

Информационные технологии и формирование умения учиться у младших школьников

Н.И. Саненко

Радикальные изменения, происходящие в сфере образования, вызваны потребностью общества в кадрах, способных принимать нестандартные решения, умеющих творчески мыслить.

Школа должна подготовить человека думающего и чувствующего, который не только имеет знания, но и умеет использовать их в жизни. Поэтому направленность на формирование инициативности, ответственности и самостоятельности ребенка должна выступить как главный приоритет с самого начала обучения. Основной формой проявления этих качеств в младшем школьном возрасте является учебная самостоятельность (умение учиться). Что это такое? Это умение:

1) планировать свои ближайшие и перспективные шаги;

2) оценивать результат своих действий;

3) оценивать свои знания и умения, обнаруживать и фиксировать свою недостаточность в чем-либо и при необходимости – обращаться за помощью, т.е. умение осуществлять рефлексию, необходимую для ответа на первый вопрос самообразования «Чему учиться?».

В начальной школе должны быть заложены основы не только предметных знаний, но и знаний о собственном незнании. Именно с действия самооценки, со способности понять, что «это я уже знаю и умею, а этого не знаю», и начинается учебная самостоятельность, переход от просто старательного ученика к человеку, умеющему учиться.

Поэтому я стараюсь организовать учебную деятельность на уроках

так, чтобы дети встречались с ситуациями, где их знания вступают в противоречие с новыми фактами. Даю невыполнимое практическое задание или задание, несходное с предыдущим, и задаю вопросы:

– Можешь ли ты выполнить это задание?

Почему?

Что тебе неизвестно?

Разбирая практическое задание, несходное с предыдущим, ученик видит неприемлемость или недостаточность старых знаний. Помогаю ему вопросами:

– Что ты хотел сделать?

– Что сделал?

– Какие знания применил?

– Задание выполнено?

– Почему не выполнено?

– Что неизвестно?

– Какова будет цель твоего дальнейшего обучения?

Иногда формулирую проблемный вопрос (сразу дать на него ответ невозможно):

– Можешь ли сразу ответить на вопрос?

– Что тебе нужно знать для того, чтобы ответить?

В ходе работы фиксируем все возникающие у детей вопросы. Именно эти трудности и являются основанием для составления **технологической карты**, в которой мы определяем цели дальнейшего обучения. Однако осознания того, «чему надо учиться», недостаточно. Ученик должен понимать, какие поисковые действия необходимы для приобретения недостающих знаний, умений.

В связи с этим встает второй вопрос самообразования: «Как научиться?» или «Каким способом достичь цели?» На него есть три ответа:

1) самостоятельно изобрести недостающий способ действия;

2) самостоятельно найти недостающую информацию в любом «хранилище»;

3) запросить недостающие данные у знатока [4, с. 66–67].

«Учебная самостоятельность разви-

того младшего школьника состоит в умении или способности инициировать совместное со взрослым действие по поиску недостающих способов решения новых задач» [8]. Высказывая догадку о недостающем способе действия, ученик начальной школы в первую очередь прибегает к помощи учителя. Учитель – это тот, кто учит самому учению. Важно научить детей не столько действовать, сколько планировать будущее действие, не давая ученику в погоне за результатом терять из вида способы достижения цели.

Одним из таких способов является составление алгоритма. Без этого трудно обойтись на этапе планирования и организации деятельности, так как необходимо установить последовательность действий для решения задачи и ответить на вопрос «Что и как делать, чтобы достичь цели?».

На этапе оценки результатов деятельности ученик отвечает на вопрос «Верный ли получен результат?». Контроль в процессе деятельности гораздо эффективнее контроля по результатам деятельности, поэтому при наличии алгоритма промежуточный контроль осуществить легче. Значимость вопросов, связанных с умением составлять, записывать и осуществлять алгоритмы, в последние годы неизменно возросла. В ряде публикаций, в частности в статьях Н.Я. Виленкина, Л.Г. Дробышева, А.В. Горячева и др., обосновывается целесообразность раннего ознакомления детей с вычислительной техникой, развития у них алгоритмического, логического мышления, освоения основ программирования. Главным аргументом является необходимость подготовки школьников к жизни в информационном обществе. На первый план выдвигается формирование у учащихся инновационной культуры. Нужно учить детей ориентироваться в информационных потоках, эффективно осуществлять поиск информации, ее обработку, классификацию. А поиск новой информации (работа с компьютером,

со словарями и т.д.) связан с алгоритмами.

