



Санкт-Петербургский
Государственный
Политехнический
Университет

Институт прикладной
математики и механики

КАФЕДРА ТЕЛЕМАТИКА

Введение в профессиональную деятельность

Лекция 12

Теоретический минимум компьютерных наук (КН):

что можно, а что нельзя вычислить ?

<https://forany.xyz/s-5>

СПб,
15 мая, 2018 г.

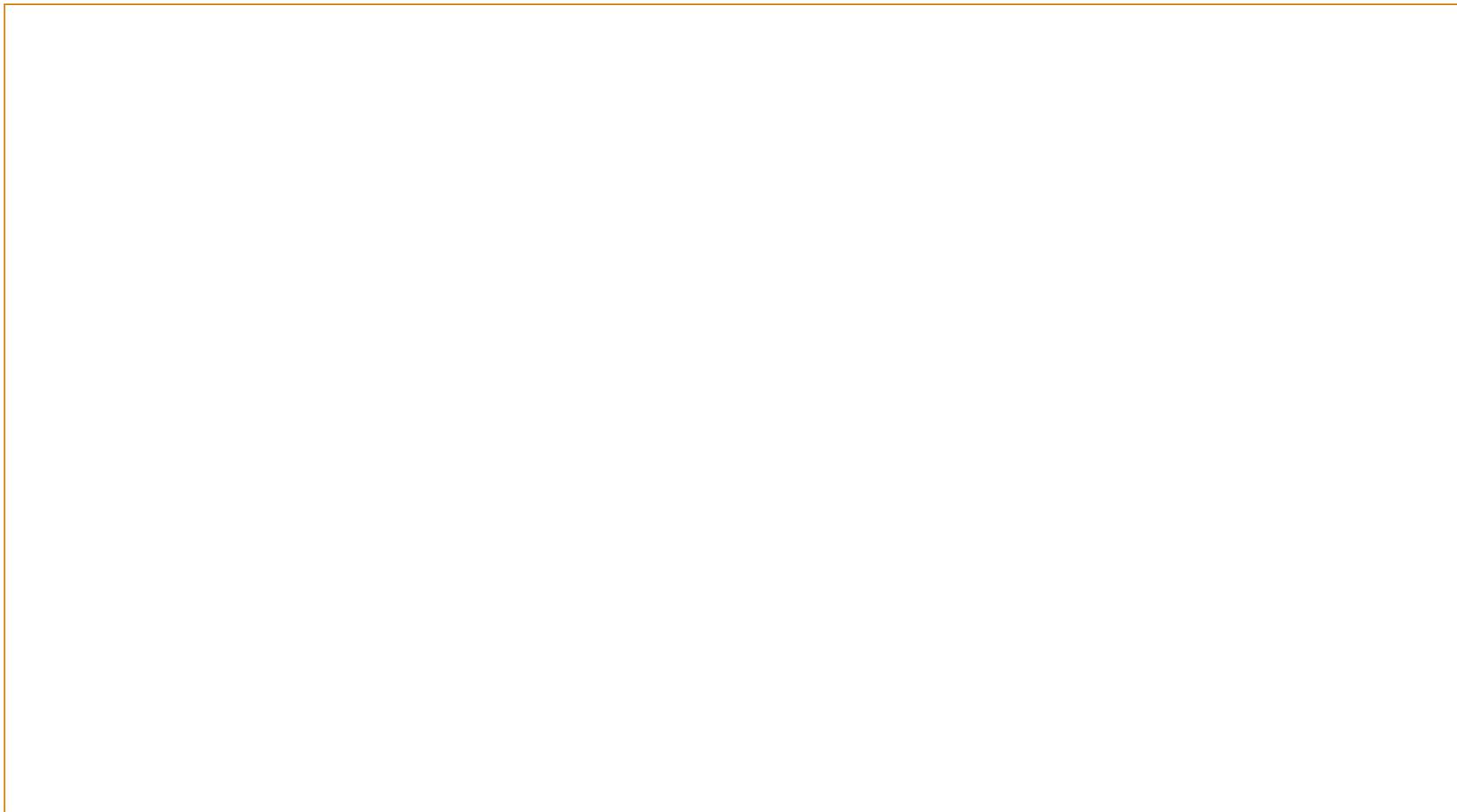
На прошлой лекции обсуждали «скрепы» науки

