

МАКСИМ КУЗНЕЦОВ, ИГОРЬ СИМДЯНОВ

# программирование: ступени успешной карьеры

ЧЕМУ И КАК УЧИТЬСЯ

СТРАТЕГИЯ ПОИСКА РАБОТЫ

КАК «РАСКРУТИТЬ» СЕБЯ

ХОРОШИЙ ТОН В ПРОГРАММИРОВАНИИ  
И ИНТЕРФЕЙСЕ

ВАЖНОСТЬ ПРОЕКТИРОВАНИЯ

ТРИМЕРЫ ДОГОВОРОВ

АВТОРСКОЕ ПРАВО В ИТ

О.К.

**Максим Кузнецов  
Игорь Симдянов**

**программирование:  
ступени  
успешной  
карьеры**

Санкт-Петербург

«БХВ-Петербург»

2006

УДК 681.3.06  
ББК 32.973.26  
К89

**Кузнецов М. В., Симдянов И. В.**

К89 Программирование: ступени успешной карьеры. — СПб.: БХВ-Петербург, 2006. — 320 с.: ил.

ISBN 5-94157-802-4

Написанная опытными профессионалами и известными авторами книга посвящена различным аспектам успешного создания карьеры в программировании и области IT-технологий. Рассмотрены как технические аспекты, такие как хороший тон в программировании, правила дизайна интерфейсов, приемы проектирования, так и вопросы, напрямую с программированием, казалось бы, не связанные: методология обучения программированию, стратегия поиска работы, специфика работы в команде и т. д. Большое внимание уделено практике применения закона об авторском праве и смежных правах в IT-области, а также грамотному подходу к составлению договоров.

*Для программистов и IT-специалистов*

УДК 681.3.06  
ББК 32.973.26

**Группа подготовки издания:**

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Евгений Рыбаков</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Игорь Цырульников</i>
Компьютерная верстка	<i>Натальи Смирновой</i>
Корректор	<i>Наталья Першакова</i>
Дизайн серии	<i>Инны Тачиной</i>
Оформление обложки	<i>Елены Беляевой</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 30.03.06.

Формат 70×100<sup>1/16</sup>. Печать офсетная. Усл. печ. л. 25,8.

Тираж 2000 экз. Заказ №

"БХВ-Петербург", 194354, Санкт-Петербург, ул. Есенина, 5Б.

Санитарно-эпидемиологическое заключение на продукцию № 77.99.02.953.Д.006421.11.04 от 11.11.2004 г. выдано Федеральной службой по надзору в сфере защиты прав потребителей и благополучия человека.

Отпечатано с готовых диапозитивов  
в ГУП "Типография "Наука"  
199034, Санкт-Петербург, 9 линия, 12

ISBN 5-94157-802-4

© Кузнецов М. В., Симдянов И. В., 2006  
© Оформление, издательство "БХВ-Петербург", 2006

# Оглавление

<b>Введение .....</b>	<b>1</b>
Для кого и о чем эта книга.....	1
Описание глав .....	1
Благодарности .....	3
<b>Глава 1. Программирование: где, чему и как учиться .....</b>	<b>5</b>
Вопросы и ответы .....	7
<b>Глава 2. Как найти работу .....</b>	<b>19</b>
Базовая техника устройства на работу.....	19
Настрой на поиск работы .....	22
Цель.....	27
Поиск информации .....	30
Сбор информации.....	32
Техника устройства на работу в нужную организацию.....	36
Составление резюме .....	38
Собеседование .....	43
Что делать, когда увольняют .....	46
<b>Глава 3. Работа в команде .....</b>	<b>49</b>
Введение в социальную психологию .....	49
Какие бывают группы .....	53
Учебно-карьеристские группы .....	54
Культурно-развлекательные группы .....	54
Алкогольно-сексуальная группа.....	56
О референтных группах.....	57
Групповые процессы .....	58
Об антилидерстве .....	59
Как найти свое место в группе .....	62
<b>Глава 4. Взаимодействие с заказчиком .....</b>	<b>65</b>
Встречают по одежке.....	66
О внешнем имидже.....	66
Встреча начинается задолго до встречи .....	68
Не думайте плохо о заказчике.....	68
Заказчик дурак?.....	70

Почему они такие? .....	70
Рожденный ползать — уйди с взлетной полосы! .....	71
Невербальное общение .....	71
Что в имени тебе моем.....	73
Основной закон психологии общения .....	75
В любом из нас спит гений. И с каждым днем все крепче .....	78
Личная встреча — лишний шаг к успеху .....	78
Не возражайте в лоб.....	78
ЯЗВа .....	79
АнтиЯЗВа.....	79
Закон об объеме оперативной памяти.....	80
Закон края (закон Эббингауза) .....	80
Закон контрастов .....	80
Не уходите от скользких вопросов .....	81
Несколько правил ведения дискуссии .....	82
Никогда не перебивайте собеседника .....	82
Не рассуждайте о том, чего сами не знаете .....	83
Не спорьте с очевидным .....	83
Да поможет нам НЛП .....	84
Мы снова говорим на разных языках.....	87
Подстройка и ведение .....	88
Влияние установок .....	90
Убеждение с игрой на некоторых слабостях .....	93
Неуверенность в себе.....	93
Медлительность.....	93
Тщеславность.....	93
Эффект ореола или эффект обобщения.....	94
Эффект близости.....	96
Важность первого впечатления .....	96
Точность — вежливость королей .....	96
Давайте делать паузы в словах .....	97
Вместо резюме или минус эмоции .....	98

## **Глава 5. Проектирование ..... 101**

Требования .....	101
Архитектура проекта.....	104
Спецификации .....	106
Планирование времени .....	107
Сложность.....	109
Масштабирование .....	112
Осознанное и случайное проектирование .....	115
Не давать пользователю почувствовать себя дураком .....	115
Осторожней с функциональностью! .....	116

<b>Глава 6. Если вы — руководитель .....</b>	<b>117</b>
Две стратегии руководства.....	117
Способы управления .....	120
Чем отличается руководство компанией от руководства командой .....	122
Но о тактике тоже нельзя забывать.....	123
Не складывайте все яйца в одну корзину.....	123
Главный психолог коллектива.....	127
Люди живут для себя.....	127
Престижность.....	127
Как превратить группу в коллектив .....	128
Поручайте своим сотрудникам только те дела, которые они могут выполнить.....	130
О критике или метод кнута и пряника .....	131
 <b>Глава 7. Стратегический маркетинг — программисту, или как найти свою нишу на рынке software.....</b>	 <b>133</b>
MySQL.....	136
Microsoft.....	136
Sun .....	138
Adobe .....	138
WinAmp .....	139
Opera.....	139
Yandex.....	140
Электронные платежные системы .....	140
Вместо заключения.....	140
 <b>Глава 8. Проектирование пользовательского интерфейса .....</b>	 <b>143</b>
Компьютер — это средство, а не цель.....	144
Виды интерфейсов.....	146
Командная строка.....	149
Графический интерфейс пользователя.....	155
Обратная связь .....	162
Эмуляция реальных предметов .....	165
Многословность программ .....	167
Психологический аспект.....	171
Распространение программы.....	173
Хороший разработчик — хороший пользователь.....	175
 <b>Глава 9. Правила хорошего тона в программировании .....</b>	 <b>179</b>
Объявления переменных и функций.....	179
Имена переменных и функций .....	182
Расстановка фигурных скобок и отступы .....	188
Пробелы вокруг символов .....	197
Комментарии.....	198