К настоящему времени подготовлено несколько программ для начальной школы по изучению информатики. Среди них хотелось бы выделить безмашинный вариант «Информатика в играх и задачах» (автор А.В. Горячев). Данный курс является составной частью УМК «Школа 2100», по которому я работаю уже 8 лет.

Авторы программы и учебников уделяют серьезное внимание развитию таких логических приемов, как анализ, синтез, сравнение, абстрагирование, обобщение. Именно эти приемы необходимы для восприятия и обработки информации и, конечно, для **составления алгоритмов. Формирование этого умения осуществляется в четыре этапа.**

На первом этапе дети знакомятся с понятиями *операция (действие), результат операции*, учатся определять результат действия.

На втором этапе узнают, что такое *программа действий* или *алгоритм*, учатся устанавливать последовательность действий, исполнять простые алгоритмы, составлять словесные алгоритмы.

На третьем этапе дети знакомятся со способами наглядного представления алгоритмов, учатся четко исполнять алгоритмы, заданные этими способами.

На четвертом этапе дети учатся составлять алгоритмы.

Этапы формирования данного умения отражены в технологической карте (см. Приложение).

На каждом этапе проводится диагностика, в ходе которой выявляется степень сформированности данного умения. В разделе «Дозирование домашней самостоятельной работы» приводятся упражнения, формирующие данные умения.

Задания репродуктивного характера (уровень 1) направлены на проверку знаний учащимися основных понятий по теме, умений исполнять готовый алгоритм.

Задания реконструктивного характера (уровень 2) предполагают проверку умений учащихся не только работать по готовому алгоритму, но и их способность найти ошибки в алгоритме, внести в него дополнения, изменения.

Задания конструктивного характера (уровень 3) предоставляют ребенку возможность найти множество вариантов решения задачи, дают свободу выбора средств для достижения цели. Ребенку дается условие и результат, которого необходимо достичь, и он сам ищет пути его достижения.

Если учащиеся выполняют задания только первого и второго уровней, то это значит, что они понимают цель учебной деятельности, при достижении которой используют частные приемы и готовые алгоритмы, а значит, можно говорить о среднем уровне развития ученической самостоятельности. Если ученик может сам составить алгоритм, то это значит, что у него высокий уровень развития учебной самостоятельности, так как он самостоятельно может ставить цели учебной деятельности, составлять план самообразования и умеет находить средства его реализации. Самостоятельное составление программ не является в начальной школе обязательным, дети должны лишь уметь пользоваться готовой программой, уметь прочитать ее, объяснить последовательность действий. Однако на уроках по всем предметам я привлекаю детей к составлению алгоритмов. Иногда задаю подобные задания на дом. Понятно, что дома программы у ребят не всегда получаются, составляют они их с ошибками. Но сам процесс обдумывания последовательности выполняемых операций оказывает самое благоприятное влияние на развитие алгоритмического мышления.

В учебниках русского языка (авторы Р.Н. Бунеев, Е.В. Бунеева, О.В. Пронина) есть готовые инструкции, алгоритмы. Но я стараюсь, чтобы дети самостоятельно открывали новые знания. Поэтому не даю детям сразу готовые алгоритмы, а организую

их работу так, чтобы они сами «додумались» до решения проблемы и смогли пошагово спланировать свои действия, т.е. составить алгоритм. Ученику начальной школы самостоятельно это сделать трудно, но под руководством учителя – возможно. Покажу это на примере **фрагмента урока русского языка во 2-м классе по теме «Сложные слова»**.