- РЕДУКЦИОНАЛИЗМ (физика) - **суть**: **сложные** явления и объекты могут быть поняты на основе законов, свойственных **простым** частям объекта.
- ЭВОЛЮЦИОНИЗМ (биология) — **суть**: критерием «улучшения» свойств объектов живой природы является «**выживаемость**» (сохранения себя) в условиях изменения среды обитания.
- РАЦИОНАЛИЗМ (осознающая себя природа) — **суть** : что существует – то разумно , а **что разумно – то существует**. Пока в рамках компьютерных наук остается без ответа фундаментальный вопрос: существуют ли в материальной Природе в «чистом виде» математические структуры, которые можно вычислить?!

Итак:



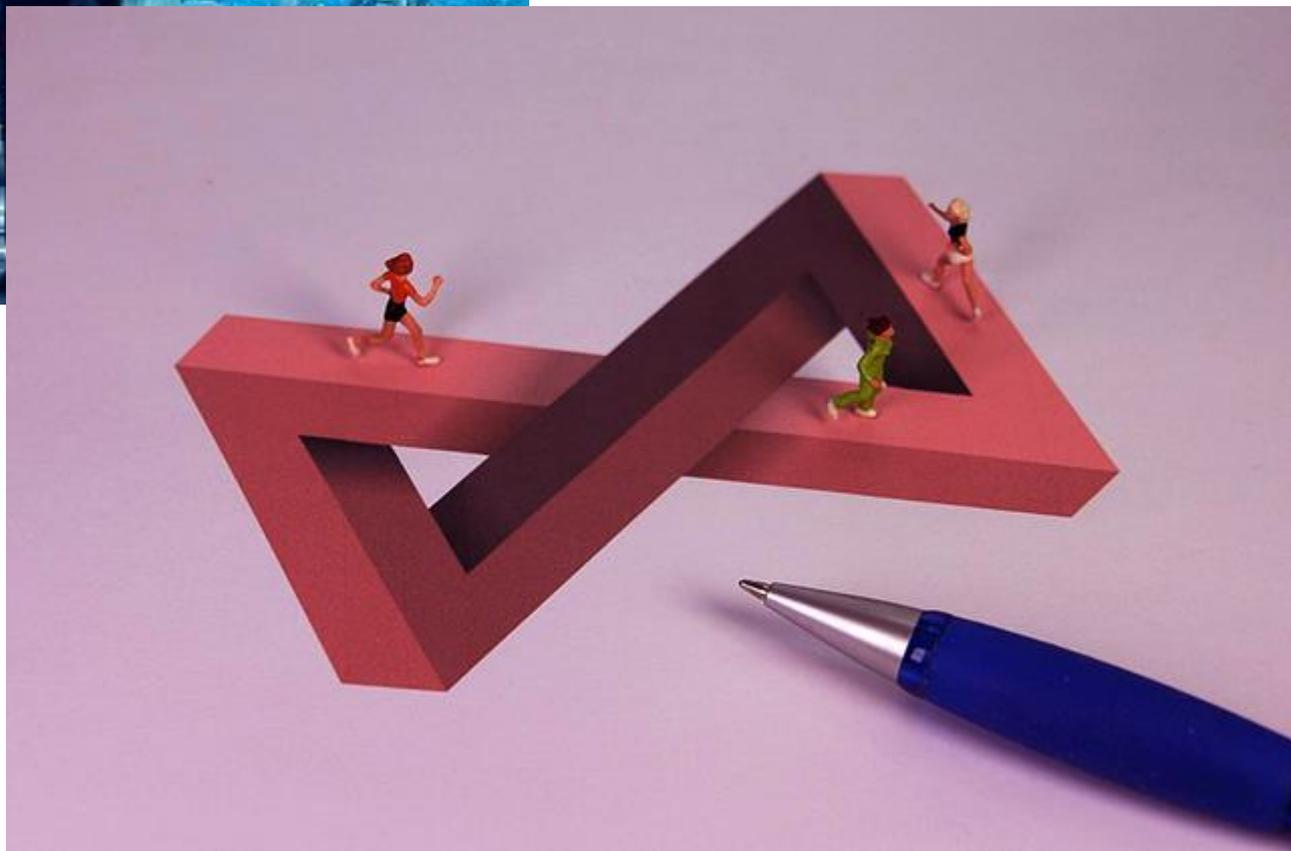
Книга знакомит вас с **вычислительным мышлением**, позволяющим решать любые сложные задачи. Научиться писать код просто, но чтобы стать профи, который будет востребован всегда и везде, **нужны фундаментальные знания**. Здесь вы найдете только самую важную информацию, которая необходима каждому разработчику и программисту каждый де



