

# Когда я стану программистом?

Это первая глава из книги «Графомания», которую можно скачать на моём сайте: <http://oleg.derevenets.com>.

«Когда я стану программистом? — втайне терзается новичок, — сколько языков мне освоить, и каких? Что востребовано теперь на рынке? Вот выучу *ЭТОТ* язык, затем *ТОТ*, а уж потом...». И что *ПОТОМ* ?

## 1.1. «Всё проходит»

*Всё проходит* — сокрушался мудрый Соломон. Будь он программистом, то поправил бы: *всё проходит очень быстро*. Ещё бы! Всего лишь три десятка лет тому пара языков программирования вкупе с небольшой практикой вполне законно давали вам титул программиста. Ныне же соискателей звания смущают, по меньшей мере, два вопроса, второй из которых таков: как долго я останусь программистом? Ведь новые языки всё плодятся и плодятся, модные технологии и направления стремительно меняют ландшафт нашей профессии. Только вчера ты освоил новинку и был «на коне», а сегодня ловишь на себе сочувствующие взгляды юных конкурентов. Коллеги не дадут соврать: удел программиста — грести против течения: чуть сбавил усилия, бросил вёсла, и уже влечёт тебя поток к опасным порогам. Где якорь, что остановит губительное движение? Где опора о твёрдое дно бурлящего потока? Когда новичок станет программистом и как долго останется им?

Проявив каплю терпения, и одолев десяток страниц, вы сами ответите на эти важные для многих вопросы. Со стороны сфера программирования видится пышным раскидистым деревом. Да, некоторые его ветви заметно увяли, зато другие бурным ростом сулят заманчивые перспективы. И всё же это только ветви... Мы же докопаемся до корней, и начнём издалека.

## 1.2. История трёх революций

Прежде, чем двинуться к цели, выясним нашу исходную позицию. Где место программиста в современных технологиях? Мы найдём его через знакомство с тремя кибернетическими революциями.

### 1.2.1. Революция первая — биологическая

Она случилась миллионы лет тому с появлением первых примитивных одноклеточных организмов. Своё пропитание — энергию, они брали, главным образом, от солнца. А надо заметить, что уже тогда, задолго до рождения Дарвина, работали его законы: мутация и естественный отбор. Мутации — это случайные изменения организма под действием внешних факторов. Мутации могут быть и полезными, и вредными — вердикт о полезности выносит естественный отбор.

И вот, вследствие мутаций и отбора, часть микробов приноровилась брать дополнительную энергию из останков своих соплеменников, что усилило их конкурентное преимущество, — так появились наши предки (животные), и был сделан первый шаг в последующей истории. Известно, однако, что под лежащий камень вода не течёт: мудро прокормиться, сидя на месте, и питаясь лишь ближайшими соседями. Второй шаг эволюции привёл к случайному появлению примитивных органов движения — ресничек. И естественный отбор утвердил новинку: ведь даже хаотическое перемещение в среде обитания существенно обогащало рацион обладателей ресничек.

На третьем этапе эволюция сотворила примитивные органы чувств — своего рода датчики, которые воздействовали прямо на органы движения. Датчики могли реагировать на освещённость, температуру среды обитания и её химический состав. Движения организмов обрели некую степень разумности: примитивное животное двигалось не куда попало, а в направлении пищи или лучших условий обитания.

Позже развились промежуточные звенья, передающие сигналы от органов чувств к органам движения — прообраз нервной системы. Этот посредник помаленьку занялся логической обработкой сигналов, что сделало животных «умнее». Нервная система усложнялась и совершенствовалась миллионы лет, что и породило современных животных и человека. В нервной системе (мозгу) животного отражены и среда существования, и его роль в этой среде. Если отражение окружающей действительности назвать *МОДЕЛЬЮ*, то мозг и нервная система, безусловно, являются *БИОЛОГИЧЕСКОЙ МОДЕЛЬЮ* окружающей среды и той роли, которую играет в ней данный организм.