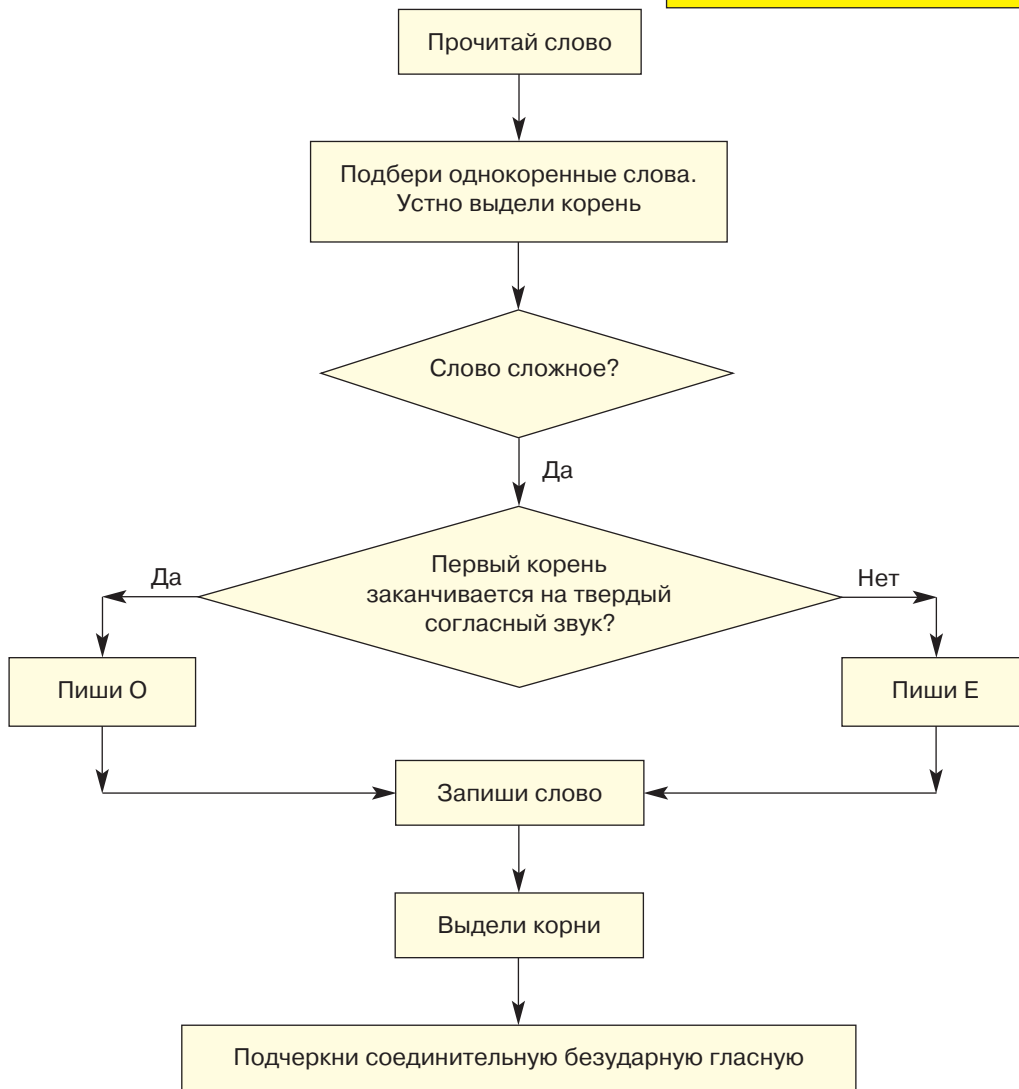
Ставлю перед детьми задачу: вставить пропущенные буквы в слове *в_д_пады*. Дети смогли вставить и объяснить написание безударной гласной в корне по ранее составленному алгоритму проверки безударной гласной в корне слова. Однако впереди их ждало затруднение: какую безударную гласную писать во втором случае?