Где проходит «теоретическая» граница виртуального, вычисляемого и реального



Будущее делается нами, но не для нас» (Стругацкие)



Все, что можно вычислить, то существует !? в :

- материальном мире
- виртуальном виде
- ментальном пространстве

Антуан де Сент-Экзюпери:

Совершенство достигается не тогда,
когда уже нечего прибавить,
но когда уже ничего нельзя отнять.

Можно ли создать совершенные системы ?

Имеются соображения, ставящие под сомнение возможности создания **совершенных компьютерных систем**, опираясь на современную систему научных знаний

- теоретически, ключевой компонентой **совершенных систем** должен стать «разумный субъект».
- **Совершенный - значит разумный**
- Как создать «разумный субъект»? –вызов современной науке

«Разумный - значит совершенный»

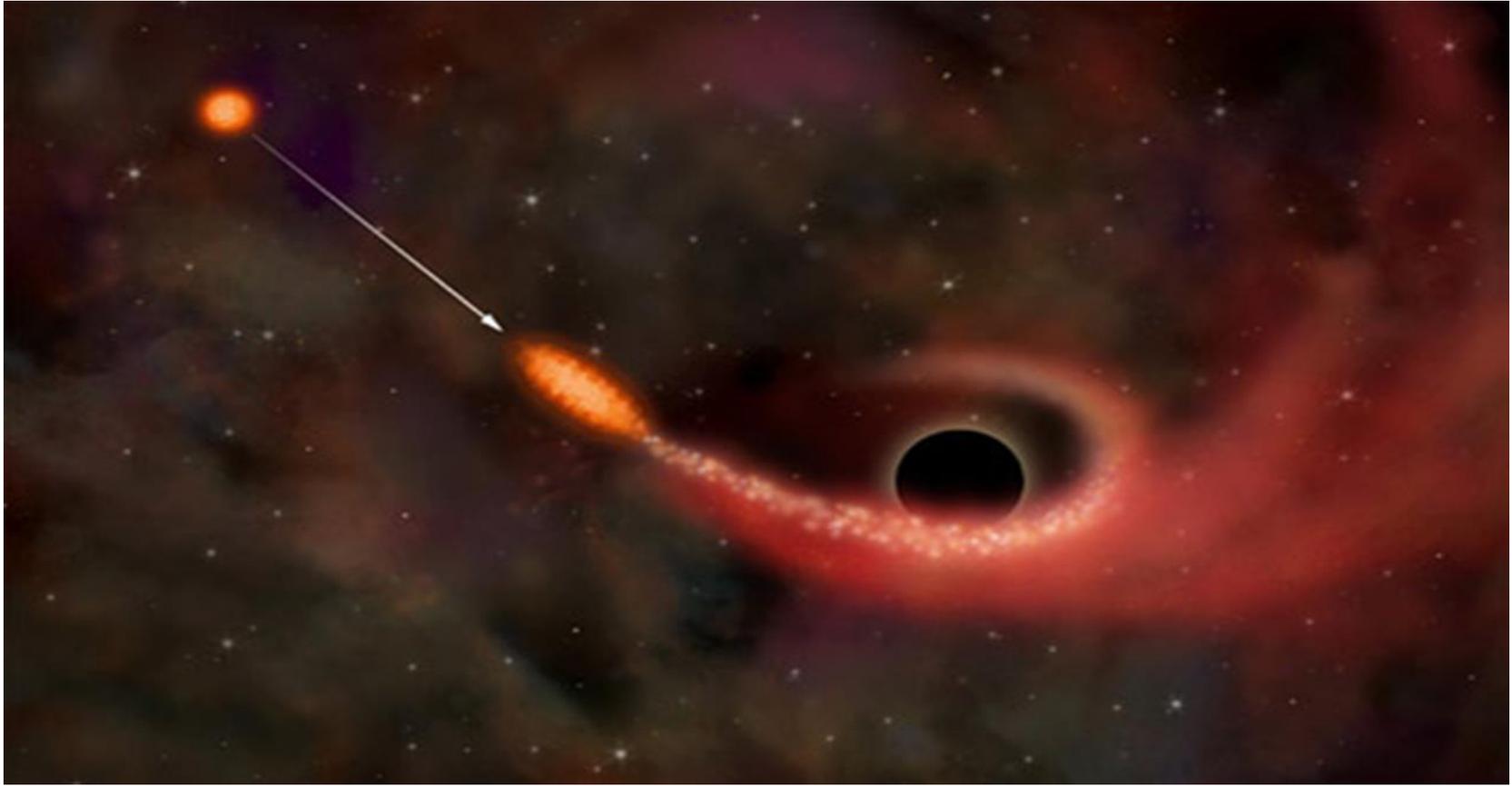
- Разумные системы совершенны, так как могут менять физическую природу своих тел, поэтому потенциально способны существовать вечно.
- Совершенный субъект обладает образом себя, который, в свою очередь обладает образом себя и т.д. (фрактальность)
- Совершенная система имеет субъекта-наблюдателя, который обладает двумя «точками зрения»: внешней (отражение, образ) и внутренней (строение, структура) моделями системы
- Совершенной системы способен к саморазвитию и, более того, к саморепродукции; другими словами, одни СС могут порождать другие сверхличности.
- Модель «совершенной системы» это математический фрактал, а физический прототип – «черная дыра»