Объектно-ориентированное программирование .....	201
Когда используется объектно-ориентированное программирование .....	202
Интерфейс класса .....	207
<b>Глава 10. Вместо заключения.....</b>	<b>213</b>
Управление временем.....	213
Мечтать не вредно?.....	215
О добре, которое нередко бывает с кулаками .....	219
Скажи мне кто твой друг, и я скажу, кем ты станешь.....	221
Упражнение: куда уходит время? .....	224
Упражнение на развитие временной интуиции.....	225
Несколько советов по организации времени .....	226
Осторожно — хронофаги! .....	227
О пользе разумного эгоизма.....	229
Самое главное условие достижения успеха .....	231
Вдох глубокий, руки шире!.....	233
Немного о виктимологии.....	237
<b>ПРИЛОЖЕНИЯ .....</b>	<b>241</b>
<b>Приложение 1. Договорные отношения.....</b>	<b>243</b>
Договор на создание сайта.....	244
Техническое задание.....	252
Вопросы и ответы .....	252
<b>Приложение 2. Авторское право .....</b>	<b>261</b>
Закон "Об авторском праве и смежных правах".....	262
Содержание закона.....	262
Тексты статей .....	265
Вопросы и ответы .....	310

# Введение

## Для кого и о чем эта книга

Эта книга о том, что необходимо предпринять, чтобы сделать успешную карьеру в области информационных технологий, и отличается от предыдущих книг авторов тем, что она — не о кодировании, а о тех вопросах, на которые многие программисты, и, особенно начинающие, не обращают должного внимания: как устроиться на работу, как вести переговоры с клиентами, как грамотно работать в команде и т. д. Статистика показывает, что большинство программистских карьер рухнуло не из-за того, что программист не умел писать код (наоборот, многие делали это блестяще), а по той причине, что авторы программ не уделяли внимания "второстепенным" вопросам, считая их не относящейся к делу "внешней атрибутикой": умению вести переговоры с клиентом, рекламе своего программного продукта, изучению положения на рынке программного обеспечения и т. д. Между тем, при нынешней большой конкуренции на рынке программных продуктов не обращать внимания на подобные "мелочи" — значит, разрушить свою программистскую карьеру в самом начале. Тому, как не допустить подобного исхода, и посвящена эта книга. В этой книге нет никаких теоретических рассуждений — все советы и рекомендации сугубо практические, основанные на опыте авторов, регулярно проводящих очные и заочные консультации по рассматриваемым в книге вопросам.

Несмотря на то, что в книге много разделов, к программированию как таковому не относящихся, все-таки эта книга для программистов. И та часть книги, в которой в той или иной мере сделан упор на психологию, тоже для них. Это как с английским языком: язык-то один, а специфика преподавания для тех или иных специальностей разная. И то, что касается психологии, авторы попытались изложить языком, понятным именно программистам.

## Описание глав

По сути, эта книга — анализ причин, почему те, кто потенциально мог бы стать хорошими программистами или IT-специалистами, ими не стали. И главы, из которых состоит эта книга, появлялись по принципу именно этого анализа. И именно по этому принципу книга виртуально разделена на две части "психологическую" и "техническую". Психологические главы —



это все, что касается взаимодействия с внешним миром: переговоров с заказчиками, работы в команде или руководства оной, продвижения собственного продукта и т. д. Техническая часть — все, что касается работы над программным продуктом. Интерфейс, проектирование, правила хорошего тона программирования. Причины, по которым некоторые программисты терпят неудачу, могут лежать как в технической сфере, так и во внешней. У одного вроде бы все хорошо, но продукт не идет, потому что интерфейс откровенно недружественный — появилась глава об интерфейсе. У другого все было хорошо до тех пор, пока не стал разрабатывать большой проект, в котором запутался, и завершить который не смог. Почему так бывает и как этого избежать — в главах о проектировании и хорошем тоне программирования. Теперь о внешней стороне. Многие успешные в техническом смысле программисты совершенно не могут продать свой продукт, потому что не владеют навыками общения с клиентом. Для них — глава о взаимодействии с заказчиком с многочисленными статьями-вставками из различных областей психологии, по сути представляющая собой небольшое пособие по ведению переговоров. В основном я приводил те приемы, которые сам использую в повседневной практике ведения переговоров. Кто-то, хорошо программируя, не сумел сработаться с командой, когда пришла пора. О том, как сделать процесс вхождения в новую команду менее болезненным — в главе о том, как работать в команде. А другой дорос до руководства командой и успешно ее развалил, в результате чего чуть не спился. О том, как разваливать команды, а наоборот сплачивать их — в главах "Работа в команде" и "Если вы — руководитель". Много вопросов у нас на форуме касается "институтской" проблемы — чему, где и как учиться программисту. Обо всем этом — в первой главе. В этой же главе мы говорим о том, что проблема, когда программировать очень хочется, а профильного образования нет — проблема надуманная. Нет такой проблемы. А сколько прекрасных программистов, что называется "от Бога", в силу тех или иных причин получив "не то" образование, загубили себе жизнь, послушав высокоумных дураков, говорящих о том, что без диплома они не программисты. Дорогой читатель! Если вы один из тех, кому так говорят, и вы уже почти в это поверили — поверьте теперь и в то, что это не так. Пока просто поверьте. А прочитав первую главу, увидите более чем убедительные реальные доказательства этого положения. Что диплом, это еще очень и очень мало. Хорошо, когда есть, ничего страшного, если нет. Не дипломами войны выигрываются, а умением. И если вы реально можете что-то хорошо делать (в нашем случае программировать), то ни один здравомыслящий руководитель ваш профильный диплом не спросит. А если очень хорошо умеете что-то делать, то, возможно, простят и отсутствие вообще какого бы то ни было диплома, а не только профильного.

Теперь о самой последней главе, которую можно было бы назвать "Фундаментальная", — от слова "фундамент", а не от нескромности автора. Просто

очень часто бывает, что говоришь с человеком, который пришел посоветоваться насчет проблемы, и видишь, что технически он еще, может, и грамотнее тебя будет, и все книжки по психологии на предмет того, как вести переговоры перечитал, но не идет дело, и все тут. Чудесный программный продукт, а продать никому не удастся. Первая мысль — неумение вести переговоры. И верно. Не умеет. Но о том, как их вести все прекрасно знает. А голос блеющий, руки потеющие... В общем, "все не так, ребята". А всему виной та самая оценочная зависимость, зависимость от чужого мнения, неверие в себя... Почти все через это прошли, но далеко не все успешно... О ней и о некоторых вариантах избавления — в "подглаве" прикладная виктимология. А у другого иная проблема. О которой он, правда, пока не догадывается. Сидит целыми днями и программирует, программирует без отдыха, без прогулок, без спорта. И вроде все хорошо, только как работник умственного труда он уже обречен. Потому что через пару-тройку лет такой жизни начнет производить умственный шлак вместо гениальных строк кода... Если вообще что-то будет в состоянии производить. Примеров — масса, исключений нет. ...Так появилась подглава о вреде нездоровья. А у третьего вообще ни на что не хватает времени. В принципе. А хочется так многого... Но ничего не успевает. О том, почему так, в разделе "Управление временем". Если у вас со временем, здоровьем и удачливостью все нормально, можете не читать эту главу. А вот если чувствуете, что что-то не так, задержитесь на ней подольше и постарайтесь решить вопросы, в ней освещаемые. С помощью этой книги, или других. Неважно. Важно то, что, не построив фундамент, постройку дома не начинают. И, к примеру, если вы не уважаете своих клиентов, а терпеть их не можете, то читать главу о том, как вести переговоры, бесполезно. Вы их все равно провалите, даже если будете безупречно соблюдать все рекомендации. И причина ваших неудач — гораздо глубже, чем неисполнение той или иной техники.

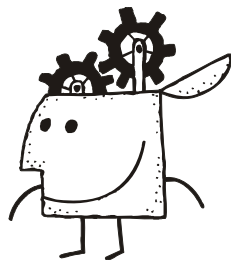
Все, что написано в этой книге, нами много раз апробировано и испытано. Это относится и к технической и к психологической стороне. О том, чего пока сами не испробовали и не знаем, мы не писали. Очень часто, особенно в главах, так или иначе касающихся психологии, повествование идет от первого лица. Поэтому, наверное, будет уместно сказать, что в основном "психологические" главы написаны М. Кузнецовым, а "технические" — И. Симдяновым.

## Благодарности

Мы признательны сотрудникам издательства "БХВ-Петербург", благодаря которым эта книга увидела свет, а также всем читателям наших предыдущих книг.



