ДОКЛАДЫ РОССИЙСКОЙ АКАДЕМИИ НАУК. МАТЕМАТИКА, ИНФОРМАТИКА, ПРОЦЕССЫ УПРАВЛЕНИЯ, 2023, том 513, с. 27–43

МАТЕМАТИКА

НОВЫЙ КУРС "АЛГЕБРА + ИНФОРМАТИКА": КАКИМИ ДОЛЖНЫ БЫТЬ ЕГО РЕЗУЛЬТАТЫ И С ЧЕГО ОН ДОЛЖЕН НАЧИНАТЬСЯ

© 2023 г. **А. В. Боровик[1,*], В. В. Кондратьев[1,**]**

Представлено академиком РАН А.Л. Семеновым
Поступило 22.11.2022 г.
После доработки 26.12.2022 г.
Принято к публикации 01.02.2023 г.

Лозунг "Программирование – это вторая грамотность" был сформулирован более 40 лет назад [13], но так и не воплотился в жизнь. Статья развивает и детализирует этот старый лозунг, предлагая объединить основное математическое образование в школах с образованием в области информатики/программирования. Необходимым условием для такого объединения является глубокая структурная реформа школьного математического образования. Мы не говорим об адаптации математики 20-го века к 21-му веку – как это описано в [10, 19], мы имеем в виду математическое образование 21-го века для математики 21-го века. Насколько нам известно, эта работа является первой попыткой начать надлежащее технико-экономическое обоснование этой реформы. Объем статьи не позволяет нам затронуть деликатные социально-политические (и финансовые) стороны реформы, мы рассматриваем только общие учебные и дидактические аспекты и возможные направления реформы. В частности, мы указываем подходы к разработке предметно-ориентированного языка (DSL) в качестве основы для всех аспектов нового курса.

DOI: 10.31857/S2686954323700297, **EDN:** CLGJVJ

1. ВВЕДЕНИЕ

В этом тексте обсуждаются некоторые аспекты учебной программы для предлагаемой замены традиционного школьного курса алгебры новым, который объединяет алгебру с информатикой/программированием. Подходя к этому вопросу с точки зрения основных принципов управления проектами, мы фокусируемся на двух моментах, с которых начинается разработка любого проекта:

(1) Каковы наши цели? Чего мы хотим достичь?

(2) Какова наша исходная позиция?

В контексте разработки учебных программ естественным образом возникают два вопроса:

Вопрос 1: Каковы цели изучения предметной области "Математика и информатика"? Что выпускники школ должны уметь делать по окончании курса?

Вопрос 2: Как новый курс будет соотноситься с арифметикой начальной школы?

При работе над проектами, направленными на работу с детьми в течение значительного периода времени (в случае учебной программы по алгебре это, возможно, 7 или более лет), пункт (2) и вопрос 2 приобретают большее значение, чем в большинстве промышленных проектов: ребенок должен быть в центре внимания, проект продолжается, и ребенок растет, все предлагаемые задания должны развиваться вместе с ним.

Оба пункта должны быть четко описаны и проиллюстрированы примерами задач такого рода, которые ученик, как предполагается, сможет решить (самостоятельно, без существенной помощи со стороны учителей) в начале и в конце своего обучения. Только тогда мы сможем приступить к написанию учебной программы, работая в направлении от ответа на вопрос 1 к ответу на вопрос 2, на каждом этапе включая только тот материал, который определенно необходим для дальнейшего продвижения на более высокий уровень.

1.1. Что касается вопроса 1, то мы формулируем следующие рекомендации

1.1.1. Самая важная проблема 21 века – это взаимоотношения между людьми и компьютерами.

Поэтому мы ожидаем, что по окончании курса учащийся должен

[1] *University of Manchester, Manchester, United Kingdom*
**E-mail: alexandre@borovik.net*
***E-mail: kondratjew239@gmail.com*

• иметь представление о возможностях и опасностях, связанных с насыщением каждого аспекта экономики, политики, военных действий, средств массовой информации, развлечений, повседневной жизни компьютерами и компьютерными системами, которые принимают решения от имени людей;

• адаптируясь к изменениям в мире, понимать, какие вопросы можно и нужно задать – и кому;

• при необходимости быть готовым изучить и использовать более технические и профессиональные аспекты информатики и программирования.

Это можно сравнить с математическими навыками, которых ожидали от выпускников средних школ в Европе в первой половине 20-го века: они должны были в достаточной степени овладеть алгеброй и тригонометрией для последующей подготовки, в случае необходимости, в качестве артиллерийских офицеров или пилотов авиации – это была эпоха массовой мобилизации.

1.1.2. Существует простой и в то же время сложный критерий овладения учащимся необходимыми навыками.

• Начиная с относительно ранней стадии курса, ответом учащегося на арифметическую или алгебраическую задачу (включая задачи, описываемые как задачи “из реальной жизни”) должен быть код, разработанный самим учащимся – без использования стандартных пакетов, таких как Mathematica– который

– решает *все* задачи одного типа;

– помогает проверить, проанализировать и обобщить решение.

Задачи, требующие *доказательства* некоторого утверждения, естественно, исключаются из этого требования – но компьютерные эксперименты следует приветствовать при построении контрпримеров и формулировании гипотез.

1.1.3. Требования к компьютерному языку, который будет использоваться в курсе, довольно высокие.

• Если более продвинутые версии программы используются на более поздних этапах обучения, она должна быть обратно совместима на всех этапах школьной алгебры – чтобы обеспечить систематический пересмотр, повторное использование и переосмысление ранее изученного материала, решенных задач и написанных кодов.

• Небольшой “стартовый” фрагмент должен обеспечить простую, безопасную и увлекательную игровую площадку для детей на первых этапах изучения алгебры.

• Язык должен включать в себя достаточно богатые элементы, обычно встречающиеся в языках высокого уровня, и обеспечивать эффективную подготовку к изучению профессиональных отраслевых языков.

• Все вычисления должны быть символьными, с использованием дробей и радикалов; если результат какого-либо промежуточного вычисления равен $\frac{1-\sqrt{3}}{2}$, он остается таким и используется при последующих расчетах. Конечно, они должны включать константы e и π. Никаких округлений, никаких вычислений с плавающей запятой.[1]

Это потребует разработки индивидуального *предметно-ориентированного языка*[2] (ПОЯ) для использования в курсе. Скорее всего, существующие языки общего назначения не подходят для этой роли.

Этот текст представляет собой первые шаги в разработке спецификаций для ПОЯ

1.1.4. Содержание курса алгебры должно измениться и отражать требования программирования и включать, например, булеву алгебру, элементы теории чисел, арифметику остатков и конечные поля. Формирование учебной программы по алгебре должно идти в ногу с развитием DSL.

1.1.5. Другие разделы математики, например геометрия, механика и статистика, нуждаются в отдельном обсуждении и здесь не затрагиваются. Кроме того, межпредметные связи с физикой заслуживают самого серьезного внимания.

1.2. Что касается вопроса 2, то наши основные предложения заключаются в следующем:

• Введение “вычислительного” мышления в алгебру должно быть подготовлено развитием алгоритмического мышления в арифметике.

• “Метод вопросов” для текстовых задач по арифметике предлагается как эффективный инструмент для овладения элементарным уровнем алгоритмического мышления – и это было сделано в арифметике до перехода к алгебре.

• Использование простого языка (который все еще нуждается в разработке) необходимо с перво-

---

[1] И ответ на вопрос “Найти остаток при делении многочлена $x^{2023}+1$ на x^2-4” должен быть $2^{2022}\cdot x+1$, без каких-либо попыток вывести на экран десятичный вид числа 2^{2022}.