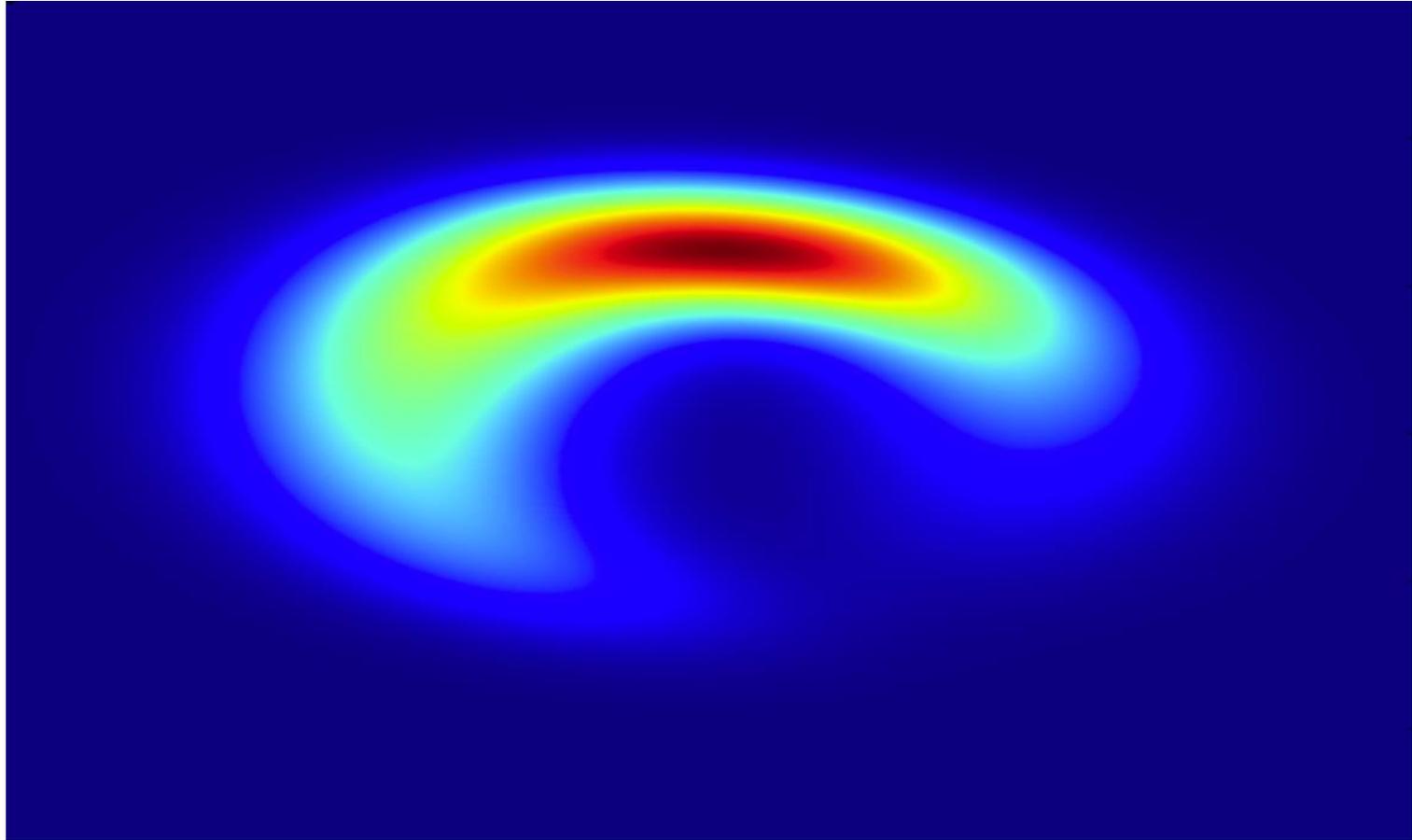
Есть ли в природе «совершенные системы»

L.Susskind, Black holes and information paradox // Scientific American, April 1997, p. 52). В «голографическую» модель Мира включена новая сущность – «черная дыра».



Поверхность черной дыры хранит информацию структуре реальности





Редукция в рамках компьютерных наук

Суть: Применение принципа «**истина в числе**»:

- «Извлечение» знаний в форме цифровых моделей из объектов природы.
- Создание программного обеспечения для «вычисления» новых знаний
- Понимание того, как **сложность** моделей связана с энтропией и **энергией**, необходимой для проведения вычислений

Эволюция в рамках компьютерных наук

Суть: понимание «нового и сложного» через эволюцию «простого» :

- Машина Тьюринга – понимание вычислений как механических (физических) процессов с неограниченными ресурсами памяти и времени вычислений
- Понять переход от логики к арифметике и далее к ... интегральным схемам
- Рассматривать архитектуру компьютеров как структуру-носитель процессов обработки информации , включая квантовые вычисления
- Развитие языков программирования от ассемблера к машинному обучению
- Движение от машины Тьюринга к вычислительным системам способным к самоорганизации и самообучению
- В чем разница в концепциях «сильный и слабый «искусственный интеллект»

