

Максим Сохацький

Топовий програміст

Поклоніння та вираження поваги

Перевірка мотивація

Коштовності та чесноти як хакарактеристики мислення

Яди мислення

Спочатку йога розуму, потім тіла

Баланс та не-медитація

Структура курсу

Гранична точність, оптимальність та складність

Самостійність

Створення власної школи

Дракон

Лямбдагарбха

Гротендік

Будда

Програмування як спорт

Баланс

Топовий програміст

Автор: Намдак Тонпа (1980)

Про Автора

Намдак Тонпа (Максим Сохацький) — доктор філософії КПІ (02070921), буддистський піп-капелан лінії передачі Лонгчен Нінгтік тибетського буддизму школи Нінгма (38778275), провідний інженер-програміст ДП «ІНФОТЕХ» (34239034) у підпорядкуванні МВС України (00032684). Автор систем «Депозити ПриватБанк» та «МІА: Документообіг» побудованих на авторських творах ERP.UNO та N2O.DEV.

Постійне посилання твору: <https://tonpa.guru/top/>
Видавець: Лонгчен Нінгтік Україна, ЄДРПОУ: 38778275
Юридична адреса: 04071, м. Київ, вул. Костянтинівська 20
Сайт релігійної організації: <https://longchenpa.guru/>
Лінія передачі: Школа Нінгма тибетського буддизму

ISBN: 978-617-8027-23-0

Збірка на замовлення моїх учнів.

© 2021 Лонгчен Нінгтік Україна

Зміст

1	Частина 1	4
1.1	Топовый программист	4
1.2	Выражение уважения	4
1.3	Проверка мотивации	5
1.4	Всеведение как источник вдохновения	7
2	Частина 2	10
2.1	Мышление	10
2.2	Характеристики чистого мышления	10
2.3	Драгоценные мышления	12
2.4	Яды	13
3	Частина 3	15
3.1	Аспекты курса	15
3.2	Предельная точность	15
3.3	Предельная оптимальность	16
3.4	Предельная сложность	16
4	Частина 4	19
4.1	Структура курса	19
4.2	Дракон	19
4.3	Лямблагарбха	19
4.4	Гротендік	20
4.5	Будда	20
5	Частина 6	21
5.1	Сначала йога ума, потом уже йога тела	21
5.2	Программирование и спорт	22
5.3	Баланс	24

1 Частина 1

1.1 Топовый программист

Если мои подписчики и просят о какой-то масштабной контрибуции, то это монографию на тему «как стать топовым программистом». Хотя такая формулировка инфантильна, она достаточно хорошо отображает суть вопрошаемого: детальное рассмотрение профессии программиста, стратегию изучения предмета исходя из личного опыта, разбавленное аутентичной философией.

Данный труд, который тяжело втиснуть в twitter формате, если и будет иметь продолжение, то тоже на страницах этого рабочего журнала в виде серии заметок, где же им еще быть!

1.2 Выражение уважения

Перед тем как начать повествование о профессии программиста прежде всего хочется выразить уважения предмету изучения и практики программирования, а именно формальным математическим вычислительным построениям, в которых возможно программирование в принципе. В последние годы стало понятно, что пространство этих построений настолько глубоко, что может поглотить не только все дискретные программы всех формальных грамматик, но и континуальную математику, в которой работа с пространством идет на другом, более фундаментальном уровне. Поэтому без лишнего преувеличения можно сказать, что само пространство рождает языковую группу языков, которые представляют собой первоначальную матрицу всех без исключения языков программирования.

Принцип глубокого уважения к предмету, который начинающий мастер должен реализовать является одним из секретных ключей восточной философии. Раз программирование рождается из пространства феноменологических построений ведущих к абстрактной классификации пространств и логик с ними связанных, то удержание в фокусе цели изучения пространства и программирования как практического человеческого про-

цесса с этим связанного является главной задачей на пути изучения. Поэтому, без недооценивания и лишнего преувеличения можно сказать, что простираение или поклонение, как проявление уважение к самому пространству, как объекту изучения, выглядит для меня логичным. Я простираюсь перед пространством.

