проводить свое свободное время. Слишком много радикальных изменений должно произойти в жизни, чтобы такое стало возможным...

И что же? К сегодняшнему дню я объехал пол-Европы, от Италии до Норвегии и от Испании до Чехии, накрутил несколько тысяч миль по Америке и совершенно спокойно чувствую себя, въезжая в до сих пор незнакомый европейский город и отыскивая в нем нужную улицу или просто «исторический центр»...

Я поступил в аспирантуру не сразу после окончания МГУ, а лет через десять, уже работая в оборонном НИИ. Диссера у меня тогда не получилось – вместо обычного текста с жёстко определённой и хорошо известной структурой я навалил какие-то фантазии «на свободную тему», которые даже для статьи не годились... В результате, когда я перешел на работу в МГУ, мне заново (из-за истечения допустимого «срока давности») пришлось сдавать кандидатский экзаме́н по специальности. Кандидатскую диссертацию я написал (за

три летних месяца, по вечерам) и защитил, только когда подступило «по самое некуда»: пригласили работать в политехнический институт в Цюрихе, и было просто неприлично ехать туда без учёной степени. Это произошло тоже в 45 лет.

Из-за такой задержки с диссером со мной однажды случился ужасный конфуз. В 1995 году, когда я участвовал в проекте ASIS и сотрудничал (в основном удаленно) с политехническим институтом в Лозанне (EPFL), мне сообщили, что в Мадридском университете состоится некий workshop, в котором приветствовались бы выступления на темы, связанные с языком Ада. И говорилось, что возможно участие за счёт приглашающей стороны. Набравшись наглости, я послал туда заявку, и, к моему немалому удивлению, она была принята. Так вот, на этом workshop'е все дружно называли меня «доктор Зуев», хотя, видит бог, никакого повода для этого я не давал. Когда ко мне обратились таким образом в первый раз, я то ли не заметил, то ли не обратил внимания, а потом поправлять говорившего просто не хватало духа. Наверное, испанцы и представить не могли, что такой солидный дядя, выступающий вместе со Столлманом (они его тоже пригласили), – никакой не доктор, а заурядный программист. Впрочем, мой внешний вид многих сходным образом обманывает... ☺

Кстати, именно на этом workshop'е я впервые в жизни делал сообщение на английском. Это было, напомним, в 41 год, и мой «английский» был еще весьма и весьма дурным. Да что там! – и на отечественных конференциях я начал выступать очень поздно (и соответствующий опыт написания статей приобрел далеко не сразу), долгое время держась в тени своего научного руководителя...

Я впервые встал на горные лыжи, когда мне стукнуло 45 лет. Это, конечно, не самое важное в жизни умение (тем более что катаюсь-то я до сих пор как типичный «чайник»), но подобная «задержка в развитии» хорошо показывает общую тенденцию, чрезвычайно для меня характерную.

Ну и так далее... Повторяющиеся ситуации позднего наступления важных событий и изменений в жизни (а их на самом деле было значительно больше, чем я описал) утвердили меня в простой мысли. Подобные «задержки» носят вполне объективный характер и, видимо, присущи мне изначально (может быть, таков Замысел). Поэтому не стоит (да и бессмысленно) расстраиваться, комплексовать по этому поводу и дёргаться, пытаюсь подстегнуть естественный ход событий. Надо просто все время *действовать* – спокойно, но неотступно,

активно, но без суеты, в различных направлениях, но целенаправленно – *и всё будет, всё чаемое наступит, всё случится!*

...Только вот все чаще приходит опасение и даже страх: это-то всё понятно и правильно, но это работает, только пока впереди остается хоть какой-то резерв времени. А его-то всё меньше и меньше – ведь странно ожидать, что мне отпущено больше, чем другим! Например, у меня до сих пор нет собственной квартиры или дома – и совершенно неизвестно, хватит ли времени занять какую-нибудь недвижимость. Я до сих пор не миллионер – с ещё менее реальными перспективами. И даже не доктор наук (в российском понимании)!..

