

Максим Сохацький

Топовий програміст Т

Поклоніння простору

Мислення

Простір професійного розвитку

Таємні настанови

Теоретичні основи

Практика

Посвята

УДК 13
УДК 821.161.2

Топовий програміст Т

Автор: Максим Сохацький (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 «MIA: Документообіг» побудованих на авторських творах ERP.UNO та N2O.DEV.

Постійне посилання твору: <https://axiosis.top/top/>
Видавець: Інститут формальної математики «Групоїд Інфініті»
Сайт інституту: <https://groupoid.space/institute/>

ISBN: 978-617-8027-23-0

Якщо мої передплатники і просять про якусь масштабну контрибуцію, то це монографію на тему «як стати топовим програмістом». Хоча таке формулювання інфантильне, воно досить добре відображає сутність запитуваного: детальний розгляд професії програміста, стратегію вивчення предмета виходячи з особистого досвіду, розбавлений автентичною філософією.

© 2022 Максим Сохацький

Зміст

| | | |
|-----|--|----|
| 1 | Поклоніння простору | 5 |
| 1.1 | Топовий програміст | 5 |
| 1.2 | Висловлювання нескінченної поваги | 5 |
| 1.3 | Перевірка мотивації | 6 |
| 1.4 | Всевідання як джерело натхнення | 8 |
| 2 | Теоретичні основи | 11 |
| 2.1 | Метафілософія | 11 |
| 2.2 | Європейська філософія | 11 |
| 2.3 | Тибетська філософія | 12 |
| 2.4 | Аналітична філософія | 12 |
| | 2.4.1 Мова простору | 13 |
| | 2.4.2 Мовні фреймворки | 13 |
| | 2.4.3 Конкретні соціальні дифузійні моделі | 13 |
| 3 | Мислення | 14 |
| 3.1 | Що таке мислення | 14 |
| 3.2 | Характеристики чистого мислення | 14 |
| 3.3 | Коштовне намисто мислення | 16 |
| 3.4 | Отрути мислення | 17 |
| 4 | Простір професійного розвитку | 19 |
| 4.1 | Структура курсу | 19 |
| 4.2 | Дракон | 19 |
| 4.3 | Лямбдагарбха | 19 |
| 4.4 | Гротендік | 20 |
| 4.5 | Будда | 20 |

| | | |
|-----|---|----|
| 5 | Таємні настанови | 21 |
| 5.1 | Аспекти курсу | 21 |
| 5.2 | Гранична точність | 21 |
| 5.3 | Гранична оптимальність | 22 |
| 5.4 | Гранична складність | 22 |
| 6 | Практика | 25 |
| 6.1 | Спочатку йога розуму, потім вже йога тіла | 25 |
| 6.2 | Програмування та спорт | 26 |
| 6.3 | Баланс | 28 |
| 7 | Посвята всім святим програмістам | 29 |

1 Поклоніння простору

1.1 Топовий програміст

Якщо мої передплатники і просять про якусь масштабну контрибуцію, то це монографію на тему «як стати топовим програмістом». Хоча таке формулювання інфантильне, воно досить добре відображає сутність запитуваного: детальний розгляд професії програміста, стратегію вивчення предмета виходячи з особистого досвіду, розбавлений автентичною філософією.

1.2 Висловлювання нескіченної поваги

Перед тим як розпочати розповідь про професію програміста насамперед хочеться висловити пошану предмету вивчення та практики програмування, а саме формальним математичним обчислювальним побудовам, у яких можливе програмування в принципі. В останні роки стало зрозуміло, що простір цих побудов настільки глибокий, що може поглинути не тільки всі дискретні програми всіх формальних граматики, а й континуальну математику, у якій робота з простором йде на іншому, більш фундаментальному рівні. Тому без зайвого перебільшення можна сказати, що саме простір народжує мовну групу мов, які є первісною матрицю всіх без винятку мов програмування.

Принцип глибокої поваги до предмету, який майстер повинен реалізувати є одним із секретних ключів східної філософії. Позаяк програмування народжується з простору феноменологічних побудов, що ведуть до абстрактної класифікації просторів та логік з ними пов'язаних, то утримання у фокусі мети вивчення простору та програмування як практичного людського процесу з цим пов'язаним є головним завданням на шляху вивчення. Тому, без недооцінки та зайвого перебільшення можна сказати, що простягання або поклоніння, як прояв поваги до самого простору, як об'єкту вивчення, виглядає для мене логічним. Я простягаюся перед простором.