[2] Из Википедии: “Предметно-ориентированный язык (англ. domain-specific language, DSL – “язык, специфический для предметной области”) – компьютерный язык, специализированный для конкретной области применения (в противоположность языку общего назначения, применимому к широкому спектру областей и не учитывающему особенности конкретных сфер знаний). Построение такого языка и/или его структура данных отражают специфику решаемых с его помощью задач.”. Известные примеры включают html и LaTeX. Несколько книг: [3, 18, 24]. Мы используем Википедию в качестве основного источника информации, потому что ни мы, ни наши предполагаемые читатели не являемся специалистами по информатике или программистами.

го дня изучения алгебры, в частности, для формулирования алгоритмов, разработанных с помощью арифметического “метода вопросов”, в компактной форме, пригодной для преобразования в компьютерный код.

• Типизированные переменные и вывод типов необходимы для языка, который будет использоваться в курсе, и к их введению необходимо подготовиться, освоив “именованные числа” в арифметике. Необходимо пояснить, что

именованные числа — это ***не*** *числа, замененные именами и символами*,

это числа единиц разного рода: 10 яблок — это не то же самое, что 10 человек и не то же самое, что 10 километров, и вы не добавляете 10 яблок и 10 человек, см. рис. 6.

• Однако существует концептуальный разрыв между

– формулировкой алгоритма и его реализацией в коде и

– представлением ответа в виде замкнутой алгебраической формулы.

Следовательно, учащемуся все равно потребуется освоить алгебраические преобразования, что концептуально связано с использованием вычислений путем вызова по имени, в отличие от вычисления путем вызова по значению, используемого в прямой реализации арифметического алгоритма.

• Большое внимание необходимо уделить *арифметике именованных чисел*. В частности, учащийся должен усвоить, что он может свободно вводить в свои решения промежуточные параметры, а их значения, если названия этих параметров (соответственно, типы переменных) являются чужеродными, не фигурируют ни в данных (условиях) задачи, ни в предполагаемом ответе. Эти дополнительные параметры исчезнут в процессе решения. Этот подход неизбежно требует, чтобы учащийся размышлял о своей работе, анализировал ее, развивал метамышление (т.е. думал о мышлении) — и может быть реализован только в том случае, если учащийся чувствует себя психологически комфортно и чувствует, что полностью контролирует проблему, ее решение, и компьютер.

Достижение всего этого является огромной задачей. Мы бы не советовали относиться к этому легкомысленно.

1.3. Предупреждение о масштабе и стоимости проекта

Первый автор имеет опыт работы над крупными и сложными исследовательскими проектами, см., например, [1], книгу объемом более 550 страниц, которую он написал в соавторстве и которая содержала доказательство одной теоремы — доказательство разрабатывалось около 15 лет и использовало критически важный вклад из неформальной команды из 10 человек. Ближе к теме “алгебра + информатика для школы” находится его работа по вычислительной символической логике, где шаг за шагом была разработана его совместная статья с Şükrü Yalçınkaya [11]. Шаг за шагом, по крайней мере, в течение 10 лет, и прогресс в решающей степени зависел от систематических компьютерных экспериментов.

Основываясь на своей оценке этого опыта, он вполне уверен, что разработка программной системы, поддерживающей предлагаемый курс и используемой в масштабах страны, реалистична, но может легко обойтись в несколько миллионов долларов, потребовать работы большой междисциплинарной команды экспертов и около 10 лет экспериментов в реальных условиях. Эта оценка не включает в себя затраты на переподготовку учителей и время, необходимое для этого.

Мы настоятельно призываем читателя вспомнить слова Г.Л. Менкена:

На каждую сложную проблему есть ответ, ясный, простой и неверный.

Мы предупреждаем:

Остерегайтесь торговцев “чудодейственными средствами”, пытающихся продать дешевые и простые рецепты оживления школьной математики с помощью “цифровых технологий”!

Мы надеемся, что наша работа может быть использована в качестве противоядия от их обещаний.

***

Теперь перейдем к более содержательному обсуждению вопроса 2:

Как предложенный курс будет связан с арифметикой начальной школы?

2. НО ЧТО ТАКОЕ АРИФМЕТИКА?

Школьная арифметика состоит из двух основных частей.

• Формальные письменные методы для арифметических действий с десятичными числами и дробями [15], т.е. письменные вычисления, основанные на рекурсивных алгоритмах.

• Решение текстовых задач.

Как показано в [6], “текстовые задачи” арифметики предполагают идентификацию математических структур и отношений реального мира и отображение их на более формализованные структуры и отношения арифметики, или, в формулировке Игоря Арнольда 1946 г., когда слова "структура" и "отношение" еще не были в моде [26],

Эти примеры наглядно показывают, что обучение арифметике включает в качестве одного из основных элементов воспитание умения ориентироваться в различных по своей конкретной природе взаимоотношениях между величинами.

Еще более важной является характеристика арифметики:

Самый метод "арифметического решения задачи" отличается от алгебраических приемов в первую очередь тем, что на всех стадиях рассуждения все сопоставления и производимые действия допускают совершенно наглядное и конкретное, осмысленное в области тех величин, о которых идет речь, истолкование.

Поэтому фазовый переход между арифметикой и алгеброй является важным поворотным моментом в изучении математики.

3. МОСТ МЕЖДУ АРИФМЕТИКОЙ И АЛГЕБРОЙ

Вот наше принципиальное утверждение:

Слияние школьной математики с информатикой должно начаться одновременно с фазовым переходом от арифметики к алгебре и должно основываться на тех аспектах арифметики, которые на самом деле относятся к информатике, но обычно не признаются таковыми.

Например, арифметика уже содержит

Абстракции: Понятие *число* уже является огромной абстракцией.

Алгоритмы: Прежде всего, деление в столбик и т.д. – это алгоритмы. Но они даются детям просто как правила, которым нужно следовать. Однако "метод вопросов" арифметики прошлых лет (ныне почти повсеместно забытый (одним из его редких четких изложений является [28]) позволяет детям разрабатывать свои собственные алгоритмы, которые решают многие типы арифметических задач.

Рекурсию: Пресловутое деление в столбик – это рекурсивный алгоритм, как и умножение в столбик, а также сложение и вычитание десятичных дробей. Алгоритм Евклида для нахождения наибольшего общего делителя двух целых чисел также является рекурсивным.

Типизированные переменные: "Именованные числа" арифметики идеально соответствуют типизированным переменным программирования.

Овеществление: Это понятие из информатики, но его игрушечная версия появляется в арифметике как введение промежуточных параметров ("полезные числа" – более подходящий термин для использования детьми), см. раздел 5.5. Или как "выдуманные единицы измерения" – см. [12, Разделы 3.1 и 3.3] и [9, Solutions and Notes to Problems 89 and 90] (в последнем случае они называются "скрытыми параметрами") для обсуждения и исторических примеров.

В информатике овеществление является широко используемым понятием. Вот как Википедия определяет овеществление:

Овеществление – это процесс, посредством которого абстрактная идея о компьютерной программе преобразуется в явную модель данных или другой объект, созданный на языке программирования. Вычислимый/адресуемый объект – ресурс – создается в системе в качестве прокси-сервера для невычислимого/адресуемого объекта.[3]

В школьной алгебре пресловутый x как обозначение неизвестного является типичным примером овеществления: это обозначение для числа, которое нам не дано, но, тем не менее, позволяет производить с ним алгебраические преобразования.

Остальная часть этого текста посвящена "методу вопросов" и алгоритмам перехода от арифметики к алгебре.

Слияние алгебры и информатики должно быть подготовлено развитием алгоритмического мышления в арифметике.

В школьной арифметике прошлых лет существовал подход к эффективному развитию алгоритмического мышления: так называемый "метод вопросов" для решения текстовых задач[4].