Отметим две особенности биологической модели. Во-первых, она передаётся только по наследству от родителей к потомкам: животные не могут передать генетический опыт иначе. Конечно, высшие животные иногда обучают потомство, организуют коллективную охоту и оборону, но это не меняет врождённую модель. Отсюда следует другая её особенность: биологическая модель крайне консервативна, она не успевает за резкими изменениями условий обитания.

### 1.2.2. Революция вторая: символические модели

Старт ей дали примитивные изображения животных и людей, вырубленные грубой рукой древнего художника. Восхищение от этих каракулей не уступало тому, что возбуждает заезжая рок-группа у обитателей провинции. Сегодня нам трудно их понять, но ведь тогда человек впервые нутром почуял возможность *САМОМУ* создать *МОДЕЛЬ* какой-то части окружающей его природы. Эта модель, в отличие от модели биологической, оказалась доступной всем зрячим соплеменникам художника: и живущим, и будущим поколениям. Модель можно было дополнять, копировать, развивать, и для этого не требовались миллионы лет эволюции или участие автора прототипа.

Первые письма (египетские, например) были рядом картинок, восточные иероглифы, вероятно, также порождены рисунками. Позже возникли современные

алфавиты, не связанные с изображениями чего-либо, а также цифры, математические знаки, нотная запись и т.д. Появились литература, науки, искусства и музыка. Всё это — современная культура, которая, подобно древним рисункам, являет совокупность *моделей* окружающего мира, но, в сравнении с наскальным рисунком, куда более совершенных. Эти модели объединяет то, что все они изображаются символами, и потому их называют *СИМВОЛИЧЕСКИМИ*. Модели хранят в себе накопленные знания и служат, как и биологические модели, всё той же цели: обретению преимуществ в борьбе за выживание. Но теперь говорим о выживании не только отдельных людей или групп, речь о государствах и человечестве в целом.

Отметим, что культура развилась с быстротой лавины: с момента её зарождения прошли не миллионы, а тысячи лет, особенно бурным выдалось предыдущее столетие. Однако накопившийся к середине 20-го века громадный объём знаний породил новую проблему.

### 1.2.3. Революция третья — «кибермозг»

Символические модели — науки, искусства — создаются людьми и оживляются ими же: даже самая удачная модель — научная теория — мертва без человека. Модель оживает только в головах тех, кто постиг её ценой обучения. Так, проявление моделей, связанных с созданием машин, возможно лишь через хорошо образованных людей: конструкторов, инженеров, рабочих.

Известно, что в деле инженера, наряду с творческой, немало и черновой работы: рутинных вычислений по известным формулам (а формулы — это тоже простые символические модели). Вычисления утомляют экономистов, бухгалтеров, и других работников умственного труда. К середине 20-го века с усложнением технологий эта рутинная работа буквально задавила специалистов, не оставляя им времени на творчество. Разверзлась пропасть между количеством и качеством научных знаний с одной стороны, и возможностью их применения с другой. Рождалось много в принципе полезных, но не осуществимых по причине их сложности моделей, — и научно-технический прогресс затормозился.

И тогда (не чудо ли?) на смену счётам явились электронные вычислительные машины (ЭВМ). Первые ЭВМ служили, главным образом, для громоздких вычислений по формулам. Не зря название одного из ранних языков программирования — ФОРТРАН — так и расшифровывается: ФОРмульный ТРАНслятор. Постепенно компьютеры и языки программирования стали приспособлять к переводу в электронную форму всё более сложных символических моделей. И теперь компьютеры понимают речь, различают образы, процессоры спрятаны в телефонах, микроволновых печах и другой технике — всего не перечислить. Куда двинулся прогресс? — в сторону перевода *СИМВОЛИЧЕСКИХ* моделей в *ЭЛЕКТРОННУЮ* форму. И программист оказался в самой гуще этой третьей стремительной кибер-революции. Какова ж его роль?

### 1.3. Модели, модели, модели...

Прежде, чем уяснить роль программиста, бегло ознакомимся со свойствами и назначением того, что мы называем *моделями*.