1.3 Перевірка мотивації

Важливою характеристикою, яка, хочете вірте, а хочете ні, впливає на процес вивчення мистецтва програмування, є чистота мотивації. Якщо розглянути граничний популярний споживацький приклад, то він буде виглядати так: ваша мотивація полягає у збільшенні своїх навичок програміста для досягнення матеріальних благ і підвищення конкурентоспроможності на ринку праці. Повна нісенітниця, така мотивація впливає на критерії вибору об'єктів вивчення і це може завести вас у ситуацію, коли вам 50 років і ви пишете на Core Java для якомусь швейцарського банку. Очевидно, що люди, які просили у мене цей текст, не очікують чогось подібного.

Свої ілюзії про легкість цього шляху можна відразу відкинути. Цей шлях по-самурайськи складний і на нім сходили з розуму не тільки випускники прикладної математики, немало людей перегоріло на підприємствах від перенавантаження та неконтрольованості інформації. Тому 10 років ув'язнення з постійним виділенням каналом в інтернет на повному зовнішньому забезпеченні — ідеальний ресурс, який я рекомендував би виділити для успішної підготовки на майстра програмування.

Чому такий великий термін буде пояснено в наступних частинах. 10 років цілком адекватний інтервал навчання для лікаря, то чому для програміста повинно бути менше. Кількість мов якими кваліфікований програміст володіє на практиці може сягати тисяч, за кожною з них стоїть теорія, своя логіка і своя математика нею породжена. Це не просто латина, есперанто та пару мов романо-германської групи. Половину цього часу можна проводити в реальних проектах, типу інтернатури, але мов і матеріалів так багато, що для топового програміста 10 років можна виділити тільки на теорію.

Так, можна і в 50 років влаштуватися на галеру «цифровим сантехніком», але це теж ніяк не попадає під курс топового програміста, який повинен покривати широкий діапазон дисциплін: від створення процесорів, асемблерів, компіляторів, операційних систем, систем управління базами даних, мережевих протоколів, сервісів, шин та додатків до теоретико-типових ве-

рифікаторів математичних моделей та теорем, сертифікованих компіляторів, систем доведення теорем.

Мотивація настільки важлива, що без правильної мотивації висувати будь-які претензії про марно втрачені 10 років життя абсолютно безрезультатно, сертифікат відкликається. Як перевірити чистоту мотивації і наскільки точні можуть бути рекомендації? Можу лиш сказати, що видо повинні бути достатньо чесним перед самим собою, адже програмування — це складний виснажливий процес, а мислення — найвища форма управління організмом, тому вади в його роботі можуть призвести до непоправних наслідків.

Якщо крім програмування ви нічого не вмієте, то непогано було би розвинутив в собі первні стратегії відступу: мінімальні техніки управління диханням та дієтою, легкий спорт без фанатизму, трохи йоги, можливо активні види спорту. Якщо ви вважаєте, що у цілому ви психічно стабільна людина, то пригугуйтеся до сюрпризів на шляху досягнення загадок простору без внутрішньої чистоти намірів.

Моєю особистою мантрою, з якою я вивчаю програмування — це посвята результатів своєї роботи людям та всім істотам, не нашкодивши нікому без виключення. Взагалі вивчення програмування мало кому може зашкодити та може мати форму глибокого відлюдництва святого монаха. Хоча є виключення, програмісти, а особливо гарні програмісти, в своїй більшості не жорсткі істоти, і їх надмірна агресія і сердитість, направлена в позитивному ключі інтроспекції є двигуном аутичного досягнення потаємностей професії програміста.

Взагалі, якщо мотивація алмазної візницьі привести усі істоти до абсолютного просвітлення знається вам занадто езотеричної, то хочи би стара етично норма інженерів минулого «не нашкодь, а краще допоможи людям» є тим мінімумом, який необхідно перевіряти перед кожною сесією програмування. Уявіть собі, що ви з рівниня лева кладете на вітар просвітлення 10 років самоосвіти в області програмування зі скрині свого життя для того аби принести користь людям та суспільству. Без подібної мотивації вам просто не бути звідки черпати енергію для щоденних вправ в програмуванні та мисленні.