Утешает одно: работаю-то я в Швейцарии, а здесь пенсионный возраст для мужчин – 65 лет ☺. Даже если я буду продолжать работать по найму, у меня ещё десять лет работы, это же прорва времени! Если отпустить на каждый жизненный или профессиональный «проект» по полгода (минимальный период, необходимый для любого маломальски существенного дела – это тоже вывод из моего опыта), то ещё очень-очень много можно успеть сделать. И, с другой стороны, ведь ужасно жалко было бы провести это время в вялом ничегонеделании (или просто в рутинной работе), не сделав ничего заметного...

Так что если по-прежнему хочется программировать – надо программировать, копать в битах и байтах, осваивать новые языки, паттерны и технологии, не смущаясь тем, что выглядишь «белой вороной» среди молодых ребят. А «стратегии», книги, лекции и прочие более «солидные» виды деятельности (к которым тоже ведь есть склонность!) успеются, тем более что до сих пор получалось совмещать их с любимым делом.

Так что – вперед!

Часть первая

.....

Я – программист

«Кто вы?»

Я – программист.

После такого начала необходимо перевести дух и кое-что пояснить. Во-первых, эта фраза (да еще и вынесенная в заголовок) может сразу вызвать ассоциации с названием знаменитой винеровской книги «Я – математик»¹. Так вот, у меня совершенно нет амбиций такого уровня, чтобы таким нехитрым способом встать рядом с основоположником кибернетики («вас здесь не стояло» – мог бы сказать он). Просто я действительно программист, больше ничего делать не умею, и заголовок честно отражает этот факт. Я хочу поделиться своим опытом, полученным в результате почти непрерывного четвертьвекового программирования – то есть чтения, придумывания, написания и отлаживания программ, а также раздумий над ними и любования ими.

Во-вторых, могу поклясться, что это название я придумал сам. Более того, долгое время (около года) файл с этим текстом состоял из этого единственного утверждения. Однажды набрав данную фразу, я немедленно погрузился в мучительные раздумья и, наконец, махнув рукой, решил отложить попытку написать что-то, отличное от очередного технического проекта. Как известно, Word автоматически присваивает новому файлу имя, взятое из первой фразы текста. Так что файл с гордым местоимением в качестве имени проболтался у меня в ноутбуке довольно долго, пока, наконец, я не вернулся к нему опять. Вот тут-то я вспомнил, что и здесь не первый. Кен Томпсон, один из авторов системы UNIX, своё выступление при вручении ему премии имени Тьюринга начал точно так же: «Я – программист. Именно так я пишу в анкетах в графе “род занятий”». Как программист я пишу программы».

Поистине шагу нельзя ступить, чтобы не толкнуть классика! Всё, решительно всё уже сказано...

¹ «Отвратительная, саморекламная книга Винера» – слова академика Дороницына (сам слышал).

Когда-то давно, еще в прошлые времена меня, новоиспеченного начальника сектора закрытого НИИ, направили на так называемые «курсы повышения квалификации руководящих работников». Там в числе прочих нам читали курс под названием «Психология управления», или что-то в этом роде. Лектор в качестве примера самоидентификации рассказывал о результатах некоего уличного анкетирования, в котором задавался единственный вопрос: «Кто вы?» По умолчанию предполагалось, что, отвечая на него, респондент непременно скажет о самом для себя главном – о том, что составляет существо его личности¹. Не помню, каковы были результаты этого опроса; приводились примеры ответов: «Я – грузин», «Я – жена Иванова». Лектор на полном серьёзе сетовал, что никто не сказал «Я – комсомолец»...

Именно тогда мне стало абсолютно ясно, как бы я ответил на такой вопрос.

Чей нос лучше?

Наверное, споры о том, какой язык программирования лучше, начались с того момента, когда был придуман второй язык программирования.

Возражат: кому какое дело до того, на чём написана моя программа, – главное, чтобы она делала то, что от неё требуется. Знал одного программера, который довольно долго писал заказные программы для Windows на языке Ада, и у заказчиков даже и мысли не возникало поинтересоваться языком реализации: программы работали надёжно и без сбоев. Однако это справедливо только для крайне ограниченного множества программных изделий. Можно согласиться, что какая-нибудь несложная утилита с интерфейсом командной строки может быть написана хоть на Перле, хоть на языке ассемблера. То же с известной натяжкой относится и к задачам чисто вычислительного характера. Но этим, по существу, круг «индифферентных» к языку задач исчерпывается. Например, решение о необходимости сколько-нибудь развитого интерфейса сразу сужает выбор инструмента до одного-двух языков. Учёт же дополнительных обстоятельств – тре-

¹ Совершенно не понятно, правда, на чем основана уверенность, что человек в несколько секунд адекватно это сформулирует, – по-моему, большинство должно ответить: «Я такой-то», просто назвав свою фамилию. Удивительно самоуверенная наука – психология.