# Глава 1



## Программирование: где, чему и как учиться

Для начала абстрагируемся от программирования и поговорим об образовании и обучении вообще. И в первой части этой главы развеем один очень распространенный миф, который не одному человеку помешал стать тем, кем он хотел бы быть. А во второй мы приведем наши ответы на вопросы читателей, которые они нам задавали по поводу обучения программированию (как в рамках вузовской программы, так и самостоятельного) и попутно развеем еще несколько мифов.

**Итак, миф, который гласит, что, не имея диплома о высшем профильном профессиональном образовании, найти себе работу программиста (дизайнера, системного администратора и т. д.) и заниматься программированием невозможно.**

Проще говоря, если вы не закончили какой-нибудь ВМК или мехмат, то о карьере программиста можете забыть. Это — не так. И я говорю об этом, имея достаточные на то основания. Да, такая теория существует, она очень живуча и не одному человеку жизнь испортила. Да, те, кто такое образование имеет, нередко делают все для того, чтобы человек "не их касты" не проник "в святая святых".

### Примечание

Сейчас такого, конечно, меньше. Но учитывайте, что "высшеобразовательный снобизм" очень свойствен людям, получившим образование в 70-х и 80-х годах.

Теперь давайте эту теорию разрушим.

### *Примечание о методологии*

Кстати, методология говорит, что любую теорию можно опровергнуть только двумя способами. Первый — это доказать ее внутреннюю противоречивость. (На этом способе основан известный метод доказательства теорем "от противного"). А второй — показать, что основные положения теории противоречат реальным фактам.

Здесь мы пойдем по второму пути и приведем несколько значимых примеров, которые эту теорию абсолютно опровергают. Кстати, это справедливо по отношению не только к программированию, но и ко многим другим отраслям деятельности, в которых, казалось бы, без профильного образования вообще никуда. И тем не менее...

К примеру, одним из основоположников гипнотерапии в современной психотерапии считается Эмиль Куэ, корифеем от психотерапии и медицины никогда не бывший, а бывший всего лишь очень скромным аптекарем.

Всемирно известный писатель-зоолог Д. Даррелл никогда не учился на зоолога. Более того, он вообще нигде не учился, и считал, что ему невероятно повезло в том, что он не получил высшего образования, тем более профильного. И во многих интервью говорил о том, что если бы он выучился на зоолога, то никогда бы не стал "всемирно известным писателем-натуралистом и основателем и директором собственного уникального зоопарка".

Всемирно известный ученый-физик академик (!) Я. Б. Зельдович тоже вообще не имел высшего образования. Он лишь два года проучился на заочном отделении физико-математического факультета Ленинградского университета, который не окончил. Но он очень много занимался самообразованием и добился того, что в 1934 г. был принят в аспирантуру ИХФ, в 1936 г. защитил кандидатскую диссертацию, а в 1939 г. диссертацию на степень доктора физико-математических наук. А потом еще и академиком стал.

А очень известный в радиотехнике метод "прыжков по частотам" придуман, между прочим, не титулованным ученым-радиотехником, а ...американской киноактрисой Х. Ламарр. Которая, кроме того, что в эпоху Второй мировой войны была известной киноактрисой, еще очень интересовалась радиоэлектроникой. Сугубо для себя, казалось бы. И вот, когда она по приглашению каких-то там чинов из ихнего министерства обороны присутствовала на учениях, на которых демонстрировались торпедные атаки, ей пришла в голову мысль, что для надежной защиты от помех систем радиоуправления торпедами нужно, чтобы источник и приемник сигнала синхронно меняли частоту. Вот именно этот метод и получил название "прыжки по частотам".

### Примечание

Так часто бывает, когда значительный вклад в развитие какой-либо научной области делают люди, от науки очень далекие. И Х. Ламмар как киноактрису мало кто помнит, а вот как первооткрывателя метода "прыжков по частотам" знают многие. Кстати, эта идея лежит сейчас в основе помехозащищенности и секретности передачи информации в цифровых системах сотовой связи, таких как GSM, DECT и т. д.

А если вернуться к программированию, то лучше всего вспомнить ненавидимого многими Билла Гейтса, у которого, как всем известно, есть только

незаконченное профильное образование. И что-то он не сильно комплексует по поводу отсутствия оного.

А вот, к примеру, совершенно недавняя ситуация с моим участием. Пришли к нам устраиваться на работу два Web-дизайнера. Один не имеет никаких дипломов (вернее имеет — об окончании ПТУ), а у второго диплом о профильном высшем образовании (специальность "Компьютерная графика"). Но первый рисует гораздо лучше второго (вернее второй не умеет рисовать вообще). И если я, как руководитель, не являюсь идиотом, конечно, я возьму (и взял) на работу первого, какие бы чудесные "корочки" не были у второго.

Думаю, примеров достаточно. Хотя приводить их можно бесконечно, — их очень много. Кстати, три четверти самых богатых людей на земле вообще не имеют никакого образования. И не очень по этому поводу расстраиваются.

Конечно, я самый первый буду за то, чтобы у вас был диплом о высшем образовании, и желательно профильном. Я не о том говорю, что диплом не нужен. Я о том, что если уж по каким-то причинам его нет, то крест на себе ставить не надо. И о том, что если он есть, это еще совершенно ничего не гарантирует. Потому что людей, которые не могут устроиться после окончания "программистских факультетов" по специальности, тоже очень много.

### Примечание

Примеров таких действительно очень много. К сожалению, многие, получившие профильное образование, считают, что если они успешно прослушали все курсы, — то все уже сделано, и можно дальше ни о чем не беспокоиться. Это не так. Да, диплом, в частности, и очень хорошее университетское образование вообще — это хорошая база. И только. А все остальное нужно сделать самому.

## Вопросы и ответы

В этом разделе приведены реальные вопросы наших читателей, касающиеся обучения программированию, которые они нам задавали на форуме и по почте, и наши ответы на них.

**Скажите, с чего вы начали? Трудно было сначала? Я прочитал три электронных учебника, просмотрел кучу примеров, но ничего не понял — это значит мне не дано понять язык PHP?**

*И. Симдянов:*

PHP не совсем удобный язык для изучения. Дело в том, что большинство PHP-программистов не изучало язык с нуля, а изучало его, уже зная либо C, либо Perl.

Поскольку он просто логически следует из Perl и широкое распространение получил именно из-за того, что программистам было легко переходить с Perl

на РНР. Кроме того, оба этих языка являются С-подобными, т. е. имеют синтаксис и конструкции языка С, являющегося одним из самых сложных языков программирования. Кроме того, для его серьезного понимания желательно знание очень многих вещей (HTML, SQL, UNIX, прикладные протоколы, хотя бы HTTP, стили программирования на С, Perl, Java).

Это не значит, что РНР сложно изучить: любой язык можно изучить, проработав хорошее руководство страниц на 900. Но так как РНР находится в начале своего развития, здорово эволюционирует от версии к версии, и еще не устоялся, хороших руководств нет. Обычно, перед изучением РНР советуют сначала изучить либо С, либо Perl, так как они уже устоялись и по ним не сложно найти хорошую книгу.

Изучение языков программирования очень индивидуально, кто-то делает это наскоком, кому-то требуется время. Но самое главное — при чтении руководств и учебников необходимо возвращаться и прорабатывать главу несколько раз, до тех пор, пока полностью ее не поймете. Так как следующие главы в основном опираются на предыдущие. При изучении С я даже откладывал руководство на полгода и потом возвращался к нему — дело после этого пошло гораздо веселее. Мне сложно давать советы по изучению РНР в качестве первого языка программирования, так для меня он был восьмым.

