

Рустэм Валеев

# ФРАНЧАЙЗИ НА ГРАНИ НЕРВНОГО СРЫВА

Как небольшой фирме-партнеру 1С  
перестать выживать и начать зарабатывать



**Рустэм Р. Валеев**  
**Франчайзи на грани**  
**нервного срыва. Как**  
**небольшой фирме-партнеру**  
**1С перестать выживать**  
**и начать зарабатывать**

*Текст предоставлен правообладателем*

*[http://www.litres.ru/pages/biblio\\_book/?art=64420357](http://www.litres.ru/pages/biblio_book/?art=64420357)*

*Франчайзи на грани нервного срыва. Как небольшой фирме-партнеру  
1С перестать выживать и начать зарабатывать: 1С-Пабблишинг;*

*Москва; 2021*

*ISBN 978-5-9677-3077-1*

### **Аннотация**

Как написать программу, за которую вас будут благодарить долгие годы? Как автоматизировать завод, запустив программу в работу всего за 10 дней? Как найти свою нишу на рынке и зарабатывать на внедрениях программ 100 миллионов рублей в год? Обо всем этом и многом другом рассказывает автор книги – программист, консультант, директор ИТ-компании с 35-летним стажем в отрасли.

Истории из личного опыта автора дополняются полезными выводами и советами. Используя их, вы сможете обойти множество «граблей» на пути автоматизатора.

Книга будет полезна всем, кто разрабатывает, продает и внедряет программные системы: сотрудникам ИТ-компаний и ИТ-отделов, подрядчикам и заказчикам больших ИТ-проектов. А также начинающим предпринимателям, рассматривающим создание ИТ-компаний.

# Содержание

Предисловие	7
Благодарности	13
Введение	15
Часть 1. Как я писал и внедрял программы до 1С	18
Глава 1. Как автоматизировать котел-утилизатор	18
Глава 2. Как научиться программировать за одну ночь, чтобы писать программу два года	33
Глава 3. Как победить болгарский дисковод	43
Глава 4. Как написать программу, за которую тебя будут благодарить 15 лет	49
Конец ознакомительного фрагмента.	51

**Рустэм Валеев**  
**Франчайзи на грани**  
**нервного срыва.**  
**Как небольшой**  
**фирме-партнеру 1С**  
**перестать выживать и**  
**начать зарабатывать**

*Все права защищены.*

*Материалы предназначены для личного индивидуального использования приобретателем.*

*Запрещено тиражирование, распространение материалов, предоставление доступа по сети к материалам без письменного разрешения правообладателей.*

*Разрешено копирование фрагментов программного кода для использования в разрабатываемых прикладных решениях.*

© Валеев Р. Р., 2021

© ООО «1С-Публишинг», 2021

© Оформление. ООО «1С-Публишинг», 2021



# Предисловие

**Борис Нуралиев, директор «1С»**

«Рустэм Валеев, давний и успешный партнер нашей фирмы, попросил написать отзыв на его книгу.

Я со своей стороны, очень благодарен Рустэму Раифгатовичу за то, что такая книга появилась.

Внедрение информационных технологий – дело интересное, творческое, и непростое. Не у всех это получается, из тех, у кого получается, не все могут и хотят сделать из этого успешный бизнес. А из тех, кто сумел создать успешный бизнес, далеко не все готовы и могут об этом рассказать.

Я прочел книгу и считаю ее содержательной, понятной и интересной. Многие совпадают с моим лично мнением и с моим опытом. Не все совпадает? но это кажется нормальным.

Я не очень верю в бизнес-гуру, которые могли бы дать однозначные рецепты развития бизнеса. Более того, считаю, что одну и ту же область деятельности, одну и ту же ситуацию можно успешно решить по-разному. С учетом деталей реальной ситуации, с учетом особенностей и даже внутренних предпочтений конкретного предпринимателя и руководителя. Поэтому сообщество «1С» сильно не только проработанными годами инструкциями и инфописьмами с пятизначными номерами, но и, одновременно, живым творчеством ру-

ководителей и сотрудников партнерских фирм в сочетании с их готовностью искренне и интеллигентно делится своим опытом, своими умозаключениями, своими творческими находками.

Кому интересно прочесть такую книгу? Мне кажется, всем кому интересна реальная автоматизация реальных предприятий – и собственникам, и руководителям, и специалистам, и студентам, которые собираются работать в этой области. Я получал высшее образование как раз по этому профилю – «Автоматизированные системы управления». К сожа-лению – тогда таких книг не было. Было много книг, по содержанию, сухости и скукоте полностью соответствующих технической документации тех же времен, было небольшое количество художественных «производственных романов» и переводных книг по организации программирования – но не внедрения. Пришлось набивать шишки на собственном опыте, что конечно тоже было интересно. И управленцам моего поколения читать эту книгу может быть приятно, потому что вызывает много похожих воспоминаний.

Но кому эта книга может быть полезна сегодня? Очевидно, руководителям внедренческих организаций, причем много ценного для тех кто участвует в тендерах на внедрения, тем кто разрабатывает свои отраслевые продукты или думает развивать у себя эти направления. Но я предложил бы ее почитать и молодым специалистам и начинающим предпринимателям. Если в начале сложно продираться через но-

стальгические воспоминания о стройотрядах и перфокартах – начните с середины или даже с последних глав, и очень вероятно, что вам захочется прочесть ее целиком. Очень ценно, что раскрывается не просто жизненная позиция и рекомендации автора, но приводятся убедительные аргументы, жизненные примеры, а иногда и несложные, понятные расчеты.

Сам я не получил бизнес-образования, и для меня в бизнесе основную роль сыграли как раз «бизнес-кейсы» – иногда из литературы, а чаще из рассказов и отчетов коллег по бизнесу. Я специально попросил прочесть книгу и дать короткий отзыв одного из начинающих молодых коллег. Его мнение: «Это хорошая книга для начинающего франчайзи, который хочет перерасти просто продажу коробок бухгалтерии и начать заниматься именно внедрениями. А для уже «подросших» – набор занятных историй, в которых можно узнать себя, научиться на чужих ошибках и посмотреть на практики других».

В итоге – книгу рекомендую, спасибо автору, спасибо тем, кто ее прочтет, спасибо сообществу «1С» за обмен опытом и соображениями!»