Мы утверждаем, что это могло бы послужить мостом между арифметикой и алгеброй, особенно если алгебра объединена с информатикой/программированием.

Мы попытаемся продемонстрировать, что метод вопросов позволяет достичь ряда дидактических целей:

• Акцент делается на выявлении и анализе структур и отношений реального мира и их представлении в математических терминах.

• Систематический анализ данных, приведенных в задаче.

[3] https://en.wikipedia.org/wiki/Reification_(computer_science).

[4] Метод вопросов более подробно обсуждается в [6], разделах 3 и 4.

• Соответствие между именованными числами в алгебре и типизированными переменными при написании кода.

• Ответ на арифметическую или алгебраическую задачу, представляющий собой исполняемый компьютерный код, который

– решает *все* проблемы одного типа;

– помогает проверить и проанализировать решение;

– использует вычисление путем вызова по имени и путем вызова по значению.

Если эти цели будут достигнуты, мы сможем утверждать, что

Ребенок контролирует проблему, ее решение и работу компьютера [5].

И мы должны достичь этого – в противном случае дальнейший прогресс невозможен.

Мы не обсуждаем здесь, почему слияние алгебры с информатикой может быть желательным в школьном образовании – это обсуждалось в [10] и [19].

4. ТИПОВАЯ ЗАДАЧА ИЗ АРИФМЕТИКИ: МИСКА С ВИШНЯМИ

Рассмотрим задачу.[5] Конечно, в школе дети должны начинать с гораздо более простых задач, состоящих из 1 или 2 шагов; задачи, описанные ниже, должны решаться на более позднем этапе. Мы используем этот пример, потому что он дает более широкий обзор границы между арифметикой и алгеброй.

Попробуем подойти к этой проблеме шаг за шагом в соответствии с методологией, которую мы только что сформулировали.

5. АНАЛИЗ УСЛОВИЯ

5.1. Разбор проблемы

Давайте теперь зададим самый важный вопрос:

Что дано и что нужно найти?

Правильный разбор задачи заключается в следующем:

Условие:	Боб наполняет миску за `8 минут`, Алиса за `24 минуты`. Теперь они работают вместе.
Вопрос:	За какое время они наполнят миску?

Мы будем называть вопрос в задаче, который мы отделили от данных, *целевой вопрос*.

5.2. Метод вопросов и вопросы

Мы приведем пример решения задачи методом вопросов. В 1960-х годах в России предполагалось, что учащиеся должны были освоить его к 11 или 12 годам и быть в состоянии составить для текстовой задачи ряд вопросов, результатом которых был ответ на целевой вопрос. Это также можно было бы описать как формулировку многоэтапного решения проблемы как совокупность решений гораздо более простых одноэтапных задач.

Одна из многих возможных последовательностей пошаговых вопросов для задачи о миске с вишнями показана в разделе 6.1. Сейчас мы попытаемся объяснить, как можно получить такую последовательность вопросов.

5.3. Общие направляющие вопросы

Обучение детей поэтапному формулированию вопросов в методе вопросов заключается в развитии способности ребенка начинать свои попытки решения с “вопросов”, задавая себе соответствующие самостоятельные вопросы. Вопросы (в русской педагогической литературе они назывались *вспомогательные вопросы*, но в Англии слова “*вспомогательный вопрос*” предполагают, что вопрос задается учителем, чтобы помочь испытывающему трудности ученику).

Общие вопросы (или мета-вопросы), рекомендуемые методом вопросов, могут быть ориентированы на данные [25, 27–29], например,

• Какие вопросы можно задать по поводу этих данных?[6]

• Что мы можем узнать из этих данных?

• И более конкретный вопрос: как математически выразить “совместную работу”?

Белошистая [28, p. 303] называет этот подход *синтетическим*. Интересно сравнить анализ Белошистой с анализом Gerofsky [17].

Есть также более сложные направляющие вопросы. Те, что встречаются в современной лите-

[5] Проблема и картинка, иллюстрирующая ее, были взяты из онлайн-беседы Ольги Москаленко “Проблема систематических ошибок в решении текстовых задач по результатам мониторинга на платформе “Учи.ру”” на Научном семинаре по методике преподавания математики в Московском государственном университете 22 сентября 2022 г. Интересно, что в тесте только 39% шестиклассников дали правильный ответ – 6 мин.

[6] В 1960-х годах в России эти вопросы можно было задавать даже в более жесткой форме: учитель писал задачу на доске, давал минуту ученикам на размышление, затем стирал целевой вопрос и спрашивал: какие вопросы можно задать про условие?

ратуре, по-видимому, взяты непосредственно из руководства по управлению проектами [29].

Проанализируйте задачу, двигаясь в обратном направлении от цели к текущей ситуации.

Действительно, взгляните сами на вопросы синтетического подхода:

• Какие дополнительные величины было бы полезно знать для поиска ответа? Этот вопрос может привести к введению “полезного числа”, см. раздел 5.5.

• Что вам нужно сделать, чтобы найти эти дополнительные количества?

• Как они могут быть получены из данных задачи?

5.4. Ключевой момент: кому были адресованы эти вопросы?

В старой традиции “метода вопросов”,

• На первых этапах изучения метода вопросов учитель или учебник задавали направляющие вопросы **учащимся** в возрасте 11—12 лет.

• На более поздних этапах ожидалось, что **учащиеся** сформулируют направляющие вопросы **сами**, таким образом создавая **свои собственные вопросы**, направленные на решение конкретной задачи.

• Таким образом, из вопросов **учащихся** получается **алгоритм** (хотя это слово не упоминалось): для решения любой задачи такого типа, но с разными данными, можно было быстро ответить, выполнив те же шаги.

• Этот алгоритм должен быть разработан учащимся самостоятельно.

5.5. Введение промежуточного параметра (или “полезного числа”))

В процессе ответа на направляющий вопрос

Как математически выразить “совместную работу”?

после некоторого размышления или обсуждения можно было бы обнаружить, что “работать вместе” означает, в математических терминах, что

производительности, или скорости сбора вишни, *суммируются*.

Это должно немедленно вызвать следующие вопросы:

• Как измерить скорость сбора вишен?

Естественно, в *`вишнях в минуту`*!

• Сколько вишен дано в задаче?

И последний вопрос подводит к *критически важному шагу*:

Полезное число:
Предположим, что миска содержит 72 `вишни`.

Этот выбор, 72, представляется произвольным и сделан только для удобства деления на 8 и на 24, что позволяет избежать дробей при дальнейшем вычислении. Скоро мы увидим, что это вполне оправданно. Кроме того, как уже упоминалось в разделе 3, это пример *овеществления*, понятия, используемого в информатике в явной форме, но неявно присутствующего и в арифметике. Позже, в разделе 6.2, будет приведено объяснение, почему этот шаг является корректным.

6. РЕШЕНИЕ

6.1. Последовательность вопросов

Следующее решение представляет собой последовательность вопросов и ответов, сформулированных на основе анализа условия и в ответ на общие руководящие вопросы, описанные в разделе 5.

□

6.2. Исключение промежуточных параметров независимых типов

Но что произойдет с ответом на задачу, если мы изменим количество вишен? (оно не было дано, оно было выбрано нами просто для удобства)

Ничего не изменится. Данные заданы в `минутах`, получается, что ответ будет тоже в `минутах`. Дополнительный показатель является независимым типом `вишня`. Ответ не зависит от числового значения данного параметра, потому что его изменения можно объяснить как *единицу измерения*: вместо 72 вишен можно сказать 32 пары, 24 ложки или 12 кружек, и т.д. Все эти показатели описывают одинаковое количество.