#### 1.3.1. Много ль в корыте корысти?

Любая модель (биологическая, символическая) нужна для *предсказаний*. Мозг хищника *предсказывает* поведение жертвы и строит на этом предсказании план охоты. А жертва пытается *предсказать* поведение хищника и планирует спасение. Мозг тренированного игрока *прогнозирует* полёт мяча. Такова роль биологических моделей.

Тому же служат и символические модели. Инженерные расчёты *предсказывают* поведение будущей машины или прочность постройки. Социально-экономические модели, пусть и не вполне точные, с некоторой вероятностью *предрекают* грядущее. Даже такая расплывчатая модель, как художественное произведение, даёт, наряду с эстетическим наслаждением, представление о сторонах жизни, с которыми мы не сталкиваемся повседневно. Входя в положение героев романа или фильма, мы невольно *прогнозируем* своё поведение в сходных обстоятельствах. Итак, назначение любой модели — *предсказание*.

#### 1.3.2. Неформализованные модели: искусство и философия

Поскольку разного рода искусства обладают некоторой *предсказательной* силой, мы причисляем их к моделям. Художественные произведения обычно фиксируют на бумаге или на других носителях посредством символов: буквами, нотами; стало быть, эти модели — символические. Но воздействуют они на нас не символами, а *образами* или *звуками*, порождаемыми символами, — именно они вызывают эмоции.

К искусствам примыкает философия, она тоже выражается символами (словами) и оперирует с образами. Однако если объектом искусств являются *конкретные* образы, взятые из природы (люди, пейзажи, звуки и т.д.), то в голове философа рождаются *абстрактные* образы — идеи, отражающие, по его мнению, общие законы природы.

Искусства и философию объединяет их *неформализованность*. Известно, что машину не увлечёшь ни фильмом, ни музыкой, её «мозг» не породит философских мыслей. Да и зачем ей это? Оставим приятное себе.

#### 1.3.3. Формализованные модели: описательные науки и математика

Обратимся теперь к точным наукам, к тому, что можно хотя бы отчасти *формализовать* и переложить на компьютерные плечи. Науки делят на описательные и теоретические, хотя чёткой грани тут нет. География, биология, химия — что это? Скорее описательные науки. Но чем больше в науке математики,

тем сильнее она смещается в сторону теоретическую, — такова физика, например. Крайней гранью является сама математика, объекты которой в природе не встречаются, хотя и навеяны ею. Подобно идеям философов, они являют плод размышлений математиков. Описательные науки соотносятся с теоретическими так же, как искусства соотносятся с философией: описательные исследуют конкретные объекты природы, а теоретические — абстрактные, воображаемые объекты (табл. 0-1).

**Табл. 0-1 — Классификация наук  
в зависимости от языка и изучаемых объектов**

Используемый язык	Характер объектов	
	Конкретные объекты	Абстрактные объекты
Неформализованный	<i>Искусство</i>	<i>Философия</i>
Формализованный	<i>Описательные науки</i>	<i>Теоретические науки</i>

#### 1.3.4. Царица наук

Царица наук — это математика, конечно. Не будучи прямо привязана к реальным объектам природы, она, тем не менее, проникает во все естественные науки, включая описательные. Все теории, претендующие на сколь-нибудь точные предсказания, не ограничиваются словами, а выражаются строгим языком математики.

#### 1.3.5. Универсальность и обратимость математических моделей

Рассмотрим следующую формулу:

$$A = B \cdot C$$

Угадайте, какую природную закономерность она выражает? Подсказки ищите в таблице, где та же формула слегка перелицована и снабжена пояснениями:

Формула	Обозначения
$S = H \cdot L$	S — площадь прямоугольника H — его ширина L — его длина
$L = V \cdot T$	L — преодоленный путь V — скорость T — время
$S = C \cdot N$	S — стоимость партии товара C — цена единицы товара N — количество товара