1.3 Проверка мотивации

Важной характеристикой, которая хотите верьте, а хотите нет, влияет на процесс изучения искусства программирования является чистота мотивации. Если рассмотреть граничный пример, то он будет выглядеть так: вашей мотивацией является увеличение своих навыков программиста для достижение материальных благ и увеличение конкурентоспособности на рынке труда. Полная чушь, такая мотивация влияет на критерии выбора объектов изучения и это может завести вас в ситуацию, когда вам 50 лет и вы пишете на Core Java для какого-то швейцарского банка. Явно люди, которые просили этот текст, не ожидают у меня чего-то подобного.

Свои иллюзии насчет легкости этого пути можете сразу отбросить. Этот путь по-самурайски сложен и на нем сходили с ума не только выпускники прикладной математики, немало людей перегорело на предприятиях от переизбытка и неконтролируемости информации. Поэтому 10 лет затвора с постоянным выделенным каналом в интернет на полном внешнем обеспечении —идеальный ресурс, который я рекомендовал бы выделить для успешной подготовки на мастера программирования. Откуда такой дикий расход с принятой методикой обучения придется раскрыть в следующих условных выпусках, так как должна оставаться интрига. Вообще 10 лет совершенно нормально временной интервал для обучения профессии врача, почему программист должен обучаться в более сжатые сроки? Ведь количество языков, которыми оперирует топовый программист может доходить до тысяч, и это в практике. Это не просто латынь, эсперанто и романо-германская группа. Половину из этого времени можно проводить в реальных проектах,

типа интернатуре, но языков и материала так много, что для топового программиста 10 лет можно выделить только на теорию.

Да, можно и в 50 лет устроиться на галеру «цифровым сантехником», но это тоже никак не попадает под курс топового программиста, который должен покрывать от создания процессоров, ассемблеров, компиляторов, операционных систем, баз данных, сетевых протоколов, сервисов, шин и приложений до теоретико-типовых верификаторов математических моделей и теорем, сертифицированных компиляторов, систем доказательства теорем. Программирование, как и математика — удел молодых!

Без правильной мотивации предъявлять претензии о зря потраченных 10 годах безрезультатно на курс Сохацкого категорически запрещается! Как проверять чистоту мотивации и насколько точны могут быть рекомендации? Могу лишь сказать, что вы должны быть предельно честны с самим собой, ведь программирование — это сложный изнурительный ментальный процесс, а мышление — это высшая форма управления организмом, поэтому сбои в его работе могут привести к фатальным последствиям. Если вы кроме программирования ничего не умеете, то неплохо было бы разработать стратегию отхода: минимальная техника управления дыханием, легкий спорт без фанатизма, немного йоги, возможно активные виды спорта. Если вы считаете, что в целом вы психически устойчивый человек, то приготовьтесь к сюрпризам на пути к постижению загадок пространства без внутренней чистоты намерений.

Моей личной мантрой, с которой я изучаю программирование — это посвящение результатов своей работы людям и всем существам, не навредив никому без исключения. Вообще изучения программирования мало кому может помешать и может иметь форму глубокого затворничества святого монаха. Хотя есть и исключения, программисты в основном не жестокие существа и именно эта излишняя энергия агрессии, развернутая в позитивном направлении интроспекции, является двигателем аутического постижения тайн профессии программиста! Вообще если мотивация алмазной колесницы привести всех

существ к полному просветлению вам кажется слишком эзотерической, то хотя бы старое доброе правило инженеров прошлого «не навреди, а лучше помоги людям», является тем минимумом, который необходимо проверять перед каждой сессией программирования. Представьте себе, что вы с рвением льва кладете на алтарь просветления 10 лет обучения программированию из сундука своей жизни для того, чтобы принести пользу людям. Без подобной мотивации вам попросту не будет откуда брать энергию для ежедневных упражнений в программировании и мышлении.

1.4 Всеведение как источник вдохновения

Главная черта характера, которая необходима в человеке, чтобы стать топовым программистом –это предрасположенность к изучению и исследованию феноменов, их анализу, синтезу и абстракции. Это желание разобрать и исследовать игрушку должно так глубоко находиться в сознании, что кажется, будто ребенок уже рождается с этим даром быстро разбираться в феноменах при должной интенсивной нагрузке на нейросеточку. Другими словами –это хакерство, если вы любите исследовать системы, разбираться в программном коде, понимаете, как работают процессоры, знаете, как работает логика и математика, то вы уже можете стать топовым программистом. Желание построить максимально точную модель феномена должно быть гипертрофированным, оно должно быть незакрытым гештальтом, который не дает вам спать по ночам пока вы его не закроете. Именно эта фанатичная одержимость конвертируется в то, что будет дровами в нашем костре просветления на пути к всеведению в мире программирования. Откуда взялось всеведение? Это вторая сторона медали главного источника хакерского вдохновения. Если при локальном рассмотрении феноменов главной мыслью должно быть построить максимально точную модель феномена, то при фокусировке в бесконечность к краям горизонта, это желание проявляется в виде максимально быстрого познания вообще всех феноменов и их универсальные принципы устройства. Такой мета-хакерский трансценден-

