Информатика: логика, алгоритмы и работа с информацией

Александр Клеймёнов

Fractal Academy
Выпуск 4 (Осень 2026), Статья 12
Серия: «Развитие ментальных механизмов через школьные предметы»
Автор: Александр Клеймёнов
ORCID: 0009-0004-9404-2400
DOI: [будет присвоен]
Дата публикации: 1.11.2026

© 2026 Александр Клеймёнов. Данная работа (на русском и английском языках) лицензирована на условиях Creative Commons Attribution-ShareAlike 4.0 International License.
Название (рус.): «Информатика: логика, алгоритмы и работа с информацией»
Название (англ.): «Informatics: Logic, Algorithms, and Working with Information»
Ссылка: https://fractal-academy.ru/articles/article-4-12/
(CC BY-SA 4.0)

Аннотация

В этой статье я показываю, как информатика как школьный предмет может стать фундаментом для развития ментальных механизмов в цифровую эпоху - при условии, что мы перестанем сводить её к обучению работе с офисными программами и заучиванию синтаксиса языков программирования. Опираясь на модель КПКЛ, представленную в предыдущих статьях модуля, и на актуальные исследования 2024-2025 годов в области информатического образования, я обосновываю, что информатика обладает уникальным потенциалом для развития построения причинно-следственных связей, критического мышления, поиска и анализа информации, прогнозирования и стратегического мышления, а также пространственного воображения и долговременной памяти. Я предлагаю конкретные примеры уроков и заданий, в которых алгоритмизация, программирование, работа с данными и цифровая гигиена служат не самоцелью, а средством развития мышления. Особое внимание я уделяю диагностике прогресса и связи предметного обучения с инфраструктурой школы - IT-лабораторией, робототехнической мастерской, медиастудией. Я также обосновываю, что перестройка преподавания информатики - это не просто методический вопрос, а стратегическая задача: именно через информатику формируется когнитивный аппарат, способный к алгоритмическому, вычислительному и системному мышлению - тому типу мышления, который становится необходимым для понимания и создания технологий, формирующих наше будущее.

Ключевые слова: информатика, ментальные механизмы, алгоритмическое мышление, вычислительное мышление, критическое мышление, программирование, работа с информацией, диагностика, инфраструктура.

Код, который начинается с вопроса

Я начну с утверждения, которое может показаться спорным, но я считаю его принципиальным. Информатика в школе - это не предмет о том, как пользоваться компьютером, а предмет о том, как устроена логика, как работают алгоритмы и как мы можем управлять информацией в мире, где её становится всё больше. Это предмет, через который человек учится мыслить системно - не как набор разрозненных фактов, а как структурированную, алгоритмически организованную реальность. Конечно, знание интерфейсов и умение работать с программами важно. Но это инструмент, а не цель. Цель - это способность разбивать сложные задачи на простые подзадачи; способность критически оценивать информацию в цифровой среде; способность находить, анализировать и структурировать данные; способность прогнозировать результаты и планировать стратегии; способность видеть абстрактные структуры и оперировать ими [1; 5].

Здесь уместно вспомнить Аду Лавлейс, которая в XIX веке работала с машиной Бэббиджа. Она увидела в ней не просто вычислитель, а инструмент, который может работать с символами, а не только с числами. Она написала первую программу - и в этом жесте было больше, чем просто набор команд. Она задала вопрос: а что, если машина сможет не только считать, но и создавать? Этот вопрос до сих пор стоит перед нами. И информатика в школе должна не просто учить писать код. Она должна учить задавать такие вопросы. Потому что код - это ответ. А вопрос - это начало. И если ребёнок выучил команду, но не понял, зачем она нужна, он выучил слово, но не понял смысл. Он знает, что такое цикл, но не знает, зачем он нужен. Он знает, что такое массив, но не знает, как он помогает решать задачи. И это самое большое поражение информатики - когда она становится набором символов, за которыми ничего не стоит. Когда она перестаёт быть языком и становится шифром. Который нужно расшифровать. Но никто не знает ключ. Потому что ключ - это не команда. Ключ - это понимание. Которое приходит не от заучивания. А от вопроса. От задачи. От ошибки.