*М. Кузнецов:*

1. Не обижайтесь, но прочтение трех электронных книг и просмотр кучи примеров — это очень мало. Книги надо долго прорабатывать, а примеры надо разрабатывать самому. Пусть для начала и очень простые. Но — свои.
2. Начните с изучения "азов", на которых строится любой язык: разберитесь с логическими операторами, порешайте алгоритмические задачки, узнайте, что такое область видимости переменных, изучите работу с массивами, файлами, строками и т. д. Действительно, РНР для этого не самый подходящий язык для изучения основ программирования вообще.
3. По поводу "не дано" — неправда. Миллионы людей смогли, а вы не сможете? Дано, но только надо много и долго работать.

**Все говорят, что программирование и математика это почти одно и то же. Ни разу не видел аналогии. Скажите, пожалуйста, кто, что думает по этой теме?**

*М. Кузнецов:*

Нет, это далеко не одно и то же. Просто иногда так считается из-за того, что программированию у нас традиционно обучают на математических факультетах вузов, и у тех программистов, которые оканчивают эти факультеты, в дипломах, как правило, написана специальность "математик". Дело в том, что на заре становления информатики в России ею занимались преимущественно математики. В отличие, скажем, от той же Америки, где спе-

специалисты по информатике инженерные науки изучают в большей степени, нежели математические. Судить, какой подход лучше, я не возьмусь, плюсы и минусы есть и там и там. Американский подход (инженерный) более ориентирован на рынок, так как предполагает быстрое решение вопроса путем применения стандартных типовых решений. Российский подход более ориентирован на фундаментальное осмысление проблемы. Другое дело, что, наверное, не очень хорошо, что у нас в вузах пока еще отдается предпочтение только одному "математическому" подходу. Было бы неплохо, если бы на ряде факультетов у студентов была возможность выбирать, так сказать, между стилями обучения. Сейчас ситуация в этой области начинает исправляться, но пока не так, как хотелось бы. Ведь в программировании есть разные задачи: есть и такие, где знание математики нужно. Это, как правило, задачи, связанные с разработкой наиболее экономичных алгоритмов. В фирме Intel научные сотрудники этим занимаются. Графы там всякие строят, ищут кратчайшие маршруты обхода этих графов и т. д. А есть и такие задачи, и их большинство, где нужен подход скорее инженерный. Я не помню, чтобы мы с того момента, как ушли из университета и стали работать в SoftTime, сталкивались в программировании с какой-то серьезной математической проблемой. "Экономичность" алгоритма в большинстве случаев меня и заказчика тоже мало волнует. Заказчику главное, чтобы он мог быстрее начать работать, и чтобы программа работала. А если я ему скажу, чтобы он подождал пару месяцев, потому что мне надо придумать более красивый алгоритм, так он меня пошлет далеко, и всё. Даже если мне надо, к примеру, знать, какой алгоритм чего-либо наиболее быстрый, то я просто открою книгу и посмотрю. А математическое доказательство этого меня совершенно не волнует.

...Если вы спрашиваете, можно ли стать программистом, не зная математики, то ответ однозначный — можно. Повторюсь — я уже не раз говорил, что знаю неслабых программистов, у которых специальность не то, что не математическая, но даже не естественнонаучная.

Если интересно, то по большому счету программисту не помешают следующие разделы математики:

- дискретная математика (дискретный анализ, комбинаторика, математическая логика, теория графов), да и вообще вся алгебра;
- методы математического моделирования;
- теория приближенных вычислений (в вузовских курсах иногда обзывается "вычислительной математикой" или "численными методами");
- в некоторых случаях — теория вероятностей и теория игр.



*И. Симдянов:*

Небольшое уточнение. Те, кто говорит, что программирование и математика это одно и то же, исходят не только из этих позиций. Дело в том, что программирование в том виде, в котором оно существует сейчас, не всегда было таким — на заре программирования программисты были сильными математиками, которые для написания более экономичного алгоритма доказывали теоремы, а алгоритму вычисления синуса угла посвящались диссертации. И по сей день тем кто занимается системным программированием, поддержанием научного ПО, библиотек алгоритмов, графических, криптографических библиотек — знание математического аппарата необходимо. Но в 99,5% случаев программист имеет дело с прикладной программой, насыщенной арифметическими действиями (причем, которые за него производит машина). Современный программист является пользователем — только на более глубоком уровне — он использует наработки предыдущих поколений: вряд ли кому-нибудь в голову придет менять алгоритм вычисления синуса — есть задачи и поинтереснее, в том числе и с применением стандартной функции синуса. Программирование — это инженерная специальность, у нас действительно программирование изучают на математических факультетах, на западе для этого имеется отдельная специальность — Computer Science: там изучают принципы операционных систем, протоколы и т. п.

На самом деле программисту не помешают деньги, на которые можно купить библиотеки, реализующие все это "добро". Математический код отлаживается очень тяжело и время, потраченное на его разработку, часто стоит много дороже, чем готовые библиотеки. Впрочем, это относится не только к математическому коду, но и вообще к программным библиотекам.

**Сам увлекаюсь комбинаторикой, в будущем собираюсь стать программистом. Мне интересно, насколько сильно нужна комбинаторика и высшая математика программисту?**

Это сильно зависит от области специализации. Системному программисту не помешает, также будет полезна прикладному программисту, если он имеет дело с инженерной предметной областью. Жизненно необходимо для создания движков игр и научного ПО.

Программист-прикладник, как правило, имеет дело со стандартным кодом. Это раньше по алгоритму вычисления синуса в калькуляторах диссертации защищали, а сейчас имеются промышленные библиотеки, с которыми и имеет дело прикладной программист. Реализация собственных хитрых алгоритмов вряд ли потребуются — они реализованы в STL и других стандартных модулях — вот их изучить придется... Математика вам потребуется, только если она будет вашей предметной областью — иначе будет дешевле использовать чужой код, так как вряд ли вы пойдете для решения какой-либо задачи на создание собственного языка программирования или библиотеки —

коллеги ничего не поймут и предпочтут выкинуть ваш код и написать так, чтобы всем все было понятно.

**Нужно ли для глубокого усвоения языков программирования высокого уровня, в частности, PHP, разбираться в высшей математике? И если да, то каких ее областей следует коснуться? Какие математические знания необходимы для Web-программиста (по образованию я — лингвист)?**

Нет. Линейная алгебра бывает очень полезна в программировании, но не в Web-программировании. Для глубокого усвоения лучше изучить несколько языков программирования, включая Perl и C++, но это необязательно. Для того чтобы сносно программировать, достаточно школьного курса математики плюс матричное исчисление (все-таки его неплохо бы знать).

При изучении (повышении уровня) о предварительном изучении математики лучше не думать (там можно надолго застрять), а лучше сосредоточиться на основах операционных систем (Windows, UNIX), в основном это дает власть над кодом, а не математика.

> по образованию я — лингвист

Кстати, Ларри Уолл, создатель языка Perl (являющегося веховым, как C++, Java), тоже является лингвистом. Создавая свой язык, он старался, чтобы тот походил на естественный язык, каждый оператор действовал в своем контексте, т. е. если что-то в списочном контексте — он ведет себя как список, если в скалярном, как скаляр и т. п. PHP во многом наследует Perl, хотя он не такой целостный (его делали программисты, и они очень торопились). Если встретите его книгу "Программирование на Perl" издательства O'Reilly, 2005 — обязательно берите, в ней, конечно, более 1000 страниц, но она вам позволит более глубоко понять как Perl, так и PHP, и, возможно, вы уже не захотите программировать больше ни на каком языке — при изучении Perl это рядовое явление. Правда, книга не из легких, но написана с юмором, продравшись через нее, вам PHP будет "по колену".

...У нас в стране не было специальности Computer Science, где как раз больше внимания уделяется операционным системам, протоколам связи и т. п. У нас преобладали либо факультеты для инженеров, которые в производственных условиях работали (в политехах; они как раз ближе к Computer Science), и факультеты ВМК (вычислительной математики и кибернетики). Первые работали на производстве и еще 20 лет назад программирование и математика были неразделимы. Это сейчас, когда создан огромный базис, все просто, а раньше защищали кандидатские диссертации по алгоритмам вычисления синуса в калькуляторе. Вторые (ВМК) создавали научные основы производства и занимались собственно кибернетикой — моделированием систем, что требует серьезной математики, поэтому на данных факультетах ей в основном и занимаются. Создание операционных систем и ПО для 3D-моделирования тоже подразумевает неслабое знание математики.