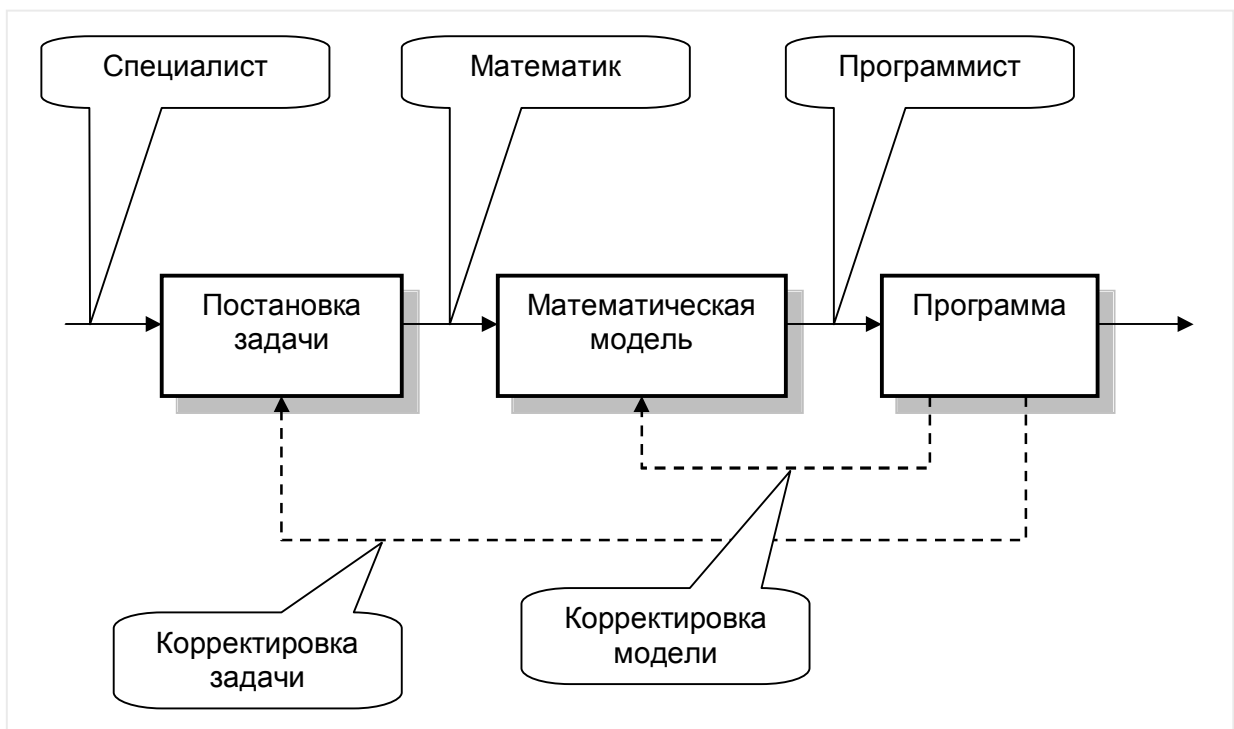
Список примеров легко продлить, но и так ясно: простейшая модель, выраженная формулой умножения, весьма востребована. Эта *универсальность* характерна почти для всей математики. Существуют, разумеется, и очень

специфические модели (например, математическая модель ядерного реактора), но и они строятся на базе тех же простых универсальных моделей.

Отметим, также, **обратимость** математических моделей. Так, из формулы, связывающей путь, скорость и время, можно через простейшие преобразования определить любой из параметров по двум остальным.

#### 1.4. На троих

Теперь мы близки к тому, чтобы указать место программиста в новой, третьей кибернетической революции. Рассмотрим упрощённую схему разработки какой-либо компьютеризированной системы. Ею может быть и сложная система управления самолётом, и сравнительно простая бухгалтерская программа.



Разработку любой такой системы начинают с *постановки задачи*, — за это отвечают специалисты в данной области. Они должны чётко и однозначно описать исходные данные и все требования к будущей системе. Затем эстафету принимают математики, которые ищут методы решения поставленных задач, подбирают готовые алгоритмы, или же строят новые, и доказывают их правильность. Одним словом, строят математическую *модель*. Программистам же остаётся перенести эту модель с бумаги в электронные «мозги» компьютера.