От удивления. Которое ты испытал, когда программа заработала. И тогда ты захотел понять. Почему? И как? И что за этим стоит? И это желание понять - оно и есть начало информатики. Не команды. А удивление. Которое делает тебя исследователем. А не исполнителем. И это самое главное. Не ответ. А вопрос. Который ты задал. Машине. И которое она тебе ответила. Не словами. А результатом. Который ты увидел. И понял. И запомнил. Навсегда. Потому что он стал твоим. Тобой открытым. Тобой понятым. Тобой пережитым. И это и есть образование. Не сумма знаний. А сумма удивлений. Которые ты испытал. И которые сделали тебя тем, кто ты есть. И тем, кем ты станешь. В следующей программе. В следующем алгоритме. В следующей жизни.

Как я уже показывал в первой статье этой серии [5], принцип двойного назначения урока означает, что каждый урок информатики должен одновременно решать две задачи: предметную - освоение информатического содержания, и метапредметную - развитие ментальных механизмов. Эти задачи не противоречат друг другу - напротив, они взаимно усиливаются. Важно подчеркнуть: информатика создаёт условия возможности для развития ментальных механизмов, но не является их достаточной причиной. Само по себе заучивание синтаксиса или работа с офисными программами не разовьёт системное мышление. Развитие происходит только через специально организованную педагогическую деятельность, где учитель выступает как фасилитатор и архитектор когнитивного аппарата [5; 6]. Исследования подтверждают: предмет информатика позволяет не только вооружить человека навыками, но и сформировать более системное мышление, то есть научить мыслить критически, вести логически обоснованные доказательства и рассуждения, а также проводить анализ и синтезировать новые знания [8]. По словам В.С. Леднёва, всякий базовый компонент общего образования включается в содержание образования двояко - в виде особого учебного предмета и в виде вкраплений во все другие предметы [8]. Информатика - именно такой предмет: она не только самостоятельная дисциплина, но и универсальный инструмент, пронизывающий все остальные. Современные исследования также подчёркивают, что в последнее десятилетие наблюдается растущий сдвиг в сторону преподавания компьютерных наук и вычислительного мышления, начиная с детского сада [9]. Однако, как отмечает Марина Умаски Берс, большинство этих усилий продиктованы спросом на технически квалифицированную рабочую силу [9]. Такой инструментальный подход, на мой взгляд, упускает главное. Преподавание информатики должно быть гуманистическим предприятием, которое продвигает самонаправленное обучение, критическое мышление и реляционные процессы [9]. Исследование Берс показало, что четыре практики являются ключевыми для продвижения гуманистического образования через обучение программированию: критическое мышление, эффективная коммуникация, личная ответственность и социальная ответственность [9].

Но я хочу пойти дальше и показать, как каждый из одиннадцати ментальных механизмов может развиваться через преподавание информатики - и как это меняет сам подход к предмету. Информатика - это не просто технический предмет. Это универсальный язык алгоритмов, без которого невозможно современное мышление. Именно поэтому информатика, на мой взгляд, является ключевым предметом для формирования коммуникативно-познавательного каркаса личности в XXI веке [1]. Более того, перестройка преподавания информатики - это не просто методический вопрос. Это часть той исторической необходимости смены педагогической парадигмы, о которой я писал в статьях [3] и [4]. Индустриальная школа сводила информатику к набору технических навыков, которые нужно заучить. Но постиндустриальному обществу нужны люди, способные мыслить, творить, адаптироваться - и информатика здесь является ключевым инструментом. Как я показывал в статье [2], предметоцентризм - одна из главных причин живучести старой модели. Преодоление этого предметоцентризма начинается с переосмысления роли информатики в образовании.