1.4 Всевідання як джерело натхнення

Головна риса характеру, яка необхідна в людині, щоб стати топовим програмістом — це схильність до вивчення та дослідження феноменів, їх аналізу, синтезу та абстракції. Це бажання розібрати і досліджувати іграшку має так глибоко перебувати у свідомості, що здається, ніби дитина вже народжується з цим даром і швидко розбиратися у феноменах за належного інтенсивного навантаження на нейросіточку. Іншими словами — це хакерство, якщо ви любите досліджувати системи, розбиратися в програмному коді, розумієте, як працюють процесори, знаєте, як працює логіка та математика, то ви вже можете стати топовим програмістом. Бажання побудувати максимально точну модель феномена має бути гіпертрофованим, воно має бути незакритим гешталтом, який не дає вам спати ночами, поки ви його не закриєте. Саме ця фанатична одержимість конвертується в те, що буде дровами у нашому вогнищі просвітлення на шляху до всезнавства у світі програмування. Звідки взялося всезнання? Це друга сторона медалі головного джерела натхнення хакера. Якщо при локальному розгляді феноменів головною думкою має бути побудувати максимально точну модель феномену, то при фокусуванні в нескінченність до країв горизонту, це бажання проявляється у вигляді максимально швидкого пізнання всіх феноменів і їх універсальні принципи пристрою. Такий мета-хакерський трансцендентальний напів-фрічний майндсет необхідний для розуміння того, наскільки абстрактними і широкими можуть бути виклики на шляху пізнання глибинних мов, якими написано наш всесвіт.

Так, як мови програмування використовуються у всіх сферах людської діяльності, то топовий програміст абсолютно точно повинен розбиратися у всіх доменних моделях, усіх типах та всіх математиках, які виникають у різних мовах програмування. Зазвичай, університетські 5 років я б рекомендував провести якраз у охопленні всіх математик та всіх видів мов програмування, перед тим як поринути у фундаментальну математику та системне програмування. Взагалі хороша сучасна освіта рівня PhD автоматично має на увазі вільне володіння мовним

та математичним забезпеченням у дослідженні всесвіту, так що нічого такого, що не вимагають топові університети, курс топового програміста в цій частині всезнавства тут не вимагає. Потрібна повна автономність на рівні полетів у космос і відновив усі знання та навички за потреби у найкоротші терміни шляхом легкого спогаду.

Щоразу поглинаючи якийсь пласт інформації ви вивільняєте величезний простір свободи, який або заповнюється новими недослідженими пластами, або звільняється абсолютно, якщо вже всі пласти поглинули. Але коли ви повністю вичерпаете всю карму, тоді буддахуд прийде автоматично, тож це вже програма максимум. Адже після того, як ви вивчили якийсь предмет і дали кілька майстер класів по ньому, ви просто гортаєте всі книги по ньому, за якими навчалися і це все для вас навіть не буквар, тому що буквар ви вже самі написали, це для вас просто шум дерев у лісі. Ви повністю вичерпали цей предмет, стали майстром у ньому, ви вже бачите всі перерізи глобулярних фазових просторів, маєте на руках кілька моделей і прототипів. Це стан всезнавства. Бажання цього стану — необхідний компонент топового програміста.

Якщо ви побудували якийсь простір феноменів, наповнивши їх змістом і залишаючись там у комфортному середовищі обмеженого знання, ви вже втрачаєте топову мотивацію як компонент всезнання. Не спрямувавши своє мислення в нескінченність, ви не зможете побачити весь ландшафт і правильно розставити пріоритети в поглинанні наукових дисциплін, щоб здійснити «стрибки Тигра» між цими пріоритетними реперними точками.

Тільки дослідники, які сповнені вродженого бажання будувати нові теорії та мовні простори, наділені насінням творчості, що веде до топової реалізації. Міждисциплінарний підхід може виникнути лише за умов широкого профілю. Ніхто ніколи не ставив завдання скласти курс для підготовки людей, яких можна було б назвати культовими хакерами, тому й вимоги до підготовки мають бути позамежними. Тільки олімпійське бажання всезнання може реально наблизити вас до нього.

2 Теоретичні основи

2.1 Метафілософія

Дуже коротко про сучасні філософії. Якщо визначити філософію предикативно, то це наука, що вивчає наступний перелік питань: 1) як жити добре в достатку та гедонізмі максимально довго всім і не померти від воєн та метеоритів; 2) реальний всесвіт та інші питання гнесеології; 3) свобода волі; 4) етика; 5) математика; 6) музика; 6) література, — це все питання, або мовні набори та форми, що цікавлять сучасних філософів.

В этой заметке мы попытаемся построить формальную систему и на ее примере показать интерпретацию трех линий передачи современной философии: европейской или континентальной философии – школы, которая задала начало глобальной интерпретации мира и реконструирования языков, в том числе и математики; восточной философии как пример особенной школы, на примере которой мы будем строить модель; и аналитической или англосаксонской философии, которая формализуется современной математикой.