тальный полу-фрический майндсет необходим для понимания того, насколько абстрактными и широкими могут быть вызовы на пути познания глубинных языков, на которых написана наша вселенная.

Так, как языки программирования используются во всех сферах человеческой деятельности, то топовый программист совершенно точно должен разбираться во всех доменных моделях, всех типах и всех математиках, которые возникают в разных языках программирования. Обычно, университетские 5 лет я бы рекомендовал провести как раз в охвате всех математик и всех видов языков программирования, перед тем как погрузиться в фундаментальную математику и системное программирование. Вообще хорошее современное образование уровня PhD автоматически подразумевает свободное владение языковым и математическим обеспечением в исследовании вселенной, так что ничего такого, что не требуют топовые университеты, курс топового программиста в этой части всеведения здесь не требует. Требуется полная автономность на уровне полетел в космос и восстановил все знания и навыки при необходимости в кратчайшие сроки путем легкого воспоминания.

Каждый раз поглощая какой-то пласт информации вы высвобождаете огромное пространство свободы, которое либо заполняется новыми неисследованными пластами, либо освобождается абсолютно, если вы уже все пласты поглотили. Но когда вы полностью исчерпаете всю карму, тогда буддахуд придет автоматически, так что это уже программа максимум. Ведь после того как вы изучили какой-то предмет и дали несколько мастер классов по нему, вы просто листаете все книги по нему, по которым учились и это все для вас даже не букварь, потому что букварь вы уже сами написали, это для вас просто шум деревьев в лесу. Вы полностью исчерпали этот предмет, стали мастером в нем, вы видите уже все сечения глобулярных фазовых пространств, имеете на руках несколько моделей и прототипов. Это состояние всеведения. Желание этого состояния –необходимый компонент топового программиста. Если вы построили какое-то пространство феноменов напол-

нив их смыслом и остаетесь там в комфортной среде ограниченного знания, вы уже теряете топовую мотивацию как компонент всеведения. Не устремив свое мышление в бесконечность вы не сможете увидеть весь ландшафт и правильно расставить приоритеты в поглощении научных дисциплин, чтобы совершить «прыжки Тигра» между этими приоритетными реперными точками.

Только исследователи, которые преисполнены врождённого желания строить новые теории и языковые пространства наделены семенем творчества, которое ведет к топовой реализации. Междисциплинарный подход может появиться только в условиях широкого профиля. Никто никогда не ставил задачу составить курс для подготовки людей, которых можно было бы назвать культовыми хакерами, поэтому и требования к подготовке должны быть запредельными. Только олимпийское желание всеведения может реально приблизить вас к нему.

2 Частина 2

2.1 Мышление

Перед тем как начинать процесс обучения неплохо было бы пару слов сказать об основном инструменте в процессе изучения –человеческом мышлении. Минуя физические стороны мышления сразу хочется поговорить о его когнитивных свойствах.

2.2 Характеристики чистого мышления

Первое и главное свойство мышления –это существенность –определяющая характеристика существа. Интегральная высшая форма, которая управляет всеми подсистемами и воспринимается существом, майндстримом или аватаром. В одном теле может жить несколько майндстримов, и некоторые из них вполне могут быть программистами! Если вы мыслите –вы существо.

Вторая когнитивная характеристика мышления, которую можно почувствовать в медитациях –это абсолютная сферическая открытость во всех направлениях и его безграничность. Такая характеристика мышления навевает мысли об изоморфизме мышления и пространства. С физической точки зрения, мышление –это сложная система квантовых полей, которые наслаиваются на квантовый, молекулярный уровень, нервную систему, поэтому долго доказывать не нужно, что мышление как квантово-механическая система, распространяется на всё пространство.