Логика, которую можно увидеть

Построение причинно-следственных связей развивается через информатику самым прямым образом. Алгоритм - это последовательность шагов, где каждый шаг логически вытекает из предыдущего и определяет следующий. Понимание алгоритма - это понимание причинно-следственной цепочки. Способность разбивать сложную задачу на последовательность простых шагов. Способность видеть, как изменение одного шага влияет на весь алгоритм. Способность выстраивать логическую цепочку от входных данных к

результату. Учитель должен сознательно акцентировать внимание на логике алгоритмов, а не только на синтаксисе. Не запомни команду, а объясни, почему эта команда должна идти после той. Не напиши программу по образцу, а выстрой логическую последовательность действий. Как отмечают исследователи, обучение программированию позволяет научить мыслить критически, вести логически обоснованные доказательства и рассуждения [8]. Информатика в этом смысле - идеальный тренажёр для причинно-следственного мышления.

Это как с рецептом. Вы можете выучить последовательность действий. Но если вы не понимаете, зачем нужен каждый ингредиент, вы не сможете приготовить блюдо, если чего-то не окажется под рукой. Вы не сможете improvise. А программа - это improvisation. Потому что каждая задача уникальна. И каждый раз нужно думать. Не повторять. А создавать. И алгоритм даёт вам язык для этого создания. Язык, на котором вы говорите с машиной. И машина понимает вас. Если вы говорите логично. Если вы говорите последовательно. Если вы не пропускаете шагов. И тогда программа работает. И вы понимаете, что вы создали что-то, что работает. Не потому, что вы повторили. А потому, что вы построили. Своим умом. Своей логикой. Своим путём. Который начался с вопроса. И закончился результатом. Который стал твоим. Потому что ты его построил. Как строитель строит дом. Из кирпичей. Из мыслей. Из логики. И это самое главное. Не команда. А путь. Который к ней ведёт. И который остаётся с тобой. Даже когда команда забыта. Потому что путь - это умение. А умение не забывается. Как не забывается умение ходить. Или говорить. Или думать. Которое ты приобрёл. И которое теперь твоё. Навсегда. И это и есть образование. Не сумма знаний. А сумма умений. Которые делают тебя тем, кто ты есть. И тем, кем ты станешь. В следующей задаче. В следующем алгоритме. В следующей жизни.

Отладка как диалог с ошибкой

Критическое мышление - это ещё один механизм, который информатика развивает уникальным образом. Информатика даёт уникальные возможности для развития критического мышления. Отладка программы требует поиска ошибок. Оценка информации из интернета требует проверки достоверности. Анализ данных требует интерпретации и оценки. Способность находить и исправлять ошибки в алгоритмах. Способность оценивать достоверность цифровой информации. Способность анализировать данные и делать обоснованные выводы. Использование проблемно-ориентированного обучения. Исследование 2025 года показало, что модель проблемно-ориентированного обучения с использованием Scratch оказала статистически значимое влияние на навыки критического мышления семиклассников [10]. Средний балл посттеста экспериментальной группы вырос с 62,4 до 83,6, с большим размером эффекта (Cohen's d) 1,24 [10]. Наибольший рост был зафиксирован по показателям вывод и объяснение [10]. Исследование также подтверждает, что информатика позволяет формировать навыки системного и критического мышления [13]. В ФГОС второго поколения подчёркивается, что формирование только предметных результатов недостаточно - необходимы личностные и метапредметные результаты, которые невозможно формировать без деятельностной составляющей процесса обучения [13].

Это как с врачом, который ставит диагноз. Он не просто смотрит на симптомы. Он ищет причину. Он проверяет гипотезы. Он ошибается. И снова ищет. И в конце находит. Или не находит. Но он не останавливается. Потому что каждая ошибка - это шаг к истине. Так и в информатике. Ученик, который находит ошибку, не просто исправляет код. Он понимает, почему ошибка возникла. И как её избежать в будущем. И это самое важное. Не исправление. А понимание. Которое делает тебя не исполнителем. А исследователем. Который умеет видеть ошибки. И исправлять их. И идти дальше. К новому коду. К новой задаче. К новой жизни. Которая всегда начинается с первого шага. Который ты делаешь. Сейчас. Здесь. В этой программе. В этой жизни. В этом мире. Который ждёт твоего шага. И твоей мысли. И твоего решения. Которое изменит всё. Или не изменит ничего. Но ты всё равно сделаешь его. Потому что ты информатик. А информатик не может не думать. Как не может не видеть. Как не может не жить. И это самое главное. Не код. А жизнь. Которая делает тебя тем, кто ты есть. И тем, кем ты станешь. В следующем шаге. В следующей мысли. В следующей жизни.