**Михаил Сорокин,**

**помощник директора «1С» по «1С: Предприятию»**

«С Рустэмом Валеевым познакомились полтора десятка лет назад в Уфе, на «Едином семинаре». С тех пор регулярно

общались на вечерних заседаниях партнерских семинаров, где Рустэм часто выступал от микрофона и дискутировал с Борисом Нуралиевым, в партнерской конференции, а в последнее время – в Фейсбуке, где появилась серия постов-кейсов, многие из которых легли в основу этой книжки. И выступления, и посты, и главы книжки отличает практическая направленность с ясной и понятной задачей? развить и усовершенствовать бизнес автоматизации на базе платформы «1С: Предприятие». Думаю, что многие руководители партнерских компаний узнают ситуации, описанные автором, и оценят принятые руководителем управленческие решения. Замечательно, что автор не использует «мемуарный» стиль, где все глуповатые, а мемуарист – рыцарь на белом коне. Нет, Рустэм откровенно и критически описывает и управленческие просчеты, делая из них выводы на будущее. Возможно, читатели предложат другие ходы или сделают другие выводы в описанных автором кейсах – это и прекрасно: книга инициирует диалог, обсуждение, как это происходило и происходит в комментариях к постам Рустэма в Фейсбуке».

**Михаил Пясковский,**  
**генеральный директор «ИнфоСофт»**

«У вас в руках прекрасная книга о бизнесе 1С: Франчайзи, написанная увлеченным своим делом человеком. В ней показано становление фирмы Рустэма, ее взлеты и падения. Книга легко читается. В виде небольших историй описаны

различные ситуации из повседневной жизни компании, занимающейся автоматизацией. Представлены простые ответы на сложные вопросы.

Книга будет полезна начинающим партнерам фирмы «1С». Рустэм открыто делится своим опытом и кейсы, представленные в книге, позволят «не наступать на те же грабли».

В то же время, опытные 1С: Франчайзи найдут в книге много подтверждений того, что они правильно поступают в тех или иных вопросах, что тоже немаловажно. Внимательные читатели смогут вынести из книги интересные детали подхода автора, и взять их на вооружение.

Партнерам, ориентированным на развитие проектного направления будет интересно прочитать про проектный опыт Рустэма.

Отдельный интерес представляет опыт по созданию тиражных решений. Показан путь от зарождения идеи, через сложности к выпуску известного и популярного решения в своей отрасли.

И, конечно, несколько глав книги посвящено продажам. Автор делает акцент на работе с корпоративным рынком. Термин «ухаживать за заказчиком» как нельзя лучше отражает подход Рустэма.

Завершающая часть книги посвящена советам по руководству ИТ-компанией. Раскрыты вопросы найма и управления персоналом, даны практические рекомендации по управлению проектами.

Я сам прочитал книгу с удовольствием, и вам ее рекомендую».

# Благодарности

Прежде всего, я хочу поблагодарить моих читателей в социальных сетях. Часть рассказов, что вошли в книгу, были опубликованы в интернете. Отзывы читателей, такие как «... ярко и замечательно написано», «автор, пишите ещё, потом может и сборник издадите» и «это как «Брат» и «Брат-2» пересматривать», и вдохновили меня написать эту книгу.

Я выражаю искреннюю благодарность Михаилу Сорокину, без советов, помощи и деятельного участия которого эта книга просто не появилась бы на свет. Я очень рад, что мои рассказы понравились человеку, который захотел увидеть их «собранными вместе».

Я очень благодарен всем своим клиентам, сотрудникам и партнерам. Именно они стали главными героями моих историй, вошедших в книгу. Руководители заводов, теплосетей и водоканалов, директора ИТ-компаний и ИТ-директора, руководители проектов, программисты и консультанты – вот люди, с которыми я работаю бок о бок уже 35 лет, и о которых хочу рассказать с уважением и любовью.

Я благодарен директору фирмы «1С» Борису Нуралиеву, без которого не было бы не только «Франчайзи на грани нервного срыва», но и вообще «франчайзи 1С». А также большому числу сотрудников этой замечательной компании, без ежедневного и упорного труда которых невозможен был

бы наш франчайзинговый бизнес.

Конечно же, я не могу не упомянуть тут своих друзей – директоров фирм-франчайзи «1С», чьи добрые отзывы на первые рассказы поддержали меня в самом начале пути.

Я хочу поблагодарить свою жену и детей за ту любовь и поддержку, что я чувствовал все время, пока писал книгу. Благодаря советам моей жены Татьяны я смог сделать свои истории и тактичнее, и мудрее.

А еще я благодарен топ-менеджерам моей компании, особенно моему партнеру Ольге, за предоставленную возможность сосредоточиться на книге. Представляю, как непросто было им замещать меня в многочисленных отпусках, которые я использовал за полгода упорного труда над рукописью.

И, наконец, я хочу поблагодарить Наталию Моисеенко за большую редакторскую работу, которая была проделана очень бережно и с огромным тактом. Благодаря ей книгу можно читать легко и с удовольствием.

*Рустэм Валеева*

# Введение

Разрешите представиться. Я – Рустэм Валеев, автор книги, которую вы держите в руках. Генеральный директор и один из собственников компании «Софт-портал». Компания – партнер-франчайзи фирмы «1С»<sup>1</sup>. Разработчик решений «1С: Управление теплосетью» и «1С: Управление водоканалом», предназначенных для автоматизации ресурсоснабжающих организаций.

Я родился и вырос в простой советской семье повара и грузчика. Пределом мечтаний мамы было, чтобы я получил высшее образование и стал инженером. В школе мне очень нравилась математика, но выучился я на инженера-строителя – по настоянию родителей, которые считали такую специальность лучшим способом заработать на квартиру. На третьем курсе института я написал первую программу, и тут же понял, в чем мое настоящее призвание. Благодаря этому я прошел путь от программиста-самоучки до руководителя ИТ-компании с годовым оборотом в 100 миллионов рублей. Мне пришлось поработать и консультантом, и руководителем проекта, и начальником абонентского отдела. Я побывал с разных сторон баррикад: и заказчиком, и подрядчиком в проектах внедрения ИТ-систем. За 35 лет автоматизации

---

<sup>1</sup> Здесь и далее в этой книге: «1С» (в кавычках) – название фирмы, 1С (без кавычек) – продукты и технологии фирмы «1С».

предприятий я попадал во множество забавных и поучительных ситуаций. Мои истории с удовольствием слушают дети, коллеги и друзья, собираясь на шашлыки в нашем загородном доме.

В этой книге я расскажу вам о том, как была создана компания «Софт-портал». Как мы искали место под солнцем. Как разработали отраслевое решение и попали в прайс-лист «1С». Как создали уникальную технологию внедрения, позволившую нам закрывать акты выполненных работ каждый месяц. Как учились и научились продавать программы по всей России. Как, вместе с фирмой «1С», внедрили ERP-систему у большого заказчика на 10 000 рабочих мест. И как, в конце концов, нашли свою нишу на рынке и перестали выживать.

