Взгляд
на эволюцию языков программирования
через призму
концепции метасистемного перехода В.Ф. Турчина

Андрей В. Климов

Институт прикладной математики им. М.В. Келдыша РАН, г. Москва

klimov@keldysh.ru

20 мая 2026 г.

Семинар «Компьютерная алгебра» факультета ВМК МГУ и ВЦ РАН

Аннотация (прямой шрифт из объявления; курсив — добавление)

- Посмотрим на эволюцию языков программирования через призму концепции метасистемного перехода (МСП) В.Ф. Турчина. Она ставит вопрос:
  - Какие произошли крупномасштабные скачки в понятиях, парадигме и устройстве языков, связанные с новыми видами деятельности над алгоритмами, программами и языками?
- Посмотрим, выполнились ли ожидания В.Ф. Турчина в 1970-е годы и что произошло, что он (и большинство программистов) не могли представить.
  - *Особенно: что произошло неожиданного!*
- Обсудим некоторые МСП за последние полвека *(список не исчерпывающий)*
  - рост числа языков
  - анализ и преобразования программ (метавычисления)
  - статическая типизация
  - верификация
  - доказательное программирование
  - параллели между современными языками и математической логикой
  - разработка программ с помощью «искусственного интеллекта»
- Также пофантазируем о будущем и попытаемся сделать прогнозы.

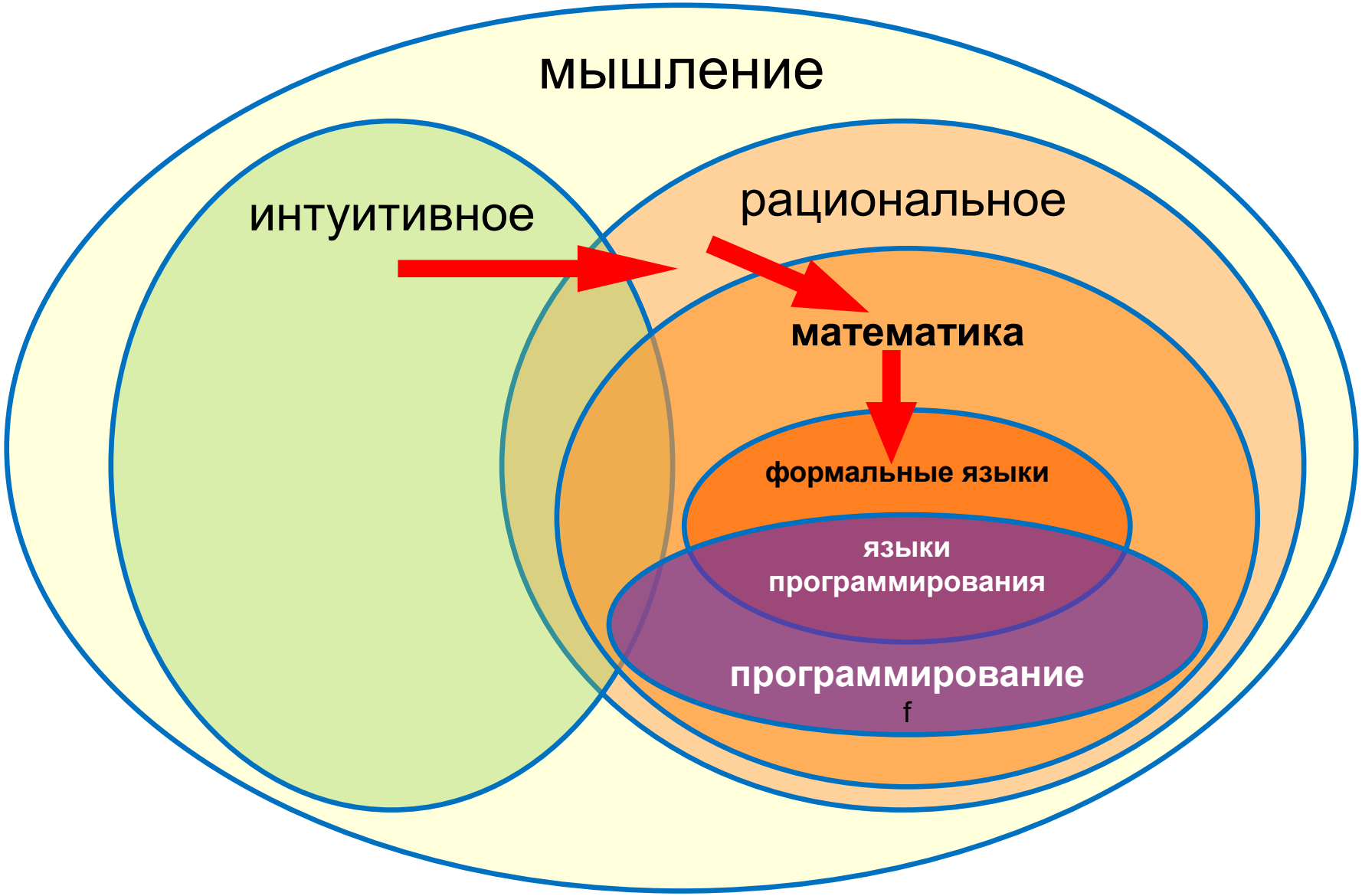
***Фокус**: что произошло в последние полвека?*