2.2 Європейська філософія

На наш взгляд главный вопрос европейской философии – это Good Life. Как жить, как жить хорошо самому, в социуме, какие цели могут стоять перед индивидом и видом, баланс этики и этика баланса. Европейская философия родила геометрию, психоанализ, научила людей не бояться свободы, трансформировать агрессию, быть более зрелым существом, и под венец своего развития поставила вопрос о языке и языковой игре, как основополагающем инструменте рефлексующего сознания.

Язык перестал иметь почву, он стал просто узорами, семантика которых утрачена, философия стала формой литературного искусства.

Представители континентальной философии: Аристотель, Платон, Кант, Декарт, Ницше, Фрейд, Юнг, Юм, Хайдеггер,

2.3 Тибетська філософія

В восточной философии центральным вопросом является освобождение, себя и других, в первую очередь от различных форм страдания. Эта философия обладает четкой системой, которая неразрывно связана с телесными и умственными практиками, и выжила на протяжении тысячелетий в законсервированном горном плато. Здесь также поднимаются вопросы этики и свободы воли, но основной упор делается на интеллектуальные и неконцептуальные упражнения ведущие к непосредственному переживанию пространства.

Некоторые формулировки восточной философии, такие как недвойственность всех феноменов поддаются формализации в гомотопической теории типов (используя методы аналитической философии), что и побудило к дальнейшим исследованиям в области формализации эзотерических теорий.

Представители восточной (тибетской) философии: Атиша, Нагарджуна, Бхававивека, Камалашила, Шантаракшита, Арьядэва, Буддхапалита, Чандракирти, Цонкапа, Мипам, Лонгченпа.

2.4 Аналітична філософія

Аналитическая философия рождена в математике, ранняя аналитическая философия начинается наверно с Лейбница, Ньютона и Эйлера. Поздняя аналитическая философия начинается с Фреге и по списку: Рассел, Уайтхед, Дедекин, Пеано, Гильберт, Фон-Нейман, Карри, Акерманн, Карнап, Сколем, Пост, Гёдель, Черч, Бернье, Тюринг, Клини, Россер, Мак-Лейн, Ловир, Гротендик, Скотт, Джояль, Тернье, Мартин-Лёф, Милнер, Жирар, Плоткин, Рейнольдс, Бакус, Барр, Барендрехт, Лерье, Сили, Кокан, Хьюет, Ламбек, Воеводский, Эводи, Шульман, Шрайбер.

Если описать в двух словах главный вопрос аналитической философии — это язык пространства. Построение языка, кото-

рый даст формальный фундамент не только математике и размышлениям, но и самой философии.

2.4.1 Мова простору

Формальные основания языка размышлений, математики (всей) и физики (вселенной).

2.4.2 Мовні фреймворки

Языковые фреймворки для менее формальных (с парадоксами) и нечетких (стохастических) систем.

2.4.3 Конкретні соціальні дифузійні моделі

Прикладная философия. Использование языковых фреймворков для описания конкретных феноменов.

3 Мыслення

3.1 Що таке мыслення

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.

3.2 Характеристики чистого мыслення

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, визуализациям. Второй раздел посвящен техникам работы с мышлением с точки зрения пространства, где мышление ассоциируется с пространством, в котором оно пребывает, неаналитиче-

ской медитации устремлении в бесконечность, медитации от-
дохновения.

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 отказаться 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 череде 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 помудрели, 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 проявит 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.

3.3 Коштовне намисто мислення

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 относиться 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 отдельный 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.

3.4 Отрути мышления

Самые три неблагоприятные формы мышления по моей личной классификации.

Инертность мышления – это колесо медитации. Будучи раз запущена 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.

4 Простір професійного розвитку

4.1 Структура курсу

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 Лямбдагарбха

Следующий уровень —это платфоообразующий уровень, который включает язык программирования, рантайм и аппаратуру. Обычно взрослые академические языки создаются сразу с рантаймом, поэтому назовем эту секцию уровень университетского 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 крышуется 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 Таємні настанови

5.1 Аспекти курсу

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.

5.2 Гранична точність

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 раскладывают саму систему до атомов, но еще и близашую 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, языкоостроение (системы типов, формальные языки программирования, языки общего назначения). Сюда также входят: любые формальные математические теории, формальные логики, язык кванторов Пи и Сиг-

ма.

5.3 Гранична оптимальність

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.

5.4 Гранична складність

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 объединений 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 слож-

ность) тоже окажется интересным объектом рассмотрения. В существе 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 демонстрация 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).

6 Практика

6.1 Спочатку йога розуму, потім вже йога тіла

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 X-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 эпинифриновой 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.

6.2 Программування та 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 приравняются. 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.

6.3 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.

7 Посвята всім святим програмістам