

## «ВЫЧИСЛИТЕЛЬНОЕ МЫШЛЕНИЕ» КАК ВТОРАЯ ГРАМОТНОСТЬ В СОВРЕМЕННОМ ЦИФРОВОМ ОБРАЗОВАНИИ И ОБЩЕСТВЕ

Т. Р. Файзрахманов<sup>1</sup> [0000-0001-5013-4523]

<sup>1</sup> *Институт разработки программного обеспечения и инжиниринга,  
Университет Иннополис, ул. Университетская, 1, г. Иннополис, Республика  
Татарстан 420500*

<sup>1</sup> tim.fayzrakhmanov@gmail.com

### **Аннотация**

В ходе невероятного роста популярности цифровых технологий и цифрового образования появилось такое понятие, как «вычислительное мышление». Его резкий спрос и популяризация, как ответ на современные нужды цифрового мира, послужили реструктуризации образования и возникновению большого количества литературы на данную тему. Не всегда ясно, чем же является вычислительное мышление, какие аспекты оно включает и с чем его сравнивать. В данной работе мы рассмотрели смысл данного понятия, его значение в современном цифровом образовании и провели аналогию наличия вычислительного мышления с навыком написанием обычного текста.

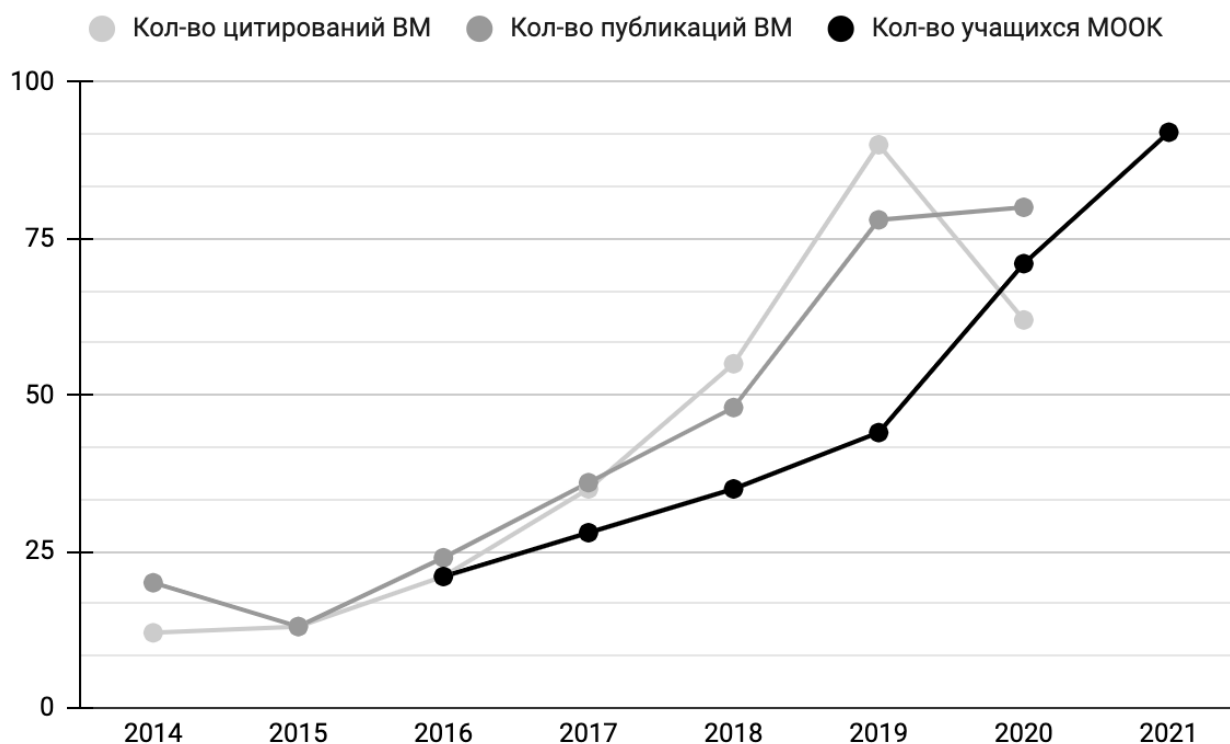
**Ключевые слова:** *вычислительное мышление, цифровое образование, кодинг, программирование, информатика, STEM.*

### **ВВЕДЕНИЕ**

Цифровое образование становится одним из ключевых факторов развития экономики из-за широкого распространения автоматизации, цифровых технологий, интернета и ограничений в связи с пандемией. По данным [1] и [2], с 2011 по 2021 годы число учащихся, охваченных массовыми открытыми онлайн-курсами (МООК), увеличилось с 58 тысяч до 170 миллионов и продолжает увеличиваться в среднем на 10–40 миллионов в год [3] внутри лишь одной платформы Coursera.