Это простое наблюдение дает

Свободу выбора вспомогательных параметров — мы должны только убедиться в том, что от их выбора не зависит ответ.

6.3. Псевдокод

В буквенных обозначениях условие задачи можно переформулировать так:

Боб наполняет миску за `B минут`, Алиса за `A минут`

Работая вместе, они наполнят миску за `T минут`.

Решение методом вопросов может быть записано с помощью псевдокода:

```
input A, B                        % Время Алисы и время Боба
input C                                      % число вишен
U := C:A;  V := C:B;    % Производительности Алисы и Боба
W := U + V;    % Производительность при совместной работе
T := C:W          % Время, за которое они наполнят миску
return T;
```

Можно записать решение в 1 строчку:

```
input A, B; T:= (A*B):(A+B); return T
```

т.е.,

$$T := \frac{AB}{A+B}?$$

Это обстоятельство заслуживает дополнительного обсуждения.[7]

В элементарной алгебре написание кода было намного проще, чем разработка алгоритма. Действительно, ответ

$$T := \frac{AB}{A+B}$$

— самое короткое выражение алгоритма — является нетривиальным.

6.4. Решение в символьных переменных

Сворачивание кода, полученного методом вопросов, в компактную формулу может быть выполнено либо вручную, либо, возможно, даже автоматически; в последнем случае код должен быть встроен в систему символьной алгебры, которая автоматически упрощает алгебраические выражения. Мы считаем, что было бы желательно достичь дидактически эффективного баланса этих двух подходов.

Обозначения: A: производительность Алисы, B: производительность Боба, C: количество вишен.

Вопрос 1. Каковы производительности Алисы и Боба?

$$\frac{C}{A} \quad \text{и} \quad \frac{C}{B}$$

Вопрос 2. Какова производительность при совместной работе?

$$V := \frac{C}{A} + \frac{C}{B}$$

Вопрос 3. За какое время они наполнят миску?

$$T := \frac{C}{V} == \frac{C}{\frac{C}{A} + \frac{C}{B}} == \frac{AB}{A+B}$$

Обратите внимание, здесь используется стратегия вычисления *“вызов по имени”*.

Также обратите внимание, что переменная C исчезла из ответа – когда была проверена, потому что ее тип не зависел от типа данных. В более раннем обсуждении этого текста был интересный вопрос об этой проблеме: “Кто съел все вишни?”.

Что ж, мы можем добавить к постановке задачи кого-то по имени Кевин, который ест вишню из расчета 1 миска за K минут. Код можно было легко настроить, выдав ответ, что чаша будет наполнена вовремя $\frac{ABK}{AK + BK - AB}$, а также выдает предупреждение о том, что если

$$K \leqslant \frac{AB}{A+B},$$

миска никогда не будет наполнена. Однако некоторые коллеги совершенно справедливо предположили, что эквивалентное неравенство, характеризующее неспособность наполнить миску

[7] При обсуждении этой проблемы на собрании Ассоциации учителей математики (Великобритания) в октябре 2022 г. один из моих коллег совершенно справедливо спросил: “Мне интересно, как ученик может узнать, что $(AB)/(A + B)$ может быть решением проблемы?” И кто-то еще предложил: “Я думаю об этом как о $1/(1/A + 1/B)$, а не как об упрощенной форме”.

Рис. 1. Условие задачи.

$$\frac{1}{K} \geqslant \frac{1}{A} + \frac{1}{B}$$

было легче интерпретировать в терминах реальной жизни.

6.5. Совсем другое решение

Разумеется, методом вопросов могут быть получены и совсем другие решения.

Вопрос 1. Во сколько раз производительность Боба больше производительности Алисы?

$$\frac{24 \text{ мин}}{8 \text{ мин}} = 3 \quad \text{или} \quad \frac{A}{B}.$$

Вопрос 2. Насколько общая производительность больше производительности Алисы?

$$3 + 1 = 4 \quad \text{или} \quad \frac{A}{B} + 1.$$

Вопрос 3. За какое время они наполнят миску?

$$24 \text{ мин}: 4 = 6 \text{ мин} \quad \text{или} \quad \frac{A}{\frac{A}{B}+1} = \frac{AB}{A+B}.$$

7. ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС

Мы надеемся, что это решение “методом вопросов” может быть естественным образом преобразовано в код с помощью удобного для детей графического интерфейса . Далее следует краткое обсуждение желаемой функциональности этого интерфейса, “пользовательской версии”, в терминологии разработки программного обеспечения, написанной в соответствии с рекомендациями *User experience design* [8].

Предполагая, что учащийся вводит все данные на каком-либо вычислительном устройстве (смартфоне, планшете, ноутбуке) с установленным программным обеспечением курса, как только вводится число, графический интерфейс начинает диалог с учащимся, спрашивающим, является ли это число исходным, или полезным числом, или ответом на проблему и каков ее тип. Кроме того, в этом диалоге разрешен ряд различных неясностей – мы опускаем здесь эти детали. В простых задачах этот диалог, скорее всего, будет очень коротким – мы объясним это в следующей статье.

Графический интерфейс присваивает внутренние переменные всем числам на экране и преобразует числа в ячейки, значения которых могут быть изменены с помощью кода или отредактированы учащимся. Значения данных и полезные числа остаются доступными для редактирования учащимся, см. рис. 2. Это исполняемый код, представленный в графическом интерфейсе. Значок “Гамбургер” представляет собой всплывающее меню со множеством дополнительных функций.

Если учащийся выбирает кнопку Run в меню и нажимает на него, ячейки становятся живыми, а графический интерфейс выглядит так, как показано на рис. 3.

Учащемуся предлагается поэкспериментировать с изменением данных и вспомогательных параметров; при этом полезное число изменяется с 72 на 48, а кнопка Run нажимается, учащийся получает другой экран, рис. 4 – с абсолютно тем же ответом!

Еще более важно: должна быть возможность заменять данные и полезные цифры буквами в ячейках (или только в одной ячейке), таким образом автоматически получая символьное решение, подобное тому, которое предлагается в разделе 6.4. На рис. 5 показано, что произойдет, если время, за которое Алиса наполнит чашу, обозначено буквой A.

8. ИМЕНОВАННЫЕ ЧИСЛА

8.1. Типы именованных чисел

Слияние алгебры с информатикой требует разработки соответствующего языка предметной области, который должен поддерживать вычисления с *именованными числами*, как в ситуации, когда мы *делим* 10 яблок между 5 людьми:

$$10\,\texttt{яблок} \div 5\,\texttt{людей} = 2\,\frac{\texttt{яблока}}{\texttt{людей}}.$$

Очевидно, что нам нужен типизированный язык, в котором переменным может быть присвоен произвольный тип, скажем, `яблоки` или `люди`, где могут быть добавлены только переменные соизмеримых (или сопоставимых) типов, скажем, $1 \text{ м} + 1 \text{ см} = 101 \text{ см}$, но не яблоки и не люди

[8] User experience design, https://en.m.wikipedia.org/wiki/User_experience_design.

Рис. 2. GUI после ввода решения. Нажатие на иконку “Гамбургер” приводит к вызову меню с дополнительными функциями.

Рис. 3. Код программы оживает. Это результат работы, который можно с гордостью показать родителям

(см. рис. 6); такое ограничение может быть реализовано[9]; однако, деление `яблок` на `людей` автоматически приводит к переменной другого типа, $\frac{\texttt{яблок}}{\texttt{людей}}$. Такие типы, как единицы СИ или валюты, должны быть стандартными, встроенными в DSL—но ученик должен уметь создавать любые другие, к примеру `дукат` или `песо` декларировать в обменный курс 1 `дукат` == 5 `песо`, так,

[9] Сиротин [22], библиотека функций и объектов языка программирования Kotlin (https://kotlinlang.org/), которая позволяет работать с переменными, значения которых выражены в Международной Системе СИ, такие как метр, секунда и т.д. Также некоторые общие единицы, а именно валюты, проценты и другие.