Поэтому традиционно на вступительных экзаменах сдают математику, хотя отличные программисты нередко получают из троечников. Но большинство факультетов являются все-таки математическими и учат там не программированию (это средство), а математике.

**Куда пойти учиться? Случилось так, что скоро получу диплом о высшем образовании, правда в химии, но что-то меня она уже не прельщает... Теперь хочется учиться по другой специальности, в области Web-программирования. Есть ли в Москве такой ВУЗ?**

В Москве вузов очень много. Если касаясь программирования — ВМиК МГУ, к примеру. Но еще раз тянуть такую ляжку... Стоит ли? Классическое образование, особенно в таких динамичных штуках, как IT, всегда отстает от того, что реально творится и делается. Хорошие вторые корочки, вы, безусловно, получите, а вот реальные знания, которые можно реально применять в повседневной работе, вряд ли. Я здесь IT-область, скорее, имею в виду, — в химии, к примеру, с этим немного попроще. По поводу Web-программирования... Конкретно ему, почти 100%, вас нигде не научат. В лучшем случае вам предложат двухсеместровый курс плюс дисциплина специализации за все время обучения. И в довесок кучу мусора, которую вы либо уже знаете, либо она вам нужна как одеяла папуасам. Конечно, при многих факультетах (и в Москве этих мест масса) есть платные факультативные курсы, на которых в течение, скажем, 4—5—9 месяцев говорят о Web-программировании, Web-дизайне и т. д. По окончании — сертификат. Это уже, конечно, не второе высшее. Но и тут есть немало "но". Если эти курсы ведут штатные преподаватели университета, ходить вряд ли стоит (это если речь идет о знаниях, а не о корочках). А вот если занятия ведет практикующий человек, каждодневно сталкивающийся в своей работе с "реальной жизнью", — это посетить не мешает.

В нашей стране сейчас по статистике более 80% выпускников работают не по специальности (цифра, скорее, занижена, некоторые приводят данные, что более 90%). Профессия по диплому это ведь как имя, которое дается нам родителями, и про которое никто не спрашивает, нравится оно нам или нет. Тут дело не всегда в родителях, а в том, что в 16 лет, когда поступаешь в вуз, сложно осознать свое истинное призвание. Это приходит попозже... Если вы беспокоитесь о том, что с вашим химическим образованием вас не возьмут работать IT-специалистом, то этого делать не стоит. Сейчас, как мы уже не раз говорили, чаще всего берут по реальным умениям. Знаю даже историков по базовому образованию, которые сейчас не без успеха программируют в серьезных конторах серьезные вещи.

**Мне 18 лет... И если честно, то я ничего не знаю почти, и хотя "комп" с 10 лет, а толку мало. И я по этому поводу очень расстраиваюсь. Конечно, есть какие-то знания, но мне кажется, этого мало, очень мало. Так сказать, сам себя критикую. Мне интересно, конечно, если это не секрет, а во сколько лет вы**

**начали и сколько лет учили тот или иной язык. И есть ли у меня перспектива в моих познаниях? Извините за пессимистичное настроение...**

1. Во-первых, не расстраивайтесь, когда чего-то не понимаете. Точнее говоря, расстраивайтесь конструктивно. Такое настроение бывает у каждого, кто что-то пытается делать. Кажется, что вот, вроде перелез через стену, все понял, и... новая стена, на которую снова надо взбираться. И это — хорошо, и огорчаться этому не стоит. Хотя, признаюсь, это первое чувство, которое возникает у меня самого, когда что-то не получается. Правда, сейчас я научился быстро это перешагивать.
2. >Так сказать, сам себя критикую.

Может я немного не в тему, но, так сказать, на будущее... Вы знаете, самое трудное, наверное, будет еще дальше... Когда критиковать себя будете уже не вы, а совсем другие люди. И критиковать очень часто — совсем не конструктивно. И люди эти будут не просто с улицы, а те, которым вы верите или верили... Вот это действительно страшно. Потому что с собой еще можно договориться, а когда человек, которому ты веришь, говорит, к примеру, что тебе не стоит заниматься программированием (и т. д.), потому что у тебя мозги под это не заточены — вот этому противостоять очень сложно. Мне так говорили, и перешагнуть через это было очень сложно. В первую очередь, потому, что я привык верить тем людям, которые это говорили, чуть было не поверил и тогда. Теперь я понимаю, почему так говорят, и знаю, как этому противостоять. (По этому поводу есть очень хорошая поговорка: "Я все сделаю для того, чтобы ты был счастливым при условии, что ты и дальше будешь оставаться несчастным"). А по "первости" можно очень просто, как Высоцкий пел, "попасть в чужую колею, глубокую" и перестать верить в себя. (Очень, кстати, хорошая эта песня. Она так и называется — "Чужая колея". Студентам ее особенно стоит послушать). А Ф. Д. Рузвельт так и говорил, что "самое сложное — поверить в себя и поверить себе". Именно поверить, а не знать, не заболеть звездной болезнью... То есть — просто хорошо знать себе цену. Не выше, не ниже — а ровно столько, сколько нужно.

3. И. Симдянов очень хорошо написал по поводу приказов (*Видел массу примеров, когда люди развивались таким образом, что могли программировать, если им приказывали: создать то-то, сделать то-то. А потом, когда они тому, кто им приказывал, уже были не нужны, они садились и смотрели телевизор в ожидании приказа, который почему-то никто не давал...*). Действительно, многие (как правило, отличники) достигают очень немало, пока с них чего-то требуют, пока их хвалят за сделанную работу. А когда ничего не требуют, то все: человек хиреет, чахнет, ничего делать уже не может... Есть очень хороший парафраз поговорки "Скажи мне, кто твой друг, и я скажу, кто ты". И звучит он так "Скажи мне, как ты переносишь одиночество, и я скажу кто ты". Я это понимаю так, что надо

учиться общаться с самим собой, быть самому себе интересным собеседником. Применительно к нашему разговору это значит, что нужно самому себе ставить задачи, решать их, а не ждать пока это сделают другие. Ведь, заметьте, — многие люди, когда уходят на пенсию (спортсмены, военные особенно), очень часто после этого чахнут и вскоре умирают. Мне думается, из-за того, что они привыкли жить "под приказом", и другой жизни просто уже не мыслят. И с другой стороны: лично хорошо знаю одного военного, который ушел на пенсию в 45 лет в звании подполковника и не помер, не спился, как многие его коллеги, а организовал свое дело. Теперь ему 52 — очень крупный бизнесмен. Вот что значит умение хорошо ладить с самим собой.

4. Программировать начал с Паскаля. Потом — Fortran 77/90, C++, Java, чего-то-еще-уже-не-помню, и вот теперь PHP и все, что с ним связано. Всегда тянуло заниматься сетевым программированием, но удалось заняться плотно только сейчас, а до этого занимались разработкой всяких программ научно-технической направленности. Заниматься программированием, как и И. Симдянов, стал в 18 лет и в том же самом месте, что и он (мы вместе учились — и программированию в том числе). Так что он говорит правильно — у вас перед нами огромная фора. Главное — не сбиться с пути. Кстати, я в школе вообще не мог программировать и очень этого боялся, а все программы за меня писал мой хороший школьный товарищ Сашка Котляревский, который потом закончил факультет ВМК. А потом как-то так получилось, что увлекся программированием.
5. > И есть ли у меня перспектива в моих познаниях?

Резюмирую. Перспектива у вас настолько громадная, что мне даже сейчас сложно предположить, чего вы сможете достичь. Главное — не растерять запал. Я очень рад, когда читаю, что те, у кого компьютер появился намного раньше, чем у меня — занимаются делом. Честно говоря, я этого даже не ожидал, поскольку большинство моих сверстников, у которых компьютеры появились достаточно рано, увы, ничего стоящего из них не выжали. Поэтому не расстраивайтесь, не отчаивайтесь — разбирайтесь со всем этим безобразием, которое сейчас называют IT-индустрией, а мы чем сможем — поможем.