Условно существует два раздела высшей медитации, первый из которых называется разделом мышления, а второй разделом пространства. Первый раздел посвящен техникам работы с феноменами, аналитической медитации, работе с мышлением с точки зрения майндстрима, глазным упражнениям, развитию ощущения перспективы, работе с воображением, визуализацией. Второй раздел посвящен техникам работы с мышлением с точки зрения пространства, где мышление ассоциируется с пространством, в котором оно пребывает, неаналитической

медитации устремлении в бесконечность, медитации отдохновения.

Третья когнитивная характеристика мышления, которую можно воспринять на опыте –это его необусловленность. Чем выше уровень развития мышления, тем выше его воля к свободе и необусловленность, к перепроверке, критическому мышлению и переоцениванию. В своей полной свободе мышление свободно выбирать направленность и интенсивность потока, без резких перепадов и гормональных фонов, двигаясь по оптимальной траектории взрослении плода мышления на пути к всеведению.

Четвертая характеристика мышления –это непрерывность. Любые попытки остановить мышление приводят в место самосознания как несущей частоте ощущения присутствия себя в этом мире, в медитации. Даже в процессе сна, мышление не спит, а переходит в другое агрегатное состояние, более разреженное, порой бесформенное, нечеткое, мерцающее. Полный контроль над непрерывностью мышления, от которой нельзя отказать, и которую нельзя прекратить –задача топового программиста на пути к освобождению ресурсов для изучения программирования. Чем больше точность дискретизации у этого контроля –тем лучше. Контроль над непрерывностью мышления называется точностью мышления.

Пятая характеристика мышления –взаимозависимость. Вы как мышление –это продукт абсорбции других фрагментов мышлений или просто феноменов, поэтому обусловлены этим наследием. Вырваться за пределы этой традиции и раскопать инсайты на пути эволюции своего мышления –истинная драгоценность как награда за труд обучения. Когда вы становитесь мастером, обусловленность пропадает, вы реструктурируете себя заново исходя уже из личного опыта, построенного на черед инсайтов, по которым вы прыгаете на пути к мастерству. И даже их вы потом сможете удалить и забыть из своего мышления оставив только память о том, как нужно сразу делать правильно, возможно и не вспомните даже, когда вас спросят, как это вы так быстро помудрили, а зачем.

Эти пять характеристик послужат вам подсказками в каком

ключе нужно мыслить о своем мышлении (первая производная) как инструменте познания, возможно для существ с высокими способностями это сразу прояснит некоторые моменты. Любая неспособность наблюдать эти характеристики в практических медитациях или размышлениях о своем мышлении, говорит о том, что их нужно развивать, либо заняться йогой, пойти к психологу, развеяться с друзьями, пойти в бар, сесть на таблетки, стакан, всё по желанию –главное, чтобы сработало! Чек лист прошли переходим к рекомендациям и индикаторам.

2.3 Драгоценные мышления

В тибетской традиции существует шесть типов мышления или программ, которые считаются, что позитивно могут повлиять в целом на процесс изучения, размышления и медитации.

Щедрость в контексте мышления означает не жадничать в процессе изучения, не хвататься за все сразу, иметь методологию, с уважением относится к любой выбранной теме, раз она уже всплыла в медитации как гештальт, который все равно придется закрыть. Способность к репликации, преподаванию, возвратной контрибуции на пути поглощения информации –это щедрость мышления.

Дисциплина означает, что мышление должно придерживаться какого-то спортивного, желательно олимпийского режима, слишком хаотичные режимы мышления не будут способствовать обучению, поэтому приступать к эволюции своего мышления нужно, когда гормональный фон может оставаться ровным значительное время, это необходимо для глубоких медитаций, без которых невозможен прогресс.

Терпение –это способность переносить трудности в процессе обучения. Есть материал, который может не закрываться годами, но к нему, все равно придется возвращаться, ведь назад дороги нет, выбран путь топового программиста. На пути может быть слишком много инсайтов и слишком много воодушевления, которое может создавать гормональный фон, который не всегда можно контролировать, пересидживать на бенче такие периоды –это терпение.

Усердие –означает с неподдельным интересом изучать предметы, поэтому правильно их расположить очень важно. Возможно, именно для вас существует своя последовательность предметов, каждый из которых в отдельным момент времени вы будете изучать с максимальным усердием. С этим придется работать, каждому индивидуально.