Я расскажу вам и о том, чем занимался до прихода в партнерское сообщество «1С». О том, как писал программы в эпоху, когда деревья и компьютеры «были большими». Так вам станет понятнее, что успех компании был не случайным. Возможно, мой опыт позволит вам избежать своих ошибок и потерь.

В последней части книги я дам несколько полезных советов по бизнесу. Расскажу о том, как правильно принимать сотрудников на работу. Как распознать талантливого руководителя проекта. Как работать с перфекционистами. Как осознать и оформить корпоративную культуру компании, чтобы каждый сотрудник знал, что верно, а что – нет.

Вы также узнаете, как побеждать в тендерах, заключать удачные договоры и успешно выполнять их.

В книге практически нет теории, кроме ссылок на книги бизнес-авторов. Я сделал упор именно на практику. Все мои советы и утверждения подкреплены примерами из жизни. Занимательные истории сделают ваше чтение нескучным. А полезные советы пригодятся на практике.

Ну, давайте уже начнем. Издалека...

# Часть 1. Как я писал и внедрял программы до 1С

## Глава 1. Как автоматизировать котел-утилизатор

– Твою мать, – крик раздался прямо из кучи.

Вася<sup>2</sup> с трудом выбрался из свежего серого бетона, весь грязный и мокрый. Отряхнулся, как мог. Стукнул по кузову самосвала кулаком. Еще раз выматерился. И побежал в сторону корпуса № 5. Там был душ. Мы с сочувствием посмотрели ему вслед и снова взялись за лопаты.

В то лето я, студент второго курса строительного факультета института, впервые попал на стройку. В составе стройотряда в 30 бойцов. Нашей задачей было строительство бетонных проездов на огромном химзаводе.

Каждый день мы вставали в 7 утра, чтобы прошагать 4 километра от нашего общежития до столовой. После завтрака отправлялись на рабочее место. Надевали каски и брезентовые перчатки, брали в руки лопаты и ждали первый самосвал с бетоном. Жидкий бетон возили в кузовах ЗИЛов объемом

---

<sup>2</sup> Здесь и далее – все имена изменены, кроме моего имени, имен моей жены Татьяны и моего партнера по бизнесу Ольги.

2,4 кубометра. Три замеса по 0,8 куба. Замесы из гравия, песка, цемента и воды делали бойцы другого стройотряда на бетонно-растворном узле неподалеку.

Если было жарко, бетон начинал застывать еще в самосвале. Частенько приходилось забираться в откинутый кузов, чтобы столкнуть застрявшую кучу и зачистить борта. Иногда куча бетона увлекала за собой и чистильщика. Если не смыть смесь сразу, она застывала, и испачканную одежду приходилось выбрасывать. Хорошо, что в цехах завода были душевые.

Кучу бетона нужно было быстро разбросать по щебеночной основе – за 30–40 минут, пока бетон не затвердел. Застывший бетон отгребали в сторону бульдозером, а за порчу материала удерживали из зарплаты. Мы работали бригадами по 4 – 5 человек. Налетали на вязкую кучу как муравьи и раскидывали ее по проезду метров на 5 – 7. После чего протаскивали по бетону виброрейку, выравнивали поверхность. Посыпали готовое полотно стружкой, поливали водой, чтобы застывающий бетон не растрескался. И садились на бордюры. Ждать следующий самосвал.

За смену мы принимали от 8 до 12 самосвалов. К вечеру спину ломило, а лопата становилась неподъемной. Каждое утро я по несколько минут разгибал сведенные пальцы, чтобы просто взять в руки зубную щетку. Боль была адская. Еще и потому, что бетонная пыль проникала через брезентовые перчатки и разъедала кожу. На руки в трещинах и кровопод-

теках было страшно смотреть.

Тем не менее, это было здорово – видеть, как на месте грязных канав появляются новые и чистые бетонные проезды. Результаты твоей работы были налицо.

Однако там я познакомился и с изнанкой строительной профессии. В те времена не было никаких GPS-навигаторов и программ управления автотранспортом. Количество рейсов, которые сделал самосвал, бригадиры бетонщиков вручную отмечали в транспортной накладной водителя. Реальное количество рейсов и то, что попадало в накладную, различались: бригадиры приписывали рейсы. В благодарность водители «не заруливали» с бетоном на сторону, а везли его сразу на стройплощадку.

Да, некоторые самосвалы приходили на четверть пустыми. Жидкий бетон был нужен не только на химзаводе. Из него получались прекрасные фундаменты и дорожки на дачах у садоводов, поэтому нечистоплотные водители шабашили – везли часть бетона в дачные товарищества.

Воспитанный на фельетонах из журналов «Крокодил» и «Фитиль», я вживую наблюдал то, что считал глубоко неправильным и с чем мне пришлось бы мириться или бороться после вуза.

Но хуже всего было другое. Качество нашей работы. То, чему нас учили в институте, и то, что я увидел на стройке, вообще не совпадало. Технология производства нарушалась везде, где только могла быть нарушена. Замес бетона делали

на глаз, поэтому он был то слишком густой, то слишком жидкий. Требования к фракциям гравия (размеру камешков) не соблюдались: нам могли привезти бетон и с булыжниками, и вообще без гравия, только с песком. Частенько не выдерживалось время замеса, в результате чего вода с цементом вытекали еще по дороге, и самосвал выгружал нам лишь кучу песка с камнями.

Ладно, это бетон. Мы не могли повлиять на его качество. Но то, что происходило дальше, было не лучше. В целом не соблюдался ни один технологический размер. Толщина бетонного полотна – 15 сантиметров. Это в теории. На практике толщина колебалась в значительных пределах: от 5 до 50 сантиметров. 50 сантиметров получались, например, тогда, когда мы отставали от плана с объемами. План в Советском Союзе надо было выполнять, несмотря ни на что. В противном случае объявлялся аврал. И самосвалы с бетоном все шли и шли, с раннего утра и до полной темноты. Но готовых щебеночных оснований не было, и бетон лили прямо на землю. Потом боковины у таких проездов присыпали землей, и со стороны казалось, что все классно.

Если же щебень был плохо выровнен или его было слишком много, никто не парился, чтобы привести все в норму. Бетон лили на то основание, что получилось, и толщина полотна тогда оказывалась очень небольшой, около 5 сантиметров. Мы старались надежно оградить такой участок, чтобы по нему не пустили КАМАЗы до тех пор, пока бетон пол-

ностью не затвердеет. Через 28 дней, когда бетон набирал крепость, проезд мог выдержать приемку работ, а там – хоть трава не расти. Ремонт некоторых проездов начинался уже через пару месяцев после завершения строительства.