***Метод**: от общего взгляда на математику, как «родину» программирования, к конкретике*

***Способ анализа, парадигма:** концепция метасистемного перехода В.Ф. Турчина*

- *Видим скачок, быстрые изменения — думаем: не МСП ли? (может, и нет)*

Мышление ⟶ математика ⟶ программирование ⟶ языки



В чем объективность математики?

- **Распространенное мнение математиков: платоновский мир**
  - Ю.И. Янов, Математика и метаматематика // Математические вопросы кибернетики. Вып. 16. — М.: ФИЗМАТЛИТ, 2007. — С. 129–154. URL: http://library.keldysh.ru/mvk.asp?id=2007-129

> Вопрос об объективности математических понятий, безусловно, выходит за рамки математики, однако, как показывает исторический опыт, большинство математиков и философов всегда считало и считает математические объекты принадлежащими в том или ином смысле **реальному миру идей,** который так же непротиворечив, как и материальный мир. В подтверждение этого мнения приведём одну цитату из [Бур]: «Каковы бы ни были философские оттенки, в которые понятие математических объектов окрашивалось у того или иного математика или философа, имеется по крайней мере один пункт, в котором они единодушны: это то, что эти объекты нам *даны* и не в нашей власти приписывать им произвольные свойства так же, как физик не может изменить какое-либо природное явление. … и даже сегодня не один математик, афиширующий непримиримый формализм, в глубине души охотно подписался бы под следующим признанием Эрмита: *Я полагаю, что числа и функции Анализа не являются произвольным созданием нашего ума; я думаю, что они существуют вне нас с такой же необходимостью, как и предметы объективной реальности, и мы их встречаем или открываем и изучаем их так же, как физики, химики и зоологи».*

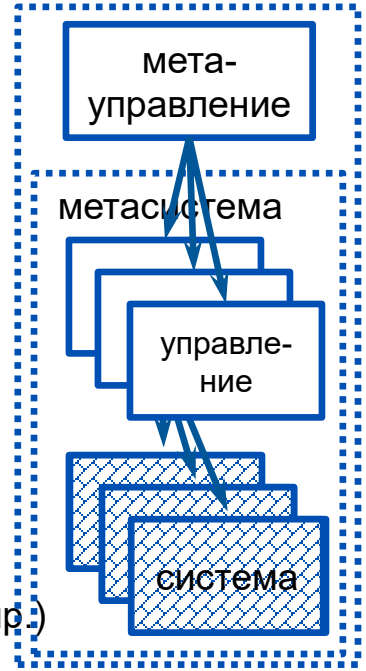
- **Мнение «кибернетика»**
  - Объектом изучения математики и ее методом являются **формально-языковые системы**, которые **объективны** — в том же смысле, в каком *единственно понятие алгоритма.*
  - **Открытие понятия алгоритма — главный математический результат XX века**