Предлагаю детям проверить свою орфографическую интуицию и предположить, какую гласную они написали бы. Мнения учащихся разделились: одни говорят, что они написали бы *о*, другие – *а*. Появилась проблема. Детям на данном этапе не хватает знаний. Нужно искать пути решения проблемы. Одни дети предлагают заглянуть в словарь, другие спрашивают, как писать, у меня, но некоторые замечают, что слово *водопады* необычное, в нем два корня, и что безударная гласная стоит между ними, и предлагают найти нужную информацию о подобных словах в учебнике.

Далее дети проводят наблюдение на материале учебника и делают вывод, что в сложных словах есть два корня, которые соединяются гласными *о* и *е*, предполагают, почему соединительные гласные получили такое название, и замечают, что гласная *о* пишется после корня, оканчивающегося на твердый согласный звук, а гласная *е* – после мягкого согласного.

Результатом этой деятельности является алгоритм, составленный детьми под руководством учителя (см. с. 45).

Подобные алгоритмы составлены нами по всем ключевым темам русского языка, изучаемым в начальной школе.

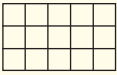

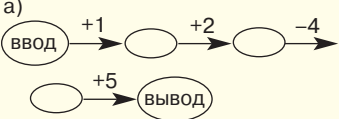
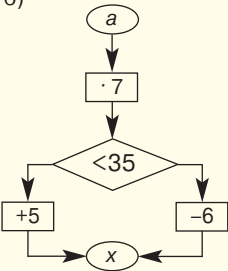

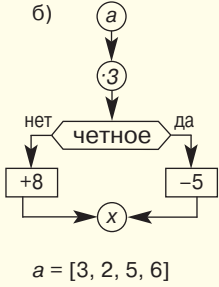


Приложение

Технологическая карта

Формирование умения учиться посредством обучения составлению алгоритмов


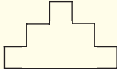
Целеполагание	Диагностика	Коррекция
<p>Ц. 1. Знать, что такое действие предметов, уметь определять результат действия.</p>	<p>Д. 1. 1) Определи результат действия: заменить букву с на д. <div style="display: flex; align-items: center; margin-bottom: 10px;"> СОМ _____ </div> 2) Какое действие привело к результату? <div style="display: flex; align-items: center; margin-bottom: 10px;"> 5 → ? → 6 </div> <div style="display: flex; align-items: center;"> бесцветный листок → красный листок </div> 3) Придумай операцию и результат. <div style="display: flex; align-items: center; margin-top: 10px;"> □ → ? </div> </p>	<div style="display: flex; align-items: center; margin-bottom: 10px;"> предмет → <div style="text-align: center; font-size: small;">операция или действие</div> → результат действия </div> <p>Например: 3 → прибавить 2 → 5</p> <p>Объектом операции может быть предмет, число, слово, арифметическое действие.</p>