**Вот хочу узнать, что нужно знать перед тем, как искать работу программиста. Мне еще только 15 лет, но я уже имею приличные знания PHP и MySQL. Планирую доучиться до 11 класса, а потом уже искать работу. Так вот, пока есть время набрать знания, что мне нужно, чтобы быть уверенным, что я не останусь без работы? Пожалуйста, расскажите, где вы учились, с какого возраста начали программировать и когда начали работать. Мне это очень важно.**

Я начал программировать по вашим меркам уже глубоким стариком — лет в 18, наверное. Или в 17. Точно помню, что уже в вузе. А вы в 15 уже знаете

PHP и MySQL. Поэтому в вашем случае, скорее, надо постараться не перегореть. Потенциал у вас хороший — главное не растерять его по дороге.

Мы в годы обучения в вузе сетевым программированием заниматься совершенно не собирались, а занимались разработкой научно-технических программ, так скажем. А получилось, что стали известными в области сетевого программирования. Это я к тому, что дело, по большому счету, не в том, что вы сейчас выучите, а в том, в какой форме будете к часу X. Потому что выучить-то можно, а технологии могут смениться. И что — менять профессию? Такого быть не должно. Поэтому — главное форма. Чтобы вы четко про себя знали, что если надо — за 3 недели выучите любой язык и минимально (для того, чтобы начать работать) освоите любую технологию. А то, что вы знаете сейчас — не суть важно. И что я знаю — тоже. Вот завтра появится новая технология — и кто ее быстрее освоит, тот и будет на коне. В этом смысле я за универсализм. Нам вот, к примеру, совершенно не нужен человек, знающий только PHP. (Вообще ведь это достаточно простой язык). Язык — это только инструмент для решения задач. А они могут быть очень разные. Значит, и инструмент в запасе желательно иметь не один, или, хотя бы, знать какие бывают. Потому что сегодня надо, к примеру, делать интернет-магазин, завтра — скрестить его с бухгалтерией 1С, после завтра — писать ПО для учета заказов, с Интернетом никак не связанное, а через неделю все это бросить и пойти делать ПО, которое считывает сигналы, посылаемые на порт собственно созданной схемой. То есть, понимаете — задачи перед вами могут стоять очень разные.

Вот вы говорите, "чтобы быть уверенным, что я не останусь без работы". Для этого надо пахать часов по 12 в сутки и знать, чего хочешь. Тогда я буду уверен, что что бы не произошло — у вас все будет прекрасно. Оба условия обязательны (много работать и знать, чего хочешь). Причем второе — даже поважнее первого: дизель может работать хоть круглые сутки, но если вхолостую — пользы от него никакой.

Ну и еще. Так сказать, техническая рекомендация. Поучите C++. И объектно-ориентированный подход. Просто для себя. Не помешает — точно. А с вероятностью 80% — не один раз пригодится.

**Здравствуйтесь! Я сейчас выбираю вуз, в который буду поступать, посоветуйте, как это лучше сделать.**

>Здравствуйтесь. Я сейчас выбираю вуз, в который буду поступать.

Обычно сначала выбирают специальность. И речь нужно вести, скорее, не о вузе, а о факультете какого-то вуза, и даже о специальности в рамках данного факультета. Сейчас на каждом факультете почти везде есть 3–4 специальности. Поэтому для того, чтобы вам кто-то что-то подсказал, нелишне знать, какую специальность вы хотите получить, принципиально ли для вас, в каком городе находится вуз, где вы сами живете, — Москва и Ново-Уральск разные города.

По поводу специальности. Это серьезный вопрос. И здесь надо сначала определиться, зачем вы идете в вуз. Как правило, грубо говоря, существуют два варианта:

- просто получить корочки и "общефилософское" образование;
- получить специальность, которая реально будет на вас работать.

Оба варианта достойны, и я, если честно, не знаю какой лучше. Можно пойти по первому варианту, получая отсрочку от армии, неплохое образование и корочки, а в свободное время изучая нужную себе специальность, программирование, скажем.

А можно — по второму варианту. Но тут надо серьезно поработать в плане выяснения того, куда лучше податься. Чтобы не скатиться к первому варианту. И стоит провести предварительную разведывательно-аналитическую работу. Допустим, вы хотите стать программистом.

Выясните, многие ли из окончивших данный факультет работают по специальности. Ах, никто? Или 5% выпускников, да и то потому, что их по блату пристроили родители? Значит, это — не ваш факультет.

Придите на лекцию по программированию на том факультете, на который собрались поступать. Потому что, если вы придете на лекцию по программированию и услышите вначале что-то типа "Основываясь на 121-м следствии из 143-ей теоремы Свигнера—Бзингера, изучению которого мы посвятили 3 предыдущие лекции, докажем, что 5 нормальная форма существует", — то это не ваш вуз, если вы хотите стать программистом. Вот если математиком — то да.

Посмотрите, кто в вашем вузе читает программирование. Если 50—60-летний профессор, то вряд ли он скажет вам что-то стоящее в смысле программирования. В смысле математики — сколько угодно.

Узнайте, работают ли сами преподаватели по той специальности, которую они преподают. Тут дело вот в чем. Многие специальности, особенно прикладного плана, в вузах преподают непрофессионалы. Ведь как обычно бывает? Вызывает декан какого-нибудь профессора по математическому моделированию в деканат и говорит: "Слушай, Вася Петрович, тут такое дело — мы открываем новую специальность, в рамках которой будет читаться двухгодичный курс лекций по программированию систем дистанционного управления. Надо бы тебе его взять на себя. Да, все понимаю — не твоя специальность. Да, понимаю, конечно, лучше пригласить спецов из НПП «На взлет», но ты же понимаешь — это деньги. И лишняя ставка. А у нас их нет и в ближайшее время не будет, даже если половина профессорско-преподавательского состава «коньки отбросит»". И идет Вася Петрович читать лекции по не своей специальности. Конечно, он их выучит. Но тонко-

стей, изюминок и изящных и простых объяснений (то единственное, ради чего стоит посещать лекции — все остальное можно в учебниках прочитать) вы от него не услышите.

У вас не должно остаться сомнений относительно того, что вуз — дело серьезное. Это 5—6 лет жизни, причем в самом ее расцвете. И от вашего нынешнего выбора зависит очень много.

**Московский государственный инженернофизический институт (МИФИ). Специальность: "Комплексное обеспечение информационной безопасности автоматизированных систем". Объясните проще, что будешь делать на работе (если работать по специальности, конечно), если у тебя будет такая специальность?**

*И. Симдянов:*

Предположу, что человек с такой специальностью должен уметь проектировать и писать ПО для всяких хитрых железок типа детекторов присутствия, систем видеонаблюдения, поиска скрытых камер, сигнализации; создание защищенных локальных систем (как на физическом, так и на программном уровне), распознавание образов, "прослушка" обычных и сотовых телефонов, а также защита оных от прослушки и т. п. Ну, я, по крайней мере, от выпускника ожидал бы, чтобы он мог приступить к решению таких задач.

*М. Кузнецов:*

Сказать что-то конкретное по этому факультету я затрудняюсь, так как лично никого с него не знаю и ни с кем не встречался — ни с преподавателями, ни с выпускниками. Поэтому про качество образования, работают ли выпускники по специальности и т. д. — точно говорить не могу. Это проще вам выяснить — возьмите водки и отправляйтесь в университетскую общагу к тем, кто заканчивает этот факультет. За пару часов непринужденной беседы вам все расскажут — как учат, кто, где и кем работает после окончания факультета и т. д. Я серьезно. Это, пожалуй, один из немногих способов узнать гарантированную информацию. Другой — посидеть на нескольких лекциях по профильным предметам и послушать, что на них говорят. Здесь принцип такой. Пришли вы, к примеру, на лекцию по внутренним датчикам систем физической защиты. И если лекцию ведет профессор, д. т. н., и начинает ее словами "Сделав обратное Фурье-преобразование, мы увидим, что...", это — профанация. Потому что ничего не увидите. Так как скорее всего профессор какой-нибудь выдающийся математик, но из датчиков физической защиты у него самого только престарелая болонка, которая, если проснется, то затыкает, когда кто-то в дверь позвонит. А вот если лекцию по этому предмету читает спец из ФСБ, ФАПСИ или серьезной фирмы с охранным профилем, который полжизни промотался по "объектам", эти самые датчики устанавливая, то тогда — да. Тогда вы по окончании факультета будете знать очень немало.