Концепция метасистемного перехода В.Ф. Турчина

- **Книга «Феномен науки», год написания ~1980**
  - демонстрация понятия МСП на материале эволюции в природе и культуре
- **Концепция МСП — это способ анализировать мир, то есть *субъективна***
  - но и *объективна*, как-то отражая реальность
    - следы МСП наблюдаются в иерархическом устройстве разных систем
  - критерий «истинности»: плодотворность
    - а не фальсифицируемость по Попперу

refal.ru

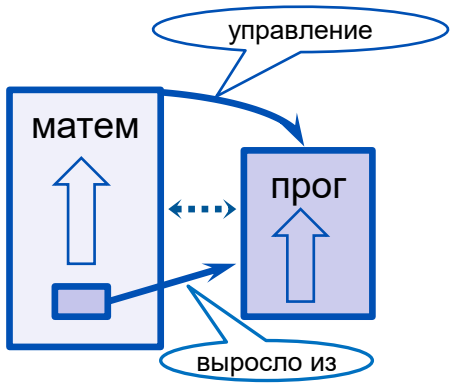
- **Интуитивно: понятие метасистемы такое же, как у «метаматематика», «метаязык»**
  - но с ним связан ряд наблюдений о типичных явлениях, помогающих использовать это понятие
  - управление
    - система есть объект манипулирования в метасистеме (анализа, преобразований, управление поведением и проч.)
  - разрастание предпоследнего уровня
    - при возникновении и развитии метасистемы происходит разрастание предпоследнего уровня в каком-нибудь части (структуры или поведения)
    - это ведущий признак совершения МСП
  - при следующем МСП: стабилизация пред-предпоследнего уровня
    - пред-предпоследний уровень становится «устойчивым материалом»
    - для эволюции с гибкостью в одной части нужна «крепость» в другой



- **Пример МСП по одной схеме: данные ⟶ переменные + операции над ними**
  - в математике: переход от арифметики к алгебре
  - в программировании: от вычислений к метавычислениям (суперкомпиляции и пр.)

Эволюция математики как метасистемы для программирования

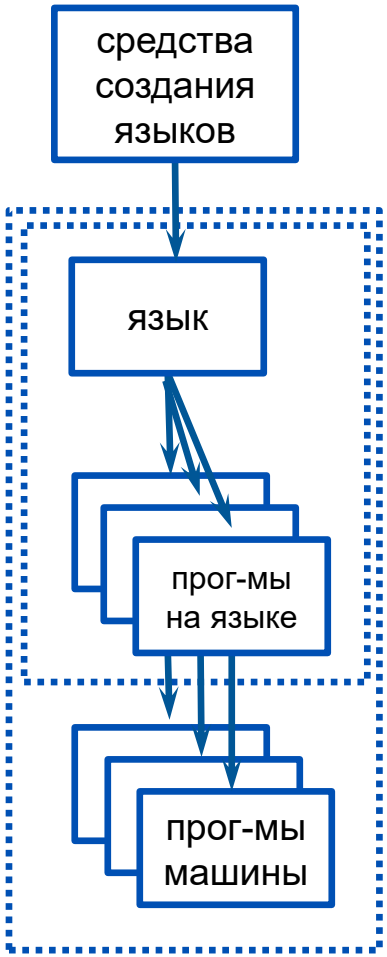
- **Программирование выросло из математики**
  - первые десятилетия (до XXI века) росли параллельно
    - математика как метасистема над программами и языками
      - метаматематика, дискретная математика, алгоритмы
    - математические семантики языков и программ
      - операционная, денотационная, логическая и т.п.
    - для верификации программ
      - вначале — логическая семантика (тройки Хоара и т.п.)