Это, значит, толщина полотна. Но с его шириной и расположением ситуация была не лучше! Ширина полотна плясала на 5–7 сантиметров. На такие мелочи никто не обращал внимания. Лента бордюров в некоторых местах напоминала кардиограмму. Глядя на нее, можно было сказать, что дорога «приболела». Угол 90 градусов получался далеко не на каждом перекрестке. Да и по высоте полотна на пересекающихся участках зачастую не совпадали.

Все это не являлось проблемой конкретной стройки – это была проблема всей строительной отрасли времен социализма. Главным для каждого рабочего, бригадира, мастера и прораба был вал. Объем работ. Качество соблюдалось ровно настолько, насколько требовалось, чтобы сдать работу.

Через год я побывал еще в одном стройотряде. Теперь уже в роли его командира. Увидев, как мой бригадир, студент 3-го курса, делает фундамент под оборудование, я окончательно понял, что стройка – это не мое. То, что получилось у него, стояло немного «не там». «Ерунда, поставят они свой станок на десять сантиметров правее». Технологические отверстия были вполовину от требуемых. «Да кто будет их измерять?! Главное – они есть». И формой фундамент напоминал валенок вместо прямоугольника. «Ну да, пара досок

опалубки отвалились под весом бетона. Но это не страшно, все равно эту дуру в монолитном полу не будет видно». Для меня это был просто кошмар. Я хотел делать работу так, как было описано в учебниках: хорошо, качественно, на совесть. На реальной стройке тогда это было невозможно.

Осенью после этого стройотряда меня посетил самый настоящий экзистенциальный кризис. Я был так разочарован тем, чем мне предстояло заниматься всю оставшуюся жизнь, что решил бросить стройфак. Завербоваться «на севера». В советские времена везде требовались неквалифицированные рабочие руки. Можно было поработать на нефтяных скважинах. Или на сейнере. Я купил пару газет с объявлениями, сходил на сборочный пункт. Написал заявление и прошел медкомиссию. Пора было ехать. И тут я решил рассказать все своему другу Руслану.

– Ты уверен, что обязательно нужно бросать институт? Ты представляешь, что делается «на северах»?

– И что же?

– Там реально некуда пойти после работы. Кругом только снег. Мужики бухают по-черному. Пара лет – и ты рискуешь просто спиться.

– А какие еще варианты? Стройка – это не мое, меня просто мутит от того, что я там увидел.

– Ладно. Тебе не нравится стройка. А что тебе нравится?

– Хороший вопрос. Математика мне нравится. Но больше всего мне нравится программировать. Помнишь, на вто-

ром курсе у нас было программирование? Я тогда половине группе программы написал.

– Ну так и сходи к завкафедрой, который вел у нас программирование. Расскажи о том, что тебе нравится. Я знаю студентов, которые работают лаборантами. Может, и тебе работа найдется. А на Север ты всегда успеешь.

– А почему бы и нет? – сказал я. И сразу же пошел на кафедру прикладной физики и химии.

Удивительное дело, им как раз требовался лаборант. Узнав, что я хочу писать программы, меня тут же взяли на ставку. 90 рублей в месяц студенту очного отделения, когда инженер с дипломом получал 118 рублей! У социализма были и свои преимущества. Зарплата неквалифицированного персонала позволяла нормально жить.

Преподаватели кафедры одновременно числились в научно-исследовательском секторе или коротко – НИС. Оправдывая свое название, НИС осваивал бюджет научно-исследовательского проекта по моделированию режимов работы котла-утилизатора. Котла, стоящего на том самом заводе, где мы два года назад бетонировали проезды! Вот это совпадение! Завод, который отпугнул меня своей стройкой, все же не хотел меня отпускать. Чем-то я ему понравился. Теперь мне представлялась возможность написать для него программу.

Котел занимался утилизацией тепла после какого-то химического процесса. Внутри огромного цилиндрического

строения высотой с 20-этажный дом вдоль внутренних стен проходили трубы. В них текла вода. В котел поступали газы из другой огромной установки. Моя задача была смоделировать в программе на языке Фортран процесс теплообмена в трубах котла. Так, чтобы на модели можно было испытать разные режимы работы котла и подобрать оптимальный. Тот, при котором отбор тепла стремился бы к максимуму, а отложения пыли на стенках труб – к минимуму.

Все, что сделал мой научный руководитель, доцент кафедры прикладной физики и химии Александр Иванович, с целью мне помочь, – показал дорогу в библиотеку. Там я и нашел кучу литературы по гидравлике, в которой описывались процессы движения жидкостей и газов в трубах. Пришлось изучить ламинарные и турбулентные потоки и, конечно же, число Рейнольдса. Но самое удивительно было то, что в библиотеке нашлась тоненькая книжица зарубежного автора, в которой были приведены системы линейных уравнений для моделирования потоков в... теплообменниках котлов!

Фортран был очень подходящим языком для моделирования системы линейных уравнений. Несложный язык, я освоил его за неделю. С ним проблем не было. Проблемы были с самим процессом написания и отладки программ.

Шел 1985 год. В богатом нефтяном институте был свой вычислительный центр, оборудованный большой машиной ЕС ЭВМ серии 1022. Он находился в главном корпусе в северной части города, а я учился на стройфаке этого ву-

за в южной части. Всего час на общественном транспорте между корпусами. В течение рабочего дня я писал программу. На листе формата А4. От руки. Красивым шрифтом и без помарок. Если где-то ошибался, то замазывал неверную строку корректором и писал заново. Вечером отвозил листки с кодом в вычислительный центр. Сдавал их в специальное окошечко. Девушка-приемщица улыбалась мне, забирала листочки и несла их операторам. Операторы переносили код с листочков на перфокарты, используя специальные устройства подготовки данных. На здоровых железных ящиках стояли клавиатуры, похожие на печатные машинки. Каждая строчка кода превращалась в перфокарту – прямоугольный кусочек картона размером примерно в четвертушку листа А4. На одной перфокарте помещалась строка кода длиной до 80 символов. Каждый символ представлял собой набор прямоугольных отверстий в нескольких позициях из 12 возможных в одном столбце.



Получив перфокарты, я сверял то, что получилось, с ко-

дом и данными на моих листочках. Если были расхождения, у меня было два пути. Первый – отдать неверно набитые строчки на повторную перфорацию. Для этого мне надо было написать их заново на другом листочке, отнести их операторам и ждать еще день. Или же я мог использовать хитрый метод корректировки перфокарт, которому меня научил Александр Иванович. Состоял он в том, что неверный символ на карте исправлялся вручную. Для этого отверстия в неправильных местах заклеивались, а в правильных – вырезались лезвием для безопасной бритвы. По шаблонной перфокарте с полным набором символов. То есть примерно от 3 до 5 минут работы на исправление одного неверного символа. Понимаете, насколько сейчас все быстрее в части корректировки набора?