Параллельно с популярностью цифрового образования и развития «цифровизации» в целом наблюдается рост популярности другого, теперь уже смежного понятия – *вычислительное мышление*. На Рисунке 1, построенном по данным [3]

и [4], можно заметить, что экспоненциальный график роста популярности цифрового образования коррелирует с ростом количества научных статей и цитирований на данную тему.



Количество публикаций на тему Вычислительного мышления (ВМ): 100% = 100 статей;  
Количество цитирований статей на тему ВМ: 100% = 1500 цитирований;  
Количество учащихся в массовых открытых онлайн-курсах: 100% = 100 миллионов;  
Измерительные точки указаны приблизительно. Источники: [3], [4]

Рис. 1. Популярность цифрового образования и понятия «вычислительное мышление»

Вычислительное мышление является подмножеством (а иногда и самим множеством) более широкого понятия – *цифровая грамотность*. У цифровой грамотности имеется множество синонимов:

- Компьютерная грамотность (компетентность, навыки);
- Цифровая грамотность;
- Информационно-технологическая грамотность;
- Вычислительная грамотность;

- Вычислительное мышление (BM) или *Computational thinking* (CT).

В ежегодном докладе «Global Skills Report» от Coursera Inc. [5], на основе данных от более 100 миллионов учащихся в более чем 100 странах, отмечено, что цифровая грамотность («digital skills») стала одним из двух основных навыков *вне зависимости* от конкретной области знаний, где второй, комплементарной грамотностью, стала общая/человеческая грамотность («human skills»).

В связи с большой популярностью и значительным количеством пересекающихся по смыслам слов и понятий, представленных выше, мы будем придерживаться следующего разделения: цифровое обучение включает в себя цифровую грамотность, которая является синонимом компьютерной грамотности. В свою очередь, цифровая грамотность включает в себя (или является) *информационно-технологической грамотностью*, где технологическая часть включает в себя вычислительное мышление.

На основе обобщений в кратком глоссарии Coursera Inc. [5] и данных ChatGPT от OpenAI Inc. [6] мы построили простую *таксономию понятий современного цифрового образования*, где разделение понятий, перечисленных выше, можно увидеть наглядно (Рисунок 2).