- **Метаматематика**
  - разные формализмы «оснований математики» и определения математических теорий
  - аксиоматико-дедуктивные системы
    - удовлетворяют современные потребности математиков, но оторваны от вычислений
    - понятие алгоритма используется для определения алгоритма проверки доказательств
  - конструктивные подходы
    - конструктивные логики: алгоритмы подразумеваются
    - алгоритмическая математика (неполна): объект = алгоритм (Л. Брауэр, А. Марков, Э. Бишоп)
    - расширение алгоритмической математики В.Ф. Турчиным (полная относительно теории мн-в)
  - зависимые типы: dependent type theory (DTT)
    - соответствие Карри–Ховарда: типы ↔ утверждения, программы ↔ доказательства
    - породило типизированные языки переднего края развития: Coq, Agda, Idris, Lean
- **На теории типов программирование «вернулось» в математику на теориях типов**
  - формализации математики на компьютерах — наконец, пошла волна работ
  - доказательное программирование — намечающаяся революция в разработке программ

Появление и рост языков программирование как МС-переходы

- **Язык программирования как метасистема**
  - «управление»: собственно язык
  - «объекты управления»: программы на языке
  - «разрастание предпоследнего уровня»:
    - рост числа программ по сравнению с машинными кодами
- **1970-е: В.Ф. Турчин ожидал дальнейший рост числа языков как МСП**
  - в первую очередь, предметно/проблемно-ориентированных
    - из-за разнообразия прикладных областей
    - DSL — domain-specific languages
  - средства описания, создание языков и манипулирования над ними
    - метаязыки; Рефал был создан как метаязык
    - макроязыки, макросистемы (модные в 1970-е, но забытые!)
- **В 1980-90 казалось, что рост числа языков замедлился**
  - «матерели» ОО-языки; появились типизированные функциональные языки
  - Неужели В.Ф. Турчин ошибся в «предсказании» разрастания числа DSL? Нет!
- **2000-... «Взрыв» DSL и интереса к ним с начала века**
  - за годы выросло много инструментов описания и создания языков
  - вернулась «мода» на DSL, теперь с многими бизнес-приложениями
- **Расширяемые языки пришли на смену «внешним» DSL**
  - это МСП внутри языков
  - DSL = средства расширения + библиотека определений
  - Лисп и его диалекты выжили «благодаря» расширяемости
  - C++ с шаблонами (templates), встроенными в типизацию
  - Объектно-ориентированные шаблоны как DSL; напр., цепочки вызовов методов
  - Haskell с программированием «высшего порядка», higher-order programming
  - Типизация помогает сделать синтаксис внутреннего DSL контекстно-зависимым



Почему выжил Лисп (с диалектами)?

- **Лисп стал расширяемым метаязыком**
  - **Минимальная база**
    - лямбда-исчисление; данные: CONS и атомы
  - **Стал языком-конструктором**
    - символьный
    - программа как данные
    - лямбда-исчисление
      - результаты в (мета)математике и семантике языков
      - стало базой для роста и развития языков
  - **Макросредства как механизм расширения языка**
    - Лисп стал удобным полигоном для конструирования языков и преподавания
    - пример: ОО-программирование осваивалось через SmallTalk и библиотеку на Лиспе
- **Поучительно сравнение с «умершим» Рефалом**
  - Рефал был создан как мета(алгоритмический)-язык
  - Lisp оказался лучше как метаязык (помимо субъективных, социальных, рыночных и пр. причин)
  - Рефал оказался плохо расширяемым?
    - слишком «специализированным»?
    - слишком «высокоуровневым»?

Пример макро на Scheme

Определение **when**
(как **if** без **else**)

Макроопределение

```
(define-syntax when
  (syntax-rules ()
    ((_ test expr ...)
     (if test
         (begin expr ...)))))
```

Макровызов

```
(when (> x 0)
  (display "x is positive")
  (newline))
```

Статическая типизация языков как МСП
(1) параметрическая типизация

- **Появление параметрической типизации**
  - начало: дополнительная информация для человека и компилятора (Фортран, Алгол и далее)
    - монотонное развитие вплоть до объектно-ориентированных языков (Симула-67 и далее)
    - типично: метасистема зарождается внутри существующей как частный вид деятельности
  - скачок: параметрическая параметризация
    - полиморфизм: System F, J.-Y. Girard (1972), J. Reynolds (1974)
    - полиморфизм Хиндли-Милнера (Hindley-Milner-Damas, 1969-1978-1982)
    - свойства программ, выводимые из типов
      - Abstraction Theorem: John Reynolds, 1983
      - Theorems for Free: Philip Wadler, 1989