Целеполагание	Диагностика	Коррекция										
<p>Ц. 2. Знать, что такое программа действий (алгоритм), и уметь определять последовательность событий.</p>	<p>Д. 2. 1) Расскажи по картинкам сказку «Репка». 2) Собери и дополни рассыпавшийся алгоритм «Режим дня». Все команды в алгоритме записаны на отдельных карточках. Нужно расставить события по порядку и вставить пропущенное действие.</p> <p><input type="checkbox"/> Завтрак <input type="checkbox"/> Школа <input type="checkbox"/> Зарядка <input type="checkbox"/> Ужин <input type="checkbox"/> Обед <input type="checkbox"/> Прогулка <input type="checkbox"/> Подъем <input type="checkbox"/> Сон <input type="checkbox"/> Уроки</p> <p>3) Составь алгоритм для робота «Съешь апельсин».</p>	<p>Алгоритм указывает, в каком порядке тебе нужно выполнять действия, чтобы получить желаемый результат. Условие. 1. _____ 2. _____ 3. _____ 4. _____ Результат.</p>										
<p>Ц. 3. Знать способы наглядного представления алгоритмов, уметь четко выполнять алгоритмы, заданные этими способами.</p>	<p>Д. 3. 1) Построй сообщение по алгоритму: кто? → что? → где? → чем? → зачем? → как? → когда? 2) Вставь пропущенные действия. Нарисуй блок-схему к алгоритму «Сделай бутерброд». Возьми хлеб. Отрежь кусок хлеба. Возьми масло. Намажь маслом хлеб. Стоп. 3) Придумай любой алгоритм и запиши его, используя любой способ представления алгоритма.</p>	<p>Помни, алгоритмы можно представлять следующим образом. 1) Словесное описание. 2) Таблица.  3) Графические схемы. Нарисуй фигуру.  4) Блок-схемы а)  б)  в) <table border="1" data-bbox="1193 1272 1300 1451"> <tr><td>x</td><td>?</td></tr> <tr><td>· 3</td><td></td></tr> <tr><td>+ 33</td><td></td></tr> <tr><td>: 9</td><td></td></tr> <tr><td>- 6</td><td></td></tr> </table> </p>	x	?	· 3		+ 33		: 9		- 6	
x	?											
· 3												
+ 33												
: 9												
- 6												
<p>Ц. 4. Знать типы алгоритмов, уметь их различать, выполнять и составлять.</p>	<p>Д. 4. 1. Укажи тип алгоритмов (линейный, разветвляющийся, циклический). а)  б)  a = [3, 2, 5, 6]</p>	<p>1. При составлении алгоритма: 1) сначала определи то, над чем будешь выполнять действие (операцию); 2) затем посмотри, какой результат нужно получить; 3) определи, какие действия ты должен выполнить, чтобы достичь желаемого результата; 4) определи, в какой последовательности ты будешь выполнять эти действия. 2. В блок-схеме действия записываются</p>										

<p>Умножь на 2 ↓ Запиши ответ ↓ Конец</p> <p>2. Восстанови пропущенное действие. <u>Заточи все сломанные карандаши:</u> 1) Собери все сломанные карандаши. 2) ... 3) Возьми сломанный карандаш. 4) Заточи его. 5) Если есть еще сломанные карандаши, перейди к строке № 3. 6) Стоп. 3. Составь алгоритм с ветвлением «Собери грибы» и заполни блок-схему.</p> <pre> graph TD Start([Начало]) --> B1[] B1 --> B2[] B2 --> D1{ } D1 -- Да --> B3[] D1 -- Нет --> B1 B3 --> D2{ } D2 --> Stop([Стоп]) </pre>	<p>ся в прямоугольнике, а стрелки показывают последовательность выполнения действий, ведущих к результату. 3. Вспомни виды алгоритмов. а) Линейный, в котором все действия идут подряд.</p> <pre> graph LR A(условие) --> B[I действие] B --> C[II действие] C --> D[III действие] D --> E(ответ) </pre> <p>б) Разветвляющийся. Действия можно выполнять по одной из двух дорожек, в зависимости от ответа на вопрос, записанный в ромбе.</p> <pre> graph TD Start([Начало]) --> D{?} D -- Нет --> B1[] D -- Да --> B2[] B1 --> B3[] B2 --> B4[] B3 --> Result([Результат]) B4 --> Result </pre> <p>в) Циклический, в котором одно и то же действие повторяется несколько раз.</p>
---	---

Дозирование самостоятельной работы

Удовлетворительно	Хорошо	Отлично						
<p>Ур. 1. Назови как можно больше действий этого предмета: книга, картошка. Назови действия врача, учителя. Определи и изобрази результат действия. ○ → уменьшить размер</p> <p>Ур. 2. Расскажи, в каком порядке ты будешь выполнять действия, укладывая куклу спать.</p>	<p>Какое действие привело к этому результату?</p> <pre> сон / \ сын стон \ / нос </pre> <p>3 → ? → 10</p> <p>Для каждого предмета подбери возможный результат и соедини слова линией.</p> <table> <tr> <td>Фрукты</td> <td>Хлеб</td> </tr> <tr> <td>Гусеница</td> <td>Компот</td> </tr> <tr> <td>Мука</td> <td>Бабочка</td> </tr> </table> <p>Правильно ли записан алгоритм «Проводи гостя»? Исправь ошибки.</p>	Фрукты	Хлеб	Гусеница	Компот	Мука	Бабочка	<p>Чем будет яйцо? Назови несколько возможных вариантов. Придумай действие, которое нужно выполнить с предметом, и изобрази результат.</p> <p>○ → ? ☆ → ?</p> <p>Придумай продолжение действия: ребенок заболел (_____).</p>
Фрукты	Хлеб							
Гусеница	Компот							
Мука	Бабочка							