Фокусировка –фокусировка, или концентрация, или медитация, или шаматха –это основной режим работы программиста. Вот вы сели за комп, поставили чашку с кофе, протерли дисплей, всосались в пиксели, запустили шелл –вы сфокусированы на работе, это медитация.

Мудрость –это система накопленных инсайтов, которая формирует новые структуры мышления, новую его топологию. Эта система может переписывать старые неэффективные и невалидные структуры, над которыми мы смеемся повзрослев. Мышление мудрости –это мышление основанное исключительно на таких проверенных рафинированных структурах, которые положены в фундамент нашего существа.

2.4 Яды

Самые три неблагоприятные формы мышления по моей личной классификации.

Инертность мышления –это колесо медитации. Будучи раз пущена некоторая привычка, уходит в автоматический режим на подсознание –это инертность. Если бы не было инертности мышления мы бы не смогли учиться. Хотя это полезное свойство мышления, иногда бывает плохо, когда плохо –нужно отлавливать. Понятно, что дебажить свое мышление, которое спрятано в подсознании не учат в школах, придется работать самому.

Лень. Слишком интенсивное мышление может перерасти в затяжную рекреационную прокрастинацию, которая сменится ленью. Наблюдение за видимым прогрессом необходимо, каким бы не был охуенный отдых нужно возвращаться за программирование, обновлять мотивацию, если нужно каждый день.

Безразличие. Корень всех ядов, жадности и прочего. Если вам все вдруг стало безразлично – это очень плохо, но не смертельно. Иногда может перерасти в экзистенциальный кризис, но мы же с вами уже договорились, что тело, йогу и таблетки и свое самочувствие вы берете на себя, с меня только рекомендации по процессу обучения. Запущенное безразличие – это тупость. Ничего выходящего за перечисление доблестей хорошего успешного студента здесь нет. Как и в первом случае постоянно применяем технику размышления, изучения и медитации к этим видам мышления, так же как к основным характеристикам мышления. Постоянно проходим валидацию своего мышления в соответствии с выбранными индикаторами.

3 Частина 3

3.1 Аспекты курса

После того, как в первых двух главах мы рассказали о топовой мотивации и топовом мышлении, перейдем к топовому курсу топового программиста. Структура этого курса программирования возможно будет очень сильно отличаться от других подходов к обучению, но этот курс является курсом, который я пытался “увидеть” сквозь ретроспективу своего профессионального опыта, касающегося непосредственно программирования, того, как я вижу топовость этого процесса.

Перед тем как начать со структуры курса, я хочу показать основные индикаторы (аспекты курса), на которые стоит ориентироваться при выборе топовых направлений в программировании. Потому, что именно эти индикаторы и определяют те предметы и их последовательность в которой вы должны поглощать информацию, чтобы приблизиться к максимальной топовости.

3.2 Предельная точность

Предельная точность означает абсолютные вычисления. В эту вымышленную категорию я включаю все области программирования и математического моделирования, которые требуют абсолютной формальной точности: системы доказательства теорем, специализированные формальные верификаторы моделей, системы символьной алгебры, системы вычислительной гомологической алгебры, т.е. те системы моделирования, которые не только раскладывать саму систему до атомов, но еще и близящую метамодель на уровень выше, которую тоже нужно формализовать, чтобы верифицировать сами модели. Сюда входят такие отрасли как процессоростроение (модели чекеры печатных плат, процессоров, специализированные языки программирования типа VHDL), языкостроение (системы типов, формальные языки программирования, языки общего назначения). Сюда также входят: любые формальные математические теории, формальные логики, язык кванторов Пи и Сиг-

ма.

3.3 Предельная оптимальность

Предельная оптимальность означает минимальное количество усилий приложенное к достижению цели. Сюда входят дискретные задачи мини-макса, линейное программирование, симплекс методы, полиэдральное многомерное симплектическое программирование, методы оптимизаций. В физике это основной принцип вариационного исчисления, минимальные геодезические линии. Ваша система не только должна быть максимально точной, в апогее абсолютно точной, но и должна быть закодирована оптимальным образом, не содержать повторяющихся частей, занимать минимальный футпринт по ограниченному количеству памяти и вычислительных мощностей. Лучше, чтобы была теорема которая доказывает эту минимальность, как например черч-кодировка индуктивных типов как естественная кодировка любых структур в лямбда исчислении, оптимальные лямбда эвалуаторы и т.д. Предельная оптимальность означает также высший пилотаж в прототипировании и создании MVP эскизов.