После того, как код и данные были выверены, пакет перфокарт попадал в следующую очередь – на исполнение. Другие операторы закладывали перфокарты в устройство ввода. Оно считывало код в оперативную память машины. Я ехал домой, а программа компилировалась и выполнялась в пакетном режиме ночью. Выходные данные распечатывались на бумажных листах размера А3. На следующее утро их уже можно было получить в окошке для выдачи результатов. Частенько, если программа зацикливалась, мне выдавали целую коробку с прекрасной мелованной бумагой. Новый код я писал на обратной стороне этих листочков.

Через несколько месяцев таких итераций у меня все же

получилась работающая программа, которую мы и сдали заказчику. И включили ее в отчет о проделанной работе по проекту.

Что же я чувствовал, когда так сложно писал свою первую серьезную программу? Я чувствовал настоящее вдохновение. Каждое утро я с нетерпением ждал поездки в вычислительный центр. Днем сидел за научными журналами, потом писал и переписывал код. Вечером я снова ехал в главный корпус, чтобы сделать следующую итерацию. При этом я твердо знал одно: качество работы моей программы полностью зависит от меня. Никаких кривых бордюров, никакого песка с водой. Только красивый код и верные результаты!

И все было бы хорошо, если бы не желание шефа сделать из этого науку. Мне пришлось сесть за написание научной статьи. При этом я быстро освоил принятый на кафедре метод написания таких статей. Сейчас его называли бы «Ctrl + C, Ctrl + V». А тогда это называлось компиляцией. Научная новизна возникала точно так же, как возникает новое блюдо у шеф-повара турецкого отеля. Утренний салат – из того, что не съели гости вечером. Моя просьба к шефу? «Давайте не будем писать статью, тут, правда, нет ничего нового» – не возымела действия. Поэтому, чтобы написать новую статью, мне пришлось прочитать два десятка старых по теме. И подойти к материалу творчески.

В статье я привел графики наших экспериментов на модели. Помню, одна из точек на таком графике никак не впи-

сывалась в формулу, которую шеф вывел для описания режимов работы котла-утилизатора.

– Что будем делать, Александр Иванович?

– Как что, ты не понимаешь?

– Доработаем формулу, чтобы учитывала и эту точку?

– Смеешься? Послезавтра сдавать статью в журнал. Сотри точку там, где она сейчас. Нарисуй там, где она должна быть по формуле.

– Как это? Это же подлог.

– Делай, что говорю. Все это очень и очень примерные формулы и модели. Никто не знает, как оно работает в реальности. Сантиметр сюда – градус туда... Неважно. А вот статья в научном журнале – очень важно. Подведем редактора – он от нас больше ни одной статьи не примет.

С тяжелым сердцем я перенес точку туда, где она подтверждала формулу. Перед сдачей статьи в журнал меня удивила еще одна вещь. Список авторов. Он включал в себя пять или шесть человек. Двоих? меня и моего шефа – я знал. Кто были остальные? Александр Иванович поморщился.

– Послушай, тут такая ситуация. Я не могу не включить своего научного руководителя. Сейчас я пишу докторскую, и он курирует и наш проект тоже. Если его не включить, он может нам «перекрыть кислород» с договорами. Кроме тебя на кафедре есть еще несколько моих аспирантов. Чтобы защитить кандидатские, им нужны публикации. Но у них очень много и другой работы. Они не успевают написать столько

статей, понимаешь?

Я, конечно же, все понимал.

Вот так появилась первая трещина в моих отношениях с наукой. Однако после института я все же распределился на кафедру прикладной физики и химии. Начал готовиться к написанию кандидатской. И проработал на кафедре пару лет. До тех пор, пока мне не предложили заниматься тем, чем я действительно хотел заниматься, и что приносило мне радость. Прикладным программированием.

И пусть я недолго занимался наукой, но все же она оказала существенное влияние на всю мою жизнь. Я понял главное, читая научные монографии и рецензируемые журналы: прежде чем что-то сделать, нужно хорошо изучить то, что сделано до тебя другими. Мы карлики на плечах гигантов. Все, что нужно – проявить уважение к предшественникам. Это железное правило. Вот как я использую его в своей работе.

ИТ-компания, которой я руковожу сейчас, развивается в динамичной окружающей среде. Чтобы выживать и быть прибыльными, нам приходится довольно быстро приспосабливаться к внешним вызовам, меняясь внутри. Изменения происходят постоянно. Но прежде чем провести какое-либо изменение в компании, мы готовим исследовательский отчет по теме. В отчет попадает информация из научных монографий и статей. Решения, подготовленные на основе таких обзоров, внедряются легче. Они учитывают опыт предшествен-

ников и позволяют избежать множества ошибок. Например, исследование систем оценки удовлетворенности работников компаний различных отраслей позволило нам быстро создать свой собственный список из 18 вопросов. Благодаря ему мы лучше понимаем состояние наших сотрудников по результатам ежемесячных опросов.

## **Глава 2. Как научиться программировать за одну ночь, чтобы писать программу два года**

Шел 1988 год. Мы, сотрудники института, работали на больших ЭВМ. Дорогой компьютер могли позволить себе только вычислительные центры и большие организации, которых было немного. Машинные залы занимали огромные площади, и были чем-то похожи на современные дата-центры. Машинное время было дефицитным ресурсом, распределяемым по разнарядке. И вдруг появились персональные компьютеры. Небольшие и не очень дорогие. Их могли купить даже средние организации.

Водоканал, в котором в то время работало около тысячи человек, купил несколько персоналок. Вообще, выбор у предприятия был небольшой: или расширять расчетный отдел бухгалтерии, в котором в то время работали четыре девушки, или как-то автоматизировать их работу. Расчетчицы совсем не успевали, в период начисления зарплаты они работали до поздней ночи.

Расчетный отдел бухгалтерии занимал одну большую комнату в небольшом здании еще дореволюционной постройки. Здание это, впихнутое между заводом и кинотеатром, расширить было нельзя, поэтому пятую девушку-расчетчи-

цу просто некуда было посадить.

И тогда предприятие пошло по пути автоматизации. Директор Водоканала купил персоналки и стал ждать. Ждать, когда зарплата начнет считаться быстро и сама собой. Месяца через два, когда ничего не изменилось, он вызвал к себе заместителя по экономике, под которым была бухгалтерия. В результате бурной дискуссии выяснилось, что только компьютеров недостаточно. Нужны еще какие-то программы.