Пример на Haskell:
Свойство из типов

```
map :: (a -> b) -> [a] -> [b]
reverse :: [a] -> [a]
          ⇓
reverse (map f xs) =
map f (reverse xs)
```

- **Статическая типизация как МСП над алгоритмическим языком**
  - разделение в структуре языка:
    - язык программирования = алгоритмический язык + язык типов
  - «управление»: язык типов
  - «объект управления»: алгоритмический язык
  - «разрастание»:
    - рост числа корректных программ, уменьшение числа ошибок
    - рост числа языков с параметрической типизацией в 1990–2000-х годах от функциональных ML и Haskell до объектно-ориентированных Java и др.
- **Программирование «вернулось» в математику на типизации языков**
  - программирование: верификация, доказательное программирование
  - математика: формализация математики на компьютерах

Статическая типизация языков как МСП
(2) параметрическая типизация → зависимые типы

- **Самое большое моё удивления за последнюю четверть века**
  - пополнение параметрической типизации зависимыми типами (dependent type theory, DTT)
  - до эквивалентности языка типов с языком логики и математики
    - соответствие Карри–Ховарда: **типы** ↔ **утверждения, программы** ↔ **доказательства**
- **Доказательное программирование**
  - программирование вместе с доказательствами корректности по каким-то требованиям
  - раньше верификация и доказательное программирование связывались с логическим подходом
    - логические условия, приписываемые к точкам программы (assert, pre-, post-, инварианты)
    - развивалось полвека и по прежнему буксует
  - теперь бурно развивается на основе типизации
    - почему стало более популярным? удобнее для человека? машины (для порождения и анализа)?
  - первые наивные ответы
    - программисты постепенно приучаются к все более широким языкам типов
    - логический подход — для программистов скачок к новому кругу понятий
    - инструменты на основе такой типизации более успешны и плодотворны
      - системы доказательного программирования от Coq до Lean
      - для анализа, преобразований и порождения программ (в т.ч. метавычислений)
- **«Прогноз» на ближайшие десять-двадцать лет**
  - массовые языки будут расширяться к higher order type systems
    - от Coq, Agda, Indris, Lean к таким как Java, C# и новым языкам

Порождение программ «искусственным интеллектом»

- **Нейросетевые языковые модели обучены на массе программного кода из интернета**
  - хорошо выдают код из учебников: лучше на популярных языках (Java), хуже на редких (Рефал)
  - врут («галлюцинируют») на задачах, отличающихся от типовых
- **Еще один МСП на программами и программированием: Vibe coding**
  - создание типовых небольших приложений по описанию на естественном языке
  - итеративно по командам человека
- **«Разрастание предпоследнего уровня»**
  - рост количества приложений
  - помощь в технической работе при квалифицированной разработке (во много раз)
  - справки (help-ы) по популярным системам (Windows, MS Office, системы программирования)
  - плодотворный «треп» с ИИ в процессе научной и прикладной работы
  - роль человека перемещается на постановку задачи, спецификацию и верификацию
- **МСП в организации самого ИИ: Сети ИИ-агентов**
  - одни агенты управляют другими агентами; разные агенты ставят задачу, кодируют, проверяют
- **Идет очередная промышленная революция**
  - социальный процесс обучения общества и перестройки бизнес-процессов
  - в бизнесе «паника»; оценки старых софтверных компаний падают; надувается пузырь ожиданий
  - начались увольнения в надежде сократить расходы; потом разочарования и наем назад
  - Робототехника: быстрый прогресс; острая конкуренция; внедрение в промышленность
- **Рост спроса на формальные методы и метавычисления из-за «галлюцинаций ИИ»**
  - ИИ будет порождать доказательства корректности программ вместе с кодом

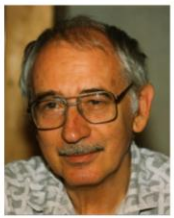
А что с метавычислениями?