Рационализм в рамках компьютерных наук

Суть: логическое обоснование существования вычислимых и не вычислимых функций:

- Разрешимые и неразрешимые множества. Примеры алгоритмически неразрешимых проблем.
- Колмогоровская сложность компьютерных моделей.
- Технологии и языки программирования
- Теорема Гёделя о неполноте формальных систем и ее прикладной смысл - рост стоимости ИС по мере накопления данных (формализуемых знаний)

Пример: фундаментальная сложность компьютерных моделей объектов «физической» реальности

Если за меру сложности компьютерных вычислений при решении прикладной задачи взять 1D (одномерную) задачу, аргумент которой разбит на 100 частей, то получим :

$$\text{«1»} = 1^{100}$$

Если в модельной задаче 1D «интервал» заменить на «квадрат» 2D , то есть увеличить размерность задача всего в 2 раза, то мера «сложности» станет

$$2^{100} = 1,2676506 \times 10^{30} \approx \text{«1»} * \infty$$

Вывод: Вычислительная сложность «реальных» т.е. многомерных задач науки и техники равна «потенциальной бесконечности». Масштабирование таких задач требует использования специальных инструментов – суперкомпьютеров, чтобы решать прикладные задачи с заданной точностью за приемлемое или «конечное время», с конечной затратой энергии.

Великие проблемы, которые без КН не могут быть решены
(А. Перлис – первый лауреат премии А. Тьюринга)



Проблема происхождения Вселенной как процесса;

Проблема природы сознания (осознания), как материального феномена

Проблема сложности численного моделирования физических объектов

Процесс цифровой трансформации знаний: есть начало, но нет ... конца... «проблема останова»)

Начало: Изучаемый физический объект или проектируемая система

Математическая модель, учитывающая законы сохранения и «сложность» объекта с выбранной точностью

Дискретная модель физических процессов и «суперкомпьютерный» решатель для выбранной СК системы

Реализация «цифровой» программной модели

Исполняемый код программы «компьютерный эксперимент»

Оценка полученных результатов

Результат: параметры, характеристики, свойства модели объекта/системы

Верификация кода программы

Валидация численных результатов

↑ уточнение требований



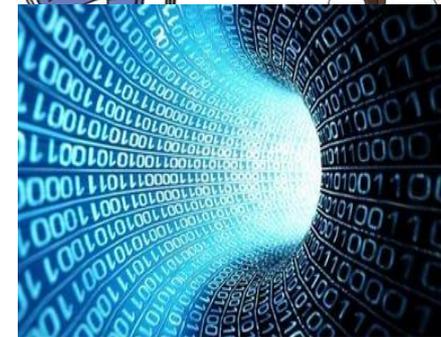
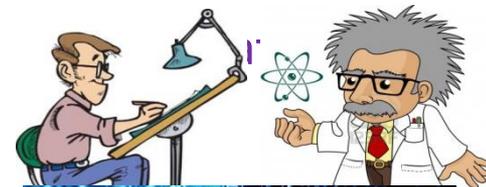
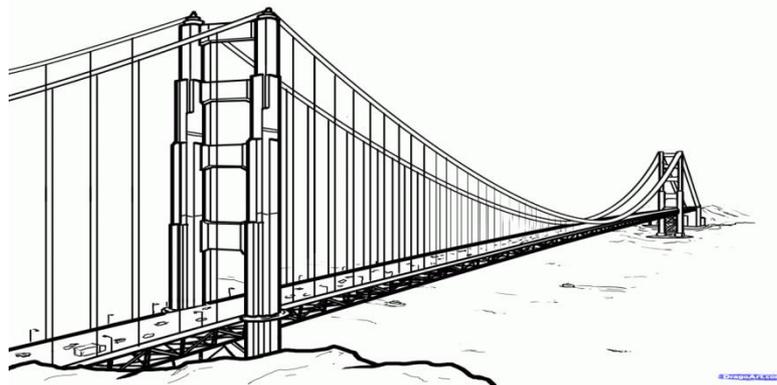
Эволюция вычислений: от точных и «медленных» вычислений к параллельной математике «больших данных»,