За программой решили обратиться в только что созданный кооператив «Вода»<sup>3</sup>, который, между прочим, не только водосчетчики поставлял, а брался вообще за все, что сулило заработок. Просто больше некуда было обратиться: большие вычислительные центры не брались за разработки на персоналках, и программистских фирм тоже еще не было.

Договор был заключен, и исполнитель немедленно найден. Им оказался Василий Петрович, преподаватель информатики того ВУЗа, который в свое время окончил заместитель директора по экономике Водоканала. Единственный известный директору кооператива человек, который писал программы.

Василий Петрович, изучив вопрос, понял, что в одиночку ему не справиться. И привлек к работе меня, молодого инженера НИС. Мне было обещано три зарплаты за эту «небольшую шабашку», как называл создание программы расчета зарплаты Василий Петрович. Никого при этом не смущало,

---

<sup>3</sup> Первые кооперативы в СССР появились в феврале 1987 года.

что я писал программы на «Фортране» для ЕС ЭВМ. А программу надо было написать на «ДВК-2М».

Я долго раздумывал, браться мне за эту разработку или нет, ведь я не знал ни предметной области, ни языка программирования. Но, как мне объяснил Василий Петрович, опыта программирования такого уровня задач на персоналках нет вообще ни у кого, а подводить клиента было нельзя. Кроме того, мне обещали поддержку моего бывшего преподавателя. И я решился. Если бы я тогда знал, через что мне придется пройти!

Мы пообщались немного с девочками-расчетчицами, которые показали нам ведомости для расчета зарплаты. Большие такие книги, размером примерно с бумажный лист формата А1. Страниц на 100 каждая. В эти книги они «разносили» таблицы и руками, с помощью счетов (те, кто постарше) или калькулятора (продвинутые пользователи), умножали оклад на время, добавляли доплаты и вычитали налоги и «исполнительные листы».

Мы с Василием Петровичем подбросили монетку и по-честному поделили работу. Мне достались начисления, ему – удержания.

Первое, что я сделал, когда добрался до компьютера, – выяснил, что на нем есть язык программирования. Бейсик. Через два дня нужно было что-то показывать заказчику, и поэтому мне пришлось изучить его очень быстро. Изучение заняло ночь. Утром я написал первую программу. Да, в этом

языке было немного команд, несколько десятков, и большинство из них оказались экзотическими и совсем ненужными. Основными были Input и Print, а также арифметические команды «плюс», «минус», «умножить» и «разделить». И еще несколько команд для поиска и форматирования данных и работы с таблицами. Десятка команд было вполне достаточно для автоматизации работы расчетной группы.

На демонстрации я показал заполнение справочников цехов, работников, табелей и расчет повременного начисления для всех работников цеха. Отработанное время умножалось на оклад и заносилось в таблицу начислений. Таблицу с расчетом можно было вывести на экран или принтер.

Василий Петрович пошел несколько другим путем. В созданной им программе «Удержания» расчет происходил следующим образом. Сперва на экране появлялось сообщение:

«Введите табельный номер работника».

Расчетчица вводила:

«17».

«Введите код начисления».

«01».

«Это подоходный налог. Введите сумму, с которой надо удержать налог, руб.»

«147».

«Подоходный налог 13 % с суммы 147 руб. равняется 19 руб. 11 коп.»

Увидев такую красоту, расчетчица спросила Василия Пет-

ровича:

– И что мне с этим делать?

– Не знаю, может, в ведомость записывать после расчета по табельному номеру?

– Двести пятьдесят человек по пять – семь удержаний?

Вводить данные и записывать результат? А спать когда?

Видимо, слухи о наших подходах к программированию дошли до директора кооператива, потому что через несколько дней я уже был единственным программистом на этом проекте.

Конечно же, я попытался применить научный подход. Но увы. Методическая литература по теме отсутствовала. Единственная книга, которую я нашел, рассказывала об автоматизации расчета зарплаты на ЭВМ серии ЕС. Описывалась следующая технология. Пользователи вручную заполняли шаблоны данных. Собирали их в пакеты и отвозили в вычислительный центр. Операторы вводили данные, и большая машина делала расчет. Информация вводилась дважды. Если программа находила расхождения, ввод пакета повторялся. Расчет начинался только после полного совпадения данных. Таким образом была реализована защита от ошибок.

Но эта технология и эти шаблоны абсолютно не подходили для работы с персональным компьютером. Пришлось все делать с нуля, подробно расспрашивая расчетчиц. Мне тогда повезло. Руководительница расчетной группы Марина, несмотря на свою горячность, была очень хорошим специ-

алистом. Она смогла грамотно поставить задачу. Мы с ней разработали свои алгоритмы расчета начислений и удержаний, откуда мат Марины был предусмотрительно удален, и, по большому счету, ничего не упустили.

Этот подход сыграл со мной в будущем злую шутку. Когда я начал заниматься автоматизацией основных средств, я также положился на руководителя другой группы. Та поставила мне задачу: рассчитать амортизацию основных средств за месяц как итоговую амортизацию прошлого месяца минус амортизацию выбывших основных средств плюс амортизацию введенных. Но при повторении ручного алгоритма в компьютере нужные результаты не получались. Мы бились месяц, но ничего не сходилось. В отчаянии я вспомнил про научный подход и отправился в библиотеку. Нашел учебник по бухгалтерскому учету, открыл раздел по учету основных средств. И сразу же понял, что нужно сделать. В итоге мы перешли к суммированию индивидуальных амортизаций по каждой карточке основных средств. Машине ничего не стоило просуммировать 10 тысяч позиций, в отличие от людей. Так мы, наконец-то, получили корректный результат.

А я получил еще один урок научного подхода. Для правильной постановки задачи программисту недостаточно общаться с пользователем. Обязательно надо изучить нормативную документацию, которая регулирует бизнес-процесс: законы и постановления правительства, положения по бухгалтерскому учету (ПБУ), положения об отделах и долж-

ностные инструкции, приказы и инструкции по предприятию. В особо сложных случаях – учебники и журнальные статьи. Чем больше источников исследует разработчик, тем более качественно будет сформулирована задача. И меньше переделок будет в коде.

Но вернемся к нашему рассказу. Не прошло и полгода, как мы провели пробный расчет зарплаты. Девочки вбили все данные, я запустил программу, мы распечатали расчетные ведомости, и расчетчицы сели за сверку.

Я был так горд собой, ведь создание программы стоило мне множества бессонных ночей. Я освоил новый язык программирования и предметную область. Усмирил болгарский дисковод (об этом – подробнее в следующей истории). Написал десятки страниц кода. Поэтому, когда я увидел рыдающую Марину, подумал: «Да, понимаю ее чувства. Не будет больше задержек на работе, бессонных ночей и красных глаз». Но оказалось, что рыдает Марина совсем по другому поводу: расчеты и близко не сходились! Ошибок было очень много. И во вводе данных, и в коде программы, и даже в алгоритмах. Поэтому пришлось еще пару месяцев дорабатывать программу, в том числе написать специальный код для защиты от неверного ввода данных, так как двойной ввод позволить себе мы не могли.