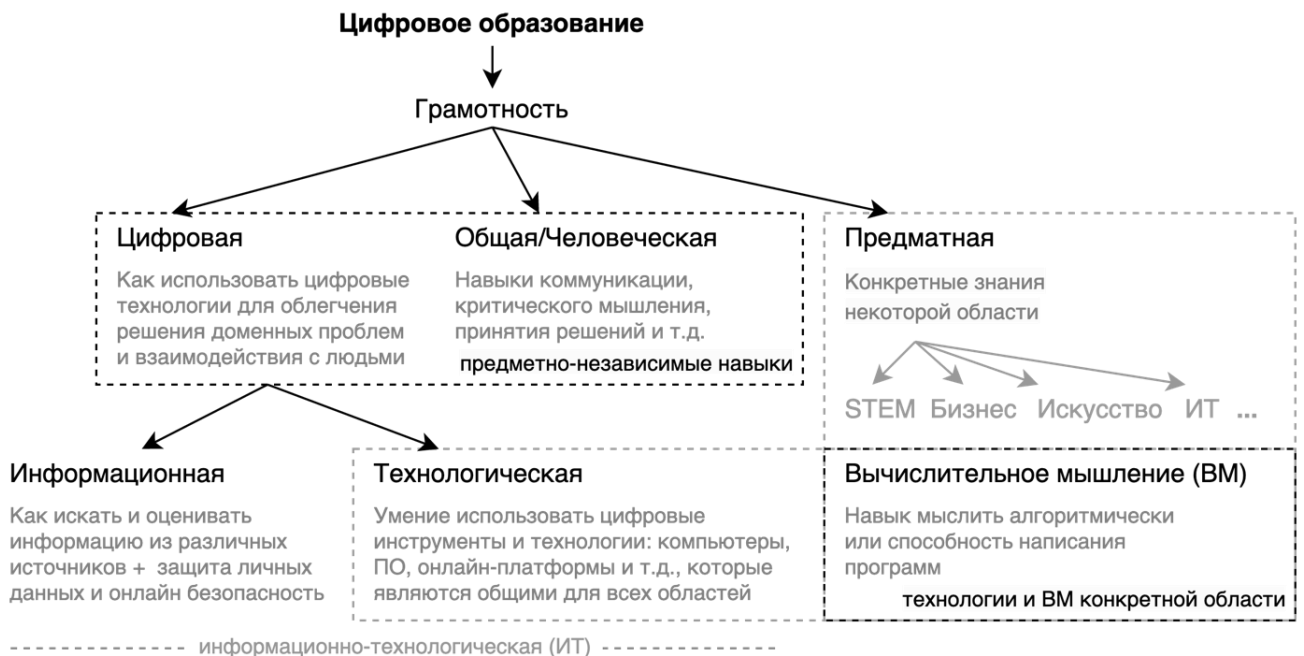


Рис. 2. Таксономия современного цифрового образования и грамотности

В данной таксономии мы видим, что цифровое образование включает в себя *доменную* и *доменно-независимую* грамотности. Доменная грамотность вполне очевидна – это навыки и терминология, необходимые для функционирования внутри некой области (естественные науки, математика, инженерия и т. д.). Доменно-независимая часть условно делится на «хард» и «софт скилы» – *человеческая грамотность* (например, «критическое мышление» или навыки коммуникации) и упомянутая ранее цифровая грамотность: *умение работать с информацией* (включая её безопасность и безопасность персональных данных) плюс работа с *цифровыми инструментами* (например, компьютером, текстовым редактором, браузером, онлайн сервисами и т. д.). Два последних объединяются в информационно-технологическую (ИТ) грамотность.

Мы хотим подчеркнуть, что «ИТ» как домен и «ИТ» как грамотность – разные вещи. Как домен, ИТ может включать в себя теоретическую информатику, профессиональную разработку ПО, машинное обучение и т. д. В то же время ИТ как грамотность – это скорее общие навыки «работы с компьютером» как для решения повседневных задач, так и для нужд автоматизации или быстрой работы с информацией внутри некой узкой области без перехода в индустриальную разработку ПО.

Фокусом данной работы является вычислительное мышление, которое мы поместили на Рисунке 2 в правый нижний угол. Мы сделали его частью технологической грамотности, т. к. появление и смысл этого понятия сложно представить в отрыве от цифровых технологий. В то же время, вычислительное мышление является и частью доменных компетенций, т. к. преподавание вычислительного мышления в отрыве от задач, решению которых оно способствует, также не имеет смысла.

Итак, на Рисунке 2 можно увидеть, что современная цифровая грамотность – это 25% навыков от всех имеющихся, а если допустить, что для коммуникации с людьми и навыками критического мышления мы овладеваем «по ходу жизни» и не проходим для этого отдельных курсов обучения, то цифровая грамотность и вовсе становится половиной современного набора компетенции.

Таким образом, можно заключить, что вычислительное мышление является ядром как информационно-технологической грамотности, так и тренда цифровой

---

грамотности в целом. Об этом, в том числе, свидетельствует повсеместная интеграция базовых понятий информатики под эгидой «Computational thinking» (Вычислительное мышление) в учебные программы обязательного школьного и дошкольного образования (K-12) западных стран [7], [8] (куда входят 29 европейских стран, США, Австралия, и Великобритания), в то время как преподавание вычислительного мышления в России и СССР давно известно под предметом «Информатика», который как правило преподается с 7-го по 11 классы. В современных же образовательных стандартах [9] присутствие информатики (именуемой как «Математика и информатика») предполагается с 1-го по 11 классы.

Итак, чем же является вычислительное мышление? На основе различных источников, куда вошли: анализ определений вычислительного мышления [10], [11], истории развития данного понятия [12], докладов Объединенного Исследовательского Центра Европейской Комиссии «European Commission’s Joint Research Centre» (JRC) [7], [13], руководств Британского сообщества для учителей информатики «Computing At School» (CAS) [14], [15] и международной образовательной программы по информатике для школьного образования «K–12 Computer Science Framework» [16] (под руководством бесчисленного количества ассоциаций, корпораций и образовательных организаций по всему миру [17]), мы выделили два простых определения. Первое, по нашему мнению, – наиболее часто встречающееся, совпадающее по смыслу с другими и просто лаконичное определение. Второе – неформальная версия первого:

1. **«Вычислительное мышление — это мыслительные процессы, связанные с выражением решений в виде вычислительных шагов или алгоритмов, которые могут быть выполнены компьютером»** [16, р. 68] или, как обобщение того, что вычисления доступны и человеку, «... выполнены человеком или машиной» [7, р. 24] (перевод).
2. **«Вычислительное мышление — это набор умственных навыков, которые преобразуют сложные, беспорядочные, частично определенные проблемы реального мира в форму, которую бездумный компьютер может решить без дополнительной помощи со стороны человека»** [15, р. 5] (перевод).

В ходе поисков были найдены и намного более обобщённые и, по нашему мнению, излишне размытые или «всеохватывающие» определения, которые, например, подчеркивают тот факт, что вычисления не являются исключительно человеческой конструкцией, но также присутствуют и в природе, например, в ДНК [13]:

*«Вычислительное мышление — это процесс распознавания аспектов вычислений в мире, который нас окружает, и применения инструментов и методов из компьютерных наук для раскрытия их сущности» [18, р. 29] (перевод).*

Также, совместно с попытками определить, что же такое вычислительное мышление, имеется и большое количество попыток определить его *аспекты* (также именуемые как «принципы», «концепции», «возможности» вычислительного мышления), включая использование различных способов их категоризации. Например, выделяют аспекты, связанные с методами работы с данными, моделированием и симуляцией, системным мышлением и т. д. [19, р. 10]. Дабы дать общее представление о наиболее часто встречающихся аспектах («ключевых словах»), ассоциируемых в литературе с вычислительным мышлением [13, р. 17], [20, р. 52], и [21, р. 14], приведем их в виде следующего списка:

1. **Абстракция.** Алгоритмы и данные как упрощённая версии явлений реального мира; решение сложных проблем путем отбрасывания ненужных деталей, сосредотачиваясь на сути.
2. **Алгоритмы.** Решение проблем через последовательность инструкций и промежуточных состояний, чтобы достигнуть конечное целевое состояние.
3. **Автоматизация.** Программы и код для выполнения задач с помощью компьютеров.
4. **Коллаборация, Параллелизм.** Разделение работы, задач и ролей с другими; использование готовых решений; написание программ с использованием возможностей параллельного вычисления.
5. **Креативность.** Использование «нестандартного мышления», творческого самовыражения и альтернативных подходов к решению проблем.
6. **Данные.** Поиск, сбор и обработка данных из различных источников; работа

- с типами данных; организация структур данных.
7. **Эффективность.** В алгоритмах нет лишних или ненужных шагов; разработанные решения просты в использовании.
  8. **Итерация.** Решения и совершенствование результатов путем нескольких/повторных подходов.
  9. **Логика, Анализ.** Аналитический склад ума и проверка гипотез / оптимальных решений перед началом написания программ.
  10. **Моделирование и дизайн.** Разработка систем, их внешнего вида и функциональности, которые впоследствии могут быть запрограммированы.
  11. **Паттерны и обобщение.** Поиск повторяющихся паттернов и формирование решений общего уровня, которые применимы к классу схожих проблем.
  12. **Декомпозиция проблемы.** Большие проблемы и артефакты распадаются на более мелкие и простые части, которые можно изучать по отдельности.
  13. **Тестирование и отладка.** Разработанные решения функционируют правильно, в соответствии с желаемым результатом; в обратном случае предполагается применение эвристики, формальных методов для выявления ошибок и их исправления.

Мы считаем, что данные аспекты излишне размыты, пересекаются друг с другом, с другими дисциплинами и вообще относятся к *мышлению* в целом.

Склонность к всеохватываемости «всё есть вычисления» или «вычисления вокруг нас», а также попытки трактовать любые когнитивные проявления как часть вычислительного мышления объясняются невероятным всплеском (или даже «бумом») автоматизации, популярности разработки ПО, интеграции машинного обучения во всевозможные дисциплины и (о чем мы уже упомянули ранее) реструктуризации образовательных процессов в школах и ВУЗах для подготовки учащихся к неизбежно надвигающемуся спросу на навыки «компьютерных наук». Этот спрос можно легко увидеть, взглянув на данные популярного сервиса по поиску работы Glassdoor Inc. В его отчете «50 лучших рабочих мест в Америке на 2022 год» [22] 6 профессий из топ 10 требуют навыки *кодинга*.

В этом контексте не будет преувеличением сказать, что вычислительное мышление становится в некотором роде «*второй грамотностью*», как это было

предсказано академиком А.П. Ершовым в докладе «Программирование – вторая грамотность» еще в 1981 году [23]. В современном же мире подобное отношение проявляется в таких заголовках, как «Вычислительное мышление: Навык цифрового века для каждого» [24] или, например, сопровождается такими цитатами, как:

*«Независимо от того, хотят ли наши дети стать фермерами, врачами, учителями или предпринимателями, им будет легче осуществить свои мечты в цифровую эпоху, если они будут иметь некоторое образование в области информатики. Нам нужно, чтобы наши дети приобрели навыки 21-го века для мира 21-го века, а коддинг учит их творчеству и умению решать проблемы, которые необходимы для успеха»* (перевод)

— Джон Тун, Сенатор США, Южная Дакота [25].

Можно заметить, что до данного момента мы использовали вычислительное мышление, программирование и коддинг как понятия, *взаимно заменяющие друг друга*. Однако, в соответствии с Планом действий по цифровому образованию (2021–2027), политической инициативы Европейского союза [26, р. 95], рекомендуется следующее разделение понятий (перевод):

- *Программирование* – это деятельность по анализу проблемы, дизайна решения и его практической имплементации.
- *Коддинг* – имплементация решений на определенном языке программирования.
- *Вычислительное мышление* – мышление как у «программиста», или способность понимать базовые понятия и механизмы цифровых технологий для формулирования и решения проблем.

Посыл же нашей работы заключается в том, чтобы свести суть вычислительного мышления, которая представлена в формальных образовательных стандартах и рекомендациях как некий абстрактный способ мышления с бесчисленными



теоретическими аспектами, такими как «абстракция», «автоматизация», «декомпозиция» и т. д. [13, p. 18] (о которых сами применяющие данное мышление возможно не догадываются), к простой и соответствующей практике **способности написания программ**. Кодинг в данном случае используется лишь как синоним программирования (где программирование, снова, – это способность написания программ) или интерпретируется как его *упрощенная версия* (например, «скриптинг» или написание цепочки простых команд *в виде кода* внутри некоторой вычислительной среды, например, MATLAB [27]).