бования производительности, взаимодействие с другими программами, переносимость, ожидаемый характер сопровождения, будущие модификации и расширения – как правило, не оставляют какой-либо альтернативы.

Я сам видел текстовый форматировщик (документатор), написанный на Фортране, и пользовался им. Конечно, можно написать компилятор на Бейсике (если очень постараться) – но насколько быстро он будет работать? Задачи сортировки удобно программировать на Borland Pascal – но как такую программу перенести под UNIX? Вполне возможно реализовать СУБД на Си++ – но как будет сопровождать и развивать эту СУБД персонал, который даже Си не знает? С другой стороны, на чём ещё, кроме как на Си, можно писать системные программы под UNIX, где всё «заточено» под этот язык? И на чём разумно программировать библиотеку утилит для анализа Ада-программ, как не на самой Аде?

Так что я могу определённо утверждать, что в девяноста случаях из ста программист реально *не волен* выбирать. Любая сколько-нибудь нетривиальная программа, предназначенная не для самоутверждения и самоудовлетворения автора (или, по крайней мере, не только для этого), а для реального использования другими людьми или другими программными системами, должна характеризоваться определенным набором свойств: конкретной функциональностью, связанной с предметной областью, адекватным задаче способом организации интерфейса, заданной средой исполнения, сроками создания, наконец. Учет этих обстоятельств почти автоматически исключает какой бы то ни было произвол и приводит к совершенно конкретному выбору – операционной системы, прикладных библиотек и, разумеется, используемого языка программирования. К тому же зачастую язык реализации явно определяется заказчиком, и обсуждать это вообще бессмысленно (тем более что вы как программист можете с заказчиком ни разу и не свидеться – между вами не один посредник).

А если программист начинает работать в некоторой организации (а в организации или, в крайнем случае, в серьёзном коллективе только и возможно сделать сколько-нибудь значимую программную систему – исключения настолько редки, что своей малочисленностью только подтверждают общую тенденцию), то он неизбежно вынужден следовать некоторым традициям, свойственным этой организации, принятым правилам и соглашениям, которые почти всегда подразумевают использование определённого ЯП. Более того, чем «круче» и респектабельнее фирма, тем с большей вероятностью она имеет то,

что называется отраслевыми стандартами, «соглашениями о стиле программирования» и т. п. Так что если даже эта контора и использует ваш любимый язык, то не очевидно, что вам понравится то, *как* она его использует.

Мне кажется, что сама готовность вступить в спор о том, какой язык лучше «вообще», без анализа требований, обсуждения сферы применения (предметной области), программной и аппаратной платформ, переносимости результирующей программы и т. п., свидетельствует об определенной незрелости программиста как профессионала. Настоящий специалист, поняв, что данная задача требует использования нелюбимого языка, лишь поморщится (может быть, пробормотав про себя: «Ненавижу» ☺), раскроет руководство и через день-два напишет и откомпилирует первый фрагмент будущей программы. Легендарный канадский хоккейный вратарь Кен Драйден сказал в одном из интервью: «Я ненавижу стоять в воротах, но это моя работа».

Иногда приходится слышать такое обоснование предпочтения одного языка другому: язык X скоро умрёт, так как появился язык Y, который «гораздо лучше». Некоторое время назад я предложил одному неглупому студенту мехмата участвовать в работе по написанию комментариев к Стандарту Си++. Поначалу с энтузиазмом согласившись, он вскоре заявил, что эта работа потеряла всякий смысл: Си++ скоро станет никому не нужен из-за появления... нового языка C#!