Но и это еще не все. Поскольку делали мы все это на ДВК-2М, уже в ходе проекта выяснилось, что его техническое оснащение не дает возможности запустить проект в пол-

ном объеме. Болгарские дисководы и очень небольшой объем ОЗУ и дискет – такая техника не выдерживала нагрузки. Ни один расчет так и не дошел до конца. Программу пришлось сдавать на примере расчета двух цехов – и спасибо, что ее приняли и оплатили, хотя и понимали, что использовать не смогут.

Поэтому все пришлось повторить после того, как заказчик приобрел новые персональные компьютеры «Искра-1030» с винчестерами и ОЗУ в 256 килобайт! После двух лет работы и освоения нового языка FoxBase программа все-таки пошла в промышленную эксплуатацию. Я выстоял, я справился со всеми трудностями и, наконец-то, увидел счастливые глаза моих расчетчиц, с которыми сроднился за это время! Программу я назвал «Роза», хотя Розы среды расчетчиц не было, просто по заглавным буквам словосочетания «Расчет зарплаты».

Но на дорожку программа преподала мне еще один урок. Первые расчетные листки распечатали и отнесли руководству Водоканала. Через некоторое время меня вызвал главный инженер.



– Что это? – спросил он, ткнув меня в какое-то начисление в расчетном листке.

– Ммм... «Пр. за нотх. по п»... А, это же единовременная премия за новую технику по приказу!

– Да? А я думал «Праздник с нотами и попами». Вы чего там, программисты, творите?

– Да у нас всего четырнадцать символов в справочнике начислений... Это же мелочи! Главное, что программа счи-

тает!

– Мелочи, говоришь? А если у тебя завтра 700 рабочих притащатся в расчетный отдел выяснять, что там с попами и нотами? Иди и исправь!

Тогда я понял, что названия важны. Имена важны. Слова важны. «Премия нов. тех.», конечно!

Основой нашего взаимодействия является речь. Которая состоит из слов. И с ними надо быть аккуратнее. Полные и однозначные названия лучше кратких. Я помню об этом всегда, даже если дело не касается названия переменных и реквизитов в программах. Мои сотрудники не отправляют клиентам файлы с названиями «КП» или «Договор». Мы готовим и высылаем клиенту «Договор № 2257 Софт-портал – Теплосеть – 1С Управление теплосетью 2 – Внедрение». Проявляя уважение к коллегам, экономя им время поиска нужного документа.

## Глава 3. Как победить болгарский дисковод

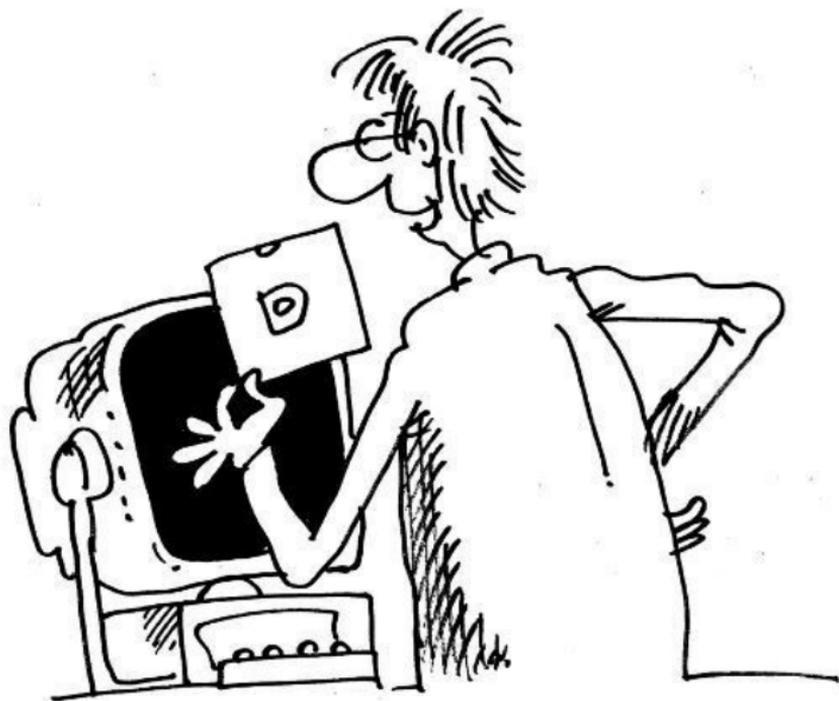
«ДВК 2М» – так назывался аппаратный комплекс, на котором создавалась программа «Расчет зарплаты». Для своего времени это был суперкомпьютер. Во-первых, он был персональным. Вполне приемлемого размера, помещался на стол. Во-вторых, у него было огромное быстродействие – целых 10 или 20 тысяч операций в секунду. Да-да, тогда быстродействие измерялось не тактовой частотой процессора, а именно операциями с двоичными числами. И в-третьих, у него было ОЗУ довольно большой емкости – целых 64 килобайта. Туда спокойно помещалась операционка, программа Бейсик, текст прикладной программы и ее данные. Чего в ДВК не было – так это винчестера. А значит, и записанной на нем операционки. К системному блоку прилагался болгарский сдвоенный дисковод размером 5,25 дюйма. В верхний слот я вставлял дискету с операционной системой и бейсиком. А в нижний – дискету с прикладной программой и данными.

Компьютер был отечественный, но сильно продвинутый. Поэтому он зависал и перегружался не чаще двух раз в день. При перезагрузке пропадали не только операционка в ОЗУ, но и, скажем, набитый за пару часов в бейсике текст новой программы. Помню, как долго я не мог прийти в себя, когда

в обед компьютер завис и унес в небытие пять страниц готовой программы! Как я в ужасе бегал вокруг стола и думал, что «ну должен же где-нибудь остаться хоть какой-то след от такого серьезного труда!». Но, увы, след оставался только в моем мозгу. Именно тогда я впервые столкнулся с таким интересным феноменом: написанная повторно по памяти программа была всегда лучше прототипа! И короче, и работала быстрее.