Рис. 4. Манипуляции с “полезным числом”: 72 вишни изменится на 48 вишен — ответ не поменялся.

чтобы `дукат` и `песо` имели одинаковую стоимость.

Более сложные манипуляции с типами необходимы, когда мы раздаем 10 яблок, по 2 яблока на человека и хотим знать, сколько людей получат свои яблоки:

$$10\ \texttt{яблок} \div 2\ \frac{\texttt{яблок}}{\texttt{людей}} = 5\ \texttt{людей}.$$

8.2. Немного истории

На самом деле, это было хорошо понятно отцу (символьной) алгебры, Франсуа Виету, который в 1591 г. написал в своем

Введении в аналитическое искусство [23, p. 16] то, что

Если одна величина делится на другую, [частное] неоднородно по отношению к первой... Большая часть туманности и неясности старого анализа объясняется тем, что они не обращали внимания на эти [правила].

Увы, эти слова остаются верными и в XXI веке.

Это грязный секрет школьной арифметики: она находится в тесной связи с арифметикой *типов*, которая тщательно скрывается от детей (и от многих учителей). Однако это хорошо известно в физике, где типы называются *размерностями*, и где *анализ размерностей* является простым, но мощным методом понимания соотношений между величинами и величинами другой природы, который, в частности, дает возможность производить быстрые, часто даже *устные* оценки величин. Например, [4, Section 8.4] содержит одностраничный вывод легендарного закона Колмогорова “5/3” для энергетического спектра турбулентного движения газа или жидкости.

8.3. Элементарный пример из физики

Давайте рассмотрим одно из более простых применений анализа размерностей в физике.

Галилео Галилей заметил, что период маятника не зависит от амплитуды его колебаний.

Таким образом, период T (типа `секунда`) маятника зависит от его длины L (типа `метр`) и ускорения свободного падения g (типа $\frac{\texttt{метр}}{\texttt{секунда}^2}$).

Единственная формула, согласованная с типом, которая может быть получена из этого, — это

$$T = C\sqrt{\frac{L}{g}},$$

где C — безразмерная константа. Даже если мы не знаем значения C (для этого требуются более тон-

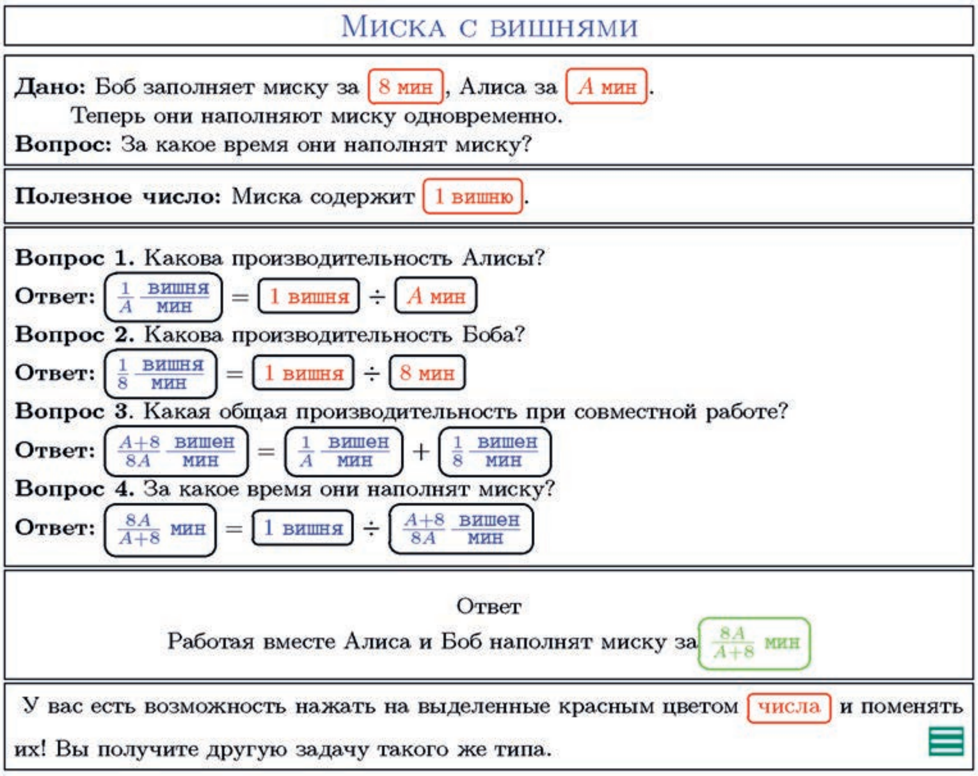

Рис. 5. Замена буквенного обозначения исходным.

кие соображения), мы можем сделать очень интересные выводы.

Например, ходьба на двух ногах может быть смоделирована как последовательность падений, в которых нога (длиной L), которая движется вперед, ведет себя как маятник с периодом T.

Следовательно, скорость V пропорциональна $\frac{L}{T}$ и

$$V \sim \frac{L}{T} \sim \frac{L}{\sqrt{\frac{L}{g}}} \sim \sqrt{gL}.$$

Следовательно, ходьба на ходулях увеличивает скорость (любой, кто пробовал ходить на ходулях, знает это), рис. 7, при ходьбе по Луне это на $\sqrt{6}$ медленнее, чем на Земле (поскольку ускорение силы тяжести на Луне составляет около $\frac{1}{6}g$).

Размерностный анализ должен быть частью школьного курса физики. Это просто, красиво и может привести к откровениям.

8.4. Анализ типов в арифметике

Давайте применим анализ типов к старой классической задаче, являющейся частью математического фольклора:

Рис. 6. Первый закон арифметики: нельзя складывать фрукты и людей. Джузеппе Арчимбольдо, *Осень*. С сайта Википедия. Общественное достояние.

Рис. 7. Ходьба на ходулях увеличивает скорость передвижения. *Обитатели Ландов*. Жан-Луи Жинтрак (1808–1886). Источник: Википедия. Общественное достояние.

Кролики и куры. У Мэри есть домашние животные, несколько кроликов и кур. Всего у них 12 голов и 32 ноги.

Сколько кроликов и кур у Мэри?

Эта задача лучше подходит для анализа типов, поскольку, очевидно, включает в себя более одного типа данных.

Прежде всего, мы должны тщательно присвоить типы каждому фрагменту (исходным данным) количественных данных, которые нам даны или которые мы знаем из нашего жизненного опыта.

Конечно, нам даны данные типов `leg` и `head`. Итак, мы должны ввести переменные LEGS и HEADS этих типов соответственно, и нам будут даны их значения.

Поскольку у каждого кролика и каждой курицы ровно по одной голове, у нас есть переменные RABBITS и CHICKEN, которые можно с уверенностью отнести к типу `head`.

Возможно, нам стоит обратиться к нашему жизненному опыту: сравнительной анатомии кроликов и кур. Это означает

$$\text{RabbitAnatomy} = 4\,\frac{\texttt{leg}}{\texttt{head}}$$

and

$$\text{ChickenAnatomy} = 2\,\frac{\texttt{leg}}{\texttt{head}}$$

Все данные, которые у нас имеются

$$\text{Heads} = 12\ \texttt{head}$$

$$\text{Legs} = 32\ \texttt{leg}$$

$$\text{RabbitAnatomy} = 4\,\frac{\texttt{leg}}{\texttt{head}}$$

$$\text{ChickenAnatomy} = 2\,\frac{\texttt{leg}}{\texttt{head}}$$

Мы должны понимать, что для ребенка такой анализ данных является сложной задачей, но именно этот навык,

способность видеть математические структуры и отношения в реальном мире

– отсутствует в основном математическом образовании. Но этому нужно учить, причем систематически.