Несмотря на анекдотичность высказывания, оно далеко не уникально – подобное приходится слышать довольно часто. На это можно возразить, что в короткой, но насыщенной истории информатики не было еще *ни одного случая*, чтобы появление нового языка привело к исчезновению какого-либо «старого». Язык PL/I вобрал в себя многие свойства Фортрана и Кобола, и что же? – эти языки выжили и чудесно себя чувствуют по сию пору! Сам PL/I умер вовсе не потому, что возникло что-то лучше него, – просто он *как таковой* оказался слишком громоздким, сложным и бессистемно построенным. Ни Си, ни Паскаль вовсе не «уничтожают» друг друга – оба вполне живы как сами по себе, так и в своих диалектах и ближайших потомках. При этом появление Си++ вовсе не отодвинуло Си на задворки программирования – язык живет, используется, развивается, для него регулярно выходят новые стандарты ISO. Ада, практически целиком включив основные концепции многих ЯП, не «отменила» ни Паскаль, ни любой другой свой прообраз. Примеры можно множить. Так что оставим броские заголовки типа «Java отменяет Си++» или «C# – убийца Java» на совести журналистов.

Могут заметить: но есть же объективный критерий того, жив данный язык или умер, – процент программирующих на этом языке и/или количество программных продуктов, реализованных на нем. Однако если разобраться, этот критерий не может считаться справедливым. Есть языки, которые имеют крайне ограниченный круг пользователей, тем не менее явно не собираются умирать, а идеи, заложенные в них, оказались чрезвычайно плодотворными для развития тех или иных аспектов «науки программирования» (в той мере, в какой программирование может вообще считаться наукой ☺). Скажем, многие ли реально программируют на ML? – я думаю, что в России об этом языке известно немногим. Однако походите по сайтам американских университетов, и вы увидите, что очень значительная (преобладающая) доля научных работ по теории типов данных и смежным сферам базируется на идеях ML и использует этот язык как инструмент либо как объект исследований. Этот язык преподается на факультетах Computer Science практически всех ведущих университетов. Так жив ML или умер? – вопрос, по-моему, риторический. В ещё большей степени это относится и к языку Scheme.

Да и вообще, проценты – вещь относительная, не так ли? Ну вот возьмем Аду (далее этот язык будет довольно часто упоминаться). Последний на момент написания книги индекс TIOBE дает Аде *всего лишь* 0,783% от общего числа упоминаний о языках программирования. Казалось бы, исчезающе малый процент. А сколько это будет в абсолютном исчислении? При поверхностном гуглении находятся (с невнятной ссылкой на некий закрытый майкрософтовский отчет) такие цифры: 10 миллионов профессиональных программистов, около 40 миллионов программирующих от случая к случаю и около 100 миллионов «любителей».

Вообще-то, к данным Майкрософта следует относиться с некоторой осторожностью: в своих оценках они вполне могут опираться на число проданных (ими же!) лицензий на свои инструменты разработки – при том что громадное количество программистов работают и на других платформах. Да и у многих программеров, особенно в развивающихся странах (да и в некоторых уже «развитых» ☺), стоят нелегальные копии средств разработки... Так что оценки Майкрософта вполне можно признать заниженными.

Тем не менее, за отсутствием иных оценок, примем майкрософтовскую и даже возьмем ее условный минимум: 10 миллионов профессионалов и столько же «прикладников» или «любителей» (и вторые, и третьи все-таки тоже программисты!). Умножив 20 миллионов на

«адский» процент, получим, что на Аде программирует примерно **150 тысяч человек**. Сто пятьдесят тысяч! – это же очень много, и уже по одному этому Аду никак нельзя признать умирающей.

...Ну а что кому больше нравится – чудесные служебные слова **begin** и **end** или отвратительные закорючки вроде '{' и '}', неподражаемая компактность и элегантность Си++, поражающая воображение простота и эффективность Ады или невиданная в природе надёжность Си, – это все личные вкусы и эмоции, характеризующие спорящих, но не имеющие отношения к существу дела. Если же говорить серьёзно, то единственное, что следует обсуждать, – насколько *данный* язык адекватен такому-то классу задач и насколько он для *этого* класса задач подходит лучше другого языка. Или, наоборот, «плясать» от языков: например, взять сходные свойства каких-либо двух языков и сравнить, насколько удобно для практического программирования эти свойства реализованы. Например, сравнить механизм исключительных ситуаций в Аде и Си++¹. Понятно, однако, что все это – предмет не для спора, а для вдумчивого исследования.