3.4 Предельная сложность

Предельная сложность означает дословно максимальную сложность системы с которой нужно работать. Если вдруг окажется, что вам сложности декартово-замкнутых категорий мало, всегда можно перейти в симметрические модальные категории в которых живут следующие языки программирования: квантовые языки программирования, конкурентные параллельные системы типа Erlang, системы линейных типов, языки для обработки тензоров. Также предельная сложность подразумевает наличие уже в системе предельной точности и предельной оптимальности, иначе без первого у вас будет просто неработающее говно, в отсутствии второго у вас будет миллионы строк дублирующего кода, которые не влияют на реальную слож-

ность проекта. Если вы хотите увидеть реальную сложность посмотрите кубические доказательства из K-теории.

Это что касается более менее чистых тем из логики, дискретной математики, и программирования. В междисциплинарном подходе, если вы хотите стать не топовым программистом, а, например, топовым биофизиком, то список тем будет другой. Предельная точность там заменяется на математическую статистику и стохастическую физику, предельная оптимальность связана с топологией пространств, а предельная сложность выражается в объединении полей и, как пример, стандартной модели. Ну а в биофизике сложность увеличивается топологическими многообразиями нейросетей как иерархических процессов работающих со тензорными потоками. Гомотопическая теория типов как язык программирования инфинити топоса является в физике аналогом теории струн. Биофизика же представляет собой более диссипативные структуры, которые теряют границы точности, таким образом задача создания AI не в полной мере соответствует абсолютной точности. Я бы сказал что AI это больше системная инженерия и прикладная область, в то время как абсолютная точность это больше сфера математического программирования. И даже машинное обучение при детальном рассмотрении сводится к статистической точности, прогнозируемой оптимальности и системной сложности.

Я, как сторонник абсолютной сложности, считаю, что любая современная PhD должна содержать междисциплинарный подход. В качестве необходимого минимума в международной научной практике предлагаю рассмотреть междисциплинарный подход, который базируется всего на двух особенных дисциплинах: УДК 51 (математика) и УДК 004 (программирование). В качестве особенных они выбраны потому, что любая другая дисциплина базируется на чистой математике, а для проведения любого моделирования (перепроверке теории), нужен фундаментальный курс программирования.

Интересно, что третий из этих индикаторов можно инвертировать и при этом оптимизационный вектор (максимальная точность, максимальная оптимальность и минимальная сложность) тоже окажется интересным объектом рассмотрения. В

существом это требования выдвигаемые к библиотекам программного обеспечения: они должны быть максимально формальными с доказательствами свойств их моделей, должны быть оптимально смоделированы или обладающие оптимальным экстрактом в другие модели, но при этом также должны быть максимально простыми или минимально сложными в общей картине компонент. Философия N2O в точности соответствует такому альтернативному оптимизационному вектору, но решая задачи технологических стеков нельзя стать топовым программистом, нужно увеличивать сложность, например в сторону сложных социо-информационных систем, как ERP системы управления предприятием, где десятки тысяч таблиц каждая по сотне полей привычная сложность для бизнес аналитиков, работающих с такими продуктами как SAP S/4.

Забавно, что в эзотерическом тибетском буддизме, эти три критерия применяются в наставлениях по визуализации в высших тантрах: вы должны максимально детально (до волос и фактуры тканей одежды —принцип максимальной точности) представить очень сложный объект (18-рукое божество в союзе с супругой, свитой и атрибутами —принцип максимальной сложности) максимально быстро (желательно мгновенно —принцип оптимальности). Ученики, которые могут это делать не "на словах а "на деле"—встраивая картинку силой мысли и воображения даже не на сечатку глаза, а прямо на кору головного мозга, выжигая изображение в своем мышлении на простыне воображаемой вселенной —считаются топовыми медитаторами. Для наглядного примера хочу показать приблизительное качество визуализации удерживаемое в пространстве своего мышления, которого может добиться средний по способностям европеоид-медитатор за 3 года упорной практики:

Наверно поэтому до сих пор не существует ни одного просветленного европейца! Эта картинка также наглядная демонстрации того, что вы можете достичь за 3 года обучения (думаю, что у отдела PhD ребят из NASA, подготовка к рендеру этой картинки заняла три года как раз).