Проблема: За 50 лет производительность компьютеров выросла более, чем в **1000 000 000** раз. Однако, выигрыш в скорости **решения** практических задач, связанный с уменьшением времени такта процессора с 2 микросекунд до 1 наносекунды, составляет лишь около **1000** раз («пар» ушел в «гудок» 😊)

Что делать ? : использовать методы **параллельных вычислений** и математику «больших данных»

Аппаратное обеспечение:
много ядер

Программное обеспечение:
много потоков



СКЦ
«Политехнический»

«цифровая» реальность:
много моделей,
масштабов, алгоритмов

Таксономия (научная систематизация) компьютерных наук

Уровень 1

Базовые знания (теоретические основы):

- Основы «математической логики» физических процессов,
- Вычислительная математика и законы физики
- Теория алгоритмов

Технологические основы (специальные знания):

- Архитектура компьютеров
- Обработка информации (теория информации, статистика, модели, поиск данных, лингвистические аспекты)
- Языки программирования (C/C++, ООП, ФЯП, параллельное программирование)
- Операционные системы (архитектура ОС, процессы, межпроцессное взаимодействие, потоки, планирование, работа с памятью и периферией, POSIX-системы)
- Базы данных и знаний

Уровень 2 (прикладной специалист)

- Разработка ПО (жизненный цикл ПО, этапы разработки, основы ведения программных проектов, инструменты);
- Анализ данных (Data Mining, машинное обучение, нейронные сети, Искусственный Интеллект);
- Компьютерные сети (по уровням стеков TCP/IP и/или ISO/OSI протоколы, сетевое программирование на C/C++);
- Языки программирования с управляемым кодом (управляемый код, виртуальные машины, сборщики мусора, юнит-тестирование, практика на C# или Java);

Уровень 3 (разработчик)

- Построения интерфейсов пользователя, компьютерная графика;
- Тестирование ПО, верификация кодов, валидация результатов (виды тестирования, инструменты);
- Веб-технологии (HTTP-протокол, веб-сервер, CGI, кэширование, клиентское программирование);
- Распределенные системы (архитектуры распределенных систем, протоколы сетевого взаимодействия компонентов, инструменты, принципы, подходы к построению высокопроизводительных вычислительных систем, отказоустойчивость, большие данные, высокие нагрузки);
- Интерпретируемые языки программирования (особенности, практика по одному-двум языкам: JS, PHP, Python, Ruby).

Экспертный уровень

- Разработка архитектур и компонент **цифровых и квантовых компьютеров**
- Разработка компиляторов
- Разработка операционных систем
- Построение распределенных программно-аппаратных комплексов, рассчитанных на особо высокие нагрузки и обработку «больших данных» в реальном масштабе времени.

Обязательный «теоретический минимум» студента СПбПУ в области КН

Это «Самостоятельно Установленный Образовательный Стандарт» (СУОС). Теоретический минимум компьютерных наук состоит из **11 экзаменов и 100 вопросов** по основным разделам компьютерных наук. Модель обучения «50:50» = 50 вопросов из лекций: 50 вопросов для самостоятельного изучения.

1. Алгебра логики
2. Вычислительная сложность
3. Теория моделей
4. Теория алгоритмов
5. Компьютерная арифметика
6. Структуры и базы данных
7. Архитектура компьютеров
8. Языки программирования
9. Компиляторы и операционные системы
10. Теория информации
11. Машинное обучение

Суть компьютерных наук: понимание того, что Вселенная это квантовый компьютер, который вычисляет сам себя

