Формирование основ алгоритмического мышления в процессе начального обучения математике

С.Е. ЦАРЕВА,

кандидат педагогических наук, профессор, Новосибирский государственный педагогический университет

Федеральные образовательные стандарты начального общего образования (ФГОС НОО) впервые в истории российского начального математического образования ввели понятие *алгоритм* в обучение математике и поставили задачу: обеспечить «овладение основами... алгоритмического мышления, ...записью и выполнением алгоритмов; ...умением действовать в соответствии с алгоритмом и строить простейшие алгоритмы». На первый взгляд кажется, что данные требования, возможно, относятся не к математике, а к информатике. Однако это не так. Алгоритм — понятие, возникшее в математике. Теория алгоритмов также разработана в математике, каждый раздел математики содержит алгоритмы. Информатика использует и развивает это понятие, поэтому перечисленные требования относятся как к информатике, так и к математике, к начальному образованию в целом, так как алгоритмы встречаются во всех областях знания и развитие алгоритмического мышления — задача всех учебных дисциплин. Данная статья посвящена характеристике понятий алгоритмической линии начального математического образования, их роли в математическом образовании младших школьников и соответствующей подготовке учителя.

Следует отметить, что в российском школьном математическом образовании понятие *алгоритм* появилось гораздо раньше, чем были утверждены ФГОС НОО. Инициатором введения алгоритмической линии в математику начальной школы был Н.Я. Виленкин (1920–1991), известный советский математик, автор научно-популярных книг и учебников по математике для школы. Еще в 70-е годы прошлого века он утверждал, что в век «умных машин» детей нужно с начальной школы готовить к работе с ними. Эта подготовка, по его мнению, должна заключаться в формировании алгоритмического мышления [3, *35–37*][1].

Н.Я. Виленкиным уже тогда были созданы учебники по математике для начальной школы, в которых рассматривались понятия *операция, программа, алгоритм*. Автор этой статьи познакомилась с ними в 1978 г., когда Н.Я. Виленкин был в Новосибирске, на семинаре в Вычислительном центре Сибирского отделения РАН, специалисты которого первыми в стране начали

[1] В квадратных скобках указан номер и страницы работы из списка «Использованная литература». — *Ред.*

подготовку учащихся начальной школы к работе с вычислительной техникой. Наум Яковлевич тогда привез ротапринтное издание своих учебников для начальной школы, в которых были развернуто представлены понятия *операция, программа, алгоритм*. Специальная тема «Программа действий. Алгоритм» сохранилась с тех пор в учебниках Л.Г. Петерсон (выросших из тех давних ротапринтных книжек), первые издания которых были подготовлены в соавторстве и под руководством Н.Я. Виленкина (в них значились два автора — Н.Я. Виленкин и Л.Г. Петерсон).

В связи с требованиями ФГОС НОО понятие *алгоритм* в том или ином виде появилось или должно появиться во всех программах, комплектах учебников, методических пособиях к учебникам по математике для начальной школы. Теперь реализация богатого образовательного потенциала этой темы зависит от качества подготовки учителя, от степени понимания им сущности понятия алгоритма и возможностей темы в достижении учащимися личностных, метапредметных и предметных планируемых результатов обучения всем учебным предметам и в первую очередь математике.

Понятия *алгоритм, алгоритмическое мышление, алгоритмическая культура* являются сложными понятиями, относительно которых в педагогической и методической литературе еще не сформировано общепринятое представление.

Алгоритм — это фундаментальное математическое понятие, продукт человеческой деятельности [6]. Алгоритм есть особый способ описания последовательности операций для осуществления определенных процессов человеком или машиной. Алгоритм — это способ удержания информации о действиях, обеспечивающих повторяемость процессов перехода от исходного состояния некоторого объекта к конечному, от исходных данных к искомым результатам.

Следует различать алгоритмы и процессы, в которых можно выделить различимые этапы, действия. Такие процессы называют *алгоритмическими*. Процессы утренних сборов в школу, чистки зубов, варки картофеля, развития лягушки и бабочки с последовательными превращениями, построения пчелами сот не являются алгоритмами, но могут быть отнесены к алгоритмическим процессам.