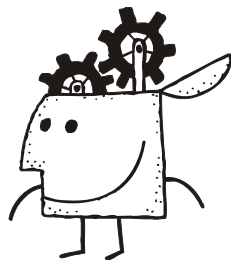


Вообще говоря, МИФИ — институт очень серьезный. И если бы вы пошли туда учиться на физика, я бы гарантировал, что физик из вас по окончании вуза выйдет хороший. Потому что МИФИ всю жизнь специализировался на "ядрениой бомбе". Этим и известен. И физика там поставлена очень серьезно. Как и математика. В части же "информационной безопасности" есть опасение очередной профанации. В очень многих вузах сейчас созданы подобные кафедры. Потому что нынче это модно и за это платят деньги. То, что диплом МИФИ это серьезный диплом — здесь вопросов нет. А вот получите ли вы какие-то реальные знания — не знаю. Немало бы сказало, если бы вы посмотрели, какие в принципе лекции читаются на данном факультете и кто их читает. Если есть возможность списать или сфотографировать лекционный план по данному факультету — сделайте это. Можете прислать мне на почту — я вам тогда скажу достаточно конкретно и с высокой вероятностью — кем вы будете и что будете уметь после окончания факультета.

А по вопросу кем придется работать и что делать... В принципе И. Симдянов вам ответил, чего бы он ожидал от выпускника с такой специальностью. Я бы ожидал примерно того же. Только чуть больше. Потому что спец по безопасности — это не просто диплом — это образ жизни. Пусть даже и по информационной безопасности. Вернее — особенно по информационной. Вот, к примеру, одна из типовых задач. В фирме (госучреждении, семье, ку-клукс-клане — нужное подчеркнуть) утечка информации. Ваша задача, как спеца по информационной безопасности, выяснить, откуда безобразие это проистекает. И если вы придете в фирму в разгар рабочего дня, обвешанный аппаратурой и сумками с дисками с программами по поиску "троянов" и во всеулышанье заявите, что пришли искать источники утечки информации... Я бы, как руководитель конторы, занимающейся безопасностью, вас тут же уволил. Для вашей же пользы. Потому что с таким подходом вы можете сильно укоротить себе жизнь.

Другая задача. Есть ВЧ. В смысле, воинская часть. И вам надо сделать проект организации систем контроля доступа. Еще пример. Сайт взломали хакеры — их надо найти. Самим. Без помощи ФСБ, МВД и т. д. Вот примерно таких умений я бы ожидал от человека, у которого в дипломе написано "Специалист в области комплексного обеспечения информационной безопасности автоматизированных систем и специалист в области программно-аппаратных методов обеспечения безопасности информации" и который прослушал курсы "Внутренние датчики систем физической защиты", "Системы контроля доступа", "Системы видеонаблюдения", "Системы сбора и обработки данных", "Внешние датчики (с полигоном периметровой системы) физической защиты", "Банковские телекоммуникационные системы и их средства защиты" и т. д.

## Глава 2



# Как найти работу

## Базовая техника устройства на работу

В этом разделе мы рассмотрим базовую (общую) технику устройства на работу, а затем более подробно остановимся на некоторых отдельных этапах.

Итак, надо устраиваться на работу. С чего начать? С собеседования? С составления резюме? С телефонного звонка секретарю? Позвольте спросить, а почему вы так считаете? Вы решили начать с составления резюме, потому что у вас ручка с бумагой рядом оказались? А вы с телефонных переговоров, потому что вам телефон ближе? А у вас та фирма, в которую вы собрались устраиваться, снимает помещение на первом этаже в вашем подъезде, поэтому вы решили зайти пособеседоваться? Ну что ж, если вы так считаете, то представьте на минутку себя верховным главнокомандующим. Представили? Сидите, вы, значит, в Первопрестольной и решили пойти войной. На восток. Стратегический интерес у вас там. Приходят к вам поутру воеводы, и докладывают:

- Царь-батюшка, войска готовы к выполнению боевой задачи. Что приказать изволите?
- Польшу брать будем, — брякаете вы.
- Как Польшу, почему Польшу? — изумляются воеводы, — мы ж на восток собрались!
- А потому что Польша ближе, — демонстрируете вы железную полководческую логику.
- Но как же так! Польша же с нами мирный договор подписала! Да и вообще мы на восток собирались!
- На восток, говоришь... Ну, тогда возьмите город Баку. Не восток, конечно, но зато я там в армии служил. Дорог он мне этот город. Как память. А восток... Восток подождет.

...Вот примерно такой же логикой руководствуются многие из нас, устраивающихся на работу. Плохая логика. Потому что все должно начинаться с цели. И устройство на работу тоже должно начинаться с цели. А не с составления резюме или собеседования. С цели, и только с нее. То есть с ответа на вопрос о том, для чего вы устраиваетесь на работу. Ответ "Потому что уже надо где-то работать" не проходит. Нужно четко определиться, зачем вам нужна работа. Потому что не стоит идти работать научным работником, если ваша цель — зарабатывать денег и извлечение прибыли. Цель хорошая. Но и средства достижения должны быть соответствующие. В науке эту цель достичь сложно. Смолоду — практически невозможно. У нас в стране, по крайней мере.

### Примечание

Да и в других странах, насколько я знаю, примерно так же. Общаясь с зарубежными коллегами, я пришел к выводу, что наукой в большинстве своем занимаются небедные люди. Которые заработали достаточно денег, чтобы позволить себе заниматься наукой.

Поэтому, если хотите зарабатывать деньги, то по большому счету есть только три честных варианта, как это можно сделать. Или заниматься бизнесом, или стать руководящим работником в какой-либо организации, ворочающей миллионами, или стать специалистом экстра-класса в своей области. Экспертом. Таким спецом, чтобы кроме вас больше не к кому было обратиться.

А если все-таки хочется работать ученым и деньги получать? Что нет никакого варианта? Есть, конечно. Если подумать, то варианты всегда найдутся. Только для того, чтобы увидеть эти варианты, надо освободить свое сознание от некоторых догм. В нашем случае это догма о том, что наука делается только в вузах и научно-исследовательских институтах. Это не так. Вернее, наука делается не только в этих учреждениях. На западе, к примеру, немалый процент научных достижений приходится на долю различных фирм. Наиболее яркие примеры всем известны: корпорации Intel, Xerox, Kodak и т. д. У нас в России тоже к этому идет: уже сейчас во многих компаниях есть научно-исследовательские отделы и лаборатории, занимающиеся научными исследованиями в рамках основной деятельности компании. И, надо сказать, достаточно успешно.

### Примечание

К примеру, научно-исследовательский отдел одной московской компании, работающей на рынке разработки устройств защиты информации, за 3 года успешно решил серьезную научную проблему, над которой несколько лет бились радиофизические факультеты не одного вуза. Суть проблемы заключалась в разработке верифицированного метода улавливания и распознавания низкоинтенсивных сигналов среди спектра шумовых сигналов много большей интенсивности. В частности, решение этой задачи нужно для улавливания очень низ-

коинтенсивных сигналов от ПЗС-матриц видеокамер, находящихся в помещениях. А последнее необходимо, к примеру, для распознавания видеокамер, ведущих скрытое видеонаблюдение.