Принимая во внимание совет Виета и естественные ограничения арифметики типов, у нас есть *очень небольшой выбор* разумных арифметических операций между этими значениями, мы подчеркиваем, *у нас очень небольшой выбор*. Одним из *очень немногих* действий, которые мы можем попробовать, является вычисление

$$\text{Heads} \times \text{RabbitAnatomy} =$$
$$= 12\ \texttt{head} \times 4\,\frac{\texttt{leg}}{\texttt{head}} = 48\ \texttt{leg}.$$

То, что мы видим здесь, – это *наследование типов*, в терминологии информатики. Но есть ли в этом реальный жизненный смысл? Да. Именно столько ног было бы у питомцев Мэри, если бы все они были кроликами. Это больше, чем заданное количество ножек. На сколько больше?

$$\textsc{ExcessiveLegs} = \textsc{Heads} \times \textsc{RabbitAnatomy} - \\ - \texttt{Legs} = 16\ \texttt{leg}.$$

Откуда берутся эти лишние ноги? От кур, каждая из которых получает дополнительные

$$\textsc{RabbitAnatomy} - \textsc{ChickenAnatomy} = \\ = 4\frac{\texttt{leg}}{\texttt{head}} - 2\frac{\texttt{leg}}{\texttt{head}} = 2\frac{\texttt{leg}}{\texttt{head}}$$

ноги. Итак, сколько всего кур?

$$\textsc{Chicken} = \frac{\textsc{ExcessiveLegs}}{\textsc{RabbitAnatomy} - \textsc{ChickenAnatomy}} \\ = 16\ \texttt{leg} \div 2\frac{\texttt{leg}}{\texttt{head}} \\ = 8\ \texttt{head}$$

Конечно, количество кроликов теперь очевидно:

$$\textsc{Rabbits} = \textsc{Heads} - \textsc{Chicken} = 12\ \texttt{head} - \\ - 8\ \texttt{head} = 4\ \texttt{head}.$$

Одним из *очень немногих* альтернативных способов решения может быть вычисление

$$\textsc{Heads} \times \textsc{ChickenAnatomy} = \\ = 12\ \texttt{head} \times 2\frac{\texttt{leg}}{\texttt{head}} = 24\ \texttt{leg},$$

но это приводит к тому же решению.

Это не самый короткий способ решить задачу, но он имеет преимущество в том, что он очень формальный, с каждым шагом приводящий к компьютерному коду, работающему с именованными числами — это решение уже почти код.

Возможны и другие подходы. Не так давно первый автор обсуждал эту задачу с небольшой группой детей беженцев из Украины (детям было от 8 до 11 лет) в GOOGLE MEET, и один из детей предложил смотреть на `конечность`, `ногу` или `крыло`, что приводит к более эффективному решению.

Действительно, всего 3 прямолинейных вопроса:

(1) Каково число конечностей?

(2) Каково количество крыльев?

(3) Каково количество кур?

приводят к решению.

9. НАМ НУЖНО СЛИЯНИЕ АЛГЕБРЫ И ИНФОРМАТИКИ, А НЕ ИХ ФОРМАЛЬНОЕ ОБЪЕДИНЕНИЕ

Статья Яна Бенсона и Джима Торпа [2] рекламирует старую статью Тревора Флетчера [14] *Думать, используя стрелки* как “концептуальную математику” и как способ “показывать и обсуждать” математическую схему.

Возможно, движимый законом номинативного детерминизма, Флетчер был заинтересован использованием стрелок в математике, преподаваемой в школах. Это фрагмент из его статьи, который Бенсон и Торп привели в качестве примера мышления с использованием стрелок.

Пусть первый кран может наполнить ванну за 3 мин, а второй за 6 мин. Как понять, что вместе они наполнят ванну за 2 мин? Аналогично, пусть оба крана, будучи включенными по отдельности, могут наполнить ванну за 10 мин. Как понять, что вместе они наполнят ванну за 5 мин? Каким образом выполняется вычисление? Будем называть операцию получения 2 из 3 и 6 *tap*, соответствующая диаграмма имеет вид

$$\begin{tikzcd}
(3,6) \arrow[r, "tap"] \arrow[d, dashed, "recip"'] & 2 \arrow[d, dashed, "recip"] \\
\left(\frac{1}{3}, \frac{1}{6}\right) \arrow[r, "+"] & \frac{1}{2}
\end{tikzcd} \tag{1}$$

Из самой диаграммы не видно, каким образом ответ был **вычислен**; мы видим только, что **ответ** на задачу **представлен** коммутативной диаграммой, которая сводит получение ответа (воспринимаемого как бинарная операция) к бинарной операции сложения положительных вещественных чисел; заметим, что слово “операция” используется самим Флетчером.

У нас есть возражения против такого подхода.

• Мы считаем, что целью математического образования должно быть обучение ребенка искусству **решать задачи** и **получать ответы**.

• **Интерпретация** ответов должна играть важную роль; однако настаивать на определенной форме **представления** ответа нежелательно; в конце концов

• Должен быть плавный переход между арифметическим/алгебраическим мышлением при решении задачи и мышлением в области информатики/программирования при представлении решения в виде алгоритма.

• Нам нужно слияние алгебры и вычислительной техники, при этом диаграмма (1) представляет собой объединение двух непересекающихся концептуальных областей.

Мы хотим продемонстрировать, что в случае класса задач на границе арифметики и алгебры, к которому естественным образом относится задача о совместной работе, коммутативное диаграммное представление решения вводит в заблуждение и контрпродуктивно.

Действительно, рассмотрим “задачу про кран” вместе с тремя другими задачами из той же серии:

зависимости пропорциональности между временем, скоростью и расстоянием (где “расстоянием”, возможно, может быть “уровень воды в ванне”).

Задача 1: Если у вас есть кран, который наполняет ванну за a минут, и кран, который наполняет ту же порцию за b минут, за сколько минут они вместе наполнят ванну? Это задача Флетчера, см. диаграмму (1).

Задача 2: Если автомобиль едет из A в B со скоростью a миль в час, а затем возвращается из B в A со скоростью b миль в час, какова средняя скорость автомобиля на протяжении всего путешествия?

Задача 3: Две машины отправились в путь на рассвете, одна из A до B, а другая из B до A. Они встретились в полдень и завершили свои поездки в час дня “а” и “б” соответственно. На сколько восход солнца был раньше полудня в тот день?

Задача 4: На реке есть два города, один находится выше по течению другого. Пароходу требуется a дней, чтобы добраться из одного города в другой, и b дней, чтобы вернуться обратно. Сколько дней потребуется плоту, чтобы проплыть от города, расположенного выше по течению, до города, расположенного ниже по течению?

Мы приводим здесь ответы на эти задачи в случайном порядке:

$$\text{(a) } \sqrt{ab}; \quad \text{(b) } \frac{ab}{a+b}; \quad \text{(c) } \frac{2ab}{|a-b|}; \quad \text{(d) } \frac{2ab}{a+b}.$$

Как ребенок может догадаться, какой ответ соответствует какой задаче? Правда ли, что единственный способ – получить эту информацию от учителя? Откуда берутся эти ответы? Коммутативная диаграмма Флетчера действительно коммутативна, но можно ли эффективно применить такой подход при работе со школьниками? Напротив, для критического этапа перехода от арифметики к алгебре мы предлагаем плавный путь от алгоритмического решения арифметической задачи к ее алгебраическому выражению и вычислений с помощью программирования. Коммутативные диаграммы могли бы появиться в игре несколько лет спустя, но не сейчас.