4 Частина 4

4.1 Структура курса

Не то, чтобы это была какая-то новость, уверен многие придерживаются такой карты топового программиста, но я возьму на себе смелость открыть это тайное знание. Начну описание курса с известной мемной картинки:

4.2 Дракон

Единорогами называют тех программистов, которые одинаково хорошо владеют CSS скажем, а также могут полностью построить любой сложности тонкий или толстый клиент не ограничиваясь HTML5, но и переходя в SVG или WPF, или DirectX или OpenGL.

Фулстек программистами называют специалистов по построению информационных систем на границе с единорогами (которые обычно не занимаются процессингом, инфраструктурой, сетями и защитой).

4.3 Лямблагарбха

Следующий уровень — это платфо́рмообразующий уровень, который включает язык программирования, рантайм и аппаратуру. Обычно взрослые академические языки создаются сразу с рантаймом, поэтому назовем эту секцию уровень университета профессора, а секцию рантайма (ОС) и аппаратуры назовем предпринимательской, так как ОС обычно продают вместе с железом и все кто это пытался продвигать на рынок можно приравнять к бодхисаттвам. Последние известные лямбдагарбхи — это древние автора первых Лисп машин и XEROX PARC.

4.4 Гротендік

На абсолютном уровне программисты (в том числе и топовые) являются математиками, поэтому тут можно отметить ядро которое было открыто Квилленом — модельные категории, в которых работали не только медалисты Филдса — Воеводский и сам Квиллен, но которые являются также основным инструментом современных теоретико-типовых математиков как Шульман. Предмет изучающий модельные категории Квиллен назвал гомотопической алгеброй, при помощи которой была построена не только модель алгебраической топологии самим Квилленом, но и A1-теория гомотопий Воеводского. Все это крышует Гротендиком, как мультидисциплинарным программистом абсолютного уровня (топ-математиком).

4.5 Будда

Без лишней скромности, любой программист который смог не только представить, но и успеть поработать за жизнь на всех уровнях, может считать себя Буддой программирования, или как мы скромно называем таких пацанов — хуй с горы.

5 Частина 6

5.1 Сначала йога ума, потом уже йога тела

Рано или поздно, все кто занимается йогой ума при должном успехе, так или иначе переходят к освоению йоги тела, которое является продолжением ума. Секрет успешной практики в том, что как и йога ума, йога тела требует еще более острой осознанности. Под йогой тела мы будем понимать здесь такого рода энергии, которые вы можете пережить только в режиме экстремального спорта, там где ревард очень высокий. В спорте это Х-спорт, в сексе это BDSM, подходит все дисциплины, где есть стоп-слово, за границей которого сразу наступает терминация существа.

Почему йога тела должна идти обязательно после укрпления в практике йоги ума? На это есть несколько причин. Считается, что основа устойчивого и взрослого мышления это правильное мировоззрение, которое должно формироваться существом в период изучения философских дисциплин, нерешенных вопросов трансгуманизма и других базовых принципов. Придерживаясь внесектарного стиля, самые базовые принципы мировоззрения топового программисты были зацементированы в первом выпуске.

Положив неправильный майндсет в основание Изучения, Размышления и Медитации вы создадите брешь, через которую в критический момент вашей жизни при встрече с Буддой ваше видения мира разрушиться как детский замок из песка после прилива. Для тренировки ума алмазной крепкости и остроты и предназначены практики формирования правильно взгляда на объект исследования свой собственный ум, за пределами которого не существуют никакие феномены.

Как и йога ума, йога тела предусматривает два режима исследования: пандита стайл (изучение теории) и йога стайл (практика). Усвоившись и укрепившись в своем сознании, существо

движущиеся главным принципом будда-таковости, стремлением к всезнанию и полной реализации, зерном которого является познание феноменов, начинает выходить за рамки ментальных феноменов и начинает осознавать себя и свое тело как часть мышления, и естественным образом начинает экспериментировать с телом, расширяя свой фронт восприятия.