- **Метавычисления = методы анализа и преобразования программ**
  - задачи: специализация программ, эффективная композиция (fusion), инверсия и т.п.
  - методы: суперкомпиляция, частичные вычисления и их развитие
  - зародились в 1970-1980-е
  - работы идут, но мало; практическое применение мало
  - еще нет «разрастания предпоследнего уровня»
  - надежды основателей еще не исполнились

Андрей Ершов (1931–1988)

Смешанные вычисления

Валентин Турчин (1931–2010)



Суперкомпиляция

Yoshihiko Futamura

Проекции Футамуры,
Generalized Partial Computation

Neil D. Jones

Частичные вычисления,
впервые: *spec*(*spec*,*spec*)

Alberto Pettorossi

Преобразования и верификация
логических программ

Состояние и перспективы метавычислений

- **МСП с метавычислениями еще не произошел**
  - нет разрастания предпоследнего уровня — роста числа приложений с глубокими преобразованиями программ
- **Интерпретационные задачи:**
  - не «взлетели»; рынок есть, но, похоже, небольшой для стимулирования
- **Задачи моделирования, симуляции**
  - мечтается давно (с Симулы-67)
  - надежда на специализацию и композицию (fusion), но еще нет никакого опыта
  - здесь рынок и спрос большой, но методы еще недостаточно наработаны
- **Доказательное программирование**
  - использование суперкомпиляции на месте нынешней нормализации (в системах от Coq до Lean)
  - автоматический вывод доказательство эквивалентностей (диссеры И. Ключникова и С. Гречаника)
- **ИИ и vibe coding**
  - пока лишь общие соображения, поскольку область молодая и должна еще заматереть
- **В перспективе: моделирование, симуляция эволюции, искусственная эволюция**
  - требование: должны «сами» возникать МСП
  - генетическое программирование — это еще не эволюция, хотя и часть ее механизма
  - нынешние нейросетки тоже часть механизма эволюции, но сами по себе оной не являются
  - можно начать с ускорения и качественного улучшения генетического программирования
- **Развитие самих метавычислений с применением ИИ и типизации**
  - применение ИИ в метавычислениях для обучения стратегиям, вывода стратегий
  - галлюцинации здесь не страшны, ибо сохраняется разделение функций
    - собственно метавычисления отвечают за эквивалентность преобразований
    - ИИ дополняет и замещает человека по управлению стратегиями
  - типизация как полезная информация о программе

В эволюции языков рассмотрены такие МСП

- **1950-е Язык программирования — это метасистема над программами**
- **1950-1960 Реализации языков: компиляторы и интерпретаторы**
  - вначале компиляторы и интерпретаторы вручную (*метасистема — сам человек*)
  - МСП: разрастание числа «универсальных» языков с 1960-х годов; управление = методы разработки
- **1960-1980 Возникают разные деятельности над языками**
  - автоматизация описания синтаксиса языков (парсеры), языки грамматик
  - средства описания отображения конструкций языков в базовые (макросредства и т.п.)
  - формальные семантики: операционная, денотационная, аксиоматическая, алгебраическая
  - разрастание предпоследнего уровня:
    - рост числа DSL с 1970-х годов: *Турчин ожидал, что будет «взрыв» DSL*
    - в 1980-90-е замедлилось: *я переживал, что «предсказание» Турчина не сбывается*
    - в 2000-е вернулась мода на DSL и новые понятия и инструменты вместо «внешних»
- **1990-2000 МСП внутри языков**
  - расширяемые языки: DSL внутри универсального языка, embedded DSL
  - определение DSL = языковые средства расширения + библиотека определений
- **2000-... Типизация: 2 революции — в основаниях математики и в устройстве языков программирования**
  - типы — от дополнительной информации к коду до формального языка математики
  - МСП, *не ожидавшийся В.Ф. Турчиным* (и большинством программистов тоже)
- **2020-... ИИ, обученные нейросети генерируют типовой код и типовые приложения (vibe coding)**
  - взрыв простых приложений и автоматизация кодирования; помощники в информационной работе
- **1970-... Метавычисления**
  - «предсказание»: роль метавычислений возрастет во времена многих МСП над языками
- **«Прогноз» развития языков и программирования на ближайшие десять-двадцать лет**
  - массовые языки будут расширяться к dependent type systems
  - доказательное программирование