Данные, которые говорят

Поиск и анализ информации - это механизм, который информатика требует и развивает. Современный мир - это мир данных. Информатика - это предмет, который учит работать с этими данными: находить, структурировать, анализировать, интерпретировать. Этот механизм относится к обеспечивающему уровню: без него ядро не получит материала для работы, но сам по себе поиск информации ещё не создаёт решение. Способность формулировать информационные запросы. Способность оценивать источники информации. Способность анализировать данные и представлять их в понятном виде. Работа с базами данных, поисковыми системами, инструментами визуализации данных. Анализ реальных данных, а не только идеальных примеров из учебника. Исследования подчёркивают, что информатика позволяет не только вооружить человека навыками, но и сформировать более системное мышление [8]. А системное мышление невозможно без умения работать с информацией на всех уровнях - от поиска до интерпретации.

Здесь уместно вспомнить Флоренс Найтингейл, которая в XIX веке использовала статистику для того, чтобы показать, что большинство солдат умирает не от ран, а от болезней. Она визуализировала данные в виде диаграмм - и это изменило медицину. Она не просто собрала данные. Она заставила их говорить. И информатика учит нас этому. Не просто собирать данные. А понимать, что они значат. И как их показать, чтобы другие тоже поняли. И тогда данные становятся не набором цифр. А историей. Которую можно рассказать. И которую можно понять. И которая может изменить мир. Если ты умеешь её рассказать. И если ты умеешь её понять. И информатика даёт тебе этот язык. Язык данных. На котором можно говорить с миром. И быть услышанным. И это самое главное. Не данные. А история. Которая за ними. И которую можно рассказать. Если научиться. И если захотеть.

Машина, которая учится думать

Прогнозирование и стратегическое мышление - это механизм, который информатика требует и развивает. Программирование - это постоянное прогнозирование. Прежде чем написать код, нужно предвидеть, как он будет работать. Прежде чем выбрать алгоритм, нужно предвидеть его эффективность. Прежде чем запустить программу, нужно предвидеть возможные ошибки. Способность предвидеть результаты работы алгоритма. Способность выбирать оптимальную стратегию решения. Способность оценивать эффективность разных подходов. Работа с алгоритмами разной сложности, анализ их эффективности, сравнение разных стратегий решения одной задачи. Проекты, требующие планирования и прогнозирования. Исследование 2024 года показывает, что вычислительное мышление включает в себя процесс решения проблем, который включает в себя проектирование решений, которые используют преимущества компьютеров [14]. А проектирование решения - это всегда прогнозирование и стратегическое планирование. Исследователи также разрабатывают модели, которые объединяют шесть критических типов мышления: логическое мышление, системное мышление, устойчивое мышление, стратегическое мышление [13]. Информатика - идеальная среда для развития всех этих типов мышления одновременно.

Это как с шахматистом, который думает на несколько ходов вперёд. Он не просто видит текущую позицию. Он видит будущие позиции. И выбирает ту, которая приведёт его к победе. А если он не видит будущее, он играет наугад. И проигрывает. Так и в информатике. Ученик, который не умеет прогнозировать, пишет код наугад. И часто ошибается. А ученик, который умеет прогнозировать, пишет код осознанно. И часто находит правильный путь. Даже если он не самый короткий. Но он ведёт к цели. И это самое важное. Не скорость. А направление. Не ответ. А путь. Который к нему ведёт.