Удовлетворительно	Хорошо	Отлично
<p>Соедини линиями предметы в правильном порядке: дом, жельдь, дуб, бревно.</p> <p>Ур. 3. Нарисуй фигуру по стрелочкам.</p>  <p>Выполни алгоритм к № 8, с. 7 учебника (<i>Горячев А.В.</i> Информатика, 3 класс. Часть 1).</p> <p>Ур. 4. Найди вопросы, на которые можно ответить «да» или «нет», и ответ на них. – Сегодня понедельник? – Тебе 10 лет? – Сколько тебе лет? – Где ты живешь? – Ты любишь рисовать?</p> <p>Запиши вопросы, на которые можно ответить «да» или «нет».</p>	<p>1) Открой дверь. 2) Попрощайся с гостем. 3) Закрой за ним дверь. 4) Помоги гостю одеться. 5) Стоп.</p> <p>Допиши команды алгоритма стрелочками.</p>  <p>1) Начало 2) * 3) ↑ 1 4) 5)</p> <p>Нарисуй блок-схему к алгоритму № 21, с. 18 учебника.</p> <p>Поиграй с другом в игру «Ты – мне, я – тебе» (проводится в парах). Один ученик придумывает по рисунку и задает другому ученику вопрос, на который можно ответить «да» или «нет», а другой ученик – вопрос, на который нельзя так ответить.</p> <p>Выполни алгоритмы к № 7, с. 6; № 14, с. 11 учебника.</p>	<p>Придумай промежуточное действие. Дано начало и конец действия. Как из муки выпечь хлеб?</p> <p>Придумай и нарисуй свою фигуру и составь алгоритм под рисунком.</p> <p>Заполни блок-схему к № 18, с. 7 учебника.</p> <p>Запиши алгоритм к № 16, с. 13 с помощью цикла № 20, с. 17 учебника.</p> <p>Заполни блок-схему № 22, с. 19.</p> <p>Нарисуй блок-схему № 21, с. 18 разветвляющегося алгоритма.</p> <p>Расставь переходы в блок-схеме № 23, с. 20.</p>

Литература

1. *Антипов И.Н.* Изучение основ информатики в школе // Начальная школа, 1985, № 11.

2. *Виленкин Н.Я., Дробышев Ю.А.* Воспитание алгоритмического мышления на уроках математики // Начальная школа, 1988, № 12.

3. *Горячев А.В., Лукашенко М.А.* Информатика в играх и задачах // Начальная школа, 1995, № 8, с. 56–59.

4. Образовательный процесс в начальной, основной и старшей школе (Рекомендации по организации опытно-экспериментальной работы) – М.: 2001.

5. *Петерсон Л.Г.* Математика? Это интересно! // Начальная школа: Прилож. к газете «Первое сентября», 1996, № 130.

6. *Цукерман Г.А.* Учебная деятельность на переходе из начальной в основную школу. <http://www.Maro.Newmail.ru/>

ped_master/innovacii_nch/in_cukerman.htm/

7. «Школа 2000». Концепция и программы непрерывных курсов для общеобразовательной школы. Вып. 1. – М.: Баласс, 1998.

8. «Школа 2100». Образовательная программа и пути ее реализации. – М.: Баласс, 1999.

Наталья Ивановна Саненко – учитель начальных классов Каргапольской начальной общеобразовательной школы № 3, р.п. Каргаполье, Курганская обл.