Если мы начнем изучать диаграммы, соответствующие ответам (a)–(d), мы легко установим, что только один из них представляется диаграммой, аналогичной рассмотренной выше (1). Действительно, если $\circ$ и $\Diamond$ – две бинарные операции на множестве положительных действительных чисел $\mathbb{R}^{>0}$, удовлетворяющих коммутативной диаграмме

$$\begin{tikzcd}
(a, b) \arrow[r, "\circ"] \arrow[d, dashed, "\phi"] & a \circ b \arrow[d, dashed, "\phi"] \\
(\phi(a), \phi(b)) \arrow[r, "\Diamond"] & \phi(a) \Diamond \phi(b)
\end{tikzcd} \tag{2}$$

для некоторого отображения $\phi : \mathbb{R}^{>0} \to \mathbb{R}^{>0}$, тогда

$$\phi(a \circ b) = \phi(a) \Diamond \phi(b), \tag{3}$$

т.е. ϕ представляет собой гомоморфизм из моноида $A = \{\mathbb{R}^{>0}, \circ\}$ в моноид $B = \{\mathbb{R}^{>0}, \Diamond\}$. В ответах (a) и (d) – вообще говоря, представляющих собой среднее геометрическое и среднее гармоническое, сами по себе довольно важные понятия – моноид *A полностью состоит из идемпотентных элементов*, т.е. $a \circ a = a$ для всех $a \in A$. Но тогда его образ $\Im(\phi)$ при отображении ϕ также представляет собой моноид, состоящий из идемпотентных элементов. Значит, $\Diamond$ не может быть, например, стандартным сложением, потому что $\{\mathbb{R}^{>0}, +\}$ не содержит идемпотентов, и если $\Diamond$ это стандартное умножение, то $\Im(\phi) = \{1\}$ и $\phi(a) = 1$ для всех $a \in A$. Это тривиальный гомоморфизм, который не несет никакой информации об операции $\circ$.

Ответ (c) вообще не соответствует никакой бинарной операции: он не определен при $a = b$. В этой точке функция имеет особенность, что не позволяет представить эту функцию диаграммой, аналогичной рассмотренной выше (1) с понятной и хорошо знакомой школьникам операцией $\Diamond$, представленной стрелкой внизу диаграммы.

Уравнение (3) позволяет построить большое количество диаграмм, аналогичных диаграмме (2). Мы начинаем с произвольной бинарной операции $\Diamond$ на множестве $\mathbb{R}^{>0}$, берем произвольное биективное отображение ϕ на $\mathbb{R}^{>0}$ и определяем

$$a \circ b = \phi^{-1}(\phi(a) \Diamond \phi(b)), \tag{4}$$

что естественным образом делает диаграмму (2) коммутативной.

Но мы ограничены в выборе: есть по крайней мере $2^{2^{\aleph_0}}$ перестановок ϕ. Остается единственный вопрос:

Сколько таких коммутативных диаграмм имеют отношение к школьному курсу алгебры и школьному курсу арифметики?

В действительности, таких диаграмм не так уж и много. Некоторые из них представляют некоторый интерес, но все равно достаточно далеки от ключевых тем школьного курса.

Например, мы можем взять $\Diamond = +$, и $\phi(x) = x^n$, n — натуральное число.

Получаем коммутативную диаграмму

$$\begin{tikzcd}
(a, b) \arrow[r, "\circ"] \arrow[d, dashed, "x \mapsto x^n"'] & \sqrt[n]{a^n + b^n} \arrow[d, dashed, "x \mapsto x^n"] \\
(a^n, b^n) \arrow[r, "+"] & a^n + b^n
\end{tikzcd} \tag{5}$$

Случай $n = 2$ изучен Флетчером, соответствующее отображение называется “отображением Пифагора”:

$$\begin{tikzcd}
(3, 4) \arrow[r, "\texttt{Pythag}"] \arrow[d, dashed, "\texttt{Sq}"'] & 5 \arrow[d, dashed, "\texttt{Sq}"] \\
(9, 16) \arrow[r, "+"] & 25
\end{tikzcd} \tag{6}$$

— при этом диаграмма никак не связана с доказательством теоремы Пифагора.

10. ЗАКЛЮЧЕНИЕ

В заключение мы хотим подчеркнуть следующее:

> В процессе решения вопросы
> должны быть сформулированы
> Учащимся, не учителем.
> **Ученик должен контролировать ситуацию**.[5]

Таким образом, решение задачи распадается на следующие этапы:

• Тщательный анализ условия задачи.

• Последовательность “вопросов”, которая естественным образом становится алгоритмом.

Кроме того, понятия из области компьютерных наук:

• Типизированные переменные, они же “именованные числа”.

• Вычисление путем вызова по имени и путем вызова по значению.

• Овеществление как способ введения промежуточных параметров.

11. НЕКОТОРЫЕ СОЦИАЛЬНО-ПОЛИТИЧЕСКИЕ НАБЛЮДЕНИЯ

Ад — это другие.
Жан-Поль Сартр, *Huis Clos*, 1943

Старая (1996!) статья Нила Коблица[10] *Дело против компьютеров в математическом образовании* [20] заслуживает чтения и сегодня. Его предупреждения по-прежнему актуальны и в полной мере применимы к возможности плохо подготовленной, недофинансированной, поспешно внедренной компьютеризации математического образования в школах:

Недостатки можно разделить на несколько широких областей:

• *недостаток ресурсов* (*денег*, *времени*, *энергии*);

• *отсутствие разработанной методологии образования*;

• *антиинтеллектуальная психологическая установка деятелей образования*;

• *коррупция в образовательной среде*.

Мы надеемся, что эта статья послужит предупреждением: правильное внедрение “вычислительного мышления” в школьную математику — чрезвычайно сложная и очень дорогостоящая задача.

Мы можем только аплодировать героическим усилиям сотен учителей в разных странах, которые пытаются что-то сделать в этом направлении, и мы считаем, что очень важно помогать им всеми доступными способами, их опыт необходимо изучать, систематизировать и использовать при разработке потенциальной реформы. Очень часто они являются одними из лучших учителей в своих странах. К сожалению, обычно это означает, что методы и подходы, открытые и разработанные ими, по умолчанию не масштабируемы и не могут быть перенесены на всю страну, и по основной причине: большинство других учителей не похожи на них, они менее образованны, менее мотивированы и часто деморализованы повседневной школьной рутинной работой.

Серьезная реформа требует систематического перевоспитания целой армии учителей и предоставления им (разумеется, оплачиваемого) времени для личностного профессионального роста. Непосредственное следствие: нам понадобится больше учителей.

Необходимо усвоить уроки реформы школьной программы по математике, проведенной Колмогоровым в России в 1970-х годах [8]. Андрей Колмогоров был всемирно известным математиком, у него были наилучшие намерения, но его реформа с треском провалилась, и одной из причин этого была недооценка роли учителей,

[10] Нил Коблиц — один из отцов-основателей эллиптической криптографии.

даже прямое пренебрежение к ним. Что действительно печально, так это то, что в России в то время была хорошо развитая и функциональная система подготовки учителей математики через многочисленные педагогические колледжи (4 года или высшее образование) и математические факультеты региональных университетов (5 лет) — но этот ресурс не был задействован. Возможно, с годами эти государственные институты постепенно начали приходить в упадок, но все же ситуация в России вряд ли будет такой же плохой, как в Великобритании, где, по словам Тони Гардинера (лучшего эксперта в области математического образования в Великобритании) [16],

Мы не знаем ни о какой другой системе, которая претендует на подготовку учителей математики, отдавая небольшие группы слушателей на откуп преподавателям, не имеющим соответствующего опыта ITE [*начального педагогического образования*], *не являющимся учителями. При этом большая часть изучаемого материала носит "общий", а не предметный характер. Англия, по-видимому, единственная среди развитых стран, кто придерживается такого подхода.*

Вновь напоминая о трагической судьбе Андрея Колмогорова и его реформе [8], мы повторяем наше предупреждение:

Любая попытка глубокой реформы основного математического образования — это огромная задача. Опасно относиться к этому легкомысленно.