Структуры в уме

Пространственное воображение и долговременная память - это механизмы, которые информатика также развивает. Информатика развивает пространственное воображение через работу с графикой, 3D-моделирование, визуализацию данных. И долговременную память через необходимость запоминать структуры данных, алгоритмы, синтаксис. Способность видеть структуры данных и алгоритмы в пространстве. Способность удерживать в памяти сложные логические конструкции. Работа с визуальными средами программирования (Scratch, блок-схемы), 3D-моделирование, создание игр и анимаций.

Исследования показывают, что обучение программированию позволяет развивать логическое, абстрактное и вычислительное мышление [8]. Абстрактное мышление тесно связано с пространственным воображением - способностью оперировать образами и структурами в уме [1]. Это как с архитектором, который смотрит на чертёж и видит здание. Он видит не линии. Он видит пространство. Он видит, как будут ходить люди. Как будет падать свет. Как будет звучать эхо. И он может изменить чертёж так, чтобы здание стало лучше. А ученик, который не умеет видеть пространство, видит только линии. И не может изменить чертёж. Потому что он не понимает, что за ним стоит. А за ним стоит мир. Который можно увидеть. Если научиться смотреть. И если знать, куда смотреть.

Урок, который начинается с задачи

Теперь я перейду к конкретике. Как выглядит урок информатики, построенный по принципу двойного назначения? Я предлагаю несколько примеров - от простых до сложных. Традиционный подход: объяснить синтаксис команд, дать задание написать программу по образцу. Подход, развивающий мышление: дать реальную задачу и предложить ученикам самостоятельно разработать алгоритм её решения. Пример задания для седьмого-восьмого класса: вам нужно написать программу, которая сортирует список чисел по возрастанию. Но прежде чем писать код, вам нужно разработать алгоритм. Анализ задачи: что значит отсортировать список? Какие у вас есть входные данные? Какой должен быть результат? Разбиение на подзадачи: на какие более простые шаги можно разбить эту задачу? Построение алгоритма: напишите алгоритм на псевдокоде или в виде блок-схемы. Объясните, почему шаги идут именно в этом порядке. Прогнозирование: что произойдёт, если вы поменяете шаги местами? Что произойдёт, если в списке будут одинаковые числа? Реализация: теперь напишите программу на выбранном языке программирования. Исследования подтверждают: разработка эффективных алгоритмов решения задач является важной частью школьного курса информатики [8]. А обучение программированию - это подход к развитию логического, абстрактного и вычислительного мышления у школьников [8]. Какие ментальные механизмы развиваются? Причинно-следственные связи - через построение логической последовательности. Прогнозирование - через прогнозирование последствий изменений. Критическое мышление - через анализ и проверку алгоритма [1]. Это как с ребёнком, который собирает конструктор. Он не просто соединяет детали. Он видит в уме, что должно получиться. Он представляет, как будет выглядеть модель. И если что-то не сходится, он меняет детали. И снова представляет. И так до тех пор, пока не получится то, что он задумал. А если он не умеет представлять, он просто соединяет детали наугад. И часто ошибается. И бросает. Потому что не видит результата. А результат - это не то, что получается. Это то, что ты видишь в уме. До того, как оно получится. И информатика учит этому. Видеть до того, как получится. Представлять до того, как построишь. И тогда программа становится не набором команд. А решением. Которое ты увидел. И которое ты теперь можешь показать другим. И это самое важное. Не результат. А решение. Которое делает тебя творцом. А не исполнителем. И это и есть образование. Не сумма знаний. А сумма решений. Которые ты создал. И которые теперь твои. Навсегда. И в следующей задаче. И в следующем алгоритме. И в следующей жизни. Которая всегда начинается с первого вопроса. Который ты задаёшь. Сейчас. Здесь. Машине. Которая не просто отвечает. Она исполняет. И ты удивляешься. И хочешь понять. И это самое главное. Не ответ. А удивление. Которое делает тебя человеком. Который умеет видеть. И понимать. И жить. В мире. Который не код. Но который можно понять. Если научиться смотреть. И если знать, куда смотреть. И информатика учит этому. Каждый день. На каждом уроке. С каждым ребёнком. С каждой программой. Которая не просто пишется. Она думается. И в этом её сила. И в этом её красота. И в этом её смысл. Который не заканчивается. Как не заканчивается информатика. Как не заканчивается мысль. Как не заканчивается жизнь.