Главный критерий который показывает можно ли вам переходить к спорту это полный контроль над дофаминовой и эпипинфриновой системой. Некоторые слишком впечатлительные спортсмены используют ТНС для суппрессии дофамина и более мягкой йога-сессии. Обычно рекомендуется входить в спорт в 40 лет, потому что в 25 и несколько лет после нужно посвятить математике и философии, ведь лучшего времени уже не будет и вернуть его будет непросто! А в 40 уже дофаминовый фон сам по себе исчезнет и останется только чистый ум и террейн. Это вторая безжалостная причина по которой спорт лучше отложить до adult возраста. Есть и другая сторона медали: раны после 40 заживают хуже, поэтому и ставки и острота и ревард в таком случае выше. Идеально это имея острый ум не совершать вообще серьезных ошибок на пути спорта. Неидеальные случаи решаются имплантацией титановых пластин.

Косвенный критерий это когда вы достигли уровня непосредственного переживания отождествления языка пространства (выпуск X), карты местности (выпуск 4) и своего мышления (выпуск 2), в таком случае выход из локальной самсары в виде темницы ума будет знаменовать выход в реальный мир на планету Земля. Alegoria которая мне видится здесь такая: мать отправляют своего сына в университет передав ему все необходимые знания, которые помогут ему жить дальше автономно.

5.2 Программирование и спорт

Гуру в спорте найти так же тяжело как и гуру в программировании. В светской жизни гуру спорта работают олимпийскими чемпионами или чемпионами в дисциплинах которые к

этому приравниваются. Само их существование уже является учением. Разбирая до мельчайших деталей покадрово трейсы топ-спортсменов на youtube вы получаете алмазные знания виртуоза-нюскулера методично оттачивая технику имея образец для верификации. Гуру спорта поменьшего калибра будут работать инструкторами на ближайшем спортивном курорте, рекламировать газировку и эквипмент или предлагать вам туры по национальным заповедникам.

Не все это говорят прямо, но в спорте важными являются картинки которые проходят через вашу сеточку и все органы чувств, поэтому ценятся картинки естественного ландшафта Земли, чтобы освободить свое мышление уже за границами тела, охватив своим мышлением всю планету и ее феномены, главный из которых гравитация, таким образом став воистину космическим ребёнком планетарного масштаба.

Однако в разговорах с локальными спортивными гуру вы получите максимум репрессивные монотонные лекции о вреде курения на Джомолунгме, ужасные команды инструкторов-обывателей, толпы туристов на своём пути. Взрослый человек овладевший йогой ума сам должен стать себе гуру и планировать каждую вылазку на встречу с гравитацией как проект с многими параметрами-переменными, от проработки которого зависит ваша жизнь.

Все эти гуру будут говорить вам, что только они понимают суть вещей, познали природу в не-мышлении, а у вас большой груз йоги-ума, который мешает вам достигать результатов, заморочки, излишняя концептуализация, чрезмерная начитанность, и другие смертные грехи. Тут действует такое же правило как и в детском саде, школе или университете "умных не любят поэтому имейте это ввиду собравшись поговорить по душам за костром с очередным спортивным гуру.

Разумный и рациональный человек всегда выберет более редкую и филигранную йогу ума вместо йоги тела, которой владе-

ет гораздо большее количество существ. Ведь получить алмаз ума гораздо сложнее чем алмаз тела, поэтому партнеры по спорту и даже локальные гуру, будут готовы в прямом смысле подсознательно вас убить на склоне, тут тоже нужен глаз да глаз.

В тибетском буддизме аналогом спорта является тайное посвящение каналов, ветров и сфер структуры вашей алмазной сети. Все эти йоги выполняются в паре с опорой на партнера и добраться до этого уровня будет затруднительно обывателю. Тем более, что таких гуру еще меньше чем спортивных.

5.3 Баланс

Главный наркотик для йоги-ума и йоги-тела это сахар. Все спортсмены как минимум висят на кока-коле, ред-булле, гелях, стимуляторах. Баланс этих веществ и правильное питание ключ к быстрому восстановлению после спортивных сессий, которые так или иначе нужно будет залечивать перед полноценными сессиями йоги ума. Игнорировать вспомогательные вещества на пути спорта глубоко, но и злоупотреблять не стоит. Главный критерий, непрерывность, вы должны планировать ваше путешествие таким образом, чтобы ваше сознание не пошатнулось от внезапного изменения гормонального фона, дефицита того или иного топливного элемента.