БЛАГОДАРНОСТИ

Первый автор бесконечно благодарен Мартину Хайланду, который побудил его принять участие в этом проекте, и благодарит Иэна Бенсона, Тони Гардинера и Виктора Сиротина за множество полезных обсуждений и исправлений. Он благодарит Рика Бута, Аню Мейер и Наташу Страбич за их понимание, поддержку и полезные комментарии. Второй автор благодарит Марию Левченко за помощь в работе над русской версией текста.

ИСТОЧНИК ФИНАНСИРОВАНИЯ

Первый автор не получил никакого финансирования. Работа В. Кондратьева выполнена при поддержке Российского фонда фундаментальных исследований (проект 19-29-14217 "Перспективные направления и формы использования компьютерных технологий в школьных программах по математике").

ОГОВОРКА

Авторы пишут в своем личном качестве, и выраженные мнения не обязательно отражают позицию их работодателей или любого другого лица, корпорации, организации или учреждения.

СПИСОК ЛИТЕРАТУРЫ

1. *Altınel T., Borovik A., Cherlin G.* Simple Groups of Finite Morley Rank, Amer. Math. Soc. Monographs Series, Amer. Math. Soc., Providence, RI, 2016.
2. *Benson I., Thorpe J.* Thinking with arrows for mathematical thought. Mathematics Teaching. Association for Teaching of Mathematics. 2022. P. 281.
3. *Boersma M.* Domain-Specific Languages Made Easy. Manning Publication, 2020.
4. *Borovik A.V.* Mathematics under the Microscope: Notes on Cognitive Aspects of Mathematical Practice. Amer. Math. Soc., Providence, RI. 2009.
5. *Borovik A.* http://bit.ly/2d6EncgBeing in control. In Understanding Emotions in Mathematical Thinking and Learning (U. Xolocotzin (ed.). Academic Press, San Diego, 2017, pp. 77–96. http://bit.ly/2d6Encg.
6. *Borovik A.V.* http://bit.ly/293orpkEconomy of thought: a neglected principle of mathematics education, in Simplicity: Ideals of Practice in Mathematics and the Arts (R. Kossak and Ph. Ording, eds.). Springer, 2017. P. 241–265. http://bit.ly/293orpk.
7. *Borovik A.V.* http://bit.ly/2qYHtstMathematics for makers and mathematics for users. In Humanizing Mathematics and its Philosophy: Essays Celebrating the 90th Birthday of Reuben Hersh (B. Sriraman ed.), Birkhauser, 2017. P. 309–327. http://bit.ly/2qYHtst.
8. *Borovik A.* The Kolmogorov reform of mathematics education in the USSR. To appear as a chapter in D. De Bock (Ed.), Modern Mathematics—An International Movement?, Springer Nature, 2022.
9. *Borovik A., Gardiner T.* The Essence of Mathematics — Through Elementary Problems.Open Book Publishers, 2019. 396 p.
10. *Borovik A., Kocsis Z., Kondratiev V.* Mathematics and mathematics education in the 21st century, 2021. arXiv:2201.08364 [math.HO].
11. *Borovik A., Yalçınkaya Ş.* Adjoint representations of black box groups $PSL_2(\mathbb{F}_q)$. J. Algebra. 2018. V. 506. P. 540–59.
12. *Borovik A.* Shadows of the Truth: Metamathematics of Elementary Mathematics. In preparation, draft: www.borovik.net/ST.pdf.
13. *Ershov A.P.* Programming, the second literacy. Microprocessing and Microprogramming. 1981. V. 8. P. 1–9.
14. *Флетчер Т.* Thinking with arrows. Mathematics Teaching. Association for Teaching of Mathematics. 1971. P. 057.
15. *Gardiner D.* Teaching Mathematics at Secondary Level. OpenBook Publishers, 2016.
16. *Gardiner D.* https://demorgangazette.files.wordpress.com/2021/02/fc116-gardiner_teacher_preparation_and_support_31aug18.pdfTowards an effective national structure for teacher preparation and support in mathematics, The De Morgan Gazette. 2018. V. 10. № 1. P. 1–10.

17. *Gerofsky S.* A linguistic and narrative view of word problems in mathematics education. For the Learning of Mathematics. 1996. V. 16. № 2. P. 36–45.
18. *Fowler M.* Domain Specific Languages. Addison-Wesley Professional, 2010.
19. *Khalin V., Vavilov N., Yurkov A.* The skies are falling: Mathematics for non-mathematicians. Submitted.
20. *Koblitz N.* The case against computers in K-13 math education (Kindergarten through Calculus). The Mathematical Intelligencer. 1996. V. 18. № 1.
21. *Semenov A., Polikarpov S., Rudchenko T.* The future of mathematics education. Mathematics in School. 2022. V. 1. № 114.
22. *Sirotin V.* SI Units, 2022. https://github.com/vsirotin/si-units.
23. *Viete F.* The Analytic Art (Translated by T. Richard Witmer.) Dover Publications Inc., Mineola, NY, 1983.
24. *Voelter M. et al.* DSL Engineering. Designing, Implementing and Using Domain-Specific Languages. 2010. http://dslbook.org.
25. *Алексеева О.В., Ищенко И.Н.* Методика обучения решению текстовых задач в начальной школе. Амурский гуманитарно-педагогический государственный университет, Комсомольск-на-Амуре, 2019.
26. *Арнольд И.В.* Принципы отбора и составления арифметических задач. Известия АПН РСФСР. 1946. Вып. 6. С. 8–28. Reprinted in Арнольд И.В. Принципы отбора и составления арифметических задач. Moscow, МЦНМО, 2008.
27. *Баженова Н.Г., Одоевцева И.Г.* Теория и методика решения текстовых задач. Флинта, Москва, 2012.
28. *Белошистая А.В.* Методика обучения математике в начальной школе. Москва, Владос, 2007.
29. ЯКласс, https://www.yaklass.ru/p/matematika/5-klass/naturalnye-chisla-13442/reshenie-tekstovykh-zadach-arifmeticheskim-sposobom-13747/Текстовые задач и их решение арифметическим способом. Accessed 11 October 2022.

A NEW COURSE “ALGEBRA + COMPUTER SCIENCE”: WHAT SHOULD BE ITS OUTCOMES AND WHERE IT SHOULD START

A. V. Borovik[a] and V. V. Kondratiev[a]

[a] *University of Manchester, Manchester, United Kingdom*

Presented by Academician of the RAS A.L. Semenov

The words “Programming is the second literacy” were coined more than 40 years ago [13], but never came to life. The paper develops and details that old slogan by proposing that the mainstream mathematics education in schools should merge with education in computer science/programming. Of course, this means a deep structural reform of school mathematics education. We are not talking about adapting the 20th century mathematics to the 21st century—s it outlined in [10, 19], we mean the 21st century mathematics education for the 21st century mathematics. To the best of our knowledge, this paper is perhaps the first known attempt to start a proper feasibility study for this reform. The scope of the paper does not allow us to touch the delicate sociopolitical (and financial) sides of the reform, we are looking only at general curricular and didactic aspects and possible directions of the reform. In particular, we indicate approaches to development of a Domain Specifiic Language (DSL) as a basis for all programming aspects of a new course.