Стены, в которых живёт код

Развитие ментальных механизмов требует не только новых методов преподавания, но и соответствующей инфраструктуры. Для информатики это означает создание пространств, где алгоритмическое и вычислительное мышление может развиваться через практическую, исследовательскую и творческую деятельность. Создавая эту инфраструктуру, мы формируем то, что Энди Кларк и Ким Стерельни называют когнитивной нишей. Инфраструктура - это не просто помещения, а материальная часть среды, которая

предоставляет скаффолдинг для развития ментальных механизмов. Без этой ниши механизмы остаются абстрактными концепциями. IT-лаборатория: пространство с современными компьютерами, где ученики могут программировать, работать с данными, создавать проекты. Исследования показывают, что интеграция проектно-ориентированного обучения, решения проблем и вычислительного мышления в учебные программы по информатике положительно влияет на критическое мышление и вычислительные навыки учащихся [13]. Робототехническая мастерская: пространство с робототехническими конструкторами, где алгоритмы становятся осязаемыми. Исследования показывают, что физические вычисления на основе Arduino положительно влияют на логическое мышление учащихся [15]. Медиастудия: пространство для создания цифрового контента - видео, подкастов, анимаций, игр. Развивает не только технические навыки, но и красноречие, творчество, командную работу [1]. Киберполигон: пространство для обучения цифровой безопасности, где ученики могут практиковаться в защите от киберугроз в безопасной среде.

Создание этой инфраструктуры - не затраты, а стратегическое вложение. Каждый ребёнок, нашедший через эти пространства свою точку входа - будь то программирование, робототехника или создание цифрового контента, - получает мощный импульс для развития всех остальных ментальных механизмов [1]. Это как с городом. В городе есть библиотеки, театры, мастерские. И каждый ребёнок может найти то, что ему ближе. А найдя, может вырасти. Не потому, что его заставляют. А потому, что среда позволяет. И это самое важное. Среда, которая позволяет. А не среда, которая требует. Потому что требовать можно много. А позволять - это искусство. И школа должна быть искусством. А не фабрикой.

Корни и ветви

Информатика не существует в изоляции. Она связана со всеми остальными предметами - и эти связи можно использовать для развития ментальных механизмов. Информатика и математика: алгебраическое и вычислительное мышление тесно связаны. Исследование 2025 года, охватившее шесть стран, показало, что алгебраическое мышление и вычислительное мышление являются ключевыми компетенциями в современном образовании, способствующими развитию навыков решения проблем и логического мышления [12]. Информатика и физика: моделирование физических процессов, обработка экспериментальных данных - прямое применение вычислительного мышления [1]. Информатика и биология: биоинформатика, обработка геномных данных - область, где информатика становится незаменимым инструментом [1]. Информатика и география: ГИС-технологии, обработка пространственных данных - использование информатики для понимания мира [6]. Информатика и искусство: генеративное искусство, компьютерная графика, создание игр - мост между технологией и творчеством [1]. Информатика и логика как новый предмет: формальная логика, теория алгоритмов - фундамент, на котором строится вся информатика [1].

Универсальный язык мышления

Я хочу закончить эту статью тем, что, возможно, является самым важным. Информатика - это не просто школьный предмет. Это универсальный язык мышления в цифровую эпоху. Она учит нас видеть мир как систему алгоритмов, структур данных и информационных потоков. Она формирует когнитивный аппарат, способный к логическому, алгоритмическому, вычислительному и системному мышлению. Именно поэтому информатика, на мой взгляд, является ключевым предметом для формирования коммуникативно-познавательного каркаса личности [1]. Когда мы учим ребёнка информатике, мы учим его не просто пользоваться компьютером или писать код. Мы учим его мыслить - разбивать сложное на простое, видеть закономерности, строить логические цепочки, предвидеть результаты. Мы учим его понимать, как работают машины. И как работает его собственный ум. Потому что программирование - это не только про компьютер. Это про то, как ты думаешь. Как ты решаешь задачи. Как ты находишь ошибки. Как ты создаёшь новое. И это самое главное. Не код. А мышление. Которое за ним. И которое можно развить. Если научиться. И если захотеть.