Известно, что слово *алгоритм* произошло от имени среднеазиатского ученого IX в. аль-Хорезми. Он «родился до 800 г., а умер после 847 г., жил и работал в Багдаде. ...Аль-Хорезми использовал индийскую позиционную систему счисления с нулем и сформулировал правила четырех арифметических действий над многозначными числами. Первоначально под алгоритмами понимали только эти правила...» [2, *173*]. Обращаем внимание читателей на эту информацию: источником понятия *алгоритм* были арифметические действия, а потому логично было бы и в формировании представлений школьников об алгоритмах значимое место отвести алгоритмам арифметических действий.

Ю.А. Макаренков и А.А. Столяр пишут, что алгоритм — понятие, возникшее как ответ на вопрос: «Существует ли общий метод, позволяющий для любой частной задачи этого класса в конечное число шагов дать требуемый ответ..?» [8, *8, 9*]. Ответы на этот и подобные вопросы позволили создать такие описания общих способов решения задач некоторого вида, что решение любой задачи этого вида становилось доступным многим «решателям», в том числе машинам, умеющим последовательно и правильно выполнять предписанные алгоритмом операции. Следовательно, понятие *алгоритм* и каждый конкретный алгоритм может быть содержательно понят, принят и даже изобретен учащимися только в связи с поиском ими общего способа решения задач некоторого вида.

Изобретение алгоритмов значительно расширило возможности людей в использовании математики. Так, сегодня каждый выпускник начальной школы умеет выполнять четыре арифметических действия с натуральными числами, в том числе с большими. Между тем Д.Д. Галанин, автор ряда методических идей, нашедших развитие в современных системах обучения математике, сообщает: монаха Кирика, жившего на Руси в XII в., причислили к лику святых за многие благие деяния, среди которых названы его первый на Руси математический трактат [10] и умение выполнять четыре арифметических действия с большими числами [4]. На Руси в XII в. для этого надо было обладать недюжинным умом и упорством, так как числа записывались в непозиционной, славянской системе и простых алгоритмов действий на Руси тогда не знали [4].

В Математическом энциклопедическом словаре понятие *алгоритм* характеризуется как понятие математики и информатики: «Алгоритм — точное предписание, которое задает вычислительный процесс (называемый в этом случае алгоритмическим), начинающийся с произвольного исходного данного... и направленный на получение полностью определяемого этим исходным данным результата. ...Алгоритмы прослеживаются в математике в течение всего времени ее существования. Общее понятие алгоритма сформировалось, однако, лишь в XX в.» [9].

В цитируемом выше пособии Ю.А. Макаренкова и А.А. Столяра [8] читаем: алгоритм — это «точное, понятное предписание о том, какие действия и в каком порядке необходимо выполнить, чтобы решить любую задачу из данного класса однотипных задач (для которого и предназначен этот алгоритм)» [8, *19*]. Эти действия называют также *шагами*, *операциями*.

Для того чтобы некоторое предписание можно было назвать алгоритмом, необходимо, чтобы оно обладало следующими свойствами:

1) для каждого шага (кроме последнего) можно указать единственный (при данном выборе исходных объектов), непосредственно следующий за ним шаг, между которыми нет других шагов (дискретность);

2) алгоритм предназначен для решения не одной задачи, а любой задачи некоторого класса однотипных задач (массовость);

3) алгоритм однозначно задает последовательность действий и результат, он гарантирует получение результата за конечное число шагов (определенность, результативность);

4) исходные объекты, промежуточные и конечные результаты алгоритма конструктивны, т.е. они имеют четкие характеристики, их можно построить, получить (конструктивность объектов);

5) действия, предписываемые алгоритмом, входят в систему действий исполнителя [8, *17–19*].

Добавим также, что алгоритм не должен содержать действий, которые не влияют на получение результата. Так, например, некорректно называть шагом алгоритма письменного деления (деления «уголком») определение количества цифр в частном, так как результат этой операции не используется в других операциях этого алгоритма, при последовательном и правильном выполнении которых частное находится и без определения количества цифр в частном.