На первый взгляд всё просто и ясно: здесь каждый делает своё дело. Но жизнь богаче и сложнее схем. Хорошо, если постановщик задачи может создать и математическую модель. В других ситуациях роль математика берёт на себя программист. Когда же роли разделяются (как на схеме), на стыках возможны трения: специалист затрудняется сформулировать задачу, математик — не всегда понимает специалиста, а программист — математика (тут сказывается уровень его математической подготовки). Как бы то ни было, конечным продуктом является

система. Если она работает не так, как надо, следует «разбор полётов». В ходе разбирательства могут исправляться и *программа*, и *математическая модель*, и даже *постановка задачи*.

Итак, тесная связь программиста с математиком очевидна. Программисту надо, по меньшей мере, владеть языком математики и понимать выражаемые им идеи. Именно это понимание, а не владение модным языком программирования, будет фундаментом и будущих ваших успехов, и профессионального долголетия.

## 1.5. Математика и математики

Раз уж мы коснулись математики, припомним школу. Основу математики составляют три сущности: аксиомы, теоремы и доказательства. *Аксиома* — это базовое утверждение, принимаемое без доказательств. *Доказательство* — это цепочка рассуждений, подчинённых нескольким простым правилам логики. Наконец, *теорема* — это утверждение, истинность которого вытекает из аксиом и доказана цепочкой рассуждений. В этом вся суть математики.

Приступая к очередной задаче, математик, прежде всего, включает интуицию и угадывает конечное или промежуточное решение. Как угадывает? — загадка. Но заподозрив решение, он должен показать его истинность, доказав одну или несколько теорем. Пока решение не доказано, это лишь догадка, *гипотеза*, а не теорема. Порой гипотеза бывает интуитивно понятной, но трудно доказуемой, некоторые доказательства даются лишь потомкам спустя десятки, сотни лет. Но, обретя доказательство, математик «умывает руки» — его работа сделана.

А что же программист? Обязан ли он доказывать теоремы? Если он изобрёл что-то новое, то придётся. Но чаще программист берёт готовые плоды математических усилий, и вправе освободить себя от перепроверки доказательств, — в конце концов, каждый грызёт свою морковку. Программисту важнее сосредоточиться на том, суть чего выражается в следующих вопросах:

- Какую реальность описывает данная математическая модель?
- Что за идеи лежат в основе решения?
- Как эффективно организовать данные и скомбинировать процедуры, чтобы превратить эту математическую модель в работающую программу?

К счастью, в отличие от изучающих математику инженеров других специальностей, у программиста есть возможность превращать изучаемый материал в работающие программы. Создание программ, оживляющих сухую математику, — прекрасный способ и освоения математики, и обретения опыта программирования, — *это наш путь*.

## 1.6. Дискретная математика

Математика включает много ветвей, крупных и мелких ответвлений. То, что применяют в инженерном деле и других науках, называют *прикладной* математикой. В ней, в свою очередь, выделяется ветвь, вплотную примыкающая к

компьютерной науке — информатике. Речь здесь о *ДИСКРЕТНОЙ* математике, грань между которой и информатикой трудно различима. Но и дискретная математика весьма обширна, и потому в этой книге коснёмся лишь двух её разделов, на мой взгляд, самых интересных и полезных для программиста: *ТЕОРИИ МНОЖЕСТВ* и *ГРАФОВ*.

У математики особый язык, и чтение математических книг, рекомендованных мною в списке литературы, предполагает владение оным. Но здесь я не буду «грузить» читателя чистой математикой, ведь эта книга для программистов. Не ищите здесь милых математикам крючков и загогулинок, — обойдёмся без них. Всё, что надо, я объясню «на пальцах». Мы не будем доказывать теорем, зато напишем море программ. Так будут убиты два зайца, которых мне не жаль: вы освоите много интересных и важных алгоритмов, а заодно научитесь создавать весьма сложные программы. В путь!

Олег Деревенец, 2016 г., Воронеж