Мы считаем, что способность писать программы (например, посредством некоторого языка программирования, такого как Python [28] или его визуального аналога, такого как Scratch [29]) – это и есть то, что в кульминации приводит к так называемому «алгоритмическому мышлению» или *дисциплине мышления* в виде четкой и ясной последовательности действий, а не, соответственно, хаотичного и неструктурированного потока. Также это является тем самым «драйвером», что движет экономику и достижения различных областей наук и жизни, о чем многократно говорят их самые разные представители [25]. Иными словами, вычислительное мышление без умения писать код/программы (или строить блочные диаграммы в средах визуального программирования), чтобы делать что-то полезное посредством компьютера, вряд ли привлекло бы такое большое внимание в мире или, в частности, в образовательной среде.

В заключение подробнее рассмотрим, чем же являются навык, грамотность или способность написания программ, определенная выше как основа вычислительного мышления (и рядом с ним стоящих), на примере аналогии с написанием *обычного текста*.

**В первом приближении**, написание программы прямо аналогично написанию обычного текста и соответственно имеет те же свойства. Например, *письменный* вид (вместо устного) требуется для *масштабирования*. Предложение, сказанное вслух, не фиксируется в воздухе и требует повторения каждый раз, где данное предложение нужно применить, в то время как письменный вид можно дублировать, пересылать, перечитывать и комбинировать с другими неограниченное количество раз. Таким образом, можно сказать, что написание программы

требуется для воспроизведения смыслов, заложенных в ней, неограниченное количество раз.

**Во втором приближении**, написание программ, как и текста, требует знания языка и его грамматики. Естественный язык (например, русский или английский) — это система ассоциаций слов (идей) с ситуациями их применения. Например, когда мы видим небо, мы быстро понимаем, что это «небо». Мы увидели человека, который бежит, значит, мы понимаем, что это «человек» и что он «бежит». Несмотря на то, что в языках есть части речи, такие как глагол, существительное, прилагательное и т. д., в конечном итоге *все слова* сводятся к некоторым категориям, паттернам или *функциям*, которые несут те самые «смыслы», что мы хотим заложить в наши текст, речь или программу.

Грамматика же тут нужна для *быстрой коммуникации* или снижения когнитивных нагрузок *во время* коммуникации. Например, предложение «ждет у человек который окна» читается с трудом и оставляет неуверенность в том, что предложение было понято верно. Соответственно, грамматика создаёт некий «стандарт», по которому мы соглашаемся о порядке слов, правилах правописания и словосочетания (или синтаксисе), чтобы кодировать и декодировать информацию быстро и с наименьшими смысловыми потерями. Таким образом, можно сказать, что написание программы требует наличие языка и его грамматики для кодирования/декодирования смыслов.

**В третьем приближении**, написание программ или текста подразумевает наличие некоторого «агента», для которого данный текст/программа предназначена. В случае обычного текста это могут быть сам автор или люди, понимающие язык, на котором текст был записан. В случае программ ситуация точно такая же. Таким агентом могут являться сам автор программы (если он хорошо помнит семантику языка, чтобы выполнить программу в уме) или *другая* программа, которая будет его *вычислительным агентом*, платформой или интерпретатором (например, компилятор для исходного кода некоторого языка программирования или виртуальная машина/операционная система для финальной скомпилированной программы). Наличие агента является очевидным, но в тоже время и ключевым. Мы можем знать сколько угодно много языков, сколь угодно сложных, и уметь писать сколь угодно сложные программы, однако без стороны, которая

примет, скомпилирует и вычислит текст программы, весь смысл наличия вычислительного мышления становится под вопросом.

Таким образом, (полная версия) **написание программы – это текстовое кодирование смыслов с помощью языка программирования для их передачи и декодирования посредством вычислительного агента (человека или машины)**. Рисунок 3 суммирует ключевые компоненты вычислительного мышления, описанные выше.