Знание учителем перечисленных свойств позволяет грамотно учить школьников распознавать и конструировать алгоритмы, овладевать алгоритмами школьного курса математики, в том числе вычислительными.

Второе свойство говорит о том, что необходимо ориентировать учащихся на поиск общего способа (а не на способ решения одной задачи) и тем более не на получение ответа задачи.

Пятое свойство алгоритма обусловливает необходимость такой актуализации у учащихся всех операций, входящих в изучаемый алгоритм, чтобы в момент знакомства или конструирования нового алгоритма каждый учащийся был бы в состоянии выполнить все операции алгоритма, пусть даже только для случаев применения алгоритма, запланированных учителем. Невозможность выполнить хотя бы одну операцию алгоритма делает его недоступным для учащегося.

Важной характеристикой алгоритма являются способы его задания. Перечислим основные способы задания алгоритмов, доступные учащимся начальной школы: а) словесное предписание (в виде памятки, инструкции, перечня шагов); б) образец выполнения, который задает алгоритм только тогда, когда исполнитель «считывает» с этой записи общий способ, а не способ решения данной в образце конкретной задачи; в) блок-схема; г) граф-схема. В качестве способов задания алгоритмов называют также алгоритмическую запись, формулу и таблицу. В информатике для представления алгоритмов используются специальные языки программирования.

Ниже приведены примеры алгоритмов курса математики начальной школы и названных выше способов их задания. Некоторые алгоритмы являются привычными для начальной школы и представлены в учебниках, некоторые изобретены учащимися, студентами и преподавателями.

1. Алгоритм вычитания для случаев вида 23 – 6 (6 > 3), 13 – 8 (8 > 3) изобретен первоклассницей на уроке заслуженной учительницы России Р.П. Шульги в школе № 41 г. Новосибирска.

23 — 6

1) 6 – 3 = 3 — узнаем, сколько единиц не хватает в единицах уменьшаемого, чтобы вычесть из них все вычитаемое;

2) 20 – 3 = 17 — вычтем недостающие единицы из десятков, получим искомую разность;

3) назовем результат: разность 23 и 6 равна 17.

2. Алгоритм вычитания для случаев вида 254 – 78 (78 > 54), 1 372 – 857 (857 > 372) изобретен студентами вместе с преподавателем.

1 372 – 857 = 1 375 – 860 = 1 315 – 800 = = 1 515 – 1 000 = 515.

1) + 3, + 3; 2) – 60, – 60; 3) + 200, + 200; 4) вычитаем; 5) читаем результат.

Словесно описать данный алгоритм можно следующим образом:

— увеличим (или уменьшим) уменьшаемое и вычитаемое на одно и то же однозначное число так, чтобы новое вычитаемое стало оканчиваться на 0;

— увеличим (или уменьшим) уменьшаемое и вычитаемое на одно и то же круглое двузначное число так, чтобы новое вычитаемое оканчивалось двумя нулями и т.д., пока вычитаемое не станет числом, записанным одной значащей цифрой и нулями;

— выполним вычитание получившихся чисел, результат будет разностью исходных чисел, назовем ее.

3. Алгоритмы вычитания однозначного числа из круглого двузначного (для случаев вида 30 – 4).

а) 30 – 4 = (20 + 10) – 4 = 20 + (10 – 4) = = 20 + 6 = 26. Это стандартный алгоритм, который представлен в учебниках математики для начальной школы.

б)

$$30 - \underset{10}{\begin{cases}4 \xrightarrow{-1} = 26\\ 6\end{cases}}$$

1. Определим число десятков разности как число, на 1 меньшее числа десятков уменьшаемого.

2. Определим число единиц разности как число, дополняющее вычитаемое до десяти.

3. Назовем разность.

Этот алгоритм изобретен детьми.

4. Алгоритм нахождения значения числового выражения (правило порядка действий в выражении).