У меня тоже есть любимый язык программирования. Какой – не скажу: для дальнейшего изложения это несущественно.

Наука и жизнь

Stephane: «We are computer scientists, aren't we?»

Eugene: «I am not a computer scientist,
I am a software developer».

Из разговора

Я поступил на факультет вычислительной математики и кибернетики МГУ в 1971 году; это был, кажется, второй набор с момента его основания. Факультет только-только отпочковался от мехмата, и, похоже, ясного понимания, чему, собственно, следует учить здесь студентов, тогда не было. Не было на самом деле того, что теперь называется computer science, или информатикой. Костяк преподавательского состава образовывали математики, раньше работавшие на мехмате и физфаке или только что их окончившие, и естественно, что основной упор делался на математический анализ, численные методы, дискретную

¹ Между прочим, интересная тема для курсовой работы! При схожести базовых принципов исключительные ситуации в этих двух языках сделаны по-разному, и могло бы быть полезным сравнить, как эта разница отражается в практических применениях.

математику и подобные уважаемые, но... не вполне «компьютерные» дисциплины. То, что действительно оказалось необходимым впоследствии, – например, теория формальных грамматик, теория сложности алгоритмов, теория графов, реляционная алгебра и подобные предметы – давалось в лучшем случае в виде спецкурсов.

К сожалению, у меня не обнаружилось особых склонностей к этой уважаемой науке – математике, хотя все экзамены я сдавал вполне прилично. Практически всё, что нам преподавали, навеки выветрилось из памяти, оставив только смутные и интуитивные ощущения того, что называется математическим методом.

С тех пор я не люблю и боюсь слова «наука». Обязательный в кандидатской диссертации раздел «научная новизна» вызывал невероятные мучения: какая, прости Господи, «научная» новизна? Если и есть новизна, то уж, конечно, не научная, а, скорее, конструкторская, инженерная: описаны и неформально обоснованы некие подходы к построению компилятора, придуманы и реализованы такие-то модели и структуры данных и такие-то алгоритмы компиляции Си++ – ну какая здесь на хрен наука?

Мне приходилось общаться с людьми многих специальностей, в обывательской жизни именуемых (и часто именующих себя) учёными. Поражает легкость употребления этого слова. А уж как охотно «учёными» бросаются журналисты! – такое впечатление, что у них каждый, кто сидит за письменным столом и/или за компьютером, не может быть назван никак иначе, кроме как «учёный»...

В связи с этим у меня есть такой образ. Представим себе землекопа, который должен вырыть яму. Чтобы сделать это, грамотный, опытный и старательный землекоп должен разметить на земле параметры будущей ямы (геодезия и картография), принять в расчёт характер почвы (геология), прикинуть объём работ, составить расписание и оценить трудозатраты (менеджмент), договориться об оплате (экономика и бухгалтерия) и, наконец, приступить к делу. Копая яму, наш землекоп знает, какой инструмент (например, штыковую или совковую лопату или, может быть, отбойный молоток) надо использовать в каждом конкретном случае, под каким углом следует вонзать лопату в землю оптимальным образом и за какое место древка лучше всего её держать (механика). Наконец, он должен следить за своим самочувствием и хорошо понимать, когда следует сделать перерыв, а когда пойти перекусить (охрана труда). И так далее... И ведь никому не придёт в голову (и ему самому тоже) считать его, скажем, геологом, медиком или физиком-экспериментатором! А ведь землекоп непо-

средственно использует в своей работе законы механики, и хороший землекоп использует их вполне осознанно!

Почему-то люди, занимающиеся сугубо прикладной деятельностью (зачастую весьма необходимой и полезной во всех отношениях) и в процессе работы привлекающие какой-нибудь научный аппарат (математические формулы, законы природы) или просто чей-нибудь опыт, выраженный в более-менее строгой форме, ничуть не сомневаясь, считают свою деятельность *научной*. По моим впечатлениям, наиболее подвержены этой страсти представители двух областей знаний – психологи и экономисты (именно тех сфер, в которых произвол в интерпретации «подведомственных» им явлений жизни достигает предела).