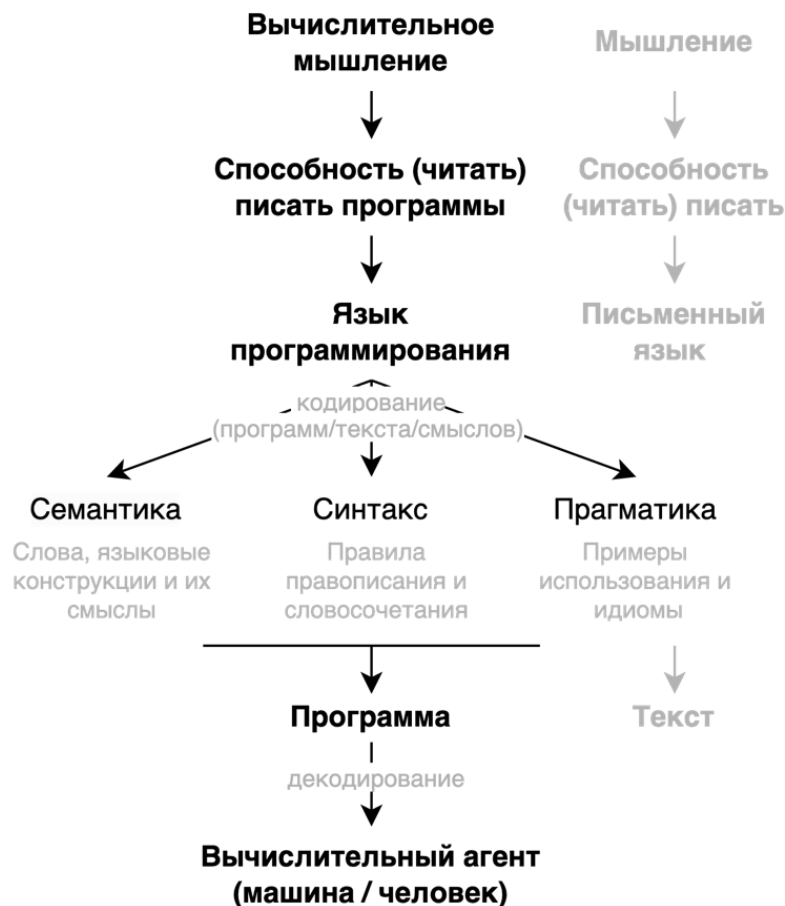


Рис. 3. Ключевые компоненты вычислительного мышления

На Рисунке 3 мы можем увидеть, что вычислительное мышление может быть рассмотрено как некая специализация обычного. Соответственно, мы можем заключить, что все те аспекты, которые затрагивают обычное мышление или использование естественных языков, легко переносимы (и применимы) в исполь-

зовании вычислительного мышления и его главного инструмента – языков программирования.

### **ЗАКЛЮЧЕНИЕ**

Мы рассмотрели смысл, значение и место «вычислительного мышления» в тренде современного цифрового образования и популяризации компьютерных наук. Мы определили, что вычислительное мышление есть ничто иное, как навык написания программ в терминах некоего языка программирования, схоже тому, как мы пишем текст в терминах естественного языка. В связи с невероятной популярностью данного понятия, всеобъемлющих определений и тренда «вычислительное мышление как вторая грамотность», мы рассмотрели простую аналогию написания программ с написанием обычного текста.

Мы считаем, что данная аналогия может послужить намного более надежным, понятным и легко применяемым в практике путеводителем разнообразных смыслов вычислительного мышления (и его основного инструмента – языка программирования), чем абстрактные определения с перечислением аспектов, которые пересекаются с друг другом, другими предметами и общими когнитивными навыками.

### **СПИСОК ЛИТЕРАТУРЫ**

1. *Goshtasbpour F.* JIME Virtual Special Collection – 2012 to 2022: The Decade of the MOOC: 1. Ubiquity Press, 2022. Vol. 2022, No. 1. P. 1.`
2. Capturing the Hype: Year of the MOOC Timeline Explained [Electronic resource] // The Report by Class Central. 2020.  
URL: <https://www.classcentral.com/report/mooc-hype-year-1/>.
3. Introducing the Coursera Impact Report 2021 [Electronic resource] // Coursera Blog. 2021. URL: <https://blog.coursera.org/coursera-impact-report-2021/>.
4. *Tekdal M.* Trends and development in research on computational thinking // Educ Inf Technol. 2021. Vol. 26, No. 5. P. 6499–6529.
5. Global Skills Reports | Coursera [Electronic resource] // Coursera’s Skills Reports. URL: <https://www.coursera.org/skills-reports/global>.
6. ChatGPT. What are the main areas digital education teaching? What includes digital literacy and computational literacy? 2023.

7. European Commission. Joint Research Centre. Reviewing computational thinking in compulsory education: state of play and practices from computing education. LU: Publications Office, 2022.

8. (Computational Thinking) The international perspective [Electronic resource]. URL: <https://www.computingschool.org.uk/resource-library/2012/september/the-international-perspective>.

9. Федеральные государственные образовательные стандарты (ФГОС) [Electronic resource] // ФГОС. URL: <https://fgos.ru/>.

10. *Selby C. C. et al.* Computational Thinking: The Developing Definition // University of Southampton (E-prints). 2013.

11. *Denning P. J., Tedre M.* Computational Thinking: A Disciplinary Perspective // Informatics in Education. 2021.

12. *Tedre M., Denning P. J.* The long quest for computational thinking // Proceedings of the 16th Koli Calling International Conference on Computing Education Research. Koli Finland: ACM, 2016. P. 120–129.

13. European Commission. Joint Research Centre. Developing computational thinking in compulsory education: implications for policy and practice. LU: Publications Office, 2016.