1) При вычислении значения выражения без скобок:

а) если в выражении есть только сложение и (или) вычитание или только умножение и (или) деление — выполнить эти действия по порядку слева направо;

б) если в выражении есть действия из групп сложение / вычитание и умножение / деление — выполнить сначала умножение и деление по порядку слева направо, затем выполнить сложение и вычитание по порядку слева направо;

в) результат последнего действия назвать значением выражения.

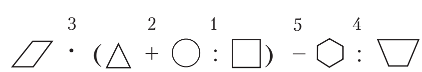
2) При вычислении значения выражения со скобками:

а) сначала выполнить действия в скобках, как в выражении без скобок (см. п. 1);

б) затем выполнить действия со значениями выражений в скобках;

в) результат последнего действия назвать значением выражения.

5. Алгоритм нахождения значения числового выражения конкретного вида.



Алгоритмы могут быть линейными и разветвленными, без циклов и циклическими. В приведенных примерах алгоритм 4 является разветвленным, остальные — линейными. Примерами циклических алгоритмов являются алгоритмы письменных вычислений — сложения, вычитания, умножения в столбик и деления «уголком».

С понятием *алгоритм* тесно связано понятие *программа*: «программа — план действий, подлежащих выполнению некоторым исполнителем, обычно автоматическим устройством; ...предписание, алгоритм» [9, *494*]. «Алгоритм является более общим понятием, чем программа, ...программа для вычислительной машины — это запись алгоритма решения некоторой задачи в виде, пригодном для данной вычислительной машины» [7, *85*].

Термин *алгоритм* закрепился в математике в связи с проблемой пошагового описания общего метода решения, а термин *программа* — в информатике в связи с проблемой разработки соответствующих форм задания алгоритмов, которые могли бы быть переданы для исполнения компьютерам.

Важной особенностью алгоритма является то, что в нем нет обоснования, что данная последовательность операций приводит к требуемому результату. Такое обоснование может приводиться в сопровождающих алгоритм текстах, оно необходимо при создании и изучении алгоритмов, в том числе при изучении алгоритмов школьного курса математики.

В методике обучения математике эта часть информации об алгоритме реализуется через понятия *теоретическая основа вычислительного приема (вычислительного алгоритма)* и *эмпирическая, предметная основа алгоритма*. Первый термин, добавив к употребляемому ранее термину *основа вычислительного приема* (Л.Н. Скаткин (1893–1981), ввела в употребление М.А. Бантова (1918–2008), один из авторов самых известных учебников математики для начальной школы, автор многочисленных статей, в том числе фундаментальной статьи «Система вычислительных навыков» [1], в которой и введено понятие *теоретическая основа вычислительного приема*. Второе понятие введено нами.

К необходимости введения понятия предметной основы вычислительных алгоритмов нас привел случай на уроке математики в I классе на тему «Вычитание вида 30 – 7» (алгоритмы 3а и 3б, представленные выше). После актуализации способов вычислений с однозначными числами и правил вычитания числа из суммы учитель создал проблемную ситуацию, предложив среди освоенных случаев вычитания и новый. Выход из затруднения первоклассники искали через моделирование вычитания на демонстрационных счетных палочках: 30 — это 3 пучка по 10 палочек, вычесть 7 — удалить 7 палочек. Для выполнения действия один пучок развязывают. Пучков становится на один меньше. Забрав 7 палочек из 10 палочек развязанного пучка, получили 3 палочки — число, дополняющее вычитаемое 7 до 10.