Поэтому, я уверен, что каждый может найти себе такую работу, которую он хочет. Главное, четко знать, что нужно (определиться с целью), а потом уже начинать сбор информации на предмет того, где находится то, что вам нужно. Этот этап не менее серьезен. Грамотно собрать информацию и ее обработать — это задача не из простых. Этому разведчиков по несколько лет учат. Но учат же, и успешно. Значит, и мы сможем. Тем более что наша задача не в пример легче тех, которые стоят перед разведчиками. Итак, вначале цель, потом — сбор информации. А уже после того, как вы четко определитесь, что же вы хотите, и где это место находится, уже можно приступать непосредственно к "штурму": к составлению резюме, звонкам в отдел кадров, собеседованиям. Хотя нет: между сбором информации и "штурмом" должен быть еще один этап: так сказать, непосредственная рекогносцировка местности. То есть определились вы, скажем, что хотите работать в фирме X, значит, надо узнать как можно больше о том, что это за фирма непосредственно на месте. Да не о том, чем эта фирма занимается и что производит, — это вы уже должны знать, а о другом: кто директор, кто заместители, как из себя выглядит секретарша, как их всех зовут и т. д. Здесь кашу маслом не испортишь, — чем больше вы знаете, тем более высоки шансы на успех. Во-первых, эта информация необходима для того, чтобы наметить наилучшие пути, по которым вы будете осуществлять "наступление". (И — отступления: всякое в жизни может произойти, и чтобы не было ситуации "вход — рубль, выход — два", лучше многое предусмотреть заранее). Во-вторых, вы еще раз проверите себя на предмет того, правильные ли выводы вы сделали из анализа первоначальной информации.

Итак, еще раз. Базовая техника устройства на работу включает в себя следующие основные этапы:

- формулировка цели;
- первоначальный поиск информации;
- непосредственный сбор информации о выбранном объекте;
- техника устройства на работу в нужную организацию.

В этом списке нет пунктов, касающихся составления резюме и собеседования, потому что они входят в последний пункт. Более того, нередко устройство на работу в выбранную организацию приходится начинать с... совершенно другой фирмы. Но об этом подробнее в соответствующем разделе.

А теперь рассмотрим каждый из этих этапов подробно. Но прежде поговорим еще об одном важном моменте: о настрое на поиск работы. О правильном психологическом настрое на поиск работы.

## Настрой на поиск работы

Во-первых, поиск работы это такая же работа. Которой надо заниматься ровно столько же, сколько вы бы занимались своей основной работой. Пахать, в общем, надо. Тогда что-то обязательно получится. Вернее, не что-то, а то, что нужно. Во-вторых, если это дело для вас новое, то стоит потратить время на освоение методологии поиска работы. А вы что хотели... Свои тонкости есть в любом деле, и если заниматься этим делом успешно, то надо знать как наилучшим образом и с наименьшими потерями достичь намеченной цели. А если не изучать методологию, а действовать абы как, если не заниматься поиском работы каждодневно, а звонить от случая к случаю по первым попавшимся телефонам, если только встать на биржу, а самому ничего не предпринимать, то ничего не выйдет. Хорошего, по крайней мере. Ну и, в-третьих, вы должны верить в успех. Обязательно. Без веры в успех ничего предпринимать нельзя. Потому что, если вы не уверены в своем успехе, его у вас и не будет. Стопроцентно. Так как здесь работает всем известный принцип: "как корабль назовешь, так он и поплывет". А для того, чтобы вам лучше поверилось, приведу несколько вдохновляющих тезисов.

**Во-первых**, катастрофической безработицы, как таковой, нет. Что бы кто не говорил, и что бы не писали газеты. Нет безработицы. А уж в IT-сфере, тем более. Я уверен в этом и постараюсь сейчас свою уверенность передать вам, приведя многочисленные примеры, эту мою уверенность подтверждающие. А уж когда я слышу, что для выпускников факультетов ВМК нет работы по специальности, я точно знаю, что это ложь. Работа-то есть. Другое дело, что есть неумение искать работу. Потому что не привыкли мы пока еще за место под солнцем бороться. Мы к сказкам привыкли. Когда лежит кто-то на печи лет тридцать, а потом "встал, увидел, победил". А иногда есть еще и нежелание работать. Встречаю недавно одного знакомого, и он жалуется мне, что его уволили, и он теперь нигде не может найти работу. Спрашивает, не найдется ли у меня местечка. Спрашиваю, где работал раньше, отвечаю, что подумаю. В принципе работник вроде неплохой, а тут как раз несколько новых направлений намечается... Отчего ж не помочь. Но, подумав, отказал. Потому что, когда позвонил навести справки в то место, где он раньше работал, узнал, почему его уволили. И понял, почему он нигде не может найти работу. А уволили за то, что он провалил переговоры, явившись на них в подпитии. И, естественно, когда кто-то из тех работодателей, к кому он обращается, наводит справки на предыдущем месте работы, откуда ему дают "волчий билет". И таких примеров я знаю десятки. То есть проблема-то, получается, не в недостатке предложений на рынке труда, а в неумении этими предложениями воспользоваться. А это уже совсем другая проблема, к безработице, как таковой, никакого отношения не имеющая.

**Примечание**

Дорогие работодатели! Запомните, пожалуйста, этот пример, потому что он и ему подобные являются красноречивой иллюстрацией правила "Не гуляйте там, где живете". Вот этому человеку теперь для того, чтобы найти работу, надо либо уезжать в другой город, либо менять специальность, либо идти на низкооплачиваемую работу. Причем первые два пункта могут и не помочь. Хотя по своему техническому уму он, в принципе, заслуживает гораздо большего. А вот жизненного ума, увы, не хватило. И еще один немаловажный момент. В IT-мире, как и в любом другом (медицинском, журналистском и т. д.) все в принципе друг друга более-менее знают. И ничего удивительного в том, когда один начальник наводит справки у другого, нет. Все стараются получить максимум информации, и, по мере возможности, объективной. Я, к примеру, иногда даю рекомендации, положительные, конечно, наиболее активным участникам форума softtime.ru. Так мне откуда только не звонили, чтобы проверить подлинность этих рекомендаций. Иногда совершенно незнакомые люди. В чем тоже нет ничего удивительного. И я звоню, и мне звонят. Запомните это, пожалуйста, и в том месте, с которого уходите, старайтесь оставить о себе благоприятное впечатление. Дверьми хлопать не надо. Даже если вас уволили незаслуженно. Наоборот, и в этом случае постарайтесь сделать так, чтобы ваш начальник по звонку другого начальника, или по запросу рекомендательного письма, дал вам хорошую характеристику. И ни в коем случае не допускайте увольнения с "волчьим билетом", как в рассказанном примере: один прекрасно проведенный вечер может обернуться месяцами пасмурных дней.

Более того, в настоящее время в нашей стране на рынке IT наблюдается скорее переизбыток рабочих мест, чем недостаток. Есть из чего выбирать. И вот тут кроется еще одна неприятная вещь, которую в народе называют "с жиру бесаться". Штука действительно неприятная, потому что многие, скажем так, утрачивают чувство реальности и начинают творить вещи, за которые их потом вполне закономерно увольняют. Работа хорошая, начальник хороший, зарплата в принципе устраивает, значит, можно творить, что хочешь. ...Будучи недавно в Венгрии, разговаривал с директором их Web-студии. Средняя зарплата по Венгрии составляет 90—95 тысяч форинтов на данный момент (форинт — денежная единица Венгрии). В пересчете на наши рубли это примерно 12 000 рублей. Работник Web-студии в Венгрии получает чуть поменьше. По нашему около 10 000 рублей.

**Примечание**

Для сравнения. Бензин в Венгрии стоит более 1 евро, а батон около 20 руб. А при зарплате в 90 000 форинтов стоимость детского садика в месяц составляет 60—65 000 форинтов. Это я к тому, что жить там, точно, в денежном смысле не проще.

То есть в среднем зарплаты примерно такие же, как у нас. А жизнь даже подороже. Так вот, при этом уровне зарплат работают раз в 5 интенсивнее. Потому что знают, что чуть что не так, их автоматически увольняют. В Венг-