14. Computing in the national curriculum - a guide for primary teachers [Electronic resource]. URL: <https://www.computingschool.org.uk/resource-library/2014/september/computing-in-the-national-curriculum-a-guide-for-primary-teachers>.

15. CAS computational thinking – A Guide for teachers [Electronic resource]. URL: <https://www.computingschool.org.uk/resource-library/2014/june/cas-computational-thinking-a-guide-for-teachers>.

16. K–12 Computer Science Framework [Electronic resource] // k12cs.org. URL: <http://k12cs.org>.

17. K–12 Computer Science Framework | Statements of Support [Electronic resource] // k12cs.org. URL: <https://k12cs.org/statements-of-support/>.

18. Shut down or restart? | Royal Society [Electronic resource]. URL: <https://royalsociety.org/topics-policy/projects/computing-in-schools/report/>.

19. *Shute V. J., Sun C., Asbell-Clarke J.* Demystifying computational thinking // Educational Research Review. 2017. Vol. 22. P. 142–158.

20. Barr V., Stephenson C. Bringing computational thinking to K-12: what is involved and what is the role of the computer science education community? // ACM Inroads. 2011. Vol. 2, No. 1. P. 48–54.

21. Fagerlund J. et al. Computational thinking in programming with Scratch in primary schools: A systematic review // Computer Applications in Engineering Education. 2021. Vol. 29, No. 1. P. 12–28.

22. Best Jobs in America | Glassdoor [Electronic resource].  
URL: [https://www.glassdoor.com/List/Best-Jobs-in-America-LST\\_KQ0,20.htm](https://www.glassdoor.com/List/Best-Jobs-in-America-LST_KQ0,20.htm).

23. Ершов А. П. Программирование – вторая грамотность | Архив академика А.П. Ершова [Electronic resource].  
URL: [https://web.archive.org/web/20210429142621/http://ershov.iis.nsk.su/ru/second\\_literacy/pred](https://web.archive.org/web/20210429142621/http://ershov.iis.nsk.su/ru/second_literacy/pred).

24. Barr D., Harrison J., Conery L. Computational Thinking: A Digital Age Skill for Everyone // Learning & Leading with Technology. International Society for Technology in Education, 2011. Vol. 38, No. 6. P. 20–23.

25. Quotes on computer science – What leaders and trendsetters have to say [Electronic resource] // Code.org. URL: <https://code.org/quotes>.

26. Digital Education Action Plan (2021–2027) | see COMMISSION STAFF WORKING DOCUMENT [Electronic resource].  
URL: <https://education.ec.europa.eu/node/1518>.

27. Inc T.M. MATLAB. Natick, Massachusetts, United States: The MathWorks Inc., 2022.

28. G. van Rossum (Guido). Python reference manual // Department of Computer Science [CS]. CWI, 1995. No. R 9525.

29. Resnick M. et al. Scratch: programming for all // Commun. ACM. 2009. Vol. 52, No. 11. P. 60–67.

---

## “COMPUTATIONAL THINKING” AS A MAIN COMPETENCY IN MODERN DIGITAL EDUCATION AND SOCIETY

Timur Fayzrakhmanov<sup>1</sup> [0000-0001-5013-4523]

*Institute of software development and engineering, Innopolis University,*

---

*Universitetskaya St, 1, Innopolis, 420500*

<sup>1</sup>tim.fayzrakhmanov@gmail.com

### **Abstract**

The incredible growth in the popularity of digital technologies and education has led to the emergence of “computational thinking”. The dramatic demand and popularization of this notion ignited the redesign of educational standards and the upsurge of literature on this subject. During the extensive literature review, we found that it is not always clear what computational thinking really is, what aspects it entails, and how to think about it in simple yet robust terms. In this paper, we have examined the meaning of this concept, its importance in modern digital education, and drawn an analogy between computational thinking and the skills of writing ordinary text to facilitate its reliable understanding.

**Keywords:** *computational thinking, digital education, coding, programming, computer science, STEM*

### **REFERENCES**

1. *Goshtasbpour F.* JIME Virtual Special Collection – 2012 to 2022: The Decade of the MOOC: 1. Ubiquity Press, 2022. Vol. 2022, No. 1. P. 1.
2. Capturing the Hype: Year of the MOOC Timeline Explained [Electronic resource] // The Report by Class Central. 2020.  
URL: <https://www.classcentral.com/report/mooc-hype-year-1/>.
3. Introducing the Coursera Impact Report 2021 [Electronic resource] // Coursera Blog. 2021. URL: <https://blog.coursera.org/coursera-impact-report-2021/>.
4. *Tekdal M.* Trends and development in research on computational thinking // Educ Inf Technol. 2021. Vol. 26, No. 5. P. 6499–6529.
5. Global Skills Reports | Coursera [Electronic resource] // Coursera’s Skills Reports. URL: <https://www.coursera.org/skills-reports/global>.
6. ChatGPT. What are the main areas digital education teaching? What includes digital literacy and computational literacy? 2023.
7. European Commission. Joint Research Centre. Reviewing computational thinking in compulsory education: state of play and practices from computing education. LU: Publications Office, 2022.