Запись 30 – 7 на доске дополнилась знаком равно и результатом вычитания: 30 – 7 = 23. Ученики нашли результат вычитания для нескольких подобных пар чисел с демонстрацией действия на счетных палочках, делая короткую запись на доске и в тетради. Затем вычисления производились без палочек и без записи. Ученики работали активно и уверенно, но вдруг учитель вспомнил, что у нее была предусмотрена и развернутая запись вычислений, отражающая теоретическую основу рассматриваемого вычислительного приема — правило вычитания числа из суммы, которое изучали на предыдущих уроках: 30 – 7 = = (20 + 10) – 7 = 20 + (10 – 7) = 20 + 3 = 23. Она решила «исправить» положение. Опираясь на изображения пучков палочек, учитель сделала нужную запись на доске, ученики записали ее в тетрадь. Учитель вновь дала задание на вычитание: из 20 вычесть 3, из 40 вычесть 6 и т.д. Однако учеников как подменили: они не поднимали руки, не называли результаты, с большим трудом следовали за подсказками учителя. При анализе урока мы вынуждены были сделать вывод, что длинная запись и правило напугали учащихся, перечеркнули достигнутое понимание происходящего. Призванное обосновать, т.е. сделать более убедительной правомерность операций соответствующего алгоритма, правило, как слишком громоздкое и сложное для первоклассников, оказало противоположное действие.

На наш взгляд, теоретическая основа может быть использована в обучении школьников более старшего возраста и лишь в некоторых случаях при обучении первоклассников. При этом нужно понимать, что чем длиннее математическая запись обосновывающего положения, тем менее информативна она для учащихся начальных классов, тем меньше она может выполнять функцию обоснования.

Теоретическое утверждение станет обосновывающим, убеждающим в достоверности чего-либо только для человека, овладевшего на некотором уровне теорией (в приведенном случае теорией арифметических действий), а также владеющего письменным математическим языком. Ни того, ни другого у первоклассников еще нет. Чем первоклассники точно владеют, так это действиями с предметами.

Предметные действия, богатый опыт которых есть у всех учеников, легко убеждают их в правомерности тех или иных операций, переходов от одних операций к другим. В связи с этим мы и ввели понятие *предметная (эмпирическая) основа вычислительного приема (алгоритма)*, которое может быть расширено до понятия *предметная (эмпирическая) основа алгоритма*. Мы утверждаем, что эмпирические, предметные основы алгоритма во многих случаях являются более убедительными, чем теоретические, а в ряде случаев будут преддверием теоретических основ.

Учителю также необходимо знать возможные способы представления алгоритмов в учебном процессе, например, такие, как в таблице на с. 11, 12.

Изобретение алгоритмов вызвано потребностью человека в сохранении удачной последовательности действий (приведшей к решению конкретных задач) для повторного применения и передачи другим людям. Обеспечить возникновение такой потребности у учащихся — значит создать действенный мотив для конструирования и усвоения ими алгоритмов, для развития алгоритмического мышления и формирования алгоритмической культуры.

Самое сложное — дать характеристику понятия *алгоритмическое мышление*. В психологической литературе такой вид мышления не выделяют. В методической литературе по информатике, в которой развитие алгоритмического мышления называют главной задачей изучения информатики, алгоритмическим мышлением называют искусство размышлять, умение планировать свои действия, способность предусматривать различные обстоятельства и поступать соответственно с ними, способность легко рассуждать об алгоритмических процессах». Часто алгоритмическое мышление понимают как *алгоритмический стиль мышления*.

Поэтому мы солидарны с утверждением А.В. Копаева, что «в современной методической литературе, посвященной обучению информатике, сложно найти материалы, в которых не упоминается об алгоритмическом стиле мышления. Но еще сложнее найти публикации, в которых определяется это понятие» [6, *1*]. Именно в его работе мы нашли, на наш взгляд, наиболее конструктивное описание алгоритмического мышления, алгоритмического стиля мышления: «...будем считать, что алгоритмический стиль мышления — это система мыслительных способов действий, приемов, методов и соответствующих им мыслительных стратегий, которые направлены на решение как теоретических, так и практических задач и результатом которых являются алгоритмы как специфические продукты человеческой деятельности» [6, *1*].

Таким образом, основным признаком алгоритмического мышления можно считать способность к конструированию алгоритмов. Но так как алгоритм задает общий способ решения задач, то формирование алгоритмического мышления тесно связано с формированием общего умения решать задачи, так как прежде, чем задавать общий способ решения задач в форме алгоритма, нужно найти общий способ, затем исследовать возможность описания этого способа в форме конструктивных, однозначно понимаемых последовательных операций.