Со спонтанными перезагрузками ничего нельзя было сделать. Более того, мы искренне радовались, что они не такие частые, как на больших машинах. Например, перезагрузки на ЕС ЭВМ случались раз в полчаса-час и считались нормой! Так вот, для того, чтобы не терять новую программу при каждой перезагрузке, нужно было ее периодически сбрасывать из ОЗУ на дискету. Но. Тут-то и начиналось самое интересное. Болгарский дисковод записывал программу на дискету. Иногда. Но не всегда. Чаще он шипел, трещал, свистел, но... не записывал! Через минуту шипения и свиста появлялось страшное сообщение «Write failure error» и можно было переходить к новой попытке. Иногда, после трех-четырех попыток, файл все-таки записывался. Однако не было никакой гарантии, что он прочитается! «Read failure error». Это было не менее ужасное и не менее редкое сообщение системы. И вот бывает, сидишь ты за компьютером, в конце рабочего дня, смотришь на листинг программы на экране и пытаешься ее хотя бы запомнить. Потому что записать ее не

получается. Ну никак! Это ужасное чувство, думаю, напоминающее то, которое должен испытывать человек, бегущий за последним вагоном электрички, уже коснувшийся поручня, и вдруг осознающий, что поезд ушел.



Водоканал, для которого я писал программу, закупил сразу пять ДВК с болгарскими дисководами. Мы с девочками

из бухгалтерии пытались найти среди них тот, который бы записывал и считывал файлы наилучшим образом. Дисковод, более надежный, чем другие, ценился на вес золота. Мы предпринимали различные способы улучшить работу болгарских дисководов. Особой популярностью пользовался такой способ. Покупался настольный вентилятор. Он ставился в коробку из-под бумаги. В коробке вырезались две большие дырки. Одна – для поступления воздуха. А вторая, квадратная, для болгарского дисковода. Он устанавливался в эту дырку и интенсивно охлаждался потоком воздуха. Кто-то свято верил в такие «усовершенствования», я же считал, что это все равно, что камлать на бубне – никакой гарантии. 50 на 50: или запишет, или нет. Вот тогда-то во мне и зародилось первое глубокое сомнение в братстве народов, СЭВ<sup>4</sup> и целесообразности социалистического пути развития. Боже мой, как же я проклинал братьев болгар – криворуких безжалостных убийц моего времени!

Операционная система, которую мы тогда использовали, называлась RT11SJ. Загружалась она секунд десять и записывалась на диск в виде пары файлов. В отличие от MS DOS, которая появилась чуть позднее, операционка эта обладала

---

<sup>4</sup> СЭВ, Совет экономической взаимопомощи – межправительственная экономическая организация, действовавшая в 1949–1991 годах. Создана по решению экономического совещания представителей Албании, Болгарии, Венгрии, Польши, Румынии, СССР и Чехословакии. Одной из задач организации было развитие высших форм экономической интеграции – производственной кооперации и специализации, научно-технического сотрудничества.

намного большим количеством удобных команд. Например, могла выдавать список файлов с подкаталогами, а DOS не умел этого делать! «Убожество» – вот первое мое ощущение от использования DOS. Однако мы все-таки перешли на DOS окончательно и бесповоротно, когда поняли, что под ней «идут» все новые программы и игрушки, а программ для RT11 становится все меньше.

64 килобайт в ОЗУ и 250 килобайт на каждой дискете делали процесс написания большой программы весьма увлекательным. Программисту требовалась определенная сноровка, чтобы правильно разбить программу и данные на части и заставить их успешно взаимодействовать. Программа расчета зарплаты «Роза» работала так. Сперва в оперативную память загружалась система. Потом Бейсик. Потом Главная программа. Она просила вставить в нижний дисковод дискету № 1 с данными цеха 1 за январь 1989 года. Расчетчица могла отредактировать данные работников, ввести таблицу и посчитать зарплату за месяц. После этого, если дисковод работал, ей даже удавалось сохранить результаты расчета! Если она хотела их распечатать, Главная программа просила вставить диск с Программой для печати, потом диск с данными цеха, потом диск с Главной программой. Потом вставлялась дискета номер 2 для цеха номер 2, и так далее – до диска номер 42. 42 цеха было в Водоканале. И ни разу нам не удалось рассчитать их все. Или болгарский дисковод перегревался и останавливался (зачастую – уже навсегда). Или

расчетчица что-то путала и записывала на диск номер 2 данные цеха номер 17. И неделю, матерясь, их восстанавливала. А что делать – дискет для архива просто не было! Или, если все шло хорошо, в каком-нибудь 27-м цехе обнаруживались новый вид расчета или ошибка в алгоритме, и все приходилось начинать заново! Помню, что благодаря тому ужасу, который вызывала необходимость дойти с такой технологией «до конца», все-таки было принято разумное решение принять программу у программистов по акту на примере двух цехов. После чего девочки продолжили считать зарплату в бумажных расчетных ведомостях, вплоть до того момента, пока программа не заработала надежно на «Искре-1030».

Вспоминая прошлое, я радуюсь тем грандиозным изменениям, что произошли в нашей работе. Надежные и производительные винчестеры, огромная оперативная память на чисто избавили программистов от необходимости задумываться о таких вещах, как дефицит вычислительных ресурсов или аппаратные сбои. Теперь программист может сосредоточиться на главном – написании кода для решения задач пользователя.

## **Глава 4. Как написать программу, за которую тебя будут благодарить 15 лет**

Программирование занимало в моей жизни все больше времени, и его совсем не оставалось на науку. Поэтому, когда заместитель директора по экономике Водоканала предложил мне ставку программиста в кооперативе «Вода», я не раздумывал. Подписал заявление об уходе в вузе, отработал две недели и перешел полностью на работу с Водоканалом.

Следующей задачей, которую мне предстояло решить, стал расчет начислений и печать счетов абонентам Водоканала.

Делалось все вручную. Расчеты выполнялись в больших книгах, «оборотках». Информация о показаниях счетчиков переписывалась из абонентских книжек на страницы книги и умножалась на тариф. В эти же книги разносились оплаты. По итогам месяца рассчитывалось сальдо. Сложности начинались в период выставления счетов. Счета и платежные требования печатались на печатных машинках на готовых бланках, в нужные места которых впечатывались только объемы и суммы. В дни выставления счетов в абонентский отдел было не зайти – машинки строчили, заглушая все вокруг. Готовые счета раскладывались по конвертам и отправлялись абоне-

нентам. Хорошо, что в то время еще не было счетов-фактур. Не представляю, как бы сотрудники отдела справлялись с двойной работой.

Первая программа, которую я написал, была не самой эффективной. Она копировала действия сотрудников, распечатывая данные из электронной ведомости в бланки счетов. Время, которое уходило на правильную установку бланка в принтер, было не меньшим, чем печать счета на машинке. И это не считая загубленных криво вставленных бланков!

# Конец ознакомительного фрагмента.

Текст предоставлен ООО «ЛитРес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на ЛитРес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.