Именно поэтому перестройка преподавания информатики - это не просто методический вопрос. Это стратегическая задача. Как я показывал в статье об эпистемологической рекурсии [7], без развития ментальных механизмов, без формирования коммуникативно-познавательного каркаса личности,

человечество не сможет решить трудные проблемы XXI века. Информатика - это тот инструмент, через который формируется способность к алгоритмическому, вычислительному и системному мышлению. И если мы хотим, чтобы наши дети стали не просто пользователями, но создателями технологий - мы должны дать им этот инструмент в полной мере. Не ради оценок. Не ради экзаменов. Ради мышления в цифровую эпоху. Ради жизни. Ради того, чтобы они могли видеть мир не как набор отдельных программ, а как систему алгоритмов. И чтобы они могли не просто жить в этом мире, а понимать его. И, может быть, менять его. К лучшему. С помощью информатики. Которая не просто считает. Она думает. Как Ада Лавлейс, которая увидела в машине больше, чем вычислитель. Как Тьюринг, который задал вопрос, может ли машина думать. Как информатик, который видит истину там, где ещё только код. И это самое главное. Не код. А истина. Которая за ним стоит. И которую можно увидеть. Если научиться смотреть. И если знать, куда смотреть. И информатика учит этому. Каждый день. На каждом уроке. С каждым ребёнком. С каждой программой. Которая не просто пишется. Она думается. И в этом её сила. И в этом её красота. И в этом её смысл. Который не заканчивается. Как не заканчивается информатика. Как не заканчивается мысль. Как не заканчивается жизнь.

В следующей статье я перейду к предмету, который, на мой взгляд, является мостом между информатикой и физическим миром, между алгоритмами и реальными действиями, - к физической культуре. Я покажу, как этот предмет может развивать не только физические качества, но и стратегическое мышление, невербалику, командную работу и эмоциональный интеллект. Это будет следующий шаг. Следующая дверь. Следующий дом. И я приглашаю читателя пройти этот путь вместе со мной. Не как наблюдатель. А как участник. Потому что школа - это не то, что делают другие. Это то, что делаем мы. Все вместе. Учителя. Ученики. Родители. Общество. И если мы хотим изменить школу, мы должны начать с себя. С каждого урока. С каждой программы. С каждой мысли. И тогда, может быть, через двадцать лет мы увидим другую школу. Других детей. Другой мир. Мир, в котором мысль не боится себя. Мир, в котором вопросы ценятся не меньше ответов. Мир, в котором образование - это не подготовка к экзамену. А подготовка к жизни. И это будет не утопия. Это будет реальность. Которую мы создали сами. Своими руками. Своими мыслями. Своими уроками. И это самое большое, что мы можем сделать. Не для себя. А для них. Для детей. Для будущего. Для жизни. Которая продолжается. В каждом алгоритме. В каждой программе. В каждой мысли. Которая ищет. И находит. И снова ищет. Потому что в этом и есть смысл. Не в ответах. А в вопросах. Не в точках. А в многоточиях. Не в конце. А в продолжении. Которое всегда возможно. Пока есть код. Пока есть алгоритм. Пока есть мысль. Пока есть жизнь.