Кроме понятия *алгоритмическое мышление* в педагогической литературе широко используется понятие *алгоритмическая культура*. М.П. Лапчик называет алгоритмической культурой человека комплекс личностных качеств, обеспечивающих понимание «алгоритмической природы понятий» [7, *87*], понимание значения алгорит-

Алгоритм «Деление с остатком: подбор остатка» (27 : 5, 3 : 4)

Теоретические и (или) предметные (эмпирические) основы алгоритма	Алгоритм	Образцы рассуждений	Образцы записей	Знания и умения, необходимые для освоения алгоритма
1. Свойство остатка: остаток при делении всегда меньше делителя (может быть равным 0). **2.** Зависимость между результатом деления с остатком и его компонентами: сумма остатка и произведения частного на делитель равна делимому. **3.** Делимость разности делимого и остатка: разность между делимым и остатком делится нацело на делитель. **4.** Существует единственное число, меньшее делителя, вычтя которое из делимого (не делящегося нацело на делитель) получим число, делящееся нацело на делитель. **5.** Предметные (эмпирические) основы: предметные действия деления предметов, количество которых равно делимому, по столько предметов (или на столько равных частей), каков делитель; предметные действия, моделирующие свойства деления с остатком	**1.** Возьмем любое число, меньшее делителя, в качестве остатка. **2.** Вычтем его из делимого. **3.** Если разность делится нацело на делитель, то делим ее на делитель; полученный результат будет частным, а подобранное число — остатком; иначе (если разность не делится нацело на делитель) возвращаемся к первому шагу и берем другое число в качестве остатка. **4.** Называем результат: частным при делении числа на число будет число (из шага 3), а остатком — подобранное число	**1.** Развернутые рассуждения. Нужно разделить 27 на 5. 27 на 5 нацело не делится. Буду делить с остатком, подбирая остаток. а) Пусть остатком будет число 3. Разность делимого 27 и остатка 3 должна делиться нацело на 5. б) Вычту 3 из 27, получится 24. 24 на 5 нацело не делится. Значит, число 3 остатком не является. Возьму в качестве остатка другое число, например 4. Вычту 4 из 27. Получится 23. 23 на 5 тоже не делится, значит, 4 также не будет остатком. Пусть остатком будет число 2. Вычту из 27 это число. 27 – 2 = 25. 25 делится нацело на 5, в частном получится 5. Значит, при делении 27 на 5 в частном получится 5, а остатком будет 2. 27 : 5 = 5 (ост. 2). **2.** Частично свернутые рассуждения. Нужно разделить 27 на 5. Нацело 27 на 5 не делится. Вычту из 27 число 1, получится 26. 26 на 5 нацело не делится. Вычту из 27 число 2, получится 25. Разделю 25 на 5, получится 5. Частное от деле-	**1.** Развернутая запись. 27 : 5 *3* 27 – 3 = 24 24 не делится на 5. *4* 27 – 4 = 23 23 не делится на 5. *2* 27 – 2 = 25 25 : 5 = 5 27 : 5 = 5 (ост. 2) **2.** Частично свернутая запись. 27 : 5 27 – 2 = 25 25 : 5 = 5 27 : 5 = 5 (ост. 2) **3.** Краткая запись. 27 : 5 = 5 (ост. 2)	**1.** Табличное умножение и деление. **2.** Табличное вычитание. **3.** Смысл деления с остатком. **4.** Свойства остатка. **5.** Свойство делимости разности делимого и остатка. **6.** Зависимость между делимым, делителем, частным и остатком $(a : b = q$ (ост. $r) \Leftrightarrow$ $\Leftrightarrow a = bq + r$, где $0 \leqslant r < b)$. **7.** Признаки делимости на 2, 3, 5, 9 и на составные числа: 4, 6, 12 и т.п.