8. (Computational Thinking) The international perspective [Electronic resource]. URL: <https://www.computingschool.org.uk/resource-library/2012/september/the-international-perspective>.

9. Federal State Educational Standards (FSES) [Electronic resource] // FGOS. URL: <https://fgos.ru/>.

10. *Selby C. C. et al.* Computational Thinking: The Developing Definition // University of Southampton (E-prints). 2013.

11. *Denning P. J., Tedre M.* Computational Thinking: A Disciplinary Perspective // Informatics in Education. 2021.

12. *Tedre M., Denning P. J.* The long quest for computational thinking // Proceedings of the 16th Koli Calling International Conference on Computing Education Research. Koli Finland: ACM, 2016. P. 120–129.

13. European Commission. Joint Research Centre. Developing computational thinking in compulsory education: implications for policy and practice. LU: Publications Office, 2016.

14. Computing in the national curriculum - a guide for primary teachers [Electronic resource]. URL: <https://www.computingschool.org.uk/resource-library/2014/september/computing-in-the-national-curriculum-a-guide-for-primary-teachers>.

15. CAS computational thinking – A Guide for teachers [Electronic resource]. URL: <https://www.computingschool.org.uk/resource-library/2014/june/cas-computational-thinking-a-guide-for-teachers>.

16. K–12 Computer Science Framework [Electronic resource] // k12cs.org. URL: <http://k12cs.org>.

17. K–12 Computer Science Framework | Statements of Support [Electronic resource] // k12cs.org. URL: <https://k12cs.org/statements-of-support/>.

18. Shut down or restart? | Royal Society [Electronic resource]. URL: <https://royalsociety.org/topics-policy/projects/computing-in-schools/report/>.

19. *Shute V. J., Sun C., Asbell-Clarke J.* Demystifying computational thinking // Educational Research Review. 2017. Vol. 22. P. 142–158.

20. *Barr V., Stephenson C.* Bringing computational thinking to K-12: what is involved and what is the role of the computer science education community? // ACM Inroads. 2011. Vol. 2, No. 1. P. 48–54.

21. *Fagerlund J. et al.* Computational thinking in programming with Scratch in



primary schools: A systematic review // *Computer Applications in Engineering Education*. 2021. Vol. 29, No. 1. P. 12–28.

22. Best Jobs in America | Glassdoor [Electronic resource].

URL: [https://www.glassdoor.com/List/Best-Jobs-in-America-LST\\_KQ0,20.htm](https://www.glassdoor.com/List/Best-Jobs-in-America-LST_KQ0,20.htm).

23. *Ershov A. P.* Programming is the second literacy | Academician A.P. Ershov Archive [Electronic resource].

URL: [https://web.archive.org/web/20210429142621/http://ershov.iis.nsk.su/ru/second\\_literacy/pred](https://web.archive.org/web/20210429142621/http://ershov.iis.nsk.su/ru/second_literacy/pred).

24. *Barr D., Harrison J., Conery L.* Computational Thinking: A Digital Age Skill for Everyone // *Learning & Leading with Technology*. International Society for Technology in Education, 2011. Vol. 38, No. 6. P. 20–23.

25. Quotes on computer science – What leaders and trendsetters have to say [Electronic resource] // Code.org. URL: <https://code.org/quotes>.

26. Digital Education Action Plan (2021–2027) | see COMMISSION STAFF WORKING DOCUMENT [Electronic resource].

URL: <https://education.ec.europa.eu/node/1518>.

27. *Inc T. M.* MATLAB. Natick, Massachusetts, United States: The MathWorks Inc., 2022.

28. *G. van Rossum (Guido)*. Python reference manual // Department of Computer Science [CS]. CWI, 1995. No. R 9525.

29. *Resnick M. et al.* Scratch: programming for all // *Commun. ACM*. 2009. Vol. 52, No. 11. P. 60–67

---

## СВЕДЕНИЯ ОБ АВТОРЕ



**ФАЙЗРАХМАНОВ Тимур Расимович** – разработчик ПО, аспирант, исследователь. Сфера научных интересов: языки программирования, представление и обработка знаний, повторное использование знаний, формализация, обобщённое моделирование систем, цифровые библиотеки, графика, среды быстрого прототипирования, блочный подход.

**Timur Rasimovich FAYZRACHMANOV** – software developer, PhD student, researcher. Research interests: computational languages, knowledge representation and processing, knowledge reuse, formalization, generic systems modeling, digital libraries, graphics, low-code development platforms, block-based approach.

email: [tim.fayzrakhmanov@gmail.com](mailto:tim.fayzrakhmanov@gmail.com)

ORCID: 0000-0001-5013-4523

*Материал поступил в редакцию 3 марта 2023 года*