Ссылки на первоисточники

1. Клеймёнов А. Основные ментальные механизмы: структура коммуникативно-познавательного каркаса личности // Fractal Academy. - 2026. - Вып. 1. - Ст. 5. https://fractal-academy.ru/articles/article-1-5/
2. Клеймёнов А. Диагностика и развитие ментальных механизмов: от теории к педагогической практике // Fractal Academy. - 2026. - Вып. 1. - Ст. 6. https://fractal-academy.ru/articles/article-1-6/
3. Клеймёнов А. Историческая необходимость изменений педагогической модели: от индустриального конвейера к постиндустриальному мышлению // Fractal Academy. - 2026. - Вып. 1. - Ст. 3. https://fractal-academy.ru/articles/article-1-3/
4. Клеймёнов А. Экономическая и эволюционная невыгодность старых моделей педагогики // Fractal Academy. - 2026. - Вып. 1. - Ст. 4. https://fractal-academy.ru/articles/article-1-4/
5. Клеймёнов А. Введение: как предметы становятся инструментами развития мышления // Fractal Academy. - 2026. - Вып. 4. - Ст. 1. https://fractal-academy.ru/articles/article-4-1/

6. Клеймёнов А. Иностранный язык: коммуникация, диалог и межкультурное понимание // Fractal Academy. - 2026. - Вып. 4. - Ст. 11. https://fractal-academy.ru/articles/article-4-11/
7. Клеймёнов А. Эпистемологическая рекурсия в когнитивной науке: ментальные механизмы как инструмент преодоления «трудной проблемы» сознания // Fractal Academy. - 2026. - Вып. 3. - Ст. 1. https://fractal-academy.ru/articles/article-3-1/
8. Каган Э.М. Обучение программированию как подход к развитию логического, абстрактного и вычислительного мышления у школьников // Вестник РУДН. Серия: Информатизация образования. - 2017.
9. Bers M.U. Computer science education as a humanistic endeavor: a model for designing technology-rich formative experiences // Education and Information Technologies. - 2025.
10. Afifudin M.A., Rusijono R., Susamo L.H., Rokhman A.Y. The Effect of Problem-Based Learning Model Assisted by Scratch in Grade VII Informatics Learning on Critical Thinking Skill // EDU-TIK: Jurnal Pendidikan Teknologi Informasi. - 2025.
11. Jeong E.S., Heo M., Jin Y.H., Kim Y.S. Effect of a Flow Chart Learning on Logical Thinking Ability and Performance Achievement in Middle School Computer Programming Class // The Journal of Korean Association of Computer Education. - 2009. - Vol. 12, No. 6. - P. 11-19.
12. Sarmasági P., Rumbus A., Bilbao J., et al. Bridging Algebraic and Computational Thinking: Impacts on Student Development in K-12 Education // Informatics in Education. - 2025. - Vol. 24, No. 2. - P. 343-376.
13. Wardani F.P., Widiyastuti S., Ferisca H.G. The Role of Informatics Teaching in Developing Critical Thinking and Computational Skills in High Schools // Indonesian Journal of Economic and Management Studies. - 2024. - Vol. 2, No. 4.
14. Grizioti M., Kynigos C. Integrating Computational Thinking and Data Science: The Case of Modding Classification Games // Informatics in Education. - 2024. - Vol. 23, No. 1. - P. 101-124.
15. Kim Y., Hong K. The Effects of Physical Computing Based Software Applications Using Arduino on Logical Thinking of Elementary School Students // The Journal of Korean Association of Computer Education. - 2016.
16. Gardeli A., Vosinakis S. Fostering computational thinking in young students through student generated challenges in tangible mobile augmented reality games // Discover Education. - 2025.

Конфликт интересов
Автор заявляет об отсутствии конфликта интересов, связанного с публикацией данной статьи.
Финансирование
Исследование выполнено без внешнего финансирования.
Заявление о доступности данных
Данные, обсуждаемые в статье, содержатся в тексте публикации. Дополнительные материалы можно запросить у автора.
Вклад автора
Александр Клеймёнов - концепция, исследование, написание текста, финальная правка.

Отказ от ответственности
Мнения, точки зрения и данные, содержащиеся в публикациях Fractal Academy, принадлежат исключительно автору(ам) и не обязательно отражают позицию издателя или редакции. Fractal Academy и редакция не несут ответственности за любой ущерб, возникший в результате использования идей, методов, инструкций или материалов, упомянутых в публикациях.
Ответственность за соблюдение авторских прав, точность цитирования и достоверность фактов лежит на авторе(ах) публикации.