Окончание

Теоретические и (или) предметные (эмпирические) основы алгоритма	Алгоритм	Образцы рассуждений	Образцы записей	Знания и умения, необходимые для освоения алгоритма
		ния 27 на 5 равно 5, а остаток равен 2. 27 : 5 = 5 (ост. 2) **3.** Свернутые рассуждения. Нужно 27 разделить на 5. Из 27 вычесть 1, будет 26. 1 не подходит. Вычту 2, получится 25. 25 : 5 — получится 5. При делении 27 на 5 частное равно 5, а остаток равен 2. 27 : 5 = 5 (ост. 2)		

мов в различных сферах деятельности и в познании. Наличие алгоритмической культуры предполагает также умение действовать по заданному алгоритму, строить описание способа решения задачи в форме алгоритма, обладать алгоритмическим мышлением, алгоритмическим стилем мышления.

Под алгоритмической культурой человека будем понимать комплекс личностных качеств и определенный уровень алгоритмического мышления, обеспечивающих понимание роли алгоритмов в различных видах деятельности, владение умениями действовать по заданному алгоритму, конструировать алгоритмы, осуществлять выбор и применение алгоритмов в своей деятельности.

Формирование алгоритмической культуры — это обеспечение понимания роли алгоритмов в различных видах деятельности на уровне, соответствующем возрасту учащихся, и формирование названных выше умений. Каждое из умений, характеризующих алгоритмическую культуру, опирается на знания из той области, для решения задач которой предназначены выполняемые и конструируемые алгоритмы. Каждое из названных умений состоит из более частных умений. Они основаны на знаниях о задачах, методах и способах их решения, на знаниях о самих алгоритмах, причинах их изобретения, назначении, возможностях и границах применения.

Разработка и использование алгоритмов для решения задач из той или иной сферы человеческой деятельности, в том числе познания, создает значительные возможности для повышения эффективности этой деятельности, а наличие алгоритмической культуры у действующего, познающего субъекта позволяет реализовать эти возможности.

Формирование алгоритмической культуры в процессе обучения математике возможно как с включением в содержание учебного предмета «математика» специальной темы «Алгоритмы», так и без такого включения. Однако ФГОС НОО ставит задачу формирования алгоритмического мышления, и потому при обучении математике необходимо явно рассматривать понятие *алгоритм*. При этом термин *алгоритм* может использоваться с первых алгоритмов, встречающихся в содержании обучения математике. Такими алгоритмами являются, например, алгоритмы написания цифр, выполнения сложения при помощи присчитывания по единице, измерения длины и др.

О качестве обучения математике судят по многим показателям, в том числе по владению учащимися определенными алгоритмами. Для того чтобы знание конкретных алгоритмов было действенным, необходимо обеспечить понимание назначения алгоритмов, их особенностей, понимание их «человеческого» происхождения, многообразия возможных алгоритмов для решения задач одного и того же класса.

Ориентация обучения математике на развитие основ алгоритмического мышления, на формирование алгоритмической культуры увеличивает результативность обучения, усиливает развивающее воздействие, так как требует овладения «общими способами действий», что, по В.В. Давыдову [5], является необходимым условием развивающего обучения.

ИСПОЛЬЗОВАННАЯ ЛИТЕРАТУРА

1. *Бантова М.А.* Система вычислительных навыков // Начальная школа. 1979. № 11.
2. *Вершинин О.Е.* За страницами учебника информатики. М., 1991.
3. *Виленкин Н.Я., Дробышев Ю.А.* Воспитание алгоритмического мышления на уроках математики // Начальная школа. 1988. № 12.
4. *Галанин Д.Д.* История методических идей по арифметике в России. Ч. I. XVIII век. М., 1915.
5. *Давыдов В.В.* Теория развивающего обучения. М., 1996.
6. *Копаев А.В.* Алгоритм как модель алгоритмического процесса. URL: http://www.rusedu.info.
7. *Лапчик М.П.* Вычисления. Алгоритмизация. Программирование. М., 1988.
8. *Макаренков Ю.А.* Что такое алгоритм? Минск, 1988.
9. Математический энциклопедический словарь. М., 1988.
10. URL: http://hbar.phys.msu.ru.