Я уж не говорю об адептах так называемых «общественных» наук. Когда я оканчивал университет и сдавал госэкзамен по специальности (это было ещё в прежние времена), в комиссии состоял один такой представитель – тогда это было обязательным. И он с великолепной самоуверенностью и даже с запальчивостью (скрывавшей, очевидно, некие внутренние сомнения) заявил, не помню, по какому поводу, буквально следующее: «научный коммунизм – это тоже точная наука!» Показательно, кстати, само разделение: «точные» науки и... неточные, что ли? А что это за наука, если она не точная?

Мне все-таки кажется, что слова «наука» и «учёный» следует относить к очень узким областям знания и крайне ограниченному кругу людей. Математика – да, это наука, и люди, ей занимающиеся, безусловно, могут называться учёными. Теория программирования – наверное, пока она сохраняет свою близость к соответствующим разделам математики. Теоретическая физика, химия – разумеется, тоже. Но вот дальше – не знаю... Пусть мои рассуждения покажутся чересчур тенденциозными, а выводы – радикальными, но я считаю, что любая область знаний может считаться наукой лишь в той мере, в какой её модели и результаты могут быть выражены нетривиальными математическими формулами, опирающимися на некоторую строгую теорию.

И пока «ученые»-обществоведы (уж не знаю, как они теперь себя называют – политологи, что ли?) не выведут формулу или систему уравнений, описывающую закономерности развития общества, и пока социологи не построят формальную модель поведения субъекта в предлагаемых обстоятельствах (и пусть при этом еще и строго специфицируют эти обстоятельства!), и пока на основе этой модели не будет дан проверяемый прогноз на будущее, и пока этот прогноз не

сбудется – давайте всё-таки считать этих уважаемых людей *исследователями*, а не учёными.

В точности то же я отношу к программистам и, разумеется, к самому себе. Моя мама до сих пор считает меня ученым-«кибернетиком», что вызывает во мне прилив жгучего стыда и желание низко опустить голову. Я признаю, что не знаю, что такое кибернетика (общие слова, которые обычно говорят по этому поводу, я сам могу сказать во множестве – это не означает понимания существа предмета). И я не учёный. Я – программист.

А если нужно подобрать более понятный эквивалент для характеристики этого рода занятий, можно предложить слово **конструктор**. Разработка компьютерных программ, мне кажется, более всего похожа на процесс конструирования какого-нибудь материального предмета (скажем, новой конструкции кресла). Осознается и вербализуется некоторая потребность, оценивается перспективность разработки и сферы применения будущего продукта, уточняются спецификации требований, функциональность и параметры будущей системы, выбираются инструменты (языки программирования, компиляторы и прочие программные средства) и материалы (алгоритмы, структуры данных, готовые компоненты), и, наконец, по определенной технологии (технологии программирования), с использованием известных методов создается сам предмет.

Кстати, я бы не смешивал конструкторскую деятельность и промышленное производство. Хотя эти деятельности тесно связаны (как правило, конструирование имеет целью промышленное производство предмета, а хорошо сконструированный материальный предмет будет технологичен в производстве), для программных продуктов аналогия не вполне корректна. По существу, производство программного продукта в собственном смысле равнозначно его *тиражированию*, а этот процесс не имеет практически никакого отношения к программе как таковой: самая совершенная операционная система (которой не существует в природе) и студенческая поделка (если кому-то придет в голову ее тиражировать) будут производиться по совершенно идентичным правилам и с теми же самыми затратами.

Где мой дом?

Я возвращался из Франкфурта в Лозанну один. Сергей уехал днем раньше – его ждала семья, а я хотел послушать tutorial по новой редакции Ады, который в последний день конференции «Ada in Europe

95» проводили Стефан и Магнус, и, главное, спокойно (в одиночестве) погулять по старому городу.

Перед поездкой во Франкфурт мы с Сергеем Рыбиным неделю просидели в Политехническом институте Лозанны, готовя статью для этой конференции и его выступление – наш первый отчет о работе по реализации стандарта ASIS для компилятора GNAT. Выступление прошло очень хорошо (один раз Сергей даже получил порцию аплодисментов), и, чувствуя себя немного причастным к этому успеху, я решил, что один свободный день – заслуженная компенсация за предшествующие усилия.

Хоть мы и были заняты подготовкой статьи и выступления, однако уик-энды оставались «нашими», и кое-что из примечательных мест Швейцарии нам посмотреть удалось – и этого оказалось достаточно, чтобы почувствовать все обаяние этой маленькой страны в центре Европы. А если учесть, что внешние изменения происходят здесь крайне неспешно (швейцарцы, как известно, никогда и никуда не торопятся), то в этот второй приезд мы уже чувствовали себя в Лозанне почти *как дома*.

Франкфуртский tutorial оказался очень информативным и легко воспринимаемым (а после него еще и халявным обедом покормили!), старый город, несмотря на окружающие его со всех сторон небоскрёбы, – очень милым и «человечным», так что в результате я почувствовал себя вполне удовлетворённым поездкой. В девятом часу вечера в хорошем настроении я загрузился в TGV, который должен был довести меня до Базеля. Там была пересадка на другой, уже обычный, поезд до Лозанны.

Контролёр посмотрел на мой билет и сказал (по-немецки) несколько слов, из которых я понял только «Базель». Да я и сам помнил, где мне выходить.

Вагон, сначала переполненный – многие, в основном молодые, сидели даже на полу (что, у них такие билеты продают – на пол?), – становился с каждой остановкой всё свободнее, и когда ближе к конечной станции появились пограничники, в нём оставалось буквально два-три человека. Я вынул паспорт и держал его наготове.

Хоть бы эти лентяи взглянули! – пусть не на мой паспорт, а хотя бы на меня! Зачем я, спрашивается, проторчал неделю в длиннейшей очереди перед германским посольством в Москве, в итоге проникнув туда за визой, только нагло обманув сотрудника на входе? Зачем я платил в Лозанне тридцать восемь франков за дополнительную визу, чтобы иметь возможность не только съездить во Франкфурт, но и

вернуться? Два типа в фуражках с кобурами на поясе вяло прошли по вагону туда и обратно, глядя... – нет, не на пассажиров. Похоже, их больше интересовало, не оставил ли кто вещи на верхних полках...

Объявили Базель. Я подхватил свои вещи и вышел. После меня в вагоне, кажется, не осталось никого. К этому времени я уже твёрдо выучил, что в Европе почти любая пересадка устроена так, что людям достаточно пересечь перрон, и там их уже ждет нужный поезд.

Соседний путь был пуст.

Я подошел к расписанию. Даже в Москве мне, честно говоря, стоит немалых трудов разобраться в расписании пригородных электричек (ну какой я после этого программист?..). Я заставил себя не прыгать глазами по большому щиту и внимательно просмотрел текст строчка за строчкой. Названия «Лозанна» нигде не было. Я покрылся испариной и похолодел.

В таких ситуациях ни в коем случае нельзя терять внутреннее ощущение темпа, иначе охватывающее тебя чувство безысходности может привести к полной апатии. Выйдя из франкфуртского экспресса, я был готов быстро идти по перрону, ища в лозаннском поезде вагон второго класса, чтобы не переходить из вагона в вагон в набирающем скорость составе. Теперь следовало обратить этот порыв на прояснение ситуации.

По перрону шел человек в форме. Я догнал его, сунул ему билет и спросил (по-английски, конечно), где мой поезд. Надо отдать ему должное: он воспринял мою сбивчивую от волнения речь и мгновенно понял ситуацию. «Вы сошли в *немецком* Базеле, а вам нужен *швейцарский*!» Только теперь я осознал, что имел в виду билетёр, напоминая мне о конечной станции, и что значит аббревиатура SBB¹, добавленная к названию «Базель» в билете.

«Что же мне делать?» Служитель подумал несколько секунд и сказал: «Садитесь в этот поезд». На месте франкфуртского TGV уже стоял другой состав. «Через пять минут будете там, где нужно». Я понял только, что ситуация как-то исправляется, и слегка успокоился. Человек отпер ключом вагонную дверь. Как бы благодаря и извиняясь, я промямлил: «Очень трудно разобраться – ведь (я вспомнил) есть еще и французский Базель!»

Ответ железнодорожника я запомнил навеки:

– У вас в Лондоне вообще пять вокзалов – вот там действительно трудно!

¹ SBB (Schweizerische Bundesbahnen) – швейцарские железные дороги.

